Model predictive control for rendezvous hovering phases based on a novel description of constrained trajectories

Paulo Ricardo Arantes Gilz[1], Mioara Joldes[1], Christophe Louembet[1], and Frédéric Camps[1]

[1]LAAS, Université de Toulouse, CNRS, Toulouse, France
[1](e-mail: prarante@laas, joldes@laas, louembet@laas, fcamps@laas)

***Abstract*—The present article proposes a predictive control law for orbital spacecraft rendezvous hovering phases. An innovative description of periodic space-restricted trajectories based on computing the envelope of a family of curves is given. This description is used to provide a model predictive controller able to minimize the fuel consumption and account convex constraints, such as periodicity, saturation of the thrusters and space restrictions. The efficiency of this control algorithm is assessed on a board simulating the performance of devices usually employed in space applications.**

***Keywords*—FPGA embedded control law; Model predictive and optimization-based control; Convex optimization; Guidance, navigation and control of vehicles; Aerospace**

1 Introduction

Orbital rendezvous missions are composed of several phases. Between each of these phases, the chaser spacecraft must keep its station while waiting for the order to proceed to the next step of the mission. These station keeping phases consist in steering and maintaining the chaser spacecraft inside a restricted zone defined in the body-fixed frame of the leader spacecraft. These are the so-called "hovering" phases (see [16]). In this work, the hovering phases are controlled by means of chemical thrusters. These control actions must be performed in a fuel-optimal way, while also accounting for several other constraints (thrusters saturation, fuel budget, visibility cone, collision avoidance, etc). Classical hovering control strategies like "teardrop" or "pogo" techniques are fuel-inefficient. A recent approach of [6] for impulsive control was developed in the hybrid framework. However, this does not take into account the same constraints as the current work, in particular the thrusters' saturation.

To address these drawbacks, we choose to adopt a model predictive control (MPC) strategy to achieve the hovering. Such an approach has already been successfully applied in the rendezvous problem (see [5] and references therein).

In literature, the rendezvous problem subject to constraints has first been formulated as Linear/Quadratic Programs (LP/QP) by discretizing the constraints (see [22, 5, 14]). However, this class of problems has a bad scaling behaviour in function of the discretization level along with systematic violation of the original constraints. A different approach of [9, 3], which overcomes the disadvantages induced by discretization, is based on Semi-Definite Programs (SDP). Nevertheless, the algorithms employed in the resolution of SDP problems present high computational burden.

Even though previous contributions were too numerically expensive to be embedded, recent works developed by [13] have presented the possibility of efficiently embedding and testing MPC algorithms on FPGAs by executing hardware-in-the-loop simulations and comparing the performance of a LP/QP-based MPC to state-of-the-art implementations.

The present work extends the research of a novel method to solve the problem, aiming to obtain an MPC algorithm that is not only capable of accounting for the exigences of the rendezvous missions, but also sufficiently efficient to be executed on a board that simulates the performance of the microprocessors and devices usually employed in space applications.

Our approach consists in providing a redefinition of the set of periodic space-constrained trajectories by evaluating convex semi-algebraic functions (computed by finding the envelope of the curves defining the boundary of the so-called admissible set) that can be used to indicate whether a given relative trajectory respects the restrictions or not (section III). This description is then embedded in the formulation of a MPC scheme characterized by the resolution of convex but non-differentiable constrained optimization problems (section IV).

The obtained controller is then coded in C and executed on a board containing a FPGA-synthesized LEON3 microprocessor that simulates the performance of devices usually employed in space applications. Bench-tests are performed against a LP method and hardware-in-the-loop simulations are performed in order to assess the efficiency and robustness of the algorithm (section V).

2 Modeling the problem

The main objective of this work is to propose a new model predictive controller for the hovering phase: the chaser is guided to and maintained in a predefined region, while minimizing the consumption and accounting for thrusters limitations. Such an objective is achieved by steering the chaser spacecraft from a given initial state to any periodic orbit included in the given subspace of interest, so-called tolerance box. Hereafter, we present the model that is required in the development of our MPC controller and that describes the relative motion of the two spacecraft orbiting the Earth and the actuators. Then, the constraints on the rendezvous problem are formulated.

2.1 Relative spacecraft motion

Fig. 1 depicts the frames used to model the relative motion between the leader S_l and the follower S_f spacecraft. The Earth-Centered Inertial (ECI) frame is given by $\left\{O, \overrightarrow{I}, \overrightarrow{J}, \overrightarrow{K}\right\}$. The moving Local Vertical / Local Horizontal (LVLH) frame is centered on the leader spacecraft at S_l and given by $\{S_l, \overrightarrow{x}, \overrightarrow{y}, \overrightarrow{z}\}$ (see [11] for details). Under Keplerian assumptions, the leader orbit is mainly described by its semi-major axis a and its eccentricity $0 < e < 1$. Then, the leader is located on its orbit by the true anomaly ν.

The relative motion is defined as the time history of the relative vector $\overrightarrow{S_l S_f}$. This motion has a state space representation with the

This work was supported by the FastRelax (ANR-14-CE25-0018-01) project of the French National Agency for Research (ANR).

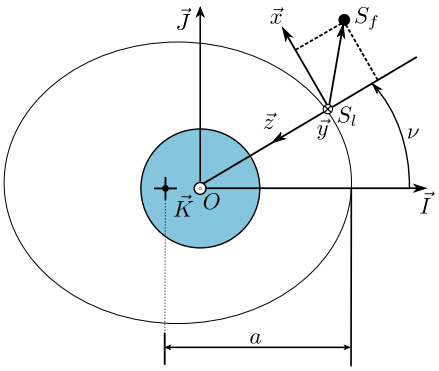

Figure 1: Inertial and relative frames.

following state vector: $X(t) = [x(t),\ y(t),\ z(t),\ \dot{x}(t),\ \dot{y}(t),\ \dot{z}(t)]^T$. Assuming relative navigation hypothesis, $\|\overrightarrow{S_l S_f}\| \ll \|\overrightarrow{OS_l}\|$, [23] provides a linearized state dynamic equation:

$$\dot{X}(t) = A(t)X(t) \tag{1}$$

By performing the state similar transformation

$$\tilde{X}(\nu) = \underbrace{\begin{bmatrix} (1+e\cos\nu)\mathbb{I}_3 & \mathbb{O}_3 \\ -e\sin\nu\,\mathbb{I}_3 & \sqrt{\frac{a^3(1-e^2)^3}{\mu(1+e\cos\nu)^2}}\mathbb{I}_3 \end{bmatrix}}_{T(\nu)} X(t) \tag{2}$$

where $\tilde{X}(\nu) = [\tilde{x}(\nu),\ \tilde{y}(\nu),\ \tilde{z}(\nu),\ \tilde{x}'(\nu),\ \tilde{y}'(\nu),\ \tilde{z}'(\nu)]^T$ and μ is Earth's gravitational constant, the relative unforced dynamics are modeled by the simplified Tschauner-Hempel equations:

$$\tilde{x}'' = 2\tilde{z}', \quad \tilde{y}'' = -\tilde{y}', \quad \tilde{z}'' = \tfrac{3}{1+e\cos\nu}\tilde{z} - 2\tilde{x}' \tag{3}$$

where $(\cdot)' = \frac{d(\cdot)}{d\nu}$ and $(\cdot)'' = \frac{d^2(\cdot)}{d\nu^2}$.

The system (3) can be represented in state space form:

$$\tilde{X}'(\nu) = \tilde{A}(\nu)\tilde{X}(\nu) \tag{4}$$

To produce a characterization of relative orbits included in the tolerance box, another variables change is performed. The ideas behind this change of variables are roughly based on Floquet theory and fully developed in [6] and references therein. Its main advantage is that the new state provides a physical description of the relative orbits contrary to the states X and $\tilde{X}$ that only give the instantaneous position and velocity (see Deaconu [8, Chapter 2] for details). The new states are defined by [9]:

$$\begin{aligned}
d_0 &= \tfrac{3ec_\nu+e^2+2}{1-e^2}\tilde{z} + \tfrac{(1+ec_\nu)^2}{e^2-1}\tilde{x}' + \tfrac{es_\nu(1+ec_\nu)}{1-e^2}\tilde{z}' \\
d_1 &= \tfrac{3(e+c_\nu)}{e^2-1}\tilde{z} - \tfrac{2c_\nu+ec_\nu^2+e}{e^2-1}\tilde{x}' + \tfrac{s_\nu(1+ec_\nu)}{e^2-1}\tilde{z}' \\
d_2 &= \tfrac{3s_\nu(1+ec_\nu+e^2)}{(e^2-1)(1+ec_\nu)}\tilde{z} + \tfrac{s_\nu(2+ec_\nu)}{1-e^2}\tilde{x}' - \tfrac{(c_\nu+ec_\nu^2-2e)}{e^2-1}\tilde{z}' \\
d_3 &= \tilde{x} + \tfrac{-3es_\nu(2+ec_\nu)}{(e^2-1)(1+ec_\nu)}\tilde{z} + \tfrac{es_\nu(2+ec_\nu)}{e^2-1}\tilde{x}' + \tfrac{e^2c_\nu^2+ec_\nu-2}{e^2-1}\tilde{z}' \\
d_4 &= c_\nu\tilde{y} - s_\nu\tilde{y}' \\
d_5 &= s_\nu\tilde{y} + c_\nu\tilde{y}'
\end{aligned} \tag{5}$$

where $c_\nu = \cos\nu$, $s_\nu = \sin\nu$.

Relation (5) can be rewritten in a matrix-vector product form as:

$$D(\nu) = C(\nu)\tilde{X}(\nu) \tag{6}$$

where $D(\nu) = [d_0(\nu),\ d_1(\nu),\ d_2(\nu),\ d_3(\nu),\ d_4(\nu),\ d_5(\nu)]^T$ is the so-called vector of Deaconu parameters and $C(\nu) \in \mathbb{R}^{6\times6}$ (see [9]). The dynamical system representing the evolution of Deaconu parameters is:

$$D'(\nu) = \underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -\frac{3e}{(1+e\,c_\nu)^2} & 0 & 0 & 0 & 0 & 0 \\ \frac{3}{(1+e\,c_\nu)^2} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}}_{A_D(\nu)} D(\nu). \tag{7}$$

The behavior of this dynamical system can also be described by its transition matrix:

$$D(\nu) = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ -3eJ_{\nu_0}(\nu) & 0 & 1 & 0 & 0 & 0 \\ 3J_{\nu_0}(\nu) & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}}_{\Phi_{\nu_0}^{\nu}} D(\nu_0) \tag{8}$$

where $J_{\nu_0}(\nu)$ is given by:

$$J_{\nu_0}(\nu) := \int_{\nu_0}^{\nu} \frac{d\tau}{(1+e\cos\tau)^2} = \sqrt{\frac{\mu}{a^3}}\frac{t-t_0}{(1-e^2)^{3/2}}. \tag{9}$$

2.2 Impulsive control

The control algorithm must perform the maneuvers while minimizing fuel-consumption and accounting for the thrusters limitations. Due to the chemical nature of the propellers, the actions performed by the actuators are modeled as impulsive velocity corrections. Given a true anomaly value ν,

$$\Delta V(\nu) = [\Delta V_x,\ \Delta V_y,\ \Delta V_z] \in \mathbb{R}^3,$$

denotes the instantaneous impulse control at ν. For sake of brevity, the impulse control at ν_i is noted ΔV_i and ΔV is the vector of all impulses such that:

$$\Delta V = \left[\Delta V_1{}^T, \ldots, \Delta V_N{}^T\right]^T. \tag{10}$$

Thus, the state right after an impulse, $X^+(\nu_i)$ can be computed by:

$$X^+(\nu_i) = X(\nu_i) + B\,\Delta V_i,\ B = [0_3\ \mathbb{I}_3]^T. \tag{11}$$

Equation (11) can be rewritten in function of D by performing the similar transformation $X \overset{T}{\to} \tilde{X} \overset{C}{\to} D$, which results in:

$$D^+(\nu_i) = D(\nu_i) + B_D(\nu_i)\Delta V_i, \tag{12}$$

where $B_D(\nu_i) = T(\nu_i)C(\nu_i)B$.

Using the representation given in (8), the vector of parameters obtained after a sequence of impulses applied at $\nu_1, \nu_2\ \ldots\ \nu_{N-1}, \nu_N$ can be expressed as follows:

$$D^+(\nu_N) = \Phi_{\nu_1}^{\nu_N} D(\nu_1) + \sum_{i=1}^{N} \Phi_{\nu_i}^{\nu_N} B_D(\nu_i)\Delta V_i. \tag{13}$$

2.3 Optimal Control Problem description

The control algorithm has to minimize a given cost while satisfying the following constraints:

2.3.1 Consumption cost

The ergols consumption is related to the norm of the impulsive control [20]. Depending on the thrusters configuration on the spacecraft body, the definition of the norm is set. In this study, the chaser is empowered by 6 thrusters mounted along each axis. Thus, the ΔV consumption is given by

$$J(\Delta V) = \sum_{i=1}^{N} \|\Delta V_i\|_1 = \sum_{i=1}^{N} |\Delta V_{i,x}| + |\Delta V_{i,y}| + |\Delta V_{i,z}| \quad (14)$$

2.3.2 Thrusters saturation

Considering that the chaser spacecraft has a pair of equivalent propellers symmetrically and oppositely disposed on each axis and assuming that the saturation limit for each propeller is $\overline{\Delta V} > 0$, the saturation constraint can be written as:

$$|\Delta V_{i,x}| \leq \overline{\Delta V}, \quad |\Delta V_{i,y}| \leq \overline{\Delta V}, \quad |\Delta V_{i,z}| \leq \overline{\Delta V}. \quad (15)$$

2.3.3 Periodicity

Orbits periodicity guarantees the relative motion to remain bounded in absence of disturbances. Since the relative trajectories described by (3) are not generally periodic, by imposing $\tilde{X}(\nu + 2\pi) = \tilde{X}(\nu),\ \forall \nu$, a necessary and sufficient periodicity condition was found in [7]: $d_0(\nu) = 0$.

A notable property of periodic trajectories is that the vector of parameters $D(\nu)$ remains constant for any value of ν since its dynamic matrix $A_D(\nu)$ has non-zero values only in its first column (see (7)) and, in this case, $D(\nu)$ is as a vector of constant values $D = [d_0{=}0,\ d_1,\ d_2,\ d_3,\ d_4,\ d_5]^T$.

In our approach, the periodicity is enforced at the end of the steering manoeuvre: $d_0^+(\nu_N) = 0$.

2.3.4 Space constraints

During the rendezvous hovering phases (or station keeping), the follower is required to remain in the interior of a certain limited region of the space. Without loss of generality, this tolerance box is assumed to be a rectangular cuboid:

$$\underline{x} \leq x(t) \leq \overline{x}, \quad \underline{y} \leq y(t) \leq \overline{y}, \quad \underline{z} \leq z(t) \leq \overline{z}\ ,\ \forall t \geq t_0 \quad (16)$$

By imposing $d_0 = 0$ (periodicity) and changing the variables $\tilde{x},\ \tilde{y},\ \tilde{z}$ back to $x,\ y,\ z$ via (2) and (6), the inequalities in (16) are rewritten as:

$$\underline{x} \leq M_x(\nu) D \leq \overline{x}, \quad \underline{y} \leq M_y(\nu) D \leq \overline{y}, \quad \underline{z} \leq M_z(\nu) D \leq \overline{z}, \quad \forall \nu, \quad (17)$$

where:

$$\begin{array}{ll} M_x(\nu) = [\ 0, & \frac{(2+e\,c_\nu)s_\nu}{1+e\,c_\nu}, \quad \frac{-(2+e\,c_\nu)c_\nu}{1+e\,c_\nu}, \quad \frac{1}{1+e\,c_\nu}, \quad 0, \quad 0 \quad] \\ M_y(\nu) = [\ 0, & 0, \quad 0, \quad 0, \quad \frac{c_\nu}{1+e\,c_\nu}, \quad \frac{s_\nu}{1+e\,c_\nu} \quad] \\ M_z(\nu) = [\ 0, & c_\nu, \quad s_\nu, \quad 0, \quad 0, \quad 0 \quad] \end{array} \quad (18)$$

Hence the admissible set, i.e. the set of periodic trajectories respecting the inequalities in (17), is given by the following definition using (18):

$$S_D^p := \left\{ D \in \mathbb{R}^6 \;\middle|\; d_0 = 0, \begin{array}{l} \underline{x} \leq M_x(\nu) D \leq \overline{x} \\ \underline{y} \leq M_y(\nu) D \leq \overline{y} \\ \underline{z} \leq M_z(\nu) D \leq \overline{z} \end{array} , \forall \nu \right\} \quad (19)$$

that hereafter is supposed to be non-empty.

Problem 1 *The problem of guiding the chaser from any initial position and velocity to any stable relative orbit included in the user-defined tolerance box can be formulated as the following optimal control problem:*

$$\min_{\Delta V \in [-\overline{\Delta V}, \overline{\Delta V}]^{3N}} J(\Delta V) \quad s.t. \begin{cases} D(\nu_1) = D_1, \\ D^+(\nu_N) \in S_D^p. \end{cases} \quad \text{(Pb1)}$$

Problem (Pb1) has been addressed in [9] by parametrizing the set S_D^p on the cone of the positive semi-definite matrices so that the numerical characterization of admissible orbits rely on SDP solvers. However, this class of solvers are cumbersome and slow in resource-constrained computation environments such as space designed devices. So far, the solution for embedding this kind of algorithm is the discrete LP/QP approach that has been implemented on a FPGA by [13].

We developed a algorithm based on a different characterization of the admissible set S_D^p that leads to the formulation of another optimization problem solvable by on-board tractable methods. Its aim is to be more efficient than the algorithm from [9] while providing the same desirable properties.

3 New admissible set description

The set of periodic and tolerance-box-included orbits is described in (19) by infinitely many constraints. Our contribution is to give a description of the admissible set different from [9] that only depends on the evaluation of closed-form expressions on the entries of the vector D and on the space constraints. The objective of this new description is to provide a formulation that guarantees the non violation of the constraints and enables an efficient computation on spacecraft compatible devices.

Let first denote by $\gamma_w(\cdot) \leq 0$ each inequality of (17) where the index $w = \{\underline{x}, \overline{x}, \underline{y}, \overline{y}, \underline{z}, \overline{z}\}$. One can remark that these inequalities describe a family of surfaces (or lines) parametrized by ν. Moreover, the boundary of the set of points respecting each of these inequalities is included in the envelope of the associated family of surfaces.

Let $\gamma(\alpha_1, \ldots, \alpha_L, \nu) = 0$ be a family of one-parameter surfaces, with $L \in \mathbb{N}, L \leq 3$ depending on the parameter ν. Its envelope is the subset of points $(\alpha_1, \ldots, \alpha_L) \subseteq \mathbb{R}^L$ for which the following system of equations is satisfied:

$$\gamma(\alpha_1, \ldots, \alpha_L, \nu) = \frac{\partial \gamma}{\partial \nu}(\alpha_1, \ldots, \alpha_L, \nu) = 0. \quad (20)$$

For each constraint $w = \{\underline{x}, \overline{x}, \underline{y}, \overline{y}, \underline{z}, \overline{z}\}$, the resolution of (20) provides an implicit equation $g_w(D) = 0$ for which the set of solutions contains a surface in space that separates the points that respect the constraint inequality from those that do not. With this, a verification method is obtained for checking that the vector D belongs to S_D^p:

$$D \in S_D^p \Leftrightarrow (d_0 = 0 \text{ and } \forall w,\ g_w(D) \leq 0).$$

Envelope $g_w(\cdot)$ expression on y and z axis: after solving (20) for indices $w = \{\underline{y}, \overline{y}, \underline{z}, \overline{z}\}$ and using the Sylvester's matrix implicitization method (see [15]), the following expressions are obtained:

$$\begin{aligned} g_{\underline{y}}(d_4, d_5) &= (d_4 - e\underline{y})^2 + d_5^2 - \underline{y}^2 \\ g_{\overline{y}}(d_4, d_5) &= (d_4 - e\overline{y})^2 + d_5^2 - \overline{y}^2 \end{aligned} \tag{21}$$

$$\begin{aligned} g_{\underline{z}}(d_1, d_2) &= d_1^2 + d_2^2 - \underline{z}^2 \\ g_{\overline{z}}(d_1, d_2) &= d_1^2 + d_2^2 - \overline{z}^2 \end{aligned} \tag{22}$$

Note that these equations (21) and (22) describe circles in the (d_4, d_5) plane and (d_1, d_2) plane respectively. For instance the frontier of the admissible set for the constraint $\gamma_{\overline{y}}$ is a circle of center $(e.\overline{y}, 0)$ and radius $\overline{y}$.

Envelope $g_w(\cdot)$ expression on x axis: finding an implicit function for x constraints requires further work. After solving the system of equations from (20), a parametric description is obtained in function of ν, d_3 and $\underline{x}$ or $\overline{x}$:

$$\begin{aligned} d_1 &= \frac{((2x_m - 2d_3)(1 + ec_\nu) + e^2 x_m c_\nu^2) s_\nu}{(2 + ec_\nu)^2} \\ d_2 &= -\frac{((2x_m - 2d_3)(1 + ec_\nu) + e^2 x_m c_\nu^2) c_\nu + e(d_3 + x_m)}{(2 + ec_\nu)^2} \end{aligned} \tag{23}$$

where x_m stands for $\underline{x}$ or $\overline{x}$.

Using the Sylvester's matrix implicitization method, equations (23) produce a multivariate polynomial in d_1, d_2, d_3:

$$\hat{g}_{x_m}(d_1, d_2, d_3) = \sum_{\xi \in \mathbb{N}^3} \theta_\xi d_1^{\xi_1} d_2^{\xi_2} d_3^{\xi_3} \tag{24}$$

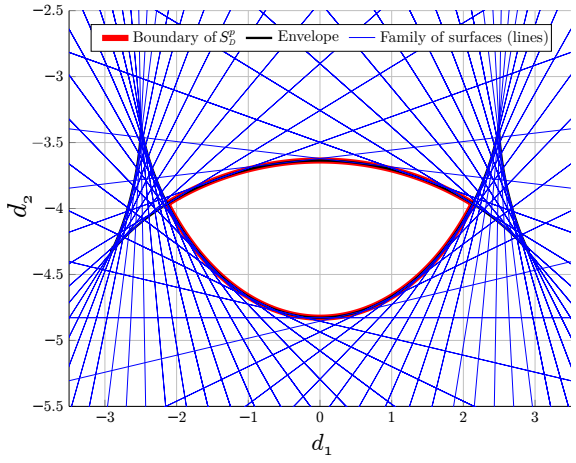

Figure 2: Family of surfaces, envelope and boundary of admissible set for $\hat{g}_{\overline{x}} = 0$, with $\overline{x} = 10, e = 0.9, d_3 = 5$.

As shown in Fig. 2, the envelope $g_{\overline{x}} = 0$ contains the boundary of the inner-convex set of points respecting the inequality $\overline{x}$ for all ν. We can reduce this set of solutions by remarking that the multivariate polynomial (24) can be seen as a fourth degree polynomial in d_3:

$$\hat{g}_{x_m}(d_1, d_2, d_3) = \sum_{0 \leq i \leq 4} \bar{\theta}_i(d_1, d_2)\, d_3^i \ . \tag{25}$$

Hence to perform the set reduction it suffices to choose the root of (25) that describes the inner convex set presented in Fig. 2. Considering $\hat{g}_{\overline{x}}(\cdot)$, the smallest real root is the one to be selected. Conversely, the largest one is chosen when considering the envelope $\hat{g}_{\underline{x}}(\cdot)$. This is done by choosing piecewisely in function of the values of d_1, d_2, e, x_m the closed-form expressions for the roots of the fourth degree polynomial in d_3 (see [1]). This procedure generates the desired g_{x_m} functions that describe the set of admissible points with respect to $\gamma_{x_m} \leq 0$:

$$\begin{aligned} g_{\underline{x}}(d_1, d_2, d_3) &= r_{\underline{x}}(d_1, d_2, e) - d_3 \\ g_{\overline{x}}(d_1, d_2, d_3) &= d_3 - r_{\overline{x}}(d_1, d_2, e) \end{aligned} \tag{26}$$

where $r_{\underline{x}}(d_2, d_3, e)$ and $r_{\overline{x}}(d_2, d_3, e)$ are the functions that return respectively the greatest real root of $\hat{g}_{\underline{x}}$ and the lowest real root $\hat{g}_{\overline{x}}$ (25).

Hence the admissible set can be redefined as:

$$S_D^p = \left\{ D \in \mathbb{R}^6 \;\middle|\; \begin{array}{c} d_0 = 0, \\ g_w(D) \leq 0, \forall w \in \{\underline{x}, \overline{x}, \underline{y}, \overline{y}, \underline{z}, \overline{z}\} \end{array} \right\} \tag{27}$$

4 Model predictive control

In this section we use the new description of the admissible set (27) to build a model predictive controller based on iteratively applying a finite number of impulsive velocities corrections at arbitrarily chosen instants in order to bring the chaser satellite to an admissible relative trajectory.

For each call of the controller, we choose N equally spaced firing instants $\nu_1, \ldots, \nu_N$ so that $\nu_k - \nu_{k-1} = \Delta\nu$. Program (Pb1) can be reformulated as the following optimization problem:

$$\begin{aligned} &\min_{\Delta V \in [-\overline{\Delta V}, \overline{\Delta V}]^{3N}} J(\Delta V) \\ \text{s.t.} &\left\{ \begin{array}{l} D(\nu_1) = D_1, \quad d_0^+(\nu_N) = 0 \\ g_w(D^+(\nu_N)) \leq 0, \quad \forall w \in \{\underline{x}, \overline{x}, \underline{y}, \overline{y}, \underline{z}, \overline{z}\} \end{array} \right. \end{aligned} \tag{Pb2}$$

Since the ℓ_1-norm in $J(\cdot)$ and the computation of roots of polynomials in $g_w(\cdot)$ are non-differentiable functions, this problem is characterized by convex but non-differentiable objective function and constraints. For this class of problems, if the feasible set is non-empty, a solution can be computed by subgradient methods (see [4]).

In particular, we use the following approach to solve (Pb2): first, a *warm-start* procedure is performed, which consists in computing a sequence of impulses minimizing the quadratic criterion $\sum_{i=1}^N \|\Delta V_i\|_2^2$ and producing an arbitrarily chosen periodic admissible trajectory D^0:

$$\begin{aligned} &\Delta V^* = \underset{\Delta V \in \mathbb{R}^{3N}}{\operatorname{argmin}} \sum_{i=1}^N \|\Delta V_i\|_2^2 \\ \text{s.t.} &\left\{ \begin{array}{l} D(\nu_1) = D_1 \\ D^0 = \Phi_{\nu_1}^{\nu_N} D(\nu_1) + \sum_{j=0}^N \Phi_{\nu_j}^{\nu_N} B_D(\nu_j) \Delta V_j \end{array} \right. \end{aligned} \tag{28}$$

The solution of this QP-constrained problem is analytically computed by performing a matrix pseudo-inverse.

After that, we apply a penalty technique to generate an unconstrained optimization problem by combining the constraints and the objective function (see [18]):

$$\begin{aligned} J_p(\Delta V) = J(\Delta V) &+ \gamma_1 |d_0^+(\nu_N)| \\ &+ \gamma_2 (\max\{g_{\underline{x}}(D), 0\} + \ldots + \max\{g_{\overline{z}}(D), 0\}) \\ &+ \gamma_3 \sum_{i=1}^{N} \sum_{x,y,z} \max\{|\Delta V_{i,w}| - \overline{\Delta V}, 0\} \end{aligned} \tag{29}$$

with $\gamma_1, \gamma_2, \gamma_3 \gg 1$. The resulting problem can then be solved by nonsmooth optimization methods.

To solve this problem, we initially perform several iterations of the quasi-Newton method proposed by [17], which consists in building local quadratic approximations of the penalized function (29) and computing descent steps using the subgradients and the inverse Hessian. Then we apply the subgradient method proposed by [21], a first-order approach that is closed to the steepest decent optimization algorithm, but uses subgradients instead of gradients.

We use both algorithms in order to take advantage of their distinct strengths: the quasi-Newton algorithm benefits of a faster decrease of the penalized objective function along the iterations, while only the subgradient method has guaranteed convergence, being used to refine the approximated solution obtained in the previous step. A pseudocode representing the described strategy is given in Algorithm 1.

Input : N, $\Delta\nu$ - number of impulses and firing interval
$\underline{X}(\nu_1)$ - initial LVLH relative state
$\overline{\Delta V}$, $\underline{x}, \ldots, \overline{z}$ - saturation and space constraints
$\bar{D}$ - periodic admissible trajectory
I_{qn}, I_{sg} - number of iterations
Output: ΔV - impulsive velocity corrections

// *Warm-start* using (28):
$\Delta V^1 \leftarrow \Delta V^*$; $\Delta V^{best} \leftarrow \Delta V^1$;
// Quasi-Newton, Lewis and Overton [17, Algorithm 2.1]:
$H_1 = \mathbb{I}_{3N}$;
for $k = 1$ **to** I_{qn} **do**
  Compute sg_k a subgradient of J_p at ΔV^k;
  $\Delta V^{k+1} \leftarrow \Delta V^k - \lambda_k H_k sg_k$, where $\lambda_k > 0$ chosen by line search;
  if $J_p(\Delta V^{k+1}) < J_p(\Delta V^{best})$ **then** $\Delta V^{best} \leftarrow \Delta V^{k+1}$;
  Update H_{k+1} as positive definite matrix satisfying secant condition $H_{k+1}(sg_{k+1} - sg_k) = -\lambda_k H_k sg_k$;
end
$\Delta V^1 \leftarrow \Delta V^{best}$;
// Subgradient, Shor [21, Theorem 2.2]:
for $k = 1$ **to** I_{sg} **do**
  Compute sg_k a subgradient of J_p at ΔV^k;
  $\Delta V^{k+1} \leftarrow \Delta V^k - \frac{0.01}{k+1} \frac{sg_k}{\|sg_k\|_2}$;
  if $J_p(\Delta V^{k+1}) < J_p(\Delta V^{best})$ **then** $\Delta V^{best} \leftarrow \Delta V^{k+1}$;
end
$\Delta V \leftarrow \Delta V^{best}$;

Algorithm 1: Resolve Pb2

5 Numerical Results

In order to evaluate the efficiency of the developed algorithm we compare its performance against another algorithm based on the discretization of the space constraints. Both methods are test-benched an AEROFLEX GAISLER GR-XC6S board containing a synthesized LEON3 microprocessor (see [19] for specifications) running a Linux 2.6 environment in order to evaluate the performances of devices usually employed in space applications (see [10]). The robustness of the MPC scheme is assessed by means of hardware-in-the-loop simulations: the embedded controller dialogs with a nonlinear Matlab/Simulink simulator of the space mechanics.

5.1 Comparing against a LP algorithm

Hereafter we compare the performance of the proposed algorithm against the (LP) problem obtained by taking N_{LP} equally spaced true anomaly values in $[0, 2\pi]$ instead of imposing infinitely many constraints as in the definition of S_D^p (19). Both algorithms (Pb2) and (LP) are coded in C (LP problems are solve via [12]), uploaded to the board and executed by the LEON3 microprocessor. The studied scenario is detailed in Table 1.

$$\text{s.t.} \quad \begin{cases} \min\limits_{\Delta V \in [-\overline{\Delta V}, \overline{\Delta V}]^{3N}} J(\Delta V) \\ D(\nu_1) = D_1, \quad d_0^+(\nu_N) = 0 \\ \underline{x} \le M_x(\nu_j)\, D^+(\nu_N) \le \overline{x} \\ \underline{y} \le M_y(\nu_j)\, D^+(\nu_N) \le \overline{y}, \quad j = 1 \ldots N_{LP} \\ \underline{z} \le M_z(\nu_j)\, D^+(\nu_N) \le \overline{z} \end{cases} \tag{LP}$$

Table 1: Scenario

$a = 7011$ km, $e = 0.4$	
$N = 5$, $\nu_1 = \pi/2$, $\Delta\nu = \pi$,	$\overline{\Delta V} = 1$ m/s
Initial relative position [m]:	$[500, 400, 10]$
Initial relative velocity [m/s]:	$[0, 0, 0]$
$[\underline{x}, \overline{x}, \underline{y}, \overline{y}, \underline{z}, \overline{z}]$ [m]:	$[50, 150, -25, 25, -25, 25]$

For the (Pb2) algorithm, 50 iterations of the quasi-Newton method (I_{qn}) and 500 iterations of the subgradient method (I_{sg}) were used. All the performed tests returned an absolute value of d_0 (periodicity constraint) lower than 10^{-6}. The obtained computing time, consumption and maximal constraint violation are presented in Table 2.

Table 2: Results

	(LP)40	(LP)80	(LP)120	(LP)160	(Pb2)
Time (s)	0.890	2.673	5.336	8.955	3.602
Cons. (m/s)	0.399	0.399	0.399	0.399	0.402
Viol. (m)	0.152	0.038	0.017	0.009	0

The (LP) algorithm violates the constraints for all tested values of $N_{LP} = \{40, 80, 120, 160\}$. This behaviour is due to the fact that the discrete approach produces a polytope outer-approximation of S_D^p and, since the solution of a LP-problem occurs at a vertex of its feasible set, the obtained solution is not expected to belong to the original set. Another disadvantage of (LP) is that the necessary number of discrete constraints to ensure an arbitrary upper bound for the constraints violation depends on the rendezvous scenario parameters and is not known *a priori*. This characteristic is undesirable for an algorithm that is supposed to be autonomous for any mission configuration.

Contrarily, the (Pb2) algorithm produces a solution that does not violate the space constraints within a computing time lower than the time needed to solve the LP-problem with 120 discrete constraints and a consumption that is only

0.8% greater (which can be explained by the fact that the feasible set for the (LP) is less restrictive).

5.2 Hardware-in-the-loop simulation

During the hardware-in-the-loop simulation, the board receives the current relative state between spacecraft and solves the optimization problem (Pb2). The computed control actions are then sent to a Matlab/Simulink simulator that computes the evolution of the relative trajectories. The transmission of the data between the board and Matlab is implemented via a user datagram protocol (UDP).

The simulator (see [2]) is based on the Gauss planetary equations for the relative motion (see, for details, [24, 25]). It takes into account the effects of disturbances such as: the atmospheric drag and Earth's oblateness; uncertainties on the measurement of the relative position and velocities; execution errors on the orientation, magnitude and time of application of the impulsive velocity corrections.

Fig. 3 presents the trajectory obtained for the scenario described in Table 1, with number of impulses $N = 3$ and firing interval $\Delta\nu = \pi/4$. Following the receding horizon principle, at each call of the control algorithm three impulses are computed, but only the first one is applied. The duration of the simulation is equivalent to 10 orbital periods. The proposed algorithm is able to produce a trajectory that respects the space constraints, even in presence of non-linearities and disturbances that are not taken into account in the used predictive model.

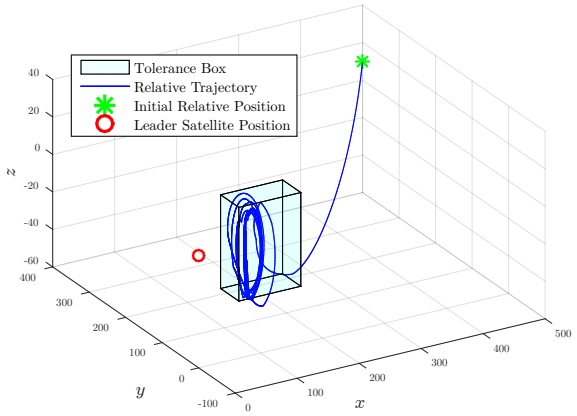

Figure 3: Hardware-in-the-loop simulation result.

6 Conclusion

This article proposes a MPC algorithm embeddable on a space-certified computing board. The algorithm is based on a new description of the subset of periodic space-constrained relative trajectories and accounts for the minimization of the fuel-consumption and the respect of space, saturation and periodicity constraints. Bench-tests have been performed against a LP algorithm on a board simulating the real performances of space dedicated devices. The robustness under disturbances has been assessed via hardware-in-the-loop simulations using a nonlinear Matlab/Simulink simulator for the relative motion. As future works, we intend to thoroughly address stability and robustness in a theoretical framework and compare with existing methods from hybrid control which feature such properties. Further extensive simulations should be carried out in order to study the influence of the scenario parameters on the propellant consumption.

References

[1] Milton Abramowitz and Irene A Stegun. *Handbook of mathematical functions: with formulas, graphs, and mathematical tables*. Number 55. Courier Corporation, 1964.

[2] Paulo Ricardo Arantes Gilz. A Matlab®/Simulink® non-linear simulator for orbital spacecraft rendezvous applications., December 2016. URL `https://hal.archives-ouvertes.fr/hal-01413328`.

[3] Paulo Ricardo Arantes Gilz and C. Louembet. Predictive control algorithm for spacecraft rendezvous hovering phases. In *Control Conference (ECC), 2015 European*, pages 2085–2090. IEEE, 2015.

[4] Stephen Boyd, Lin Xiao, and Almir Mutapcic. Subgradient methods. *lecture notes of EE392o, Stanford University, Autumn Quarter*, 2004:2004–2005, 2003.

[5] Louis S Breger, G. Inalhan, M. Tillerson, and J. P. How. Cooperative spacecraft formation flying: Model predictive control with open- and closed-loop robustness. In Pini Gurfil, editor, *Modern Astrodynamics*, volume 1 of *Elsevier Astrodynamics Series*, pages 237 – 277. Butterworth-Heinemann, 2006.

[6] M. Brentari, D. Arzelier, C. Louembet, S. Urbina, and L. Zaccarian. A hybrid control framework for impulsive control of satellite rendezvous. In *2016 American Control Conference (ACC)*, pages 7414–7419, July 2016. doi: 10.1109/ACC.2016.7526843.

[7] G. Deaconu. Constrained periodic spacecraft relative motion using non-negative polynomials. *American Control Conference*, pages 6715–6720, 2012.

[8] G. Deaconu. *On the trajectory design, guidance and control for spacecraft rendezvous and proximity operations*. PhD thesis, Univ. Toulouse 3 - Paul Sabatier, Toulouse, France, October 2013.

[9] Georgia Deaconu, Christophe Louembet, and Alain Théron. Designing continuously constrained spacecraft relative trajectories for proximity operations. *Journal of Guidance, Control, and Dynamics*, 38(7):1208–1217, 2015.

[10] ESA. Onboard computer and data handling, May 2014. URL `http://www.esa.int/Our_Activities/Space_Engineering_Technology/Onboard_Computer_and_Data_Handling/Microprocessors`. Accessed Oct. 10, 2016.

[11] Wigbert Fehse. *Automated rendezvous and docking of spacecraft*, volume 16. Cambridge university press, 2003.

[12] GLPK. GLPK (GNU linear programming kit v4.6), June 2012. URL `http://www.gnu.org/software/glpk`. Accessed Oct. 10, 2016.

[13] Edward N Hartley and Jan M Maciejowski. Predictive control for spacecraft rendezvous in an elliptical orbit using an FPGA. In *Control Conference (ECC), 2013 European*, pages 1359–1364. IEEE, 2013.

[14] Edward N Hartley, Paul A Trodden, Arthur G Richards, and Jan M Maciejowski. Model predictive control system design and implementation for spacecraft rendezvous. *Control Engineering Practice*, 20 (7):695–713, 2012.

[15] Hoon Hong. Implicitization of curves parameterized by generalized trigonometric polynomials. In *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes*, volume 948, pages 285–296. Springer Berlin Heidelberg, 1995. ISBN 978-3-540-60114-2.

[16] David J Irvin, Richard G Cobb, and Thomas A Lovell. Fuel-optimal maneuvers for constrained relative satellite orbits. *Journal of guidance, control, and dynamics*, 32(3):960–973, 2009.

[17] Adrian S Lewis and Michael L Overton. Nonsmooth optimization via quasi-Newton methods. *Mathematical Programming*, 141(1-2): 135–163, 2013.

[18] David G Luenberger and Yinyu Ye. *Linear and nonlinear programming*, volume 2. Springer, 1984.

[19] PENDER. GR-XC6S Development Board - User Manual, July 2011. URL `http://www.pender.ch/docs/GR-XC6S_user_manual_i1r0_20110713.pdf`. Accessed Oct. 10, 2016.

[20] I. Michael Ross. 6 - Space Trajectory Optimization and L1-Optimal Control Problems. In Pini Gurfil, editor, *Modern Astrodynamics*, volume 1 of *Elsevier Astrodynamics Series*, pages 155 – 188. Butterworth-Heinemann, 2006.

[21] Naum Zuselevich Shor. *Minimization methods for non-differentiable functions*, volume 3. Springer-Verlag, 1985.

[22] M. Tillerson and J. P. How. Formation flying control in eccentric orgits. In *AIAA Guidance, Navigation, and Control Conference (GNC)*, August 2001. URL `http://acl.mit.edu/gnc02.html`.

[23] J. Tschauner. Elliptic orbit rendezvous. *AIAA Journal*, 5(6):1110–1113, 1967.

[24] D. A. Vallado. *Fundamentals of astrodynamics and applications*, volume 12. Springer, 2001.

[25] M. J. H. Walker. A set of modified equinoctial orbit elements. *Celestial mechanics*, 38(4):391–392, 1986. ISSN 1572-9478.