A computationally efficient model predictive control scheme for space debris rendezvous

Alexander Korsfeldt Larsén [*], **Yutao Chen** [**], **Mattia Bruschetta** [***], **Ruggero Carli** [***], **Angelo Cenedese** [***], **Damiano Varagnolo** [****], and **Leonard Felicetti** [†]

[*] *Luleå University of Technology, Kiruna, Sweden, email: alexander.korsfeldt@gmail.com*
[**] *Eindhoven University of Technology, Eindhoven, The Netherlands, email: y.chen2@tue.nl*
[***] *University of Padova, Padova, Italy, emails: mattia.bruschetta@unipd.it, carlirug@dei.unipd.it, angelo.cenedese@unipd.it*
[****] *Norwegian University of Science and Technology, Trondheim, Norway, email: damiano.varagnolo@ntnu.no*
[†] *Cranfield University, Cranfield, United Kingdom, email: leonard.felicetti@cranfield.ac.uk*

Abstract:
We propose a non-linear model predictive scheme for planning fuel efficient maneuvers of small spacecrafts that shall rendezvous space debris. The paper addresses the specific issues of potential limited on-board computational capabilities and low-thrust actuators in the chasing spacecraft, and solves them by using a novel MatLab-based toolbox for real-time non-linear model predictive control (MPC) called MATMPC. This tool computes the MPC rendezvous maneuvering solution in a numerically efficient way, and this allows to greatly extend the prediction horizon length. This implies that the overall MPC scheme can compute solutions that account for the long time-scales that usually characterize the low-thrust rendezvous maneuvers. The so-developed controller is then tested in a realistic scenario that includes all the near-Earth environmental disturbances. We thus show, through numerical simulations, that this MPC method can successfully be used to perform a fuel-efficient rendezvous maneuver with an uncontrolled object, plus evaluate performance indexes such as mission duration, fuel consumption, and robustness against sensor and process noises.

Keywords: Non Linear Model Predictive Control, Non-cooperative Rendezvous, Space Debris Removal, Low-Thrust Maneuvers, Constrained Optimization Problem, Real-time Control

1. INTRODUCTION

Space debris is becoming a major threat for current and future space missions. Collisions among pieces of debris and satellites are not anymore infrequent, and collision avoidance maneuvers are frequently performed to avoid space junk, with a sensible increase of space mission costs (Schaub et al., 2015). Recent studies show that relying on only mitigation strategies is no longer enough to solve the space debris problem, and active debris removal missions thereby appear to be a necessity (Liou et al., 2010).

Since the threat posed by each piece of debris is a product of its mass and probability of collision, today's efforts are mainly focused on removing the larger pieces of debris (Liou, 2011). While there are many methods being considered, the vast majority of them depends on being able to get one or more controlled spacecrafts close enough to the debris to perform a docking maneuver, and then use on-board thrusters to move the debris away (Shan et al., 2016). To reduce the financial costs involved in these operations and make them economically feasible, the current trend is to use small satellites, such as CubeSats (Lucken et al., 2017; Larbi et al., 2017). This comes however with weak actuation capabilities, limited on-board computation power, and lower quality sensors (Udrea and Nayak, 2015). The low thrust values turn even simple orbital maneuvers into complex optimization problems, as the control actions must be planned for a long period ahead. Moreover, the appeal of being able to solve these control problems using the limited on-board computers, rather than having it done by a ground station, puts strict requirements on the computational efficiency of the chosen control strategy. Finally, low quality sensors imply the need for robust control schemes.

Thus, the following tasks for the the on-board computational capabilities must be set for executing rendezvous

[⋆] The research leading to these results has received funding through an ERASMUS TRAINEESHIP SCHOLARSHIP (SMP) as part of the Erasmus+ Mobility for Traineeships Programme

maneuvers to pieces of debris through small chasing spacecrafts:

(1) the on-board computer shall estimate the relative position of the debris with respect to the chasing spacecraft by using measurements from on-board sensors such as cameras, lidars, etc.;
(2) the on-board computer shall plan the trajectory that the chaser should follow;
(3) the on-board computer shall also calculate how to actuate the thrusters so to follow as closely as possible the reference trajectory in the point above.

This paper is mainly focused on solving the second point in a computationally efficient way, while keeping in mind the implications and constraints caused by the third.

We thus cast the problem of computing and updating which trajectory a satellite chasing a debris should follow as an Model Predictive Control (MPC) problem. The suggested approach allows for taking into account physical constraints such as limited thrusting capabilities when generating a trajectory, and strives to get a trajectory that is as close as possible to a feasible one, while ignoring the effects of disturbances that are difficult to predict.

In particular, our main contribution lies on the fact that the suggested MPC formulation takes especially care about the fact that completing a rendezvous maneuver may take several days. This means that minimizing the fuel spent on the mission requires computing reference trajectories that may span for very long time horizons, and that in its turn requires that the relative motion between the chasing and chased objects can be predicted for an adequately long time. This prediction problem is naturally non-linear, and traditionally the issue with using non-linear MPC is of being subject to large computational burdens. The contribution of this paper is thus on using MPC techniques that are amenable to fast computations, so that longer prediction horizons can be achieved. Beyond proposing this technology, the manuscript also characterizes its computational and suitability performance through in-silico experiments.

The remaining part of the paper is organized as follows: a brief overview of the rendezvous scenario and a characterization of the adopted simulation model is presented in section 2. Section 3 explains how the controller is designed and implemented. The simulation environment used to assess the performances of the MPC strategy is described in section 4 and section 5 shows the main results obtained through opportune sets of simulations. Section 6 finally draws some conclusions and future potential research directions.

2. SYSTEM MODELING

We consider the scenario represented in Figure 1. A piece of debris d has been identified by a spacecraft c as a potential target for an active debris removal operation. The chaser c needs to plan and perform a rendezvous maneuver to d in such way that the relative distance between the two orbiting objects is close enough and the relative attitude is sufficiently synchronized to allow for the removal operation. The following quantities can be used to kinetically and dynamically characterize the state of the objects involved in the scenario:

• the orbital position of the chaser and of the debris, denoted with ${}^i\boldsymbol{r}_c$ and ${}^i\boldsymbol{r}_d$ respectively, whose components are defined with respect to the Earth Centered Inertial (ECI) reference frame, as denoted by the left subscript i;

• the orbital velocity of the chaser and of the debris, denoted with ${}^i\boldsymbol{v}_c$ and ${}^i\boldsymbol{v}_d$ respectively;

• the quaternions representing the attitudes of the chaser and of the debris, denoted with $\boldsymbol{q}_c$ and $\boldsymbol{q}_d$ respectively, defined as in (Curtis, 2014, Chap.9);

• the angular velocities of the chaser and of the debris with respect to the ECI frame, denoted with ${}^c\boldsymbol{\omega}_c$ and ${}^d\boldsymbol{\omega}_d$ respectively, whose components are defined with respect their respective body coordinate frames, as highlighted by the left subscripts c and d, respectively.

In Figure 1 the ECI reference frame is denoted by using the letter i, the chaser's body reference frame with c and the debris's body reference frame with d. The Local Horizontal Local Vertical (LHLV) reference frame, denoted in the same figure by using the letter l, is centered to the debris's center of mass and has the $\hat{\boldsymbol{x}}_l$ axis aligned along the radial direction, the $\hat{\boldsymbol{y}}_l$ axis aligned along the in-track direction, and the $\hat{\boldsymbol{z}}_l$ axis aligned along the normal direction of the the debris's orbit plane.

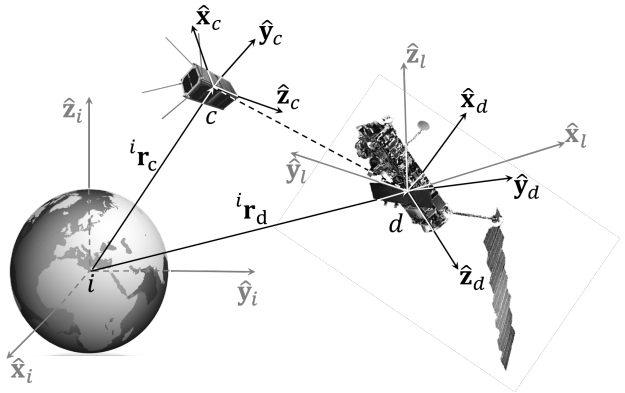

Fig. 1. Schematic representation of the reference frames considered in our space debris rendezvous scenario.

2.1 Dynamics of the environment

With the intent of providing a fairly realistic test environment, the model used for computing the orbital dynamics of both the chaser and the debris is based on the equations in (Curtis, 2014). More precisely, we include the perturbations due to J_2 Earth's oblateness (Curtis, 2014, Chap.4), atmospheric drag, solar radiation pressure, and Luni-Solar attraction (Curtis, 2014, Chap.12). The attitude dynamics of both the chaser and the debris are then simulated by using the equation of motion of a rigid body (Curtis, 2014, Chap.9) subjected to environmental torques (Curtis, 2014, Chap.10).

2.2 Dynamics of the chaser

We assume that the chaser spacecraft is a small box-shaped satellite with a homogeneous mass distribution and

with no protruding objects, so that the radiation pressure and drag coefficients are independent with respect to the attitude motion. The chaser is also assumed to have a total of six electrical thrusters, each mounted at the center of each surface, aligned with the center of mass and therefore producing no torque. Thus, the applied thrust to the chaser can be simply modelled by the vector ${}^c\boldsymbol{T}_c$.

The chaser is also assumed to embed three reaction wheels; one mounted along each axis of the body reference frame. For simplicity we do not model the reaction wheels themselves, but instead just model their output torques ${}^c\boldsymbol{\tau}_c$ - a more accurate modeling would indeed require the inclusion of a momentum dumping procedure in the overall control scheme. In the formulations within this paper this has thus been deemed to be an unnecessary complication; however the framework presented in the manuscript can be opportunely extended to include such details.

2.3 Sources of noise

The following phenomena are embedded in our simulation environment to compute the orbital dynamics of the various objects; however these sources are considered as *noises* for the control scheme computing the rendezvous maneuver. In other words, we assume that our control scheme does not embed forecasters or observers of these sources of uncertainty:

• *Orbital perturbations:* while both the simulated objects and controller itself are built with the equations outlined in Section 2.1, the controller dynamics have been simplified so to completely ignore the contributions caused by atmospheric drag, solar radiation pressure, and the Luni-Solar attraction. This reduces the complexity of the problem that the controller must solve, which in turn lowers the computational effort required by the on-board computers. The dynamics that are not taken into account by the controller can therefore be seen as fairly realistic sources of noise. Note that the J_2 effect is however included in the controller dynamics, as its contributions are both significant and can be easily estimated.

• *Model uncertainties:* we assume that the chaser has a constant and uniformly distributed mass, i.e., we assume that the centre of mass of the spacecraft is at its geometrical center. In reality there is often a small distance between these two points, and we may not have exact knowledge about this, due to non-perfect knowledge about the mass distribution of the object. Note moreover that mass consumption due to the usage of actuators also makes this mass distribution a virtually time-varying quantity. However, due to the high specific impulse associated with electrical thrusters, the total effects of the mass consumption from the planned maneuvers is here neglected.

• *Actuation uncertainties:* physical limitations or imperfections in the actuators, such as slight misalignments of the thrusters or small irregularities in their output profiles imply that the effective amount of thrust and torque produced by the thrusters will not be exactly that one that the controller would otherwise signal.

• *Measurement errors:* it can be safely assumed that the chaser will use a variety of different sensors, each with its own unique noise profile, to measure the states of both itself and the target. Although many signal filtering techniques can be used to obtain accurate statistical estimates of the quantities of interest, some level of measurement noise is inevitably going to be present and affect the control scheme. Simulating this accurately requires assumptions regarding the specific load-out of the chaser. For the sake of simplicity we model these measurement errors by using, in the control scheme, estimates of the state that are typically [1] computed as the actual state plus a white heteroskedastic Gaussian random vector with independent components, where the heteroskedasticity model is linear, i.e., of the kind for which the overall standard deviation of the noise grows linearly with the state itself as in

$$\hat{x}_i(k) = x_i(k) + \sigma x_i(k)\nu_i(k) \tag{1}$$

where σ depends on the actual state under consideration (see Section 5 for the actual values used in the simulations) and where k denotes a discrete time index (as described more verbosely in Section 3.3).

3. DESIGN OF THE CONTROL LAW

This section describes the control scheme that we propose as a potential solution to solve the chasing problem defined in Section 1.

3.1 Definition of the control requirements

The main requirement of the mission is to let a small spacecraft (such as, e.g., a CubeSat) be able to change its whatever initial orbit into a second orbit that is suitable for docking with a piece of debris. This orbital change should be performed while simultaneously taking into account: *a)* the limited actuation capabilities of the satellite, *b)* fuel efficiency considerations, and *c)* the fact that the computation of the reference trajectory should be not only performed on board, but also using measurements on the attitude / orbit of the debris that are provided by on-board sensors. In other words, requirement *c* above implies that the involved computational algorithms must be fast enough to be handled by an on-board computer. Moreover, since the on-board measurements will be of limited accuracy, the overall strategy should also be robust against a great variety of disturbances.

3.2 Composition of the maneuvers

The main assumption that the chaser is a small spacecraft implies that it will likely be launched as a secondary payload. It is therefore meaningful to expect that the chaser will begin its mission in a different orbit / orbital phase than that of the target. Assuming that the launch and early orbit phase procedures (such as de-tumbling) have been completed (Felicetti et al., 2016), the first step of the mission should be then to change the orbital parameters of the chaser so to make them match those of the target. After this, docking operations can start.

Consider then the following two key facts:

- during the orbit change phase, the chaser has access to information of limited accuracy about the

[1] Some important exceptions are described more precisely in Section 4.

target's motion. Thus, precision in following exactly the computed reference trajectory may be relaxed to the purpose of increasing the fuel-efficiency of the maneuver. This phase may thus last several days, implying that the orbits prediction horizon should span a considerably long time;
- when the chaser is close enough to initiate the docking procedure, it should make optimal use of the on-board resources and emphasize precision instead of fuel-consumption, as docking is a very delicate process.

In other words, we distinguish two intrinsically different situations, with the latter expected to temporally last for only a fraction of the former, where the lengths of the prediction horizons are different, and where the requirements on the precision in following the reference trajectories / in saving fuel are vastly different.

Thus, the requirements on the controller change in several ways when the chaser passes from being hundreds of kilometers away from its target to just within a few meters. *For this reason we devise to divide the mission in two parts, one for the orbital change maneuver, and one for the approaching-for-docking maneuver.* We solve both through a dedicated MPC formulation where the considered dynamics are practically the same, but where also the prediction horizon and control costs are different.

3.3 State space representation of the system

To formulate the control problem we describe the chaser using a state space representation where the state is

$$\boldsymbol{x} := \begin{bmatrix} {}^{i}\boldsymbol{r}_c^T & {}^{i}\boldsymbol{v}_c^T & \boldsymbol{q}_c^T & {}^{c}\boldsymbol{\omega}_c^T & {}^{c}\boldsymbol{T}_c^T & {}^{c}\boldsymbol{\tau}_c^T \end{bmatrix}^T, \tag{2}$$

and the input

$$\boldsymbol{u} := \begin{bmatrix} {}^{c}\dot{\boldsymbol{T}}_c^T & {}^{c}\dot{\boldsymbol{\tau}}_c^T \end{bmatrix}^T, \tag{3}$$

that is, the rate of variation of the thrusts and torques applied to the chaser.

In Korsfeldt Larsén (2018), one can find the expression of the map f relating $\dot{\boldsymbol{x}}$ to $\boldsymbol{x}$ and $\boldsymbol{u}$, that is,

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{u}) \tag{4}$$

Since the mission requires the chaser to match its orbit with that of the debris, we also define the relative position and velocity of the chaser with respect to the debris as

$${}^{l}\boldsymbol{d}_c := {}^{l}\boldsymbol{R}_i\left({}^{i}\boldsymbol{r}_c - {}^{i}\boldsymbol{r}_d\right) \tag{5}$$

$${}^{l}\dot{\boldsymbol{d}}_c := {}^{l}\boldsymbol{R}_i\left({}^{i}\boldsymbol{v}_c - {}^{i}\boldsymbol{v}_d\right) + {}^{i}\boldsymbol{\omega}_l \times {}^{l}\boldsymbol{R}_i\left({}^{i}\boldsymbol{r}_c - {}^{i}\boldsymbol{r}_d\right) \tag{6}$$

where ${}^{c}\boldsymbol{R}_i$ is the rotation matrix from the ECI to the LHLV reference frame, and ${}^{i}\boldsymbol{\omega}_l$ is the angular velocity of the LHLV with respect to the ECI reference frame.

Further, controlling the attitude of the chaser requires defining the error quaternion, i.e.,

$$\boldsymbol{q}_{err} := \boldsymbol{q}_{c,ref} \otimes \boldsymbol{q}_c^*, \tag{7}$$

where $\otimes$ indicates the quaternion multiplication, $\boldsymbol{q}_c^*$ is the conjugate of the measured/predicted quaternion, and $\boldsymbol{q}_{c,ref}$ is the quaternion representing the target attitude that the chaser has to point to during the maneuver.

3.4 Designing the solution via a Model Predictive Control formulation

Intuitively, the control action shall allow trading-off the speed in performing the rendezvous task with minimizing the associated fuel consumption. To this goal, it is convenient to introduce the output of our system as

$$\boldsymbol{y} := \begin{bmatrix} {}^{l}\boldsymbol{d}_c & {}^{l}\dot{\boldsymbol{d}}_c & \overline{\boldsymbol{q}}_{err}^T \end{bmatrix}, \tag{8}$$

where $\overline{\boldsymbol{q}}_{err}$ in $\boldsymbol{y}$ denotes the vectorial part of the error quaternion. Note that $\boldsymbol{y}$ depends on the state $\boldsymbol{x}$ as well as the parameters ${}^{i}\boldsymbol{r}_d$, ${}^{i}\boldsymbol{v}_d$, and $\boldsymbol{q}_{c,ref}$ and, hence, we can write

$$\boldsymbol{y} = h\left(\boldsymbol{x}, {}^{i}\boldsymbol{r}_d, {}^{i}\boldsymbol{v}_d, \boldsymbol{q}_{c,ref}\right), \tag{9}$$

for a suitable map h. Observe that reaching the rendezvous task means that $\boldsymbol{y} = 0$.

A natural approach to trade off $\boldsymbol{u}$ with $\boldsymbol{y}$ above is to consider MPC formulations, that formalize the aforementioned aim by encoding the control design step as a receding horizon optimization problem. In our specific setting, the optimal control problem to be solved at time t, given the current state $\boldsymbol{x}(t)$ is formulated as

$$\begin{aligned} \min_{\hat{\boldsymbol{u}}} \quad & \int_0^T \left(\|\hat{\boldsymbol{y}}\|_Q^2 + \|\hat{\boldsymbol{u}}\|_R^2\right) d\tau \\ \text{s.t } & \hat{\boldsymbol{x}}_0 = \boldsymbol{x}(t), \quad \dot{\hat{\boldsymbol{x}}} = f(\hat{\boldsymbol{x}}, \hat{\boldsymbol{u}}), \quad \hat{\boldsymbol{y}} = h\left(\hat{\boldsymbol{x}}, {}^{i}\hat{\boldsymbol{r}}_d, {}^{i}\hat{\boldsymbol{v}}_d, \boldsymbol{q}_{c,ref}\right), \\ & {}^{c}\hat{\boldsymbol{T}}_c \in \left[\underline{{}^{c}\boldsymbol{T}_c} \ \overline{{}^{c}\boldsymbol{T}_c}\right], \quad {}^{c}\hat{\boldsymbol{\tau}}_c \in \left[\underline{{}^{c}\boldsymbol{\tau}_c} \ \overline{{}^{c}\boldsymbol{\tau}_c}\right], \\ & {}^{c}\hat{\boldsymbol{\omega}}_c \in \left[\underline{{}^{c}\boldsymbol{\omega}_c} \ \overline{{}^{c}\boldsymbol{\omega}_c}\right], \ {}^{c}\hat{\dot{\boldsymbol{T}}}_c \in \left[\underline{{}^{c}\dot{\boldsymbol{T}}_c} \ \overline{{}^{c}\dot{\boldsymbol{T}}_c}\right], \ {}^{c}\hat{\dot{\boldsymbol{\tau}}}_c \in \left[\underline{{}^{c}\dot{\boldsymbol{\tau}}_c} \ \overline{{}^{c}\dot{\boldsymbol{\tau}}_c}\right], \end{aligned} \tag{10}$$

where the weighted norms $\|\cdot\|_Q$ and $\|\cdot\|_R$ act as mechanisms to prioritize between state variables and control actions through the weights included in the diagonal matrices Q and R. The formulation moreover embeds physical constraints under the form of minimum and maximum admissible values on the components of the state ${}^{c}\boldsymbol{T}_c$, ${}^{c}\boldsymbol{\tau}_c$ and ${}^{c}\boldsymbol{\omega}_c$, and on the inputs ${}^{c}\dot{\boldsymbol{T}}_c$ and${}^{c}\dot{\boldsymbol{\tau}}_c$.

To numerically solve this problem we adopt MATMPC, a MatLab based toolbox for real-time Nonlinear Model Predictive Control (Chen et al., 2019). MATMPC is written in MatLab C API with a MatLab simulation interface. It has been designed to facilitate the controller design while maintaining a friendly user experience and efficient implementation in terms of computational burden. MATMPC supports several algorithms that can be easily interchanged to adapt to different applications. It exploits direct multiple shooting to discretize the continuous optimal control problem (10) into a Nonlinear Programming Problem (NLP), which is solved with sequential quadratic programming (SQP). The NLP problem is built with a discrete version of the model dynamics into N subintervals over the prediction horizon, obtained through a 4th order Runge-Kutta integrator by using the simplified set of Equations of Motion (EOM), as explained in Section 2.3. For completeness, we report that we used the tool HPIPM (Frison et al., 2014) as QP solver in MATMPC. Note that MATMPC requires continuous time model equations and takes advantage of efficient implementations of numerical differentiation and integration to speed up the computational time. The actual MPC step optimizes the NLP discrete problem computing discretized inputs and outputs, thus resulting in a discrete control scheme.

For the specific application considered in this paper, being able of balancing between the optimizer accuracy and computational time is critical. More specifically, the possibility of trading off a long prediction horizon length N (that would make the optimally computed trajectory appealing from a rendezvous perspective) against the accuracy of the

computed solution (that is expected to be lower specially after initializing the procedure and after impulsive, sudden and non-expected disturbances affect the system) is a crucial aspect to make the proposed control strategy effective. The solution to the optimal control problem at a given step k, that is, at time $t = k\tau$ where $1/\tau$ denotes the frequency at which the control inputs are applied, consists of the dicretized optimal values $\hat{\boldsymbol{u}}_0, \ldots, \hat{\boldsymbol{u}}_{N-1}$. According to the receding horizon principle, the input value $\boldsymbol{u}(t) = \hat{\boldsymbol{u}}_0$ is applied for $t \in [k\tau, (k+1)\tau)$, and the procedure is repeated again at the subsequent time step $k+1$.

Importantly, the optimization problem in (10) is characterized by a dynamics $f(\cdot,\cdot)$ that is highly non-linear. Combining this with the necessity of considering long prediction horizon due to the long-time scales that characterize the low-thrust rendezvous problem as hinted in Section 3.1 generally translates into high computational efforts to numerically find the optimal actuation reference. It appears then evident that a computationally efficient solution plays a key role for making the proposed technology applicable in real life conditions.

4. SIMULATION SETUP

To check the viability and assess the performance of the MPC strategy within our framework we propose to use the simulation & control scheme schematized in Figure 2.

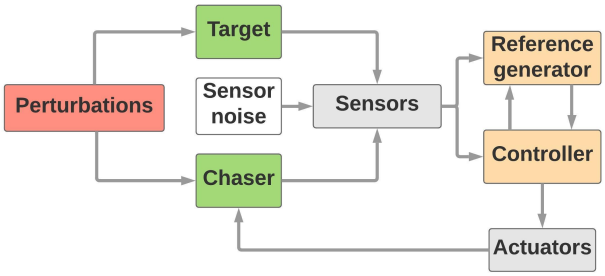

Fig. 2. Block diagram schematizing the simulation setup used to evaluate the implemented MPC strategy.

More specifically, the *Chaser* and *Target* blocks in Figure 2 perform for each time interval the integration of the EOM of the (tumbling) debris and of the controlled chaser. The integration is performed in a MatLab environment and takes into account the environmental actions described in Section 2.1.

Beyond these disturbances, the dynamics of the chaser are also driven by the control inputs actions provided by the *Actuators* block. This block embeds into the control reference signals the uncertainties highlighted in Section 2.3. These reference signals are in their turn computed by the *Controller* block, whose purpose is to numerically solve the optimization problem defined in (10). Note that this block is the one actually implementing the NMPC algorithm.

In parallel with the *Controller* block, the *Reference generator* block predicts the future positions and velocities of the target within the prediction horizon, generating in this way the ideal reference trajectory that the chaser should follow.

The *Sensors* block is instead responsible to add the measurement noise generated in the *Sensor noise* sub-block to the simulation scheme, something that is essential to verify that the control scheme is robust not only w.r.t. process noises but also the errors introduced by the sensors. More specifically, the effect of this block is to transform the real state vector by adding in general a white heteroskedastic noise as in Equation (1). Important exceptions to this general rule are the following:

- The errors on the measurements of the attitude of the chaser are simulated by adding to the actual attitude a zero-mean random rotation ϕ_{me} around a zero-mean random axis $\hat{\boldsymbol{j}}_{me}$. In formulas, we thus generate a quaternion $\boldsymbol{q}_{me}$ defined as

$$\boldsymbol{q}_{me} = \left[\cos\left(\frac{\phi_{me}}{2}\right) \quad \hat{\boldsymbol{j}}_{me}^T \sin\left(\frac{\phi_{me}}{2}\right)\right]^T. \tag{11}$$

and then use, as the attitude measurement, the vector

$$\boldsymbol{q}_m = \boldsymbol{q}_c \otimes \boldsymbol{q}_{me}. \tag{12}$$

- The chaser is assumed to be equipped with a GPS-receiver, and measurements of its own position and velocity states are therefore made by adding Gaussian noise of constant standard deviations $\sigma_{\boldsymbol{r}_c}$ and $\sigma_{\boldsymbol{v}_c}$ to the respective states.

- The noise on the measurements of the position and velocity of the debris are generated differently depending on the considered phase of the mission (i.e., Long Range (LR) or Short Range (SR)). This choice is due to the fact that it is likely that in the LR phase the chaser uses external measurements of the target's motion (e.g., information coming from Earth stations). In this case we use thus constant Gaussian distributions with standard deviations $\sigma_{\boldsymbol{r}_d}$ and $\sigma_{\boldsymbol{v}_d}$ respectively. In the SR phase, instead, we assume that the chaser is performing on-board measurements of the relative position of the target, e.g., using camera-based measurements. It is in this specific case that we add a heteroskedastic noise as in (1). I.e., we let the along-range-error have a per-unit standard deviation $\overline{\sigma}_{r_t}$ and the transverse-error (defined as the uncertainty on the plane perpendicular to the range) have a per-unit standard deviation $\overline{\sigma}_{t_t}$. We moreover let the measurements errors for the velocity follow the same structure, with per-unit standard deviations for the along-range and for the transverse-errors be respectively $\overline{\sigma}_{vr_t}$ and $\overline{\sigma}_{vt_t}$.

The following tables (1, 2 and 3) complete the description of our simulations setup by describing which parameters we have been actually using in our in-silico experiments.

Measurement noise parameters

$\sigma_{\phi_{me}}$	1 deg
$\sigma_{\boldsymbol{\omega}_c}$	1 % of $\boldsymbol{\omega}_c$
$\sigma_{\boldsymbol{r}_c}$	1 m
$\sigma_{\boldsymbol{v}_c}$	1 mm/s
$\sigma_{\boldsymbol{r}_d}$	10 m
$\sigma_{\boldsymbol{v}_d}$	10 mm/s
$\overline{\sigma}_{r_t}$	1 % of r_t
$\overline{\sigma}_{t_t}$	0.01 % of r_t
$\overline{\sigma}_{vr_t}$	0.001 % of r_t
$\overline{\sigma}_{vt_t}$	0.000001 % r_t

Table 1. List of the variances and parameters used to simulate the measurement noises.

Control and state bounds

	Min Value	Max value
${}^c\boldsymbol{T}_c^T$ $[\mu N]$	-30	+30
${}^c\dot{\boldsymbol{T}}_c^T$ $[\mu N/s]$	-10	+10
${}^c\boldsymbol{\tau}_c^T$ $[\mu Nm]$	-0.05	+0.05
${}^c\dot{\boldsymbol{\tau}}_c^T$ $[\mu Nm/s]$	-0.01	+0.01
${}^c\boldsymbol{\omega}_c^T$ [mrad/s]	-5	+5

Table 2. List of the constraints imposed on the state and actuation signals considered in the MPC optimization problem.

Cost function weights

	Long Range	Short Range
${}^l d_{c,x}$ [Mm]	7e-5	10
${}^l d_{c,y}$ [Mm]	7e-5	10
${}^l d_{c,z}$ [Mm]	70	100
${}^l \dot{d}_{c,x}$ [km/s]	7e-5	10
${}^l \dot{d}_{c,y}$ [km/s]	7e-5	10
${}^l \dot{d}_{c,z}$ [km/s]	70	100
$\bar{q}_{err,1}$	0.05	0.5
$\bar{q}_{err,2}$	0.05	0.5
$\bar{q}_{err,3}$	0.05	0.5
${}^c\dot{T}_{c,x}$ [mN/s]	5,000	500
${}^c\dot{T}_{c,y}$ [mN/s]	5,000	500
${}^c\dot{T}_{c,z}$ [mN/s]	5,000	500
${}^c\dot{\tau}_{c,x}$ [μNm/s]	6,000	6,000
${}^c\dot{\tau}_{c,y}$ [μNm/s]	6,000	6,000
${}^c\dot{\tau}_{c,z}$ [μNm/s]	6,000	6,000

Table 3. List of the parameters defining the cost function used in the MPC formulation.

5. EXPERIMENTAL RESULTS

To test the controller's capabilities we design a mission where the objective is to rendezvous and dock with an uncontrolled piece of debris flying in a circular Low Earth Orbit at an altitude of 300 km and inclination of 30 degrees. The chaser start with the same orbital parameters as the target, but with an anomaly difference of -10 degrees, putting the initial distance between the two objects at 1164 km, and a set of initial states as summarized in Table 4. The chaser is a 3U CubeSat with a mass of 4 kg and the three principal moments of inertia are $I_x = 0.0333 kgm^2$, $I_y = 0.0067 kgm^2$ and $I_z = 0.0333 kgm^2$.

Initial states - Long Range

	Chaser	Target
x-pos [km]	6576	6678
y-pos [km]	1004	0
z-pos [km]	579	0
x-vel [km/s]	1.341	0
y-vel [km/s]	6.589	6.690
z-vel [km/s]	3.804	3.862
$\boldsymbol{q_4}$	1	-
$\boldsymbol{q_1}$	0	-
$\boldsymbol{q_2}$	0	-
$\boldsymbol{q_3}$	0	-
$\boldsymbol{\omega_x}$ [rad/s]	0	-
$\boldsymbol{\omega_y}$ [rad/s]	0	-
$\boldsymbol{\omega_z}$ [rad/s]	0	-

Table 4. Initial states of the chaser and target in the Long Range scenario.

As for the description of the mission, as said above it starts in the Long Range (LR) phase, whose aim is primarily to perform orbital change maneuvers, while we recall that the second phase, the Short Range (SR) one, is focused on the approaching-for-docking maneuvers. The two phases are simulated with sampling rates of 0.01 and 0.1 Hz, respectively.

Starting by considering the LR phase, we note that the performance of the maneuver are dependent to the predictions time horizon, since the latter influences the control action decided by the control scheme. This implies the existence of a trade-off between the fuel consumption and mission time versus the computational effort required for solving the model-predictive optimization problem. To decide the best choice our methodological approach is to consider simulations of the system, and then select the best parameters by graphically inspecting the associated results (here shown in Figure 3 and Figure 4).

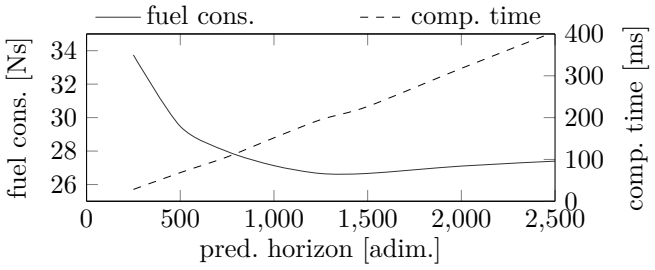

Fig. 3. Fuel consumption and average computation time per sample, as a function of the number of prediction intervals.

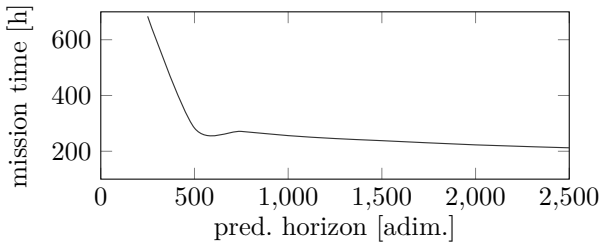

Fig. 4. Mission time as a function of the number of prediction intervals.

Figure 3 shows that –as expected– the computational effort increases nearly linearly with the prediction horizon, while the time taken to reach the target generally decreases. The fuel consumption is the lowest around $N = 1250$, indicating the existence of a "sweet spot/global minimum" with regards to fuel efficiency when choosing the length of the prediction horizon. These optimal values (prediction horizon and fuel consumption) are obviously obtained with a specific set of parameters, and are in general a function of the considered scenario. Figures 3 and 4 should thus be intended as an illustration of our methodological approach.

We then assess the robustness of the proposed MPC strategy against the noisiness of the sources discussed in Section 2.3. Figures 5 and 6 show then how fuel consumption and mission time is affected when the standard deviations presented in Table 1 are multiplied with a constant factor of various magnitudes (with the prediction horizon set at

$N = 1000$). Moreover, to account for different modeling errors and actuation uncertainties, we artificially increased I_x by 5%, decreased I_y and I_z by 5%, and decreased the mass by 50 grams in the chaser model.

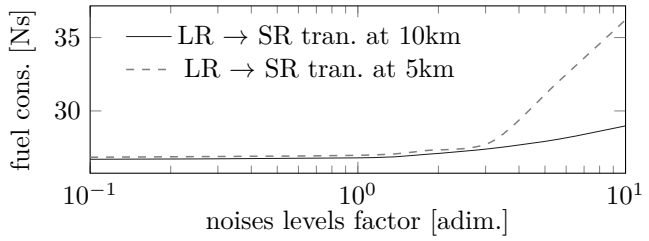

Fig. 5. Fuel consumption as a function of sensor noise level.

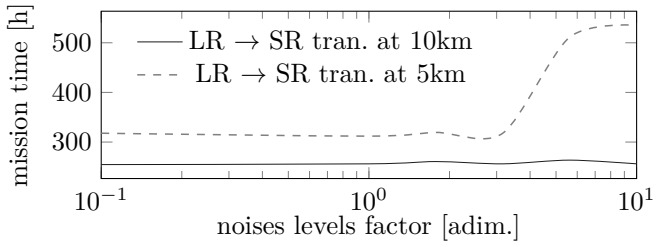

Fig. 6. Mission time as a function of sensor noise level.

Figures 5 and 6 show the results for two possible choices of the distance governing the LR to SR transition phase. Clearly, the performance indexes "fuel consumption" and "mission time" seem to be not too heavily affected by the noise levels (indicating a good noises rejection capability for the overall scheme) for low noise levels. Smaller LR to SR transition distances, however, are less resilient to big noises levels. Our interpretation is that when the transition distances are small but the noises are high, then the system has more difficulties in entering in the SR phase (i.e., reaches its border but orbits around it for longer time).

The final test case is therefore aimed at showing how the choice of transition point between the LR and SR phases affect the overall performance of the rendezvous maneuver in terms of total fuel consumption and total mission time. We then compare the effect of choosing different transition distances while using a noise level factor of $10^{0.75}$ and a prediction horizon $N = 1000$ to obtain Figure 7.

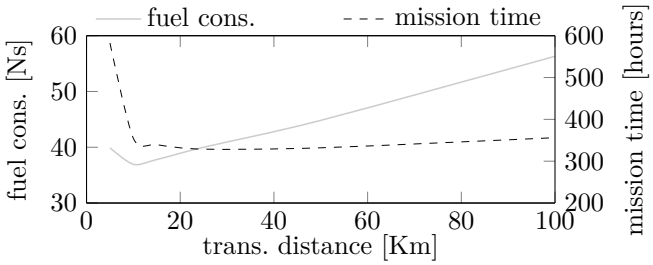

Fig. 7. Total fuel consumption and mission time as a function of the LR → SR transition distance.

Importantly, the results indicate that both mission time and fuel consumption may increase dramatically if the transition distance is not chosen carefully. Obviously the optimal parameter in general depends on many factors, so the indications from Figure 7 shall be considered only as methodological.

6. CONCLUSIONS

This paper shows how MATMPC can be used to implement a non-linear MPC strategy for performing a rendezvous maneuver between a small controllable spacecraft and an uncontrollable piece of debris. The computational efficiency of MATMPC allows to generate maneuver strategies that account for inherently low thrust values of small spacecrafts through enabling embedding long prediction horizons in the model predictive optimization problem. Promising fuel efficiency and noise rejection capabilities of the proposed scheme have been demonstrated through simulations. Future studies may include a more detailed modeling of the actuators and a closer examination of the docking phase.

REFERENCES

Yutao Chen, Mattia Bruschetta, Enrico Picotti, and Alessandro Beghi. Matmpc-a matlab based toolbox for real-time nonlinear model predictive control. In *2019 18th European Control Conference (ECC)*, pages 3365–3370. IEEE, 2019.

Howard D. Curtis. *Orbital Mechanics for Engineering Students (Third Edition)*. Butterworth-Heinemann, Boston, third edition edition, 2014.

Leonard Felicetti, Paolo Gasbarri, Andrea Pisculli, Marco Sabatini, and Giovanni B. Palmerini. Design of robotic manipulators for orbit removal of spent launchers' stages. *Acta Astronautica*, 119:118 – 130, 2016.

G. Frison, H. H. B. Sørensen, B. Dammann, and J. B. Jørgensen. High-performance small-scale solvers for linear model predictive control. In *2014 European Control Conference (ECC)*, pages 128–133, June 2014.

Alexander Korsfeldt Larsén. Non-linear model predictive control for space debris removal missions. Master's thesis, Luleå University of Technology, Space Technology, 2018.

Ben Larbi, Grzesik M. K., Radtke B., C. J., Trentlage, and E. Stoll. Active debris removal for mega constellations: Cubesat possible? *Proceedings of the 9th International Workshop on Satellite Constellations and Formation Flying IWSCFF2017, Boulder, Colorado*, 2017.

J.-C. Liou. An active debris removal parametric study for leo environment remediation. *Advances in Space Research*, 47(11):1865 – 1876, 2011.

J.-C. Liou, N.L. Johnson, and N.M. Hill. Controlling the growth of future leo debris populations with active debris removal. *Acta Astronautica*, 66(5):648 – 653, 2010.

R. Lucken, N. Hubert, and D. Giolito. Systematic space debris collection using cubesat constellation. *Proceedings of the 7th European Conference for Aeronautics and Aerospace Sciences (EUCASS), Milan, Italy*, 2017.

Hanspeter Schaub, Lee E.Z. Jasper, Paul V. Anderson, and Darren S. McKnight. Cost and risk assessment for spacecraft operation decisions caused by the space debris environment. *Acta Astronautica*, 113:66 – 79, 2015.

Minghe Shan, Jian Guo, and Eberhard Gill. Review and comparison of active space debris capturing and removal

methods. *Progress in Aerospace Sciences*, 80:18 – 32, 2016.

B. Udrea and M. Nayak. A cooperative multi-satellite mission for controlled active debris removal from low earth orbit. In *2015 IEEE Aerospace Conference*, pages 1–15, March 2015.