Design and Application of Thrust Manoeuvres in a Constrained Spacecraft Rendezvous Context Using Optimal Control Techniques

Afonso Botelho, *Instituto Superior Técnico*

***Abstract*—The design of a model predictive control (MPC) algorithm for performing orbital rendezvous manoeuvres in elliptical orbits is addressed. The MPC algorithm computes a trajectory between an initial and desired states, minimizing fuel and taking into account limited thrust authority and passive collision safety. The MPC prediction model is based on the Yamanaka-Ankersen equations with Ankersen's zero-order hold particular solution.**

Finite-Horizon MPC is the standard formulation for this application, allowing for generating fuel-optimal trajectories via linear programming given a pre-defined manoeuvre duration. We extend it by presenting two new approaches for formulating passive collision avoidance constraints, which are naturally non-convex, as linear constraints. We also contribute with two new robustness techniques which improve the controller performance in the presence of disturbances while maintaining the computational complexity.

I. Introduction

ORBITAL rendezvous is a highly useful procedure in which two separate spacecraft meet at the same orbit, therefore approximately matching their orbital velocity and position [1]. Rendezvous missions have been performed successfully hundreds of times for multiple purposes and using different control strategies. In this context, we consider in this work the use of Model Predictive Control (MPC) [2] for performing rendezvous manoeuvres, which is a widely successful optimal control strategy that naturally considers the system dynamics and can handle various different operational constraints, such as fuel consumption minimization, limited thrust authority, and passive collision safety. The use of MPC for this purpose can grant more autonomy to the spacecraft and increase the optimality of the approach trajectories, when compared to the traditional techniques.

The literature for MPC applied to rendezvous is now quite considerable, and this remains an active area of research. Despite this, MPC has been tested in real spaceflight only once, to the best of the author's knowledge, by the PRISMA mission [3]. The main difficulty with the use of MPC for a real rendezvous mission is that it requires a considerable online computational effort, which can prove challenging given the typically limited computing power available onboard. Furthermore, there is not yet a standard approach for robustness in face of all disturbances present in a rendezvous mission which is both feasible to implement in real-time and maintains good operational performance, and thus more research into this topic is required.

An onboard automatic control system for a rendezvous mission contains three sub-systems tasked with the execution of thrust manoeuvres: guidance, navigation and control (GNC) [1]. Model Predictive Control can simultaneously handle both guidance and control functions, while navigation is not considered in this work. Furthermore, because the translational and attitude control are typically decoupled in rendezvous operations, only translational control is dealt with.

The motivation for this work is the PROBA-3 mission by ESA, scheduled for launch in 2021, in which two satellites will be launched together into a highly elliptical Earth orbit to test new formation flying technologies. A secondary objective of this mission is to perform a rendezvous experiment (RVX), which is led by DEIMOS [4]. Thus, this work was motivated as a feasibility study for DEIMOS in the use of MPC for the guidance and control systems in the PROBA-3 RVX.

We present a new method for formulating passive safety constraints with online linear programming, relying on offline non-linear optimization. A variation of this technique which relies purely on iterative linear optimization is also presented. Finally, we contribute with new robustness techniques, with the use of a terminal quadratic controller for a more accurate and robust final braking manoeuvre, and the dynamic relaxation of the terminal constraint in order to avoid the overcorrection of disturbances and waste of fuel.

In Section II, we first cover general MPC theory with linear system models. The relative orbital dynamics between two satellites, which are crucial to understand for the design of a rendezvous mission, are covered in Section III, presenting the linearised model which is later used as the MPC prediction model. Section IV applies the MPC framework to the rendezvous problem and our main contributions are presented. Finally, Section V presents rendezvous simulations using MPC.

II. Model Predictive Control

Model Predictive Control is a Control design method based on iterative online optimization [2]. The strategy is to obtain a control decision by solving an optimization problem which factors in future states of the system in a finite horizon, predicted using a discrete system model. This approach is illustrate in Fig. 1.

Part of this work was funded by project HARMONY, Distributed Optimal Control for Cyber-Physical Systems Applications through POR-Lisboa under LISBOA-01-0145–FEDER-031411 and by FCT through INESC-ID plurianual funds under contract UID/CEC/50021/2019.

Deimos Engenharia provided the working framework for Afonso Botelho.

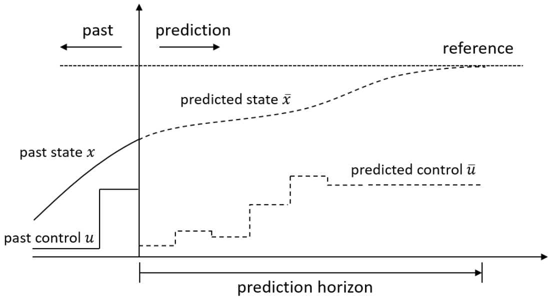


Fig. 1. Illustration of the Model Predictive Control strategy.

At each time step, the problem is solved with the most recent state measurement or estimate as the initial condition for the prediction, and a control strategy for future steps within the prediction horizon is obtained. The first control value in the obtained sequence is executed, and the problem is solved again in the next time step, with an updated state and with the prediction horizon shifted forward. For this reason, this method is also known as Moving/Receding Horizon Control.

Since MPC is formulated as an optimization problem, it allows for the inclusion of control and state constraints. This is a powerful tool and one of the major advantages of MPC in respect to other control methods, since it allows to limit the control action and to model complex state restrictions, such as safety constraints. Also, MPC naturally considers the system dynamics and can easily handle multivariate systems. On the other hand, because the optimization problem must be solved online, it requires a great computational effort which limits the use of MPC for systems with very fast dynamics or with reduced computation power.

A. General Formulation

A discrete-time system, with state variables x and input u, is described by the difference equation

$$x^+ = f(x,u), \tag{1}$$

where x^+ is the system state at the next time step and $f(x,u)$ is the system model.

At a time-instant t, MPC solves the following open-loop optimal control problem

$$\begin{aligned} \min_{\substack{\bar{u}_0,\ldots,\bar{u}_{N-1}\\ \bar{x}_0,\ldots,\bar{x}_N}} \quad & \sum_{i=0}^{N-1} l(\bar{x}_i,\bar{u}_i) \;+\; V_f(\bar{x}_N) && \text{(2a)}\\ \text{s.t.} \quad & \bar{x}_0 = x_t, && \text{(2b)}\\ & \bar{x}^+ = f(\bar{x},\bar{u}), && \text{(2c)}\\ & \bar{x}_k \in \mathcal{X}_k,\; k=0,\ldots,N, && \text{(2d)}\\ & \bar{u}_k \in \mathcal{U}_k,\; k=0,\ldots,N-1, && \text{(2e)} \end{aligned}$$

where $\bar{x}$ and $\bar{u}$ are the predictions of x and u, N is the length of the prediction horizon, and $l(\cdot\,,\cdot)$ and $V_f(\cdot)$ are cost functions. The minimization is subject to (”s.t.”) constraints (2b-e). The constraint in (2b) sets the initial condition for the prediction, and the state and control predictions are subject to the system model in constraint (2c). The sets $\mathcal{X}$ and $\mathcal{U}$ in (2d) and (2e) represent constraints on the state and control variables, respectively. Solving this open-loop problem yields an optimal control sequence $\bar{u}^*$, of which only the first is applied, meaning that $u_t = \bar{u}^*_t$. The problem is then solved again at the next time-step, at $t+1$, with the updated state measurement/estimate x_{t+1} that is the response to the applied control, thus closing the control loop.

B. Linear-Quadratic MPC

If the MPC prediction model in (2c) is linear, it can instead be described by the state-space model

$$x^+ = Ax + Bu. \tag{3}$$

The use of a linear prediction model instead of a nonlinear one grants a computational advantage, since the latter results in a non-linear optimization constraint that makes the optimization process harder. On the other hand, nonlinear models can more realistically describe the system, allowing for better state predictions and operating the system closer to the boundary of the admissible operating region [7].

A common and effective cost for the MPC optimization problem is the quadratic function, where the cost functions become

$$\begin{aligned} l(x,u) &= x^\top Q x + u^\top R u,\\ V_f(x) &= x^\top Q_f x, \end{aligned} \tag{4}$$

where Q and Q_f are the state and terminal state cost matrices, which are positive semi-definite, and R is the positive definite input cost matrix. These cost matrices are used to tune the controller: increasing the elements in R relative to Q and Q_f increases the penalization of the control variable in the cost function, and so the optimal solution will have limited actuator action.

Although the quadratic function is most commonly used, other cost functions are available. In particular, the use of the ℓ_1-norm is advantageous since it results in a sparser control profile, which is desirable in applications with limited fuel such as orbital rendezvous. On the other hand, the quadratic cost presents better robustness.

C. Reference Tracking

To control the system to a reference set point x_{ref} instead of to the origin, the tracking error must penalized. With the quadratic function, the cost functions become

$$\begin{aligned} l(x,u) &= (x-x_{ref})^\top Q (x-x_{ref}) + u^\top R u\\ V_f(x) &= (x-x_{ref})^\top Q_f (x-x_{ref}). \end{aligned} \tag{5}$$

Another method for reference tracking is through the use of a terminal state hard-constraint, as will be shown in Section IV.

D. Optimization Methods

Since the computational complexity is the greatest limitation for MPC, the method for optimization is greatly important. For optimal control problems which are convex and lack inequality constraints, analytical optimization is possible via solving the Karush-Kuhn-Tucker (KKT) [8] conditions, which becomes a very fast approach. In the presence of inequality constraints, analytical optimization easily becomes infeasible since the search for the solution grows exponentially with the number of constraints.

Typically, numerical optimization algorithms [9] are utilized instead. In a real-time MPC context, a fast and robust algorithm is desirable, while accuracy may be less valued. Furthermore, it is also advantageous that the algorithm has good early termination properties, in case the optimization has to be interrupted due to real-time constraints, and that it can be warmstarted with the solution of previous optimizations, which can be exploited given the iterative nature of MPC. The computational complexity depends on the type of optimization problem, *i.e.* linear, quadratic, nonlinear, which in turn depends on the prediction model and constraints used. It also depends on the length of the prediction horizon, which defines the number of control decisions. For linear system models, the complexity can be reduced by eliminating the state as an optimization variable, by writing it as a function of the initial condition and the control decisions via the prediction model. Alternatively, some algorithms specific for MPC optimization, for example [10], exploit the structure between the control and state variables, allowing for even faster optimization.

A different optimization approach is Dynamic Programming (DP), where the optimization problem is solved recursively and explicitly as a function of the initial condition. This is desirable because it allows for performing most of the computation offline, while the online work simply becomes applying the solution to the current state. However, the (DP) approach does not handle inequality constraints well, and these render the solution intractable.

An equivalent explicit solution approach which yields a simpler result is Explicit MPC [11], where the optimal control problem is solved with multi-parametric programming. This yields a piecewise solution which is a look-up table of control laws, each valid in a different set of the state-space known as *critical region*. In the specific case of linear-quadratic MPC, the control laws become affine and the critical regions are convex polytopes. In general, however, the number of critical regions grows exponentially with the number of inequality constraints, which makes the use of Explicit MPC infeasible for anything but small problems, although several complexity reduction techniques exist.

III. Relative Orbital Dynamics

Relative orbital dynamics refers to the relative motion of two satellites orbiting the same body, and it is imperative to consider them in order to successfully accomplish rendezvous missions. Typically in these missions, one of the satellites is in free motion, designated *target* spacecraft, while the other, *chaser*, performs the thrust manoeuvres to close their relative positions. In this context, it is convenient to consider relative positions and velocities, centred at one of the spacecraft, rather than absolute coordinates centred in the massive body.

Denoting a position vector as $\mathbf{r}$ with magnitude r, the relative position of the satellites is given by $\mathbf{s} = \mathbf{r}_c - \mathbf{r}_t$, where the subscripts c and t are associated to the chaser and target spacecraft, respectively. Assuming an uniform gravitational field and that the the satellite masses are negligible, the relative motion between the satellites is easily derived from Newton's law of gravitation and second law of motion, yielding

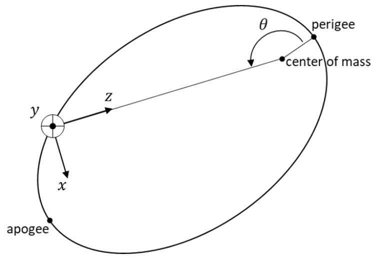


Fig. 2. The target local orbital frame (F_{lo}).

$$\ddot{\mathbf{s}} = \mu\left(\frac{\mathbf{r}_t}{r_t^3} - \frac{\mathbf{r}_c}{r_c^3}\right) + \frac{\mathbf{F}}{m_c}, \tag{6}$$

where μ is the standard gravitational parameter, $\mathbf{F}$ is the force applied by the chaser thrusters, and m_c is the chaser mass. Unlike in the case of only one unperturbed satellite, this problem has no closed-form solution and must be solved numerically or approximated with a linearisation.

A. Target Local Orbital Frame

When the distance between the two spacecraft is small, it is convenient to consider the non-inertial *target local orbital frame*, illustrated in Fig. 2, where θ is the true anomaly of the target spacecraft. This frame is also known as *local-vertical/local-horizontal* frame (LVLH), given that it is centred in the position of the target spacecraft.

Axis x_{lo} is in the general direction of the velocity vector, although it is not always aligned with it, and is commonly known as *V-bar*. Axis y_{lo} is orthogonal to the orbital plane, in the opposite direction of the angular momentum, and is also known as *H-bar*. The axis z_{lo}, known as *R-bar*, is always directed at the center of mass of the central body. For this reason, the frame rotates with the orbital angular velocity ω, and thus is non-inertial.

B. Approximate Equations of Relative Motion

To make use of the nonlinear relative dynamics in (6), a linearisation is performed around the target position $\mathbf{r}_t$ [1]. Furthermore, the relative position is converted to the LVLH frame, yielding the linearised equations of relative motion (LERM)

$$\ddot{x} - \omega^2 x - 2\omega\dot{z} - \dot{\omega}z + k\omega^{\frac{3}{2}}x = \frac{F_x}{m_c}, \tag{7a}$$

$$\ddot{y} + k\omega^{\frac{3}{2}}y = \frac{F_y}{m_c}, \tag{7b}$$

$$\ddot{z} - \omega^2 z + 2\omega\dot{x} + \dot{\omega}x - 2k\omega^{\frac{3}{2}}z = \frac{F_z}{m_c}\ , \tag{7c}$$

where (x, y, z) is chaser position relative to the target in the LVLH frame, $\mathbf{F} = [F_x, F_y, F_z]^\top$, and k is the constant $k = \mu/h^{\frac{3}{2}}$, with h as the magnitude of the target orbit specific angular momentum.

The LERM equations are linear with respect to relative position, velocity and acceleration, although they are not so with respect to the orbital angular velocity ω. Because this parameter is time-varying for elliptical orbits, in general the equations are linear and time-varying, and thus a solution is not trivial. A special case emerges for a circular target orbit, where the orbital angular velocity becomes constant with time and thus the dynamics become time-invariant. In this case, the LERM equations become the well-known Hill equations, and their solution, known as the Clohessy-Wiltshire equations, is trivial. Finally, notice that the motion in H-bar, which is referred to as *out-of-plane* motion, is decoupled from that in V-bar and R-bar, designated *in-plane* motion. This simplifies the problem, since the two can be solved and analysed independently. However, note that this is directly due to the linearisation, and that in the nonlinear dynamics the two are in fact coupled.

C. Simplification of the General Equations

To solve the LERM equations in the general elliptic orbit case, a change of the independent variable from time t to the true anomaly θ is employed. Furthermore, a coordinate transformation is also applied

$$\tilde{\mathbf{s}} = \rho(\theta)\mathbf{s}, \tag{8}$$

where $\tilde{\mathbf{s}}$ is the relative position vector in the new coordinate system and $\rho(\theta)$ is a time-variant auxiliary function defined as

$$\rho(\theta) = 1 + e\cos(\theta), \tag{9}$$

where e is the orbit eccentricity. Denoting the derivative of $\mathbf{s}$ with respect to the true anomaly θ as $\mathbf{s}'$, the transformation for velocity is

$$\tilde{\mathbf{s}}' = -e\sin(\theta)\mathbf{s} + \frac{1}{k^2\rho(\theta)}\dot{\mathbf{s}}. \tag{10}$$

Applying these transformations, the LERM equations simplify to

$$\tilde{x}'' - 2\tilde{z}' = \frac{F_x}{m_c k^4 \rho^3}, \tag{11a}$$

$$\tilde{y}'' + \tilde{y} = \frac{F_y}{m_c k^4 \rho^3}, \tag{11b}$$

$$\tilde{z}'' - \frac{3}{\rho}\tilde{z} + 2\tilde{x}' = \frac{F_x}{m_c k^4 \rho^3} \ , \tag{11c}$$

which are known as the Tschauner–Hempel equations and can be more easily solved.

D. Homogeneous Solution

For the homogeneous solution of the Tschauner–Hempel equations ($\mathbf{F} = 0$), we consider the Yamanaka-Ankersen equations [5]. Given initial conditions $\tilde{\mathbf{x}}_0 = [\tilde{x}_0, \tilde{y}_0, \tilde{z}_0, \tilde{x}_0', \tilde{y}_0', \tilde{z}_0']^\top$ for an initial true anomaly θ_0, the homogeneous solution becomes

$$\tilde{\mathbf{x}}_h(\theta) = \phi(\theta)\phi^{-1}(\theta_0)\tilde{\mathbf{x}}_0, \tag{12}$$

where $\phi(\theta)$ is a transition matrix not presented here for brevity.

E. Particular Solution

We will consider Ankersen's particular solution [6], which assumes a constant force during the sampling period and thus results in a zero-order hold (ZOH) discretization. The particular solution thus becomes

$$\tilde{\mathbf{x}}_p(\theta) = \Gamma(\theta_0, \theta)\mathbf{F}, \tag{13}$$

where input matrix $\Gamma(\theta_0, \theta)$ is also not presented here.

F. State-Space Model

Representing the coordinate transformations in (8) and (10) with the transformation matrix $\Lambda(\theta)$ and the inverse transformation with $\Lambda^{-1}(\theta)$, we get the state-space model in the original coordinates

$$\mathbf{x}_{k+1} = A_k^{k+1}\mathbf{x}_k + B_k^{k+1}\mathbf{u}_k, \tag{14}$$

with the input vector $\mathbf{u} = \mathbf{F}$, and such that the system at a discrete time k has true anomaly θ_0 and at time $k+1$ the true anomaly θ. Matrix A_k^{k+1} is the state transition matrix from time k to $k+1$

$$A_k^{k+1} = \Lambda^{-1}(\theta)\phi(\theta)\phi^{-1}(\theta_0)\Lambda(\theta_0), \tag{15}$$

while B_k^{k+1} is the input matrix, which becomes

$$B_k^{k+1} = \Lambda^{-1}(\theta)\Gamma(\theta_0, \theta). \tag{16}$$

Given two arbitrary positions and a transfer duration, the Yamanaka-Ankersen state transition matrix A_k^{k+1} can be used to generate two-impulse manoeuvres, by solving for the initial and final changes in velocity required, designated ΔV's. These manoeuvres are often used to generate trajectories in traditional rendezvous guidance techniques.

IV. Rendezvous with MPC

In traditional rendezvous mission design, guidance trajectories are designed offline, and so manoeuvres are performed in open-loop, often times with punctual mid-course correction boosts determined online from the trajectory deviation. In this context, an increasing amount of research has been dedicated to applying Model Predictive Control (MPC) to the rendezvous problem [12], in order to perform these thrust manoeuvres online and in full closed-loop.

A. Relative Dynamics Sampling

The relative dynamics are time-varying for elliptical orbits, which presents a difficulty for the sampling of the MPC prediction horizon. For example, in the conditions of the PROBA-3 mission, the orbital velocity is nearly ten times greater at perigee than at apogee, which also translates to the velocity of the relative motion. Because in MPC there is a limited amount of samples, associated with the length of the prediction horizon, these must be allocated appropriately along the orbit in order to get the best performance, since the point at which the thrust is applied is important for the optimality of the trajectory.

If the dynamics are sampled with constant time intervals, the samples will concentrate on apogee because the orbital velocity is greater there, which is opposite to what is desired since the dynamics are faster on perigee. Thus, in this work we sample the dynamics with constant eccentric anomaly intervals, which results in longer time intervals between samples at apogee than at perigee, in way that samples are uniformly distributed in space.

B. Fixed-Horizon MPC

The receding-horizon strategy which is standard for MPC is not appropriate for the rendezvous scenario. The fact that the prediction horizon slides forward every sample means the system can never commit to one specific trajectory, which prohibits achieving a fuel-optimal closed-loop trajectory. Thus, the standard approach is to decrement the prediction horizon every sample, such that its edge is always at the same time-instant, and thus known as Fixed-Horizon MPC (FH-MPC) [13].

To achieve a fuel-optimal formulation, the cost function must also be chosen appropriately. First, no intermediate cost $l(\cdot\,,\cdot)$ is used for the state variables, thus allowing the controller to better plan ahead since it is not penalized for not being on the reference state while halfway through the manoeuvre. Also, for the input variable cost, the ℓ_1-norm is used. Because the absolute value of the thrust is proportional to the fuel consumption, this cost function allows for the direct minimization of the fuel required for the manoeuvre. On the other hand, the use of the ℓ_1-norm does not allow for a computationally efficient optimization.

To simplify the cost function, the input matrix in the prediction model can be extended, such that input forces are split into its positive and negative components $F = F^{+} - F^{-}$. This increases the number of optimization variables, which is disadvantageous, but now each input variable can only take positive values, which makes its absolute value equal to itself, thus turning the ℓ_1-norm into a linear cost function. To simplify the problem further, the terminal state cost $V_f(\cdot)$ is removed and replaced with a terminal state constraint, which allows for the cost function to become completely linear. Thus, the standard FH-MPC formulation, as first presented in [13], becomes

$$\min_{\substack{\bar{u}_0,\dots,\bar{u}_{N-1}\\ \bar{x}_0,\dots,\bar{x}_N}} \quad \sum_{i=0}^{N-1} \Delta t_i \mathbf{1}^\top \bar{u}_i \tag{17a}$$

$$\text{s.t.} \quad \bar{x}_0 = x_t, \tag{17b}$$

$$\bar{x}_{k+1} = A_k^{k+1}\bar{x}_k + B_k^{k+1}\bar{u}_k, \tag{17c}$$

$$0 \leq \bar{u}_k \leq u_{max}, \quad k = 0,\dots,N-1, \tag{17d}$$

$$\bar{x}_N = x_{ref}, \tag{17e}$$

which is a linear program (LP) and can be solved very efficiently. Notice that in (17a) the input variables are weighted with the time intervals between samples Δt, which is important to maintain the cost function proportional to the fuel required, since a ZOH discretization is used and the system is not sampled with constant time intervals. Furthermore, the constraint in (17d) limits the control to the maximum thrust u_{max}, and also lower-bounds it with zero in order to maintain the integrity of the input variable split. Finally, constraint (17e) is the terminal state constraint, where x_{ref} is the reference state. This constraint sets the manoeuvre duration, which is defined by the prediction horizon N at the first iteration, and thus must be defined prior to the optimization.

C. Variable-Horizon MPC

It is also desirable to optimize the manoeuvre duration as well as fuel consumption, which is performed by adding the prediction horizon N as an integer optimization variable, in what is known as Variable-Horizon MPC (VH-MPC). However, this makes the terminal constraint (17e) nonlinear, turning the problem intro a mixed-integer nonlinear program, which is computationally expensive to solve.

A method for transforming the problem into a mixed-integer linear program (MILP) presented in [13] requires two binary optimization variables per time-instant. Variable v_k is 1 if the manoeuvre is completed exactly at instant k, while p_k is 1 while the manoeuvre is not completed and 0 afterwards. The VH-MPC MILP formulation then becomes

$$\min_{\substack{\bar{u}_0,\dots,\bar{u}_{N_{max}-1}\\ \bar{x}_0,\dots,\bar{x}_{N_{max}}\\ p_0,\dots,p_{N_{max}}\in\{0,1\}\\ v_1,\dots,v_{N_{max}}\in\{0,1\}}} \quad \gamma \sum_{i=0}^{N_{max}} p_i + \sum_{i=0}^{N_{max}-1} \Delta t_i \mathbf{1}^\top \bar{u}_i \tag{18a}$$

$$\text{s.t.} \quad \bar{x}_0 = x_t, \tag{18b}$$

$$\bar{x}_{k+1} = A_k^{k+1}\bar{x}_k + B_k^{k+1}\bar{u}_k, \tag{18c}$$

$$0 \leq \bar{u}_k \leq u_{max}, \tag{18d}$$

$$-(1-v_k)h \leq x_k - x_{ref} \leq (1-v_k)h, \tag{18e}$$

$$p_{k+1} = p_k - v_{k+1}, \tag{18f}$$

$$p_{N_{max}} = 0, \tag{18g}$$

$$\sum_{k=1}^{N_{max}} v_k = 1. \tag{18h}$$

where the new term in the cost function is proportional to N, and thus γ is a parameter used to tune the trade-off between fuel consumption and manoeuvre duration, while N_{max} defines the maximum manoeuvre duration possible. Constraint (18e) is the new terminal state constraint, where h is a large enough number; at the instant the manoeuvre is completed, v_k is 1 and thus the terminal state constraint is active, otherwise v_k is 0 and the constraint is loose. Constraints (18f) and (18h) maintain the integrity of the binary variables, while (18g) forces the manoeuvre to be completed at least by the end of the maximum prediction horizon.

The computational load for this formulation is greater than for the FH-MPC, since MILP problems are harder to solve. Thus, in a real-time scenario, it is preferable to predetermine offline the manoeuvre duration and use the FH-MPC formulation instead. However determining the manoeuvre duration offline can be performed in an optimal way by using the VH-MPC formulation. The optimal transfer time may change slightly along the way due to disturbances, but the time determined offline can still be expected to remain approximately optimal.

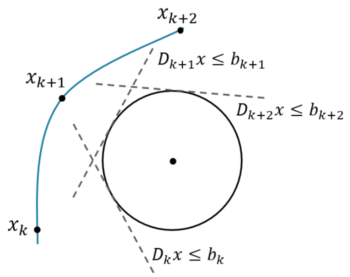


Fig. 3. Illustration of obstacle avoidance with linear constraints.

D. Passive Safety Constraint

It is a requirement in rendezvous missions that, besides ensuring that the nominal trajectory does not cause a collision between the spacecraft, the free-drift motion from any point in the trajectory also remain collision-free within a specified time-horizon. Designing trajectories in this way ensures that, in case a thruster fails to fire, the two spacecraft will not collide due to the natural drift, and thus this is known as passive safety.

To formulate the passive safety constraint, the failure trajectories must be propagated with the prediction model, as first proposed by Breger and How [14], and hard-constrained to avoid the obstacle. If a total thruster failure occurs at time k, the resulting free-drift failure trajectory x_{F_k} is described by

$$x_{F_{k,t}} = A_k^t x_k, \; t > k, \tag{19}$$

where A_k^t is the dynamic matrix that transitions the state from instant k to t. The passive safety constraint then becomes

$$\begin{aligned} x_{F_{k,t}} \notin \mathit{Obstacle}, \quad & k \in \{1, \ldots, N\} \\ & t \in \{k+1, \ldots, k+S\}, \end{aligned} \tag{20}$$

where failures at all discrete instants in the trajectory are considered and tracked for S samples, where S is the safety horizon.

This requires the addition of $N \times S$ optimization constraints, which have a significant computational burden, requiring an efficient implementation for feasible online use. However, collision avoidance constraints are naturally non-convex, which greatly increases the complexity of the optimization. It is possible to achieve obstacle avoidance with linear constraints, by subjecting each state in the trajectory to a different linear constraint tangent to the original obstacle, as illustrated in Fig. 3, and where the linear constraints have to be determined before the optimization. However, the methods found in the literature for determining the linear constraints are limiting and not appropriate for use in a passive safety constraint (*e.g.* [14], [15]). Thus, we propose two new methods for achieving obstacle avoidance with linear constraints.

1) Obstacle Avoidance with Offline Nonlinear Programming: One way to determine the linear obstacle constraints for online optimization is to perform a single offline optimization with the original nonlinar optimization constraints. Then, the planes tangent to the obstacle and facing each of the states are determined, and used to define the linear obstacle constraints of the online LP. This method will be referred to as *obstacle avoidance with offline nonlinear programming* (OAONP).

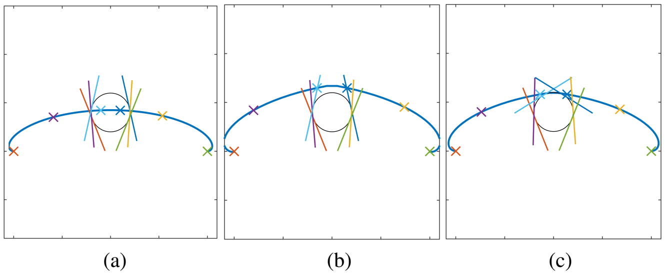


Fig. 4. Determining the linear obstacle avoidance constraints with iterative optimization with linear constraints.

2) Obstacle Avoidance with Iterative Linear Programming: Alternatively, the tangent planes can be determined with the result from the unconstrained LP optimization, which naturally may violate the constraint, as illustrated in Fig. 4a). Optimizing again with the linear constraints determined from the unconstrained problem yields a trajectory that necessarily satisfies the real constraint, although it may be conservative, as illustrated in Fig. 4b). However, the process can be repeated, determining again the linear obstacle constraints and optimizing successively until the trajectory converges, as illustrated in Fig. 4c).

Thus, this technique will hereby be referred to as *obstacle avoidance with iterative linear programming* (OAILP). It can be feasible to use it online, since it relies purely on linear optimization. Alternatively, this algorithm may also be applied only once offline to determine the linear constraints, which are then used for online optimization, much like the OAONP strategy.

E. Robustness Techniques

There are many sources of disturbances in a real rendezvous mission scenario to which the controller must be robust. This not only means that it remains stable, but also that it can still accurately converge to the reference, in approximately the specified manoeuvre duration and without a very significant increase in fuel consumption, in what we address here as *robust performance*. Furthermore, it is necessary that state constraints, such as passive safety, are not violated due to disturbances, which is known as *robust constraint satisfaction*. Finally, the disturbances can often push the system to a state that renders the optimal control problem infeasible, and thus the controller must have *robust feasibility*.

There is some inherent stability robustness for MPC [2], resulting from the fact that it is a closed-loop control strategy. Sometimes this is enough, though often robust strategies must be employed to increase performance and also to ensure state constraints are not violated due to a disturbance. In order to maintain a feasible real-time implementation, these strategies should not significantly increase the computational complexity of the problem. We consider a technique for ensuring robust feasibility of the terminal state constraint previously found in the literature, and suggest two new techniques for improving robust performance. Robust satisfaction techniques are not considered, although the chance-constrained technique presented in [16] is promising, since it uses real-time disturbance estimation for constraint-tightening and maintains the optimization problem as an LP.

1) Feasible Terminal Box: When the prediction horizon becomes short, as it does with the FH strategy, disturbances can push the system to a state from which the reference is not reachable in the remaining steps, given the system dynamics and other control and state constraints. This renders the terminal state constraint and the optimization problem infeasible, which should be avoided at all costs in a real-time application.

This can be minimized by relaxing the terminal state constraint into an inequality, thus introducing the concept of *terminal box*. The terminal constraint (17e) then becomes

$$-\delta \leq \bar{x}_N - x_{ref} \leq \delta, \tag{21}$$

where $\delta \in \mathbb{R}^6$ defines the bounds for the box. However, improving the guarantee of feasibility requires increasing the size of the terminal box, which in turn this worsens the accuracy of the controller. Furthermore, there is no guarantee that the chosen dimensions for the terminal box will always ensure the existence of a solution for all possible scenarios.

To improve this approach, the box bounds δ can be included as optimization variables, as first suggested in [17], which maintains the constraint linear. Furthermore, the new optimization variables are included in the cost function

$$V(\cdot) \quad = \quad \sum_{i=0}^{N-1} \Delta t_i \mathbf{1}^\top \bar{u}_i \; + \; \sum_{j=1}^{6} h_j \delta_j, \tag{22}$$

where h_j is large enough as to ensure the controller only relaxes the terminal constraint to ensure feasibility and not to save fuel. Thus, this *feasible terminal box* will always have the minimum size that guarantees feasibility.

2) Dynamic Terminal Box: The terminal state constraint forces the controller to set the fate state exactly on the reference at every iteration. In the presence of stochastic disturbances, however, this may cause an overcorrection of the trajectory and results in the waste of propellant. This may be minimized by loosening the terminal box, allowing the controller to perform fewer corrections, although at the cost of decreased manoeuvre accuracy.

Thus, we propose the inclusion of a time-varying bound ε_t to the terminal box constraint

$$-\delta - \varepsilon_t \leq \bar{x}_N - x_{ref} \leq \delta + \varepsilon_t, \tag{23}$$

where t is the time instant at which the optimization problem is being solved. The bound ε_t is decreased with time, such that it reduces the controller overcorrection but still maintains accuracy as it approaches the reference. This technique is not mutually exclusive with the feasible terminal box method.

3) Terminal Quadratic Controller: The sparse thrust profile of the FH-MPC formulation is not appropriate for executing accurate manoeuvres in the presence of disturbances, since crucial ΔV's are performed in one sample only, while planning under imperfect state information and with imperfect execution of the ΔV. Thus the final braking thrust at the end of manoeuvres for cancelling the remaining relative velocity tends to not be very precise, resulting in a poor manoeuvre accuracy.

Therefore, we propose the use of a linear-quadratic MPC controller, such as that in (5), to substitute the final sample of the FH-MPC and perform the final breaking thrust. The terminal controller has a prediction horizon N_T that has the same timespan as the last sample of the nominal controller and is decremented in the same FH strategy, in order to maintain the manoeuvre duration initially specified. Furthermore, a terminal state cost is used instead of the terminal constraint, and the intermediate state cost is zero. The non-sparse thrust profile resulting from the use of the quadratic cost and the increased control decisions result in a more accurate braking manoeuvre.

V. Experiments and Results

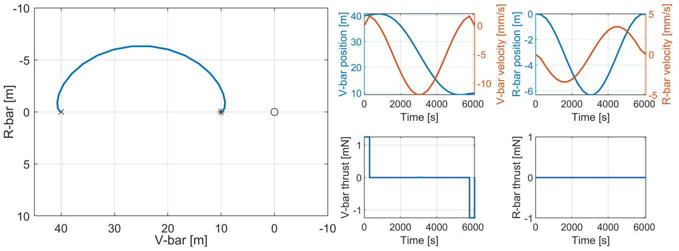
This section features several rendezvous simulations with the techniques presented. Simulations are performed for Earth satellites, with the perigee height and chaser mass for the PROBA-3 RVX of 600 km and 211 kg, respectively. The orbit eccentricity e simulated is either 0, for experiments in circular orbits, or with the PROBA-3 eccentricity of 0.8111. The initial true anomaly θ_0 is indicated for elliptical orbit experiments, and is irrelevant for circular orbits. The orbital orientation parameters are disregarded, since a uniform gravitational is assumed. For circular orbits, the sampling period T_s is presented, while for elliptical orbits the sampling eccentric anomaly E_s is shown. Finally, the fuel expenditure is analysed via the total ΔV applied by the chaser thrusters, which have a max thrust of 1 N in each direction.

The MPC optimization problem is solved with the MATLAB Optimization Toolbox, where function *linprog* is used for linear programming for the FH-MPC formulation, function *intlinprog* is used to solve the MILP in the VH-MPC formulation, and *fmincon* is used for solving the nonlinear program that arises in the OAONP technique. Because these algorithms do not take advantage of the MPC problem structure, the state-substitution technique briefly mentioned in Section II-D is utilized. The worst-case computation times t_{max} for solving the optimization problem is presented, solving with a 4th Generation 2.4GHz Intel-i7 Processor.

A. FH-MPC

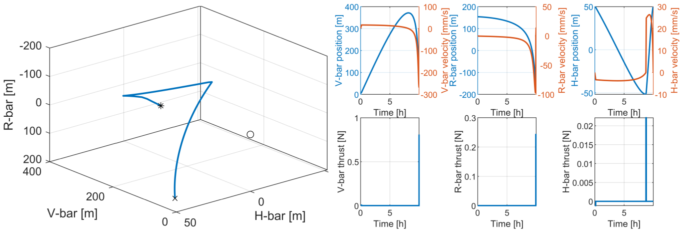
Applying the FH-MPC controller to a V-bar transfer scenario with an initial prediction horizon of just over one orbital period yields the result from Fig. 5, where the manoeuvre is performed with only two V-bar thrust actions. This resembles the ideal V-bar transfer with two horizontal impulses [1], for which the required ΔV exactly matches that obtained in this experiment. Thus, this validates the fuel-optimality of the FH-MPC formulation. The computation time was 7.97 ms in the worst case, which is very small comparing to the sampling time.

Considering an elliptical orbit and including the H-bar dimension, we simulate a manoeuvre from the PROBA-3 RVX, which starts approximately at apogee and ends near perigee. The ΔV required to perform this manoeuvre with the



(a) (b)

Fig. 5. V-bar transfer manoeuvre in one orbital period with FH-MPC. (a) Trajectory (b) State and control variables. Simulation parameters: $N = 21$, $T_s = 290\,\mathrm{s}$. Results: $\Delta V = 3.45\,\mathrm{mm/s}$, $t_{max} = 7.97\,\mathrm{ms}$.



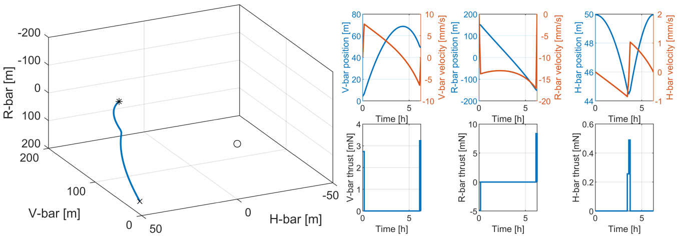
(a) (b)

Fig. 6. Manoeuvre from the PROBA-3 RVX with FH-MPC. (a) Trajectory (b) State and control variables. Simulation parameters: $N = 100$, $E_s = 1.70\,\mathrm{deg}$, $\theta_0 = 179\,\mathrm{deg}$. Results: $\Delta V = 407.4\,\mathrm{mm/s}$, $t_{max} = 9.52\,\mathrm{ms}$.

traditional two-impulse transfer is 481 mm/s, while FH-MPC yields a manoeuvre with 407 mm/s, thus requiring only 85% of the fuel that would typically be expended. This is possible due to the fact that the traditional two-impulse manoeuvres are constrained to one thrust action at beginning and another at end of the manoeuvre, while MPC may take intermediate control decisions. Thus, via the intermediate thrust action seen in H-bar toward the end, a more efficient manoeuvre is possible, which further shows the fuel-optimality of the FH-MPC formulation. Finally, notice that despite the fact that the the prediction horizon was increased by almost five times, the worst-case computation time only increased by 19.4%, thus demonstrating the benefit of formulating the optimization problem as a linear program.

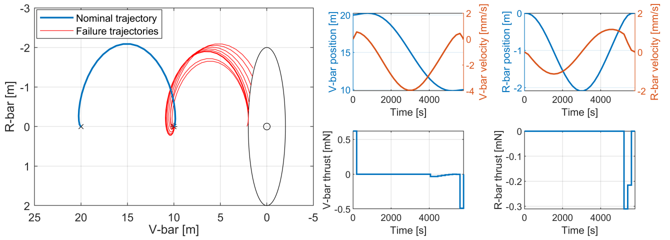
B. VH-MPC

Applying now the VH-MPC formulation to the previous PROBA-3 manoeuvre with a maximum transfer time of one orbital period yields the result from Fig. 7. Parameter γ is set to zero in order to only minimize the required fuel. The optimal transfer time obtained is 41% of an orbital period, yielding a ΔV of 48.16 mm/s, which is only 11.8% as that required for the half-orbit transfer in Fig. 6, thus validating the VH-MPC formulation. However, a transfer time of one orbital period would actually have been even more efficient, and is within the maximum duration specified. The reason VH-MPC did not yield that solution is that the ΔV is not a convex function of the transfer time, and the algorithm used to solve the MILP does not optimize globally, and thus it converges to this local minimum.



(a) (b)

Fig. 7. Manoeuvre from the PROBA-3 RVX with VH-MPC. (a) Trajectory (b) State and control variables. Simulation parameters: $N_{max} = 100$, $\gamma = 0$, $E_s = 3.6\,\mathrm{deg}$, $\theta_0 = 179\,\mathrm{deg}$. Results: $N = 41$, $\Delta V = 48.16\,\mathrm{mm/s}$, $t_{max} = 710\,\mathrm{ms}$.



(a) (b)

Fig. 8. V-bar transfer manoeuvre in one orbital period with FH-MPC and passive safety constraint with OAONP. (a) Trajectory (b) State and control variables. Controller parameters: $N = 30$, $T_s = 193\,\mathrm{s}$, $S = 30$. Simulation results: $\Delta V = 1.62\,\mathrm{mm/s}$, $t_{max} = 21.7\,\mathrm{ms}$, $t_{offline} = 1.31\,\mathrm{s}$.

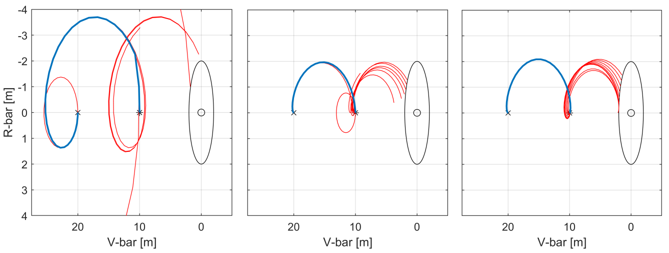
Finally, notice that the worst-case computation time required to solve the MILP is significantly greater than that of the LP in the FH-MPC formulation. Therefore, it may be infeasible to use the VH-MPC formulation online, especially with the inclusion of the computationally heavy passive safety constraint.

C. Passive Safety

Considering passive safety, Fig. 8 presents the same type of V-bar transfer manoeuvre presented previously in Fig. 5 with the inclusion of this constraint with a one-orbit safety horizon, formulated with the OAONP technique. Without it, a thruster failure in the final burn would result in a violation of the target safety region, defined here as a 2 metre circle, after less than an orbit. With the passive safety constraint, some extra actuation at the end of the manoeuvre ensures that the failure trajectories stop at the edge of the safety region exactly after one orbital period.

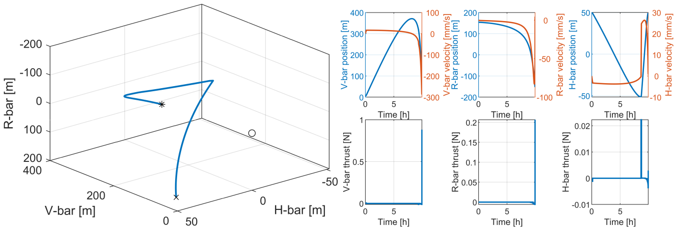
This comes at the cost of a 17% increase in fuel with respect to the non-passively safe trajectory. Furthermore, the computation time for optimizing with the nonlinear constraints is 1.31 s. However, the OAONP technique allows for this optimization to be performed just once offline, and the online computation time, although having increased due to the addition of 900 linear optimization constraints, remains 2 orders of magnitude smaller than this.

Using the OAILP technique instead for the same manoeuvre yields the result from Fig. 9. Although one iteration ensure the trajectory is passively safe, it is extremely inefficient, with a ΔV over ten times higher than that obtained in Fig. 8. After



(a) (b) (c)

Fig. 9. V-bar transfer manoeuvre in one orbital period with FH-MPC and passive safety constraint with OAILP. (a) One iteration, $\Delta V = 20.76\,\text{mm/s}$ (b) Two iterations, $\Delta V = 2.039\,\text{mm/s}$ (c) Three iterations, $\Delta V = 1.62\,\text{mm/s}$. Controller parameters: $N = 30$, $T_s = 193\,\text{s}$, $S = 30$. Simulation results: $t_{max} = 17.8\,\text{ms}$, $t_{offline} = 212.4\,\text{ms}$.



(a) (b)

Fig. 10. Manoeuvre from the PROBA-3 RVX with FH-MPC with feasible terminal box and simulated with the nonlinear dynamics. (a) Trajectory (b) State and control variables. Simulation parameters: $N = 100$, $E_s = 1.70\,\text{deg}$, $\theta_0 = 179\,\text{deg}$. Results: $\Delta V = 497.5\,\text{mm/s}$, $t_{max} = 10.1\,\text{ms}$, $e_{pos} = 25.44\,\text{cm}$, $e_{vel} = 4.746 mm/s$.

two iterations the result improves drastically, although the ΔV is still above the optimal trajectory. With three iterations the result almost exactly matches that with the previous method, and the ΔV is also approximately the same. Similarly to the OAONP strategy, the OAILP algorithm was run only once offline, and the computation time is significantly less than the former.

The satisfaction of non-convex obstacle avoidance constraint via purely linear constraints is promising and desirable, although there currently is no guarantee that the trajectory converges to a local minimum of the original nonlinear problem, which warrants further research.

D. Robustness Experiments

So far simulations have been performed with the same linearised model used for the MPC prediction model, and thus there are no modelling errors. Simulating the PROBA-3 manoeuvre with the real nonlinear dynamics yields the result from Fig. 10. The trajectory is slightly different from that obtained in Fig. 6, with some extra actuation at the end of the manoeuvre in an attempt to correct the disturbance that results in an increase of the ΔV of 22%. Furthermore, the feasible terminal box technique from Section IV-E1 is applied, having no significant increase in computational time, otherwise the last iteration would become infeasible and the manoeuvre incomplete. There is also a residual deviation of the final state from the reference in position e_{pos} and velocity e_{vel}, due to the loosening of the terminal constraint and the imperfect precision model. While increasing the number of samples in the prediction horizon decreases this error and the ΔV, such an approach is not effective in the presence of stochastic disturbances.

We now consider an additive Gaussian error to the state received by the controller with a standard deviation of 10 cm for position and 1 mm/s for velocity, thus modelling navigation errors. Also, an error in thrust magnitude with a standard deviation of 10% and in orientation of 0.5 deg in each direction is applied, modelling thruster errors. Finally, control commands under 1 mN are ignored in order to model the minimum thrust possible via pulse-width modulation of the on/off thrusters. Because the conditions are now stochastic, 20 simulations are performed and the results averaged. In these conditions yields the result from Fig. 11.

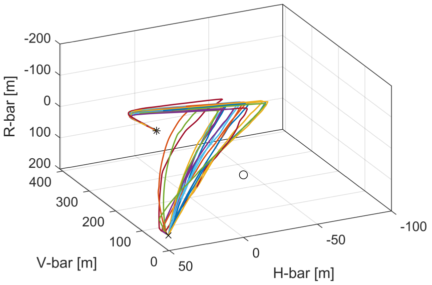


Fig. 11. Manoeuvre from the PROBA-3 RVX with FH-MPC with feasible terminal box, simulated 20 times with the nonlinear dynamics, navigation and execution errors. Simulation parameters: $N = 100$, $E_s = 1.70\,\text{deg}$, $\theta_0 = 179\,\text{deg}$. Results: $\Delta V_{avg} = 577\,\text{mm/s}$, $t_{max_{avg}} = 9.72\,\text{ms}$, $e_{pos_{avg}} = 1.08\,\text{m}$, $e_{vel_{avg}} = 28.1 mm/s$.

The effect of the disturbances can be verified by the dispersion in the different trajectories. Despite this, the controller can still robustly converge to the reference, although with a significant increase in ΔV and terminal error. Increasing the number of samples in the prediction horizon would decrease the terminal error, although the ΔV increases due to increased overcorrection of the trajectory.

Thus, we apply the dynamic terminal box method presented in Section IV-E2 in order to make the controller less sensitive to stochastic errors. With an initial terminal box of 5 m for position and 5 mm/s for velocity, and a linear decreased with time such that the box dimensions are zero the last iteration, yields the result from Fig. 12. The trajectories are slightly more dispersed due to less correction and the ΔV has slightly decreased. Despite loosening the terminal constraint, the terminal error has not increased, due to the dynamic tightening of the terminal box bounds. Although the fuel gain with the use of this technique is not very significant in this simulation, it validates the approach, warranting further research for improvement.

The terminal error for the previous simulations remains significant, especially for the velocity with 18.4 mm/s, which is considerably higher than the navigation uncertainty. To improve the manoeuvre accuracy, the terminal quadratic controller described in Section IV-E3 is applied with a prediction horizon N_T of 10 samples, yielding the result from Fig. 13. The terminal errors have been reduced both for position and velocity, and are now comparable to the navigation uncertainty.

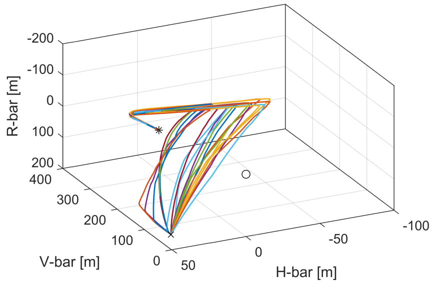


Fig. 12. Manoeuvre from the PROBA-3 RVX with FH-MPC with feasible terminal box and dynamic terminal box, simulated 20 times with the nonlinear dynamics, navigation and execution errors. Simulation parameters: $N = 100$, $E_s = 1.70\,\mathrm{deg}$, $\theta_0 = 179\,\mathrm{deg}$. Results: $\Delta V_{avg} = 545\,\mathrm{mm/s}$, $t_{max_{avg}} = 9.98\,\mathrm{ms}$, $e_{pos_{avg}} = 79.7\,\mathrm{m}$, $e_{vel_{avg}} = 18.4 mm/s$.

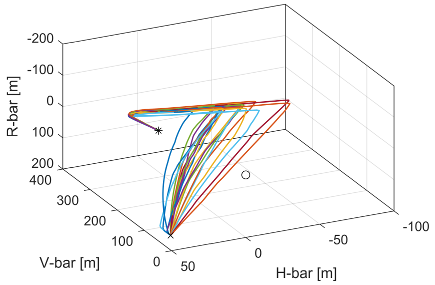


Fig. 13. Manoeuvre from the PROBA-3 RVX with FH-MPC with feasible terminal box, dynamic terminal box, and terminal quadratic controller, simulated 20 times with the nonlinear dynamics, navigation and execution errors. Simulation parameters: $N = 100$, $E_s = 1.70\,\mathrm{deg}$, $\theta_0 = 179\,\mathrm{deg}$. Results: $\Delta V_{avg} = 557\,\mathrm{mm/s}$, $t_{max_{avg}} = 9.54\,\mathrm{ms}$, $e_{pos_{avg}} = 28.6\,\mathrm{m}$, $e_{vel_{avg}} = 3.57 mm/s$.

The ΔV has also slightly increased, although less than the decrease in residual velocity. A Better performance might be achieved with further tuning of the terminal horizon N_T or the terminal cost matrices, although this result validates this approach for achieving better accuracy.

VI. Conclusion

The FH-MPC formulation presents advantages in fuel-consumption over traditional guidance techniques, which usually rely on two-impulse manoeuvres, and is implemented efficiently via linear programming. The use of MPC allows for the explicit modelling of operational constraints, such as maximum thrust and passive safety collision. The latter is naturally formulated with non-convex constraints, and thus the OAONP and OAILP techniques presented here allow for satisfying this constraint while maintaining the LP formulation.

The controller must be robust to disturbances, such as linearisation errors, navigation uncertainty, and execution errors. The dynamic terminal box technique presented allows for reducing trajectory overcorrection due to stochastic disturbances, while the terminal quadratic controller allows for a more precise breaking burn, improving the manoeuvre accuracy. Other important disturbances not addressed here include the effect of the geopotential anomaly, solar radiation pressure and atmospheric drag.

The feasibility of implementing MPC in a real rendezvous mission depends on the specific hardware for that mission, since computational complexity is its greatest downside. Also, there is a need to analyse weather MPC indeed offers fuel benefits over the traditional approach, taking into account all disturbances present in a rendezvous mission. This ties into the need for a standard approach for achieving robustness that remains feasible to implement in real-time, which does not yet exist and thus calls for further research.

References

[1] W. Fehse. *Automated Rendezvous and Docking of Spacecraft*. Cambridge University Press, 2003. ISBN 0521824923.

[2] J. Rawlings, D. Mayne, and M. Diehl. *Model Predictive Control: Theory, Computation, and Design*. Nob Hill Publishing, 2nd edition, 2017.

[3] P. Bodin, R. Noteborn, R. Larsson, and C. Chasset. System test results from the GNC experiments on the PRISMA in-orbit test bed. *Acta Astronautica*, 68(7):862 – 872, 2011. ISSN 0094-5765. doi:10.1016/j.actaastro.2010.08.021.

[4] P. Rosa, P. F. Silva, B. Parreira, M. Hagenfeldt, A. Fabrizi, A. Pagano, A. Russo, S. Salvi, M. Kerr, S. Radu, et al. Autonomous close-proximity operations in space: The PROBA-3 rendezvous experiment (P3RVX). In *69th International Astronautical Congress (IAC 2018)*. International Astronautical Federation, 2018.

[5] K. Yamanaka and F. Ankersen. New state transition matrix for relative motion on an arbitrary elliptical orbit. *Journal of Guidance Control and Dynamics*, 25:60–66, 2002. doi:10.2514/2.4875.

[6] F. Ankersen. *Guidance, Navigation, Control and Relative Dynamics for Spacecraft Proximity Maneuvers: Ph.D. Thesis*. Institut for Elektroniske Systemer, 2010. ISBN 9788792328724.

[7] R. Findeisen and F. Allgöwer. An introduction to nonlinear model predictive control. January 2002.

[8] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge university press, 2004.

[9] J. Nocedal and S. Wright. *Numerical Optimization*. Springer Science & Business Media, 2006.

[10] Y. Wang and S. Boyd. Fast model predictive control using online optimization. *IEEE Transactions on control systems technology*, 18(2): 267–278, 2009.

[11] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos. The explicit linear quadratic regulator for constrained systems. *Automatica*, 38(1):3–20, 2002.

[12] E. N. Hartley. A tutorial on model predictive control for spacecraft rendezvous. pages 1355–1361, July 2015. doi:10.1109/ECC.2015.7330727.

[13] A. Richards and J. How. Performance evaluation of rendezvous using model predictive control. *AIAA Guidance, Navigation, and Control Conference and Exhibit*, November 2003. doi:10.2514/6.2003-5507.

[14] L. S. Breger and J. P. How. Safe trajectories for autonomous rendezvous of spacecraft. *Journal of Guidance, Control, and Dynamics*, 31(5):1478–1489, 2008.

[15] J. B. Mueller and R. Larsson. Collision avoidance maneuver planning with robust optimization. In *7th International ESA Conference on Guidance, Navigation and Control Systems, Tralee, County Kerry, Ireland*, June 2008.

[16] F. Gavilan, R. Vazquez, and E. F. Camacho. Chance-constrained model predictive control for spacecraft rendezvous with disturbance estimation. *Control Engineering Practice*, 20(2):111–122, 2012.

[17] M. Tillerson, G. Inalhan, and J. P. How. Co-ordination and control of distributed spacecraft systems using convex optimization techniques. *International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal*, 12(2-3):207–242, 2002.