Variable-Horizon Guidance for Autonomous Rendezvous and Docking to a Tumbling Target

Mirko Leomanni,* Renato Quartullo,† Gianni Bianchini,‡
Andrea Garulli,§ and Antonio Giannitrapani¶
University of Siena, 53100 Siena, Italy

https://doi.org/10.2514/1.G006340

In this paper, the trajectory planning problem for autonomous rendezvous and docking between a controlled spacecraft and a tumbling target is addressed. The use of a variable planning horizon is proposed in order to construct an appropriate maneuver plan, within an optimization-based framework. The involved optimization problem is nonconvex and features nonlinear constraints. The main contribution is to show that such problem can be tackled effectively by solving a finite number of linear programs. To this aim, a specifically conceived horizon search algorithm is employed in combination with a polytopic constraint approximation technique. The resulting guidance scheme provides the ability to identify favorable docking configurations, by exploiting the time-varying nature of the optimization problem endpoint. Simulation results involving the capture of the nonoperational EnviSat spacecraft indicate that the method is able to generate optimal trajectories at a fraction of the computational cost incurred by a state-of-the-art nonlinear solver.

I. Introduction

SPACECRAFT rendezvous and docking (RVD) technologies were tested for the first time in the 1960s within the Gemini and Soyuz programs and later brought to operational status with the advent of manned space stations [1]. In these programs, RVD was accomplished through manual or semi-automated procedures involving a tight cooperation among the vehicles, heavy instrumentation, and man- or ground-in-the-loop interaction to ensure successful maneuvering. In recent years, a demand for new RVD technologies has emerged in the context of small multipurpose servicing vehicles, which will be capable of autonomously performing a number of complex tasks, such as refueling, in-orbit repair/assembly, and orbital debris capture. Several technology demonstration missions have been carried out for servicing a three-axis stabilized spacecraft, including JAXA's ETS-VII [2], NASA's DART [3], AFRL's XSS-11 [4], and DARPA's Orbital Express [5]. However, autonomous RVD to an uncontrolled and possibly tumbling (i.e., rotating) target has yet to be fully demonstrated in orbit, and servicing or capturing an uncooperative target still involves a number of open problems. In particular, there is a need for RVD techniques accounting for rotational motion of the target, which optimize meaningful performance indexes and are easy to implement onboard the spacecraft.

Achieving autonomous RVD involves many complementary operations: inspection, pose estimation, maneuver planning, attitude synchronization, and relative motion control. From a guidance and control perspective, the main challenge to be faced when the target is uncooperative is that the docking point position may vary over time. This leads to the formulation of trajectory planning problems in which both the endpoint and the constraints are time varying. A wide

variety of optimization-based techniques have been proposed in the literature to tackle such type of problems, which in general entail nonlinear optimization methods. In [6], minimum-time and minimum-energy rendezvous trajectories are obtained via direct collocation and validated against the first-order optimality conditions provided by the Pontryagin minimum principle. The approach is refined in [7] to limit the computational burden. In [8], a model predictive control (MPC) strategy based on nonlinear programming (NLP) is proposed. The primary advantage of these NLP methods is that they can account for nonlinearities in the system model, nonconvex constraints, and a free final time. However, NLP is affected by a number of well-known drawbacks, including the lack of convergence guarantees and the requirement of complex solvers. To mitigate these issues, researchers in the field have focused on sequential convex programming (SCP) [9–13]. Within this approach, all nonconvex elements of the trajectory optimization problem are linearized and the resulting convex problem is solved in a trust region where the linearization is accurate. The process is repeated iteratively until a stopping criterion is met. For certain classes of problems, the asymptotic convergence to a local optimum has been proved [14]. However, for general nonlinear problems, the solution sequence may converge to an infeasible trajectory [15]. This is one of the salient limitations of SCP, and an intensive research is ongoing to overcome this obstacle (see, e.g., [16,17]). At present, SCP provides an effective and flexible way to perform rapid trajectory optimization trade studies (see, e.g., [18]).

To facilitate online optimization, a great deal of research has been directed toward problem formulations that are inherently convex. Convex formulations are usually obtained by linearizing the spacecraft relative motion dynamics, exploiting suitable convex approximations of the RVD path constraints, and adopting a fixed planning horizon [19,20]. Most of the studies in this area focus on RVD to a cooperative target, assuming that the docking point is static (see, e.g., [21–23]). Some important contributions have addressed the uncooperative RVD problem. In [24], a convex description of the path constraints is introduced and shown to provide much faster solutions compared with a mixed-integer linear programming (MILP) formulation of such constraints (see, e.g., [25]). In [26], a linear MPC strategy is applied to a planar RVD problem with a tumbling target. This is extended to the three-dimensional case in [27,28]. Although such techniques have nowadays proven to be suitable for implementation onboard a spacecraft (see, e.g., [23]), their application to RVD missions still faces remarkable challenges. For instance, the use of a fixed planning horizon may not be consistent with the mission requirements, as it prevents from taking into account the maneuver time in the cost function. This has been first pointed out in [29,30], where a variable-horizon formulation

is proposed in order to improve the regulation performance, for RVD maneuvers involving a three-axis stabilized target. However, to the best of our knowledge, variable-horizon approaches tailored to the case of tumbling targets have not been developed to date. This is a serious limitation because the planning horizon dictates the optimization problem endpoint (being the docking point time varying) and thus the whole maneuver geometry. Therefore, an improper choice of this parameter can lead to severe performance degradation or even infeasibility. In light of these considerations, and taking into account that the characteristics of the target rotational motion are generally not known beforehand, it is reasonable to expect that the horizon length will have to be tuned on orbit. This makes it necessary to adopt a variable-horizon strategy, to be run in real time onboard the spacecraft.

Variable-horizon optimal control problems can be addressed either in a continuous-time or in a discrete-time setting. In the former, a free-final-time problem is converted into a fixed-final-time one by normalizing the time variable. The resulting optimization problem is nonlinear, even for linear dynamic systems. The latter approach amounts to solving a sequence of fixed-horizon problems, in which linearity of the dynamics is preserved. This simplifies the convergence analysis. However, treating the horizon length as an additional decision variable leads to mixed-integer optimization problems that are difficult to solve (see, e.g., [29]).

The contribution of this paper is to provide an effective discrete-time solution to the variable-horizon guidance problem, for RVD to a tumbling target. The only source of nonconvexity in the proposed formulation is due to the variable horizon, which is weighted in the cost function of the trajectory optimization problem. For any fixed value of the horizon length, the optimization problem is formulated as a linear program (LP). This is achieved by suitably approximating the RVD constraints. In particular, nonconvex keep-out-zone constraints are approximated by a set of linear time-varying inequalities, using a variant of the so-called rotating hyperplane strategy (see, e.g., [20]). Then, the solution to the variable-horizon problem is computed by solving a finite number of LPs. The use of rotating hyperplanes is instrumental to mitigate the computational burden. In fact, for any given horizon length, the proposed approach requires to solve a single LP, whereas alternative (and less conservative) methods based on constraint linearization (see, e.g., [14]) involve the solution of a sequence of convex programs. Another key element of the proposed solution strategy is the construction of a convenient initial guess for the planning horizon, around which a local search is performed. The resulting optimization algorithm ensures convergence to a local optimum in a finite and typically small number of steps. A parametric study shows the advantages of this approach with respect to other solution techniques commonly employed for variable-horizon optimization.

The guidance scheme is demonstrated on two simulated maneuvers inspired by the capture of the nonoperational EnviSat spacecraft [31]. Simulation results show that the method is able to generate safe RVD trajectories to the tumbling target at a fraction of the computational cost incurred by a state-of-the-art nonlinear solver. Moreover, the obtained results indicate that the employed constraint approximation scheme, although conservative, does not lead to a significant loss in terms of maneuver performance. These features make the proposed approach attractive for autonomous RVD applications, in which the solution to the guidance problem must be computed onboard the spacecraft.

The paper is organized as follows. The variable-horizon guidance problem is formulated in Sec. II, and the RVD constraint model is presented in Sec. III. The proposed solution strategy is discussed in Sec. IV. In Secs. V and VI the performance of the method is evaluated numerically and the EnviSat RVD case studies are detailed, and conclusions are drawn in Sec. VII.

A. Notation

The adopted notation is fairly standard. The sets of real, non-negative real, positive integer, and nonnegative integer numbers are denoted by $\mathbb{R}$, $\mathbb{R}^+$, $\mathbb{N}$, and $\mathbb{N}_0$, respectively. The time derivative of a vector x is denoted by $\dot{x}$. The p-norm of x and the direction of x are indicated by $\|x\|_p$ and $\vec{x} = x/\|x\|_2$, respectively. The symbol $\times$ indicates the cross-product operation. The pseudo-inverse of a matrix

M is denoted by $M^\dagger$. The set difference operation is denoted by \ and the empty set is denoted by $\varnothing$. The matrix describing a rotation about the axis $a_x \in \mathbb{R}^3$ by an angle $\theta \in \mathbb{R}$ is denoted by $R(a_x, \theta)$.

II. Problem Formulation

The considered spacecraft maneuvering problem is that of RVD between an actively controlled servicer vehicle and a tumbling target. Herein, the focus is on guidance aspects, i.e., on the generation of safe RVD trajectories to the target. We restrict our attention to the motion of the center of mass of the servicer (sCM) relative to a time-varying docking point modeling the rotational motion of the target. Hence, the servicer attitude motion is neglected. The rationale behind this approach is that, under reasonable assumptions (namely, the availability of omnidirectional thrust), the translational dynamics of the servicer can be decoupled from its own attitude dynamics [32], thus resulting in a simplified guidance algorithm design.

The reference coordinate frame employed in this work is the radial–transverse–normal (RTN) frame centered at the target. The R axis is aligned to the target radius vector, the N axis points toward the target orbit normal, and the T axis completes a right-handed triad. According to standard design rules, the dynamics of the sCM with respect to the center of mass of the target (tCM) are expressed in terms of relative position and velocity, by using the normalized discrete-time Hill–Clohessy–Wiltshire equations [33]:

$$x(k + 1) = Ax(k) + Bu(k)$$
$$= e^{A_c \tau_s} x(k) + \left(\int_{\tau=0}^{\tau_s} e^{A_c \tau} \, d\tau \right) B_c u(k) \qquad (1)$$

where $k \in \mathbb{N}_0$ is the discrete time index, $x(k) \in \mathbb{R}^6$ is the system state, $u(k) \in \mathbb{R}^3$ is the control input, $\tau_s \in \mathbb{R}^+$ is the sampling interval, and

$$A_c = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 3 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & -2 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \end{bmatrix} \qquad B_c = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad (2)$$

The following properties pertain to this representation. The control input is defined as $u(k) = a(k)/a_{\max}$, being $a(k) \in \mathbb{R}^3$ the actual acceleration, expressed in RTN coordinates, and $a_{\max}$ the maximum acceleration deliverable by the servicer along each axis of the RTN frame ($a_{\max}$ is assumed constant). Notice that, in this setting,

$$\|u(k)\|_\infty \leq 1 \qquad (3)$$

The state vector is defined as $x(k) = [x_p^T(k) \quad x_v^T(k)]^T$, where $x_p(k) \in \mathbb{R}^3$ describes the RTN components of the relative position vector multiplied by $\eta^2/a_{\max}$, and $x_v(k) \in \mathbb{R}^3$ describes the RTN components of the relative velocity vector multiplied by $\eta/a_{\max}$, being η the target mean motion. In Eq. (1), a scaled time variable $\tau = \eta t$ is employed, where $t \in \mathbb{R}^+$ is the actual time. Then, $t = k\tau_s/\eta$ at the sampling instants. The linear time-invariant model [Eqs. (1) and (2)] is appropriate for spacecraft in circular orbits.

The docking specifications are modeled by defining a docking axis that is rigidly attached to the target and specifying a suitable docking point along this axis, as illustrated in Fig. 1. The docking point describes the desired position of the sCM upon docking. Taking into account the target rotational motion, it follows that the position of the docking point evolves on a sphere of constant radius. The linear position $p^d \in \mathbb{R}^3$ and velocity $v^d \in \mathbb{R}^3$ of this point relative to the tCM satisfy the differential equation
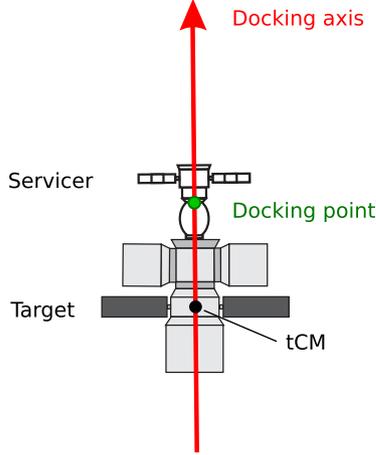
Fig. 1 Definition of the docking point.

$$\dot{\boldsymbol{p}}^d(t) = \boldsymbol{v}^d(t) \quad \boldsymbol{v}^d(t) = \boldsymbol{\omega}(t) \times \boldsymbol{p}^d(t) \qquad (4)$$

where $\boldsymbol{\omega}(t) \in \mathbb{R}^3$ is the instantaneous angular velocity of the target body frame relative to the RTN frame. All vectors in Eq. (4) are expressed in the RTN frame. The time-varying signal $\boldsymbol{\omega}(t)$ is assumed to be known (e.g., determined on orbit before maneuvering). The docking-point trajectory $\boldsymbol{x}^d(k) \in \mathbb{R}^6$ for the guidance problem is obtained by sampling $\boldsymbol{p}^d(t)$, $\boldsymbol{v}^d(t)$ and applying the same normalization used to obtain $\boldsymbol{x}(k)$, resulting in

$$\boldsymbol{x}^d(k) = \begin{bmatrix} \boldsymbol{x}_p^d(k) \\ \boldsymbol{x}_v^d(k) \end{bmatrix} = \begin{bmatrix} \frac{\eta^2}{a_{\max}} \boldsymbol{p}^d\left(\frac{k\tau_s}{\eta}\right) \\ \frac{\eta}{a_{\max}} \boldsymbol{v}^d\left(\frac{k\tau_s}{\eta}\right) \end{bmatrix} \qquad (5)$$

The RVD maneuver objective is stated as follows: steer the state $\boldsymbol{x}(k)$ of system (1) from a given initial condition $\boldsymbol{x}_0$ at the initial time step k_0 to the docking state $\boldsymbol{x}^d(k_0 + N)$ at the final time step $k_0 + N$, while minimizing a tradeoff between fuel consumption and maneuver time, and satisfying suitable state and input constraints. The proposed guidance scheme achieves this objective through the solution of the following variable-horizon discrete-time optimal control problem:

$$\min_{N, \boldsymbol{u}_N} \quad J_N = N + \gamma \|\boldsymbol{u}_N\|_1$$

$$\text{s.t.} \quad \boldsymbol{x}(k_0) = \boldsymbol{x}_0$$

$$\boldsymbol{x}(k+1) = \boldsymbol{A}\boldsymbol{x}(k) + \boldsymbol{B}\boldsymbol{u}(k)$$

$$\boldsymbol{x}_p(k) \in \mathcal{X}(k, N) \quad k = k_0, \ldots, k_0 + N - 1$$

$$\|\boldsymbol{u}_N\|_\infty \leq 1$$

$$\boldsymbol{x}(k_0 + N) = \boldsymbol{x}^d(k_0 + N)$$

$$N \in \mathbb{N} \qquad (6)$$

In problem (6),

$$\boldsymbol{u}_N = \begin{bmatrix} \boldsymbol{u}(k_0 + N - 1) \\ \vdots \\ \boldsymbol{u}(k_0) \end{bmatrix}$$

is a control sequence of variable length N to be optimized. The cost function J_N is the same used in [29] and involves the sum of the number of steps N required to steer $\boldsymbol{x}_0$ toward $\boldsymbol{x}^d(k_0 + N)$ and the normalized fuel consumption $\|\boldsymbol{u}_N\|_1$, which is weighted by the scalar parameter $\gamma \geq 0$. The constraint $\|\boldsymbol{u}_N\|_\infty \leq 1$ enforces a bound for each component of the control sequence according to Eq. (3). The state constraint sets $\mathcal{X}(k, N)$ are allowed to depend explicitly on both k and N. They are assumed to be convex and polytopic for any fixed k and N. The particular structure of $\mathcal{X}(k, N)$ is detailed in Sec. III. Problem (6) is

a nonstandard one in discrete-time optimal control theory, because the number of decision variables and constraints is dictated by the optimization variable N.

III. RVD Constraints

To safely achieve the RVD objective, collisions must be avoided. Moreover, the servicer position must be confined within a suitable visibility region during the final part of the maneuver. A common approach to address such requirements is to introduce a separate set of constraints for the rendezvous and the docking phases. To this aim, we find it convenient to model the safety constraints enforced on the whole maneuver as $\boldsymbol{x}_p(k) \in \mathcal{X}(k, N)$, where

$$\mathcal{X}(k, N) = \begin{cases} \mathcal{R}(k, \lambda_N), & k < k_0 + \lambda_N \\ \mathcal{D}(k), & k \geq k_0 + \lambda_N \end{cases} \qquad (7)$$

In Eq. (7), $\mathcal{R}(k, \lambda_N)$ and $\mathcal{D}(k)$ denote the (time-varying) constraint sets for the rendezvous and the docking phase, respectively, and λ_N is the number of sampling instants allocated to the rendezvous phase, within the planning horizon. The latter is treated as a horizon-dependent variable, by setting

$$\lambda_N = N - N_d \qquad (8)$$

where the parameter $N_d \in \mathbb{N}$ indicates a predefined number of time steps allocated to the docking phase, at the end of the planning horizon. Note that N_d is fixed a priori because docking operations must be usually completed within a fixed amount of time.

A. Rendezvous Constraints

In the rendezvous phase, collision avoidance constraints are typically modeled by enforcing a keep-out zone of radius r (see Fig. 2), i.e.,

$$\|\boldsymbol{x}_p(k)\| \geq r \qquad (9)$$

The above constraint is nonconvex and hence not compatible with Eqs. (6) and (7) [$\mathcal{X}(k, N)$ is assumed to be convex]. One way to convexify Eq. (9) is to employ the so-called rotating-hyperplane method [20]. Within this method, the hyperplane rotation rate is treated as a parameter to be tuned heuristically. The heuristic proposed herein amounts to parameterizing the hyperplane rotation in terms of both the time index k and the horizon length N. In particular, the following safety constraint set is enforced:

$$\mathcal{R}(k, \lambda_N) = \{\boldsymbol{\xi} \in \mathbb{R}^3 : \boldsymbol{\xi}^T \vec{\boldsymbol{\nu}}(k, \lambda_N) \geq r\} \qquad (10)$$

which defines a half-space by means of a separating plane passing through the point $r\vec{\boldsymbol{\nu}}(k, \lambda_N)$ with outward unit normal $\vec{\boldsymbol{\nu}}(k, \lambda_N)$. The vector $\vec{\boldsymbol{\nu}}(k, \lambda_N)$ is obtained from the geodesic joining the projections on the unit sphere of the initial position $\boldsymbol{x}_p(k_0)$ and the reference position at the end of the rendezvous phase $\boldsymbol{x}_p^d(k_0 + \lambda_N)$, as

$$\vec{\boldsymbol{\nu}}(k, \lambda_N) = \boldsymbol{R}(\boldsymbol{a}_x(\lambda_N), \theta(k, \lambda_N))\vec{\boldsymbol{x}}_p(k_0) \qquad (11)$$

where

$$\boldsymbol{a}_x(\lambda_N) = \vec{\boldsymbol{x}}_p(k_0) \times \vec{\boldsymbol{x}}_p^d(k_0 + \lambda_N) \qquad (12)$$

and

$$\theta(k, \lambda_N) = \frac{k - k_0}{\lambda_N} \arccos(\vec{\boldsymbol{x}}_p^T(k_0)\vec{\boldsymbol{x}}_p^d(k_0 + \lambda_N)) \qquad (13)$$

The set $\mathcal{R}(k, \lambda_N)$ in Eqs. (10–13) has the property that the constraint $\boldsymbol{x}_p(k) \in \mathcal{R}(k, \lambda_N)$ implies Eq. (9). With respect to [19,20], the half-space $\mathcal{R}(k, \lambda_N)$ has been designed in such a way that its gradual rotation favors the transition from the initial position $\boldsymbol{x}_p(k_0)$ to the reference position $\boldsymbol{x}_p^d(k_0 + \lambda_N)$, where the docking phase will start. Notice that the time scale of such rotation, and consequently the conservativeness of
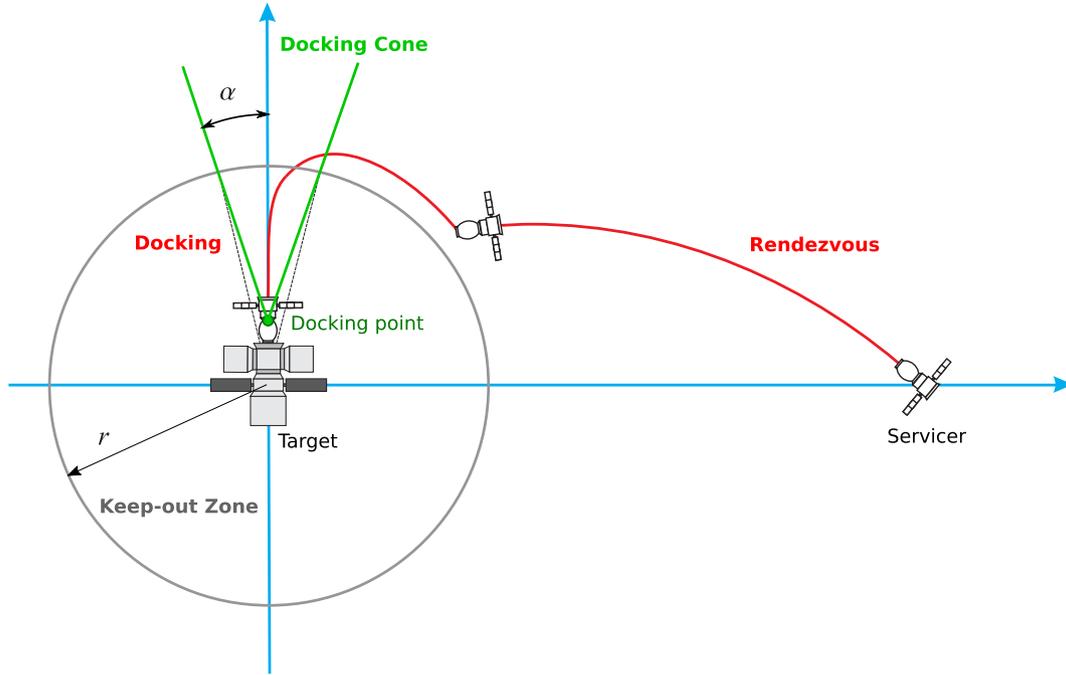
Fig. 2   Illustration of a typical rendezvous and docking maneuver (adapted from [1]). For the sake of illustration, the docking point is assumed to be static.

the keep-out-zone approximation, is affected by the horizon-dependent variable λ_N. For any fixed $\boldsymbol{x}_p^d$, the approximation gets better for increasing values of λ_N (see Fig. 3), and thus of N [see Eq. (8)].

B. Docking Constraints

To define the constraints for the docking phase, let us consider the time-varying set

$$\mathcal{C}(k) = \left\{ \boldsymbol{\xi} \in \mathbb{R}^3 : \|\boldsymbol{\xi} - [\boldsymbol{\xi}^T \vec{\boldsymbol{x}}_p^d(k)]\vec{\boldsymbol{x}}_p^d(k)\|_2 \le \tan(\alpha)[\boldsymbol{\xi} - \boldsymbol{x}_p^d(k)]^T \vec{\boldsymbol{x}}_p^d(k) \right\} \tag{14}$$

which consists of a cone stemming from the docking point $\boldsymbol{x}_p^d(k)$, whose half-angle amplitude is defined by the parameter α (see again Fig. 2). The visibility condition during docking can be defined as $\boldsymbol{x}_p(k) \in \mathcal{C}(k)$, which represents a convex quadratic constraint. For the sake of computational performance, however, it is convenient to

devise a polyhedral constraint set implying $\boldsymbol{x}_p(k) \in \mathcal{C}(k)$. To accomplish this, we exploit the inequality

$$\|\boldsymbol{y}\|_2 = \|\boldsymbol{T}\boldsymbol{y}\|_2 \le \sqrt{\rho}\|\boldsymbol{T}\boldsymbol{y}\|_\infty \tag{15}$$

which holds for any $\boldsymbol{y} \in \mathbb{R}^3$ and any orthogonal matrix $\boldsymbol{T} \in \mathbb{R}^{3\times3}$, where ρ is the number of nonzero elements in $\boldsymbol{T}\boldsymbol{y}$. Let us consider the (orthogonal) rotation matrix $\boldsymbol{T}(k)$ that takes $\boldsymbol{x}_p^d(k)$ to a basis vector of the RTN frame. Then, $\boldsymbol{T}(k)\{\boldsymbol{\xi} - [\boldsymbol{\xi}^T \vec{\boldsymbol{x}}_p^d(k)]\vec{\boldsymbol{x}}_p^d(k)\}$ has at most two nonzero elements, since $\boldsymbol{\xi} - [\boldsymbol{\xi}^T \vec{\boldsymbol{x}}_p^d(k)]\vec{\boldsymbol{x}}_p^d(k)$ and $\vec{\boldsymbol{x}}_p^d(k)$ are orthogonal. Therefore, introducing the polyhedral set

$$\mathcal{D}(k) = \left\{ \boldsymbol{\xi} \in \mathbb{R}^3 : \|\boldsymbol{T}(k)(\boldsymbol{\xi} - [\boldsymbol{\xi}^T \vec{\boldsymbol{x}}_p^d(k)]\vec{\boldsymbol{x}}_p^d(k))\|_\infty \right.$$
$$\left. \le \frac{\tan(\alpha)}{\sqrt{2}}[\boldsymbol{\xi} - \boldsymbol{x}_p^d(k)]^T \vec{\boldsymbol{x}}_p^d(k) \right\} \tag{16}$$
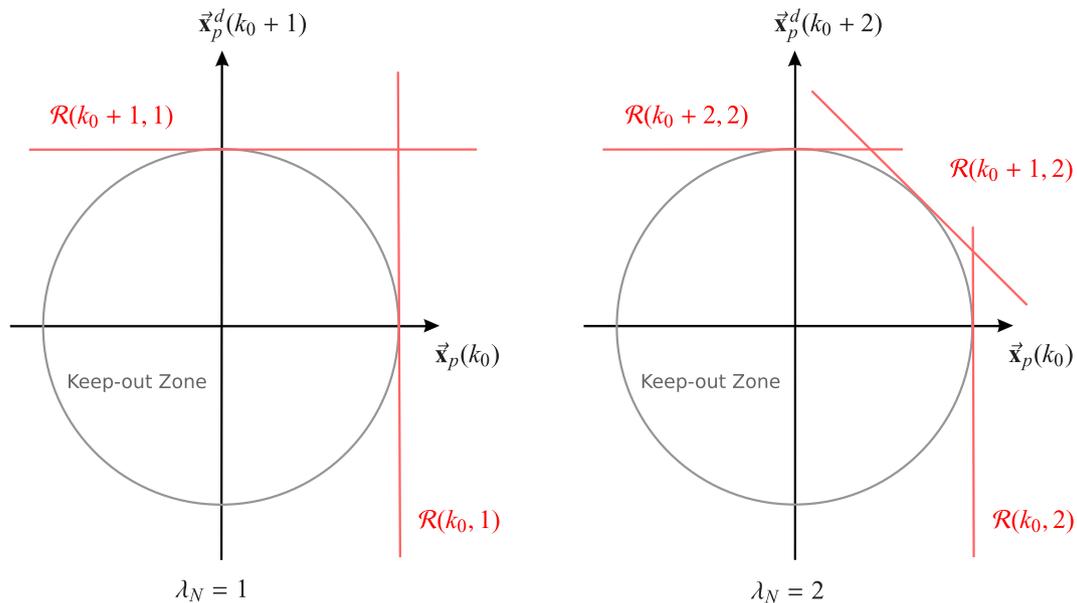
Fig. 3   Illustration of the keep-out-zone approximation scheme on a two-dimensional example, for different values of λ_N.
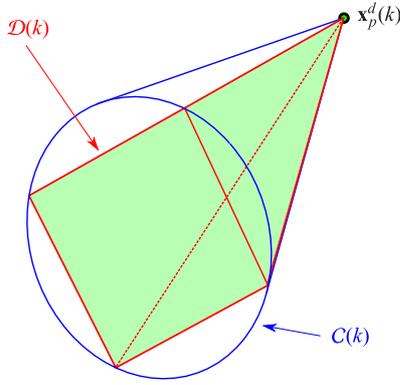
Fig. 4 Illustration of the polyhedral set $\mathcal{D}(k)$ used to approximate the docking cone $\mathcal{C}(k)$.

Fig. 5 Profile of J_N^* versus N for the RVD scenario detailed in Sec. V, with $\gamma = 4$. The presence of multiple local minima is due to the target rotation. The global optimum N^* is marked by an asterisk. The problem is infeasible for $N \leq 25$.

we get from Eq. (15) that $\mathcal{D}(k) \subseteq \mathcal{C}(k)$, as illustrated in Fig. 4. Hence, the linear constraint set $\boldsymbol{x}_p(k) \in \mathcal{D}(k)$ implies the visibility condition $\boldsymbol{x}_p(k) \in \mathcal{C}(k)$. In this work, the matrix $\boldsymbol{T}(k)$ is chosen as

$$\boldsymbol{T}(k) = \boldsymbol{R}(\vec{\boldsymbol{x}}_p^d(k) \times [\,1\,0\,0\,]^T, \arccos([\,1\,0\,0\,]\vec{\boldsymbol{x}}_p^d(k))) \qquad (17)$$

Although not discussed here for conciseness, other types of maneuver constraints naturally fit the proposed formulation. For instance, plume-impingement constraints can be accommodated for by enforcing a time-varying bound on the control input magnitude (see [24]). Note that plume impingement is less likely to occur when the docking point is spinning, as the servicer thrust is mostly directed away from the target during the final approach, in order to compensate for centrifugal effects.

IV. Solution Strategy

Problem (6) is nonconvex due to the integer optimization variable N. Nevertheless, it can readily be tackled by solving a sequence of linear programming problems of the form

$$\min_{\boldsymbol{u}_N} \quad J_N = N + \gamma \|\boldsymbol{u}_N\|_1$$

$$\begin{aligned}
\text{s.t.} \quad & \boldsymbol{x}(k_0) = \boldsymbol{x}_0 \\
& \boldsymbol{x}(k+1) = \boldsymbol{A}\boldsymbol{x}(k) + \boldsymbol{B}\boldsymbol{u}(k) \\
& \boldsymbol{x}_p(k) \in \mathcal{X}(k, N) \quad k = k_0, \ldots, k_0 + N - 1 \\
& \|\boldsymbol{u}_N\|_\infty \leq 1 \\
& \boldsymbol{x}(k_0 + N) = \boldsymbol{x}^d(k_0 + N) \qquad (18)
\end{aligned}$$

where the planning horizon N is now a fixed parameter. We denote the minimizer of Eq. (18) by $\boldsymbol{u}_N^*$, and the corresponding optimal cost by J_N^*. If the problem is infeasible, the cost is set by definition to $J_N^* = \infty$. To find an optimal solution of problem (6), one can solve Eq. (18) to obtain the function $J_N^*:\mathbb{N} \to \mathbb{R}^+$, and then minimize J_N^* with respect to $N \in \mathcal{I} = \{1, \ldots, N_{ub}\}$, where N_{ub} is an upper bound on the optimal horizon N^* of Eq. (6). Being J_N unbounded for $N \to \infty$, the upper bound N_{ub} is guaranteed to be finite.

The profile of J_N^* versus N is reported in Fig. 5 for an example of RVD maneuver. It can be seen that the horizon length N has a profound impact on maneuver performance, and that optimizing over N is an inherently nonconvex problem. Hence, searching for the global optimum N^* turns out to be as hard as solving Eq. (18) for all $N \in \mathcal{I}$. To rule out some of the values of N, the following result can be employed.

Proposition 1: Consider the linear system $\boldsymbol{x}(k+1) = \boldsymbol{A}\boldsymbol{x}(k) + \boldsymbol{B}\boldsymbol{u}(k)$, where $\boldsymbol{x}(k) \in \mathbb{R}^n$ and $\boldsymbol{u}(k) \in \mathbb{R}^m$, and let $\boldsymbol{x}^d(k_0 + N) \in \mathbb{R}^n$ be a target state to be reached from the initial state $\boldsymbol{x}(k_0) = \boldsymbol{x}_0$. Define the N-step reachability matrix $\boldsymbol{R}_N = [\boldsymbol{B}\,\boldsymbol{A}\boldsymbol{B} \ldots \boldsymbol{A}^{N-1}\boldsymbol{B}]$ and the unconstrained minimum energy (least-squares) input sequence $\boldsymbol{e}_N = \boldsymbol{R}_N^\dagger(\boldsymbol{x}^d(k_0 + N) - \boldsymbol{A}^N \boldsymbol{x}_0)$. Moreover, let $\mathcal{I} \subseteq \mathbb{N}$ and

$$\mathcal{F} = \left\{ N \in \mathcal{I} : \begin{array}{l} \boldsymbol{x}^d(k_0 + N) = \boldsymbol{A}^N \boldsymbol{x}_0 + \boldsymbol{R}_N \boldsymbol{e}_N \\ \|\boldsymbol{e}_N\|_2 \leq \sqrt{mN} \end{array} \right\} \qquad (19)$$

Then, the feasibility problem

$$\begin{aligned}
\text{find} \quad & \boldsymbol{u}_N \\
\text{s.t.} \quad & \boldsymbol{x}(k+1) = \boldsymbol{A}\boldsymbol{x}(k) + \boldsymbol{B}\boldsymbol{u}(k) \\
& \boldsymbol{x}(k_0) = \boldsymbol{x}_0, \qquad \boldsymbol{x}(k_0 + N) = \boldsymbol{x}^d(k_0 + N) \\
& \|\boldsymbol{u}_N\|_\infty \leq 1 \qquad (20)
\end{aligned}$$

has no solution for $N \in \{\mathcal{I} \setminus \mathcal{F}\}$.

Proof: The equality in Eq. (19) stems from the equality constraints of problem (20). To prove necessity of the inequality in Eq. (19), observe that $\boldsymbol{e}_N$ is the minimum energy input sequence that drives the system from state $\boldsymbol{x}_0$ at time k_0 to state $\boldsymbol{x}^d$ at time $k_0 + N$. Hence,

$$\|\boldsymbol{e}_N\|_2 \leq \|\boldsymbol{u}_N\|_2 \leq \sqrt{mN}\|\boldsymbol{u}_N\|_\infty \qquad (21)$$

A necessary condition for problem (20) to be feasible is that $\|\boldsymbol{u}_N\|_\infty \leq 1$. This can happen only if $\|\boldsymbol{e}_N\|_2 \leq \sqrt{mN}$. $\qquad \square$

Remark 1: By applying Proposition 1 to system (1), it follows that problem (18) has no solution for $N \in \{\mathcal{I} \setminus \mathcal{F}\}$. Moreover, the set $\mathcal{I}$ in Eq. (19) is chosen as $\mathcal{I} = \{1, \ldots, N_{ub}\}$. Then, the smallest element of $\mathcal{F}$ is a lower bound on the minimum horizon $N = N_{\min}$ for which problem (18) is feasible.

The minimum energy control for different values N of the horizon length can be evaluated very efficiently, given that the reachability matrix $\boldsymbol{R}_N$, its pseudo-inverse $\boldsymbol{R}_N^\dagger$, and matrix $\boldsymbol{A}^N$ can be precomputed offline. However, solving problem (18) for all $N \in \mathcal{F}$ is still prohibitively complex when the number of elements in $\mathcal{F}$ is large. To mitigate this issue, one has to settle for a locally optimal solution. More specifically, our aim is to solve a small subset of problems (18), in which the choice of N is guided by a local search within the set $\mathcal{F}$. To this purpose, the following three-step solution method is proposed:

1: Compute an initial guess N_1 of the horizon length by evaluating the cost associated with the minimum energy control $\boldsymbol{e}_N$, i.e.,

$$N_1 = \operatorname*{argmin}_{N \in \mathcal{F}}[N + \gamma \|\boldsymbol{e}_N\|_1] \qquad (22)$$

If multiple minima are found, take the one with the smallest N (among equivalent solutions, the one featuring the smallest N is preferred). Notice that $N = N_1$ is not guaranteed to be feasible for problem (18).

2: Starting from $N = N_1$, perform a local search within the set $\mathcal{F}$ until a feasible solution to problem (18) is found (see Algorithm 1). This step amounts to solving a sequence of problems (18) with

Algorithm 1:   Returning a local optimum $\hat{N}$ of problem (6)

---

let $\mathcal{F}(q)$ denote the qth largest integer in the set $\mathcal{F}$

if $\mathcal{F} = \varnothing$ **then**
    return problem (6) infeasible
else   ▷ Step 1.
    let $N_1 = \underset{N \in \mathcal{F}}{\operatorname{argmin}}[N + \gamma \|E_N\|_1]$ ▷ Step 2.
    let q_1 be such that $\mathcal{F}(q_1) = N_1$, let $i = 0$
    repeat
        solve problem (18) with $N = \mathcal{F}(q_1 + i)$ and $N = \mathcal{F}(q_1 - i)$,
        $i = i + 1$
    until a feasible solution is found or both the endpoints of $\mathcal{F}$ are reached
    if a feasible solution is found **then**
        let N_2 be the horizon length
    else if two feasible solutions are found at $i = \zeta$ **then**
        if $J^*_{\mathcal{F}(q_1+\zeta)} < J^*_{\mathcal{F}(q_1-\zeta)}$ **then**
            let $N_2 = \mathcal{F}(q_1 + \zeta)$
        else
            let $N_2 = \mathcal{F}(q_1 - \zeta)$
    end if
    else
        return problem (6) infeasible
    end if
                                                                            ▷ Step 3.
    let q_2 be such that $\mathcal{F}(q_2) = N_2$, let $i = 1$
    if $N_2 > N_1$ **then**
        repeat
            solve problem (18) with $N = \mathcal{F}(q_2 + i)$, $i = i + 1$
        until an index ς is found such that $J^*_{\mathcal{F}(q_2+\varsigma)} > J^*_{\mathcal{F}(q_2+\varsigma-1)}$
        let $\hat{N} = \mathcal{F}(q_2 + \varsigma - 1)$
        return $\hat{N}, J^*_{\mathcal{F}(q_2+\varsigma-1)}$
    else if $N_2 < N_1$ **then**
        repeat
            solve problem (18) with $N = \mathcal{F}(q_2 - i)$, $i = i + 1$
        until an index $i = \varsigma$ is found such that $J^*_{\mathcal{F}(q_2-\varsigma)} > J^*_{\mathcal{F}(q_2-\varsigma+1)}$
        let $\hat{N} = \mathcal{F}(q_2 - \varsigma + 1)$
        return $\hat{N}, J^*_{\mathcal{F}(q_2-\varsigma+1)}$
    else
        let $\underline{J} = \min\{J^*_{\mathcal{F}(q_1-1)}, J^*_{\mathcal{F}(q_1)}, J^*_{\mathcal{F}(q_1+1)}\}$
        if $\underline{J} = J^*_{\mathcal{F}(q_1-1)}$ **then**
            proceed as for $N_2 < N_1$
        else if $\underline{J} = J^*_{\mathcal{F}(q_1+1)}$ **then**
            proceed as for $N_2 > N_1$
        else
            let $\hat{N} = N_1$
            return $\hat{N}, J^*_{\mathcal{F}(q_1)}$

---

different values of N in a neighborhood of N_1. If a feasible solution is found, denote by N_2 the corresponding horizon length. If no feasible solution is found for any $N \in \mathcal{F}$, then mark problem (6) as infeasible.

3: If $N_2 > N_1$, solve a sequence of problems (18) with increasing horizon length within the set $\mathcal{F}$, starting from $N = N_2$. Stop when the optimal cost of Eq. (18) does not decrease anymore. Similarly, if $N_2 < N_1$, apply the same procedure but with decreasing horizon length. If $N_2 = N_1$, decide whether to increase or decrease the horizon length by comparing the cost $J^*_{N_1}$ with two neighboring solutions of Eq. (18) within the set $\mathcal{F}$. The optimized horizon length resulting from this step is denoted by $\hat{N}$.

Algorithm 1 formalizes the proposed method.

Let us remark that steps 1–3 of Algorithm 1 are motivated by the practical need to trade off performance and computational efficiency. In step 1, the initial guess N_1 is obtained based on the observation that in the domain $\mathcal{F}$ the profile of $N + \gamma \|e_N\|_1$ is often close to that of $N + \gamma \|u^*_N\|_1$, provided by Eq. (18). In steps 2 and 3, a local search is performed in the neighborhood of N_1. By virtue of step 2, a feasible solution of problem (6) is always found, if it exists. Moreover, according to step 3, convergence to a local minimum $\hat{N}$ of J^*_N is

guaranteed. Global optimality (i.e., $\hat{N} = N^*$) is ensured if one of the following conditions is met:
    i) J^*_N has no local minima in the domain $\mathcal{F}$.
    ii) $\gamma = 0$, in which case N_1 is a lower bound on $N_{\min}$ [see Eq. (22) and Remark 1], and steps 2 and 3 are guaranteed to find $\hat{N} = N^* = N_{\min}$.

It is also worth observing that the proposed approach can be easily extended to general linear time-varying systems. Such an extension may allow one to deal with RVD maneuvers featuring elliptical or J_2-perturbed orbits (see, e.g., [30,34]).

The performance of Algorithm 1 is evaluated in the next section.

V.   Search Algorithm Validation

A qualitative assessment of the proposed solution strategy has been carried out by testing Algorithm 1 on a specific RVD scenario, for different values of the parameter γ in Eq. (18). The motivation of this parametric study is that larger values of γ usually correspond to longer planning horizons and thus to an increased computational load. This allows one to draw some conclusions regarding the computational feasibility of the method. The RVD maneuver parameters are summarized in Table 1, where t_0 denotes the initial time. They are consistent with the specifications of a small satellite mission tailored to the removal of a debris object in low Earth orbit (see, e.g., [22]).

Within this setup, the performance of Algorithm 1 has been compared with that of full enumeration and binary search methods. The full enumeration approach amounts to solving problem (18) for all $N \in \mathcal{I}$, and it is guaranteed to find the global optimum N^* of problem (6). The binary search method minimizes J^*_N with respect to N by bisection on the interval $\mathcal{I}$. Its complexity is logarithmic in N_{ub}, and in general it returns a local minimum of J^*_N. In this study, the horizon upper bound is set to $N_{ub} = 128$ samples.

The obtained results are reported in Figs. 6–8 for different values of γ in the range $\gamma \in [1, \quad 15]$. Figure 6 shows the value function J^*_N, evaluated at the horizon length returned by the three compared methods. It can be seen that the cost $J^*_{\hat{N}}$ incurred by Algorithm 1 is either equal or very close to the optimal cost $J^*_{N^*}$ obtained via full enumeration. Conversely, the cost $J^*_{N_{bs}}$ provided by the binary search algorithm is always far from global optimality except for small values of γ. This is not surprising, because the binary search method is known to work well for the minimization of unimodal functions, whereas the unimodality condition is not met for the problem at hand (this is evident in Fig. 5, which depicts the profile of J^*_N corresponding to the parameters in Table 1, for $\gamma = 4$). Figures 7 and 8 report, respectively, the optimized horizon length and the normalized fuel consumption $\|u^*_N\|_1$ incurred by each solution strategy. It can be seen that the binary search solution N_{bs} tends to underestimate N^*, while requiring a much higher fuel consumption with respect to the other approaches. Conversely, the horizon length $\hat{N}$ provided by Algorithm 1 is very close to N^* in many instances. When $\hat{N} > N^*$, the corresponding fuel cost is such that $\|u^*_{\hat{N}}\|_1 \leq \|u^*_{N^*}\|_1$. Figure 7 also depicts the initial guess N_1 defined by Eq. (22). It can be seen that the initial guess often falls reasonably close

Table 1   RVD test maneuver parameters

Parameter	Value
Initial docking point position	$p^d(t_0) = [1, 0, 0]^T$ m
Target angular velocity	$\omega(t) = [0, 0, 0.01]^T$ rad/s
Target mean motion	$\eta = 0.001$ rad/s
Servicer maximum acceleration	$a_{\max} = 0.001$ m/s²
Sampling interval	$\tau_s = 2\pi/256$ rad/sample
Initial relative position	$\frac{a_{\max}}{\eta^2} x_p(k_0) = [0, -100, 0]^T$ m
Initial relative velocity	$\frac{a_{\max}}{\eta} x_v(k_0) = [0, 0, 0]^T$ m/s
Docking cone half-angle	$\alpha = 20$ deg
Keep-out zone radius	$\frac{a_{\max}}{\eta^2} r = 5$ m
Docking phase duration	$\frac{\tau_s}{\eta} N_d = 220.9 s (N_d = 9)$
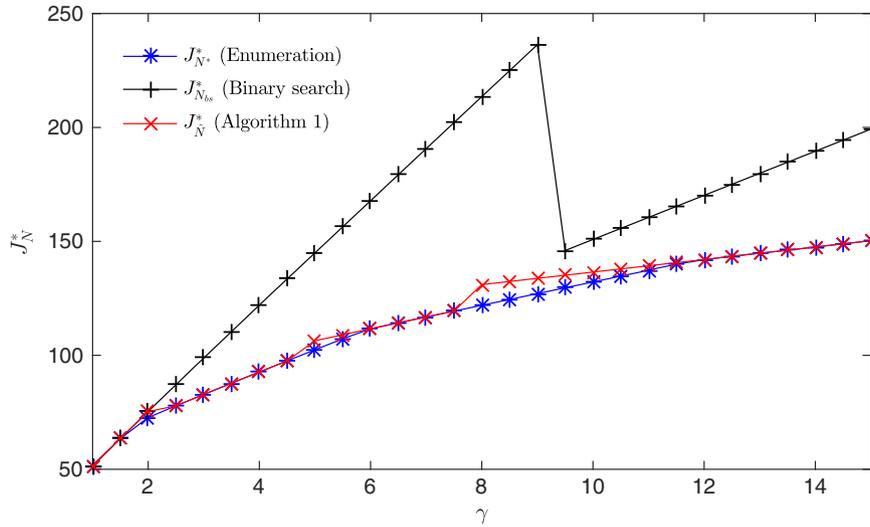
Fig. 6   Maneuver cost incurred by the considered search strategies for different values of γ.
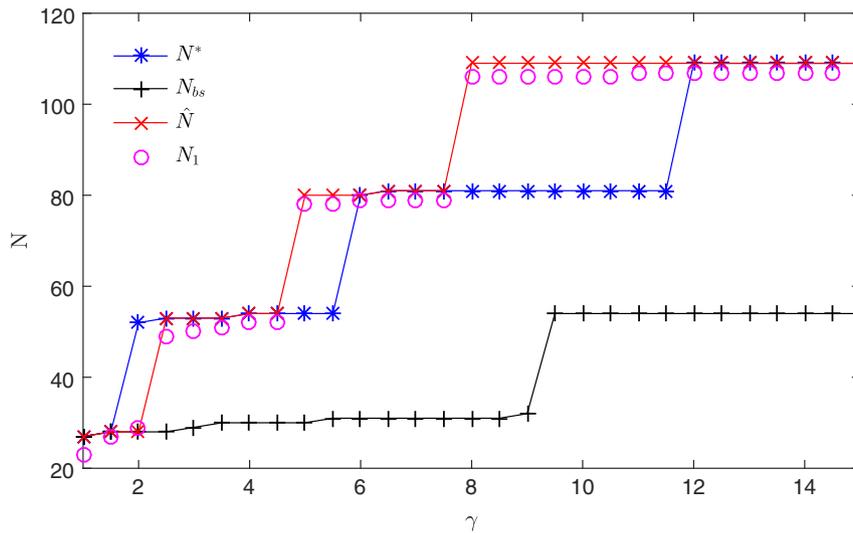
Fig. 7   Optimized planning horizon provided by the considered search strategies for different values of γ. The initial guess N_1 of Algorithm 1 is also reported.
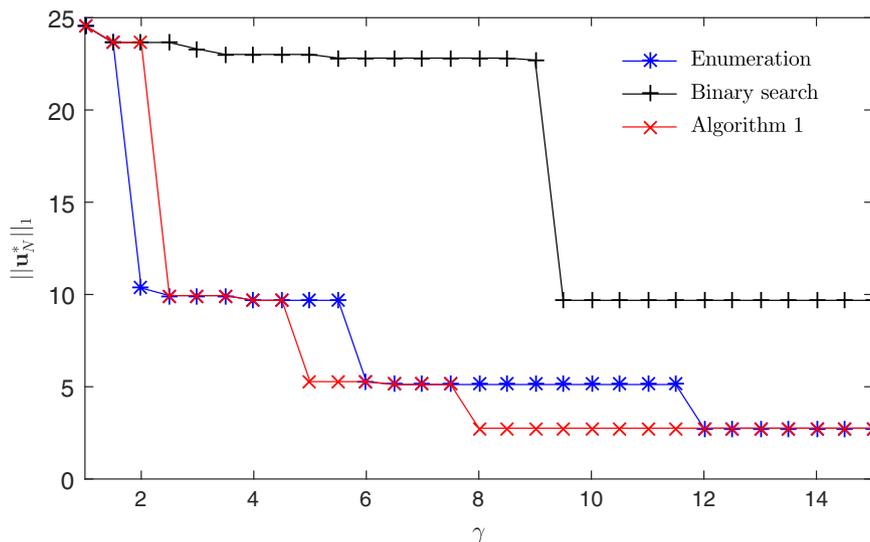
Fig. 8   Fuel consumption incurred by the considered search strategies for different values of γ.

Table 2   Computational burden of binary search and Algorithm 1

CPU time	Binary search	Algorithm 1
Minimum, s	0.069	0.020
Average, s	0.094	0.104
Maximum, s	0.126	0.165

to a local optimum. Table 2 reports the minimum, maximum, and average CPU time over the considered values of γ for the binary search and Algorithm 1 solutions, showing that the computational burden of the two methods is on a similar level. This is a remarkable result, given that $\hat{N} \gg N_{bs}$ in most problem instances (see again Fig. 7). The CPU time of the full enumeration procedure is by far higher than that of these two approaches and amounts to approximately 3 s, regardless of the value of γ. From these results, it can be concluded that the proposed solution strategy provides an excellent tradeoff between performance and computational efficiency. Note that the CPU time reported in Table 2 for Algorithm 1 includes the time required to compute the initial guess N_1.

Finally, it is worth recalling that variable-horizon problems can often be cast as a MILP; see, e.g., [29]. A MILP formulation of problem (6) has been tested, but the obtained results turned out to be unsatisfactory. In part, this is because the feasible region (7) is parameterized by an explicit function of N. To cope with this issue, one has to construct a MILP including all realizations of $\mathcal{X}(k, N)$, obtained for $N \in \mathcal{I}$ and $k = k_0, \ldots, k_0 + N - 1$, resulting in a number of state constraints which is proportional to N_{ub}^2. Hence, the problem rapidly becomes untractable as N_{ub} grows. In the considered scenario, even by removing the state constraints, the MILP solution time is in the order of seconds. All computations have been performed on a standard laptop, via a direct call from MATLAB of the commercial solver Gurobi [35].

VI.   RVD to the EnviSat Platform

In recent years, a number of studies have focused on in-orbit servicing missions dedicated to capturing and de-orbiting the European Space Agency (ESA) EnviSat platform; see, e.g., [36]. The EnviSat operational life ended on April 8, 2012, following the unexpected loss of contact with the spacecraft. After this event, the spacecraft lost the ability to hold its Earth-pointing attitude and started to tumble. Because of its huge size and its particular orbital configuration, EnviSat is currently regarded as a potential trigger for space debris proliferation in low Earth orbit. In the following, the proposed guidance scheme is demonstrated on an RVD scenario inspired by the capture of EnviSat.

A schematic view of EnviSat is reported in Fig. 9. The spacecraft was not designed with servicing in mind, and features an elongated shape with many protruding elements. Consequently, the determination of a
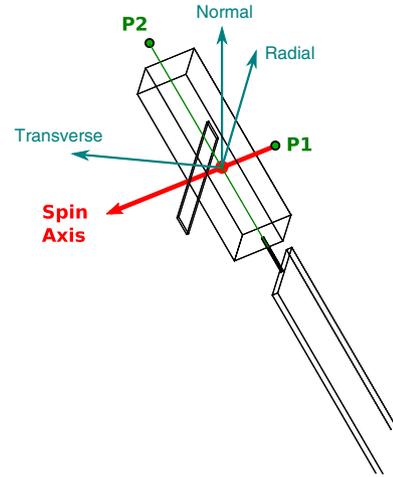
Fig. 10   Characterization of the docking points (P1, P2) and of the spin axis in the RTN frame at the initial time t_0.

suitable docking configuration is nontrivial. Following a review of the literature available on the topic (see, e.g., [37,38]), two favorable docking points have been identified: the first (P1) is located above the center of mass (tCM), in the direction opposite to the ASAR antenna; the second (P2) lies along the spacecraft long axis, in the direction opposite to the solar panel, as depicted in Fig. 10. It is worth recalling that these points describe the desired position of the sCM upon

Table 3   EnviSat RVD maneuver parameters

Parameter	Value
Initial docking point position	P1: $\boldsymbol{p}^d(t_0) = [-0.0360, -2.6451, 1.4149]^T$ m P2: $\boldsymbol{p}^d(t_0) = [-0.1683, 3.5384, 6.6107]^T$ m
Initial angular velocity	$\boldsymbol{\omega}(t_0) = [0.0003, 0.0252, -0.0145]^T$ rad/s
Target mean motion	$\eta = 0.001045$ rad/s
Servicer maximum acceleration	$a_{\max} = 0.005$ m/s^2
Sampling interval	$\tau_s = 2\pi/512$ rad/sample
Initial relative position	$\frac{a_{\max}}{\eta^2}\boldsymbol{x}_p(k_0) = [0, -200, 0]^T$ m
Initial relative velocity	$\frac{a_{\max}}{\eta}\boldsymbol{x}_v(k_0) = [0, 0, 0]^T$ m/s
Docking cone half-angle	$\alpha = 20$ deg
Keep-out zone radius	$\frac{a_{\max}}{\eta^2} r = 22$ m
Docking phase duration	$\frac{\tau_s}{\eta} N_d = 187.8$ s ($N_d = 16$)
Weighting parameter γ	$\gamma = 4$
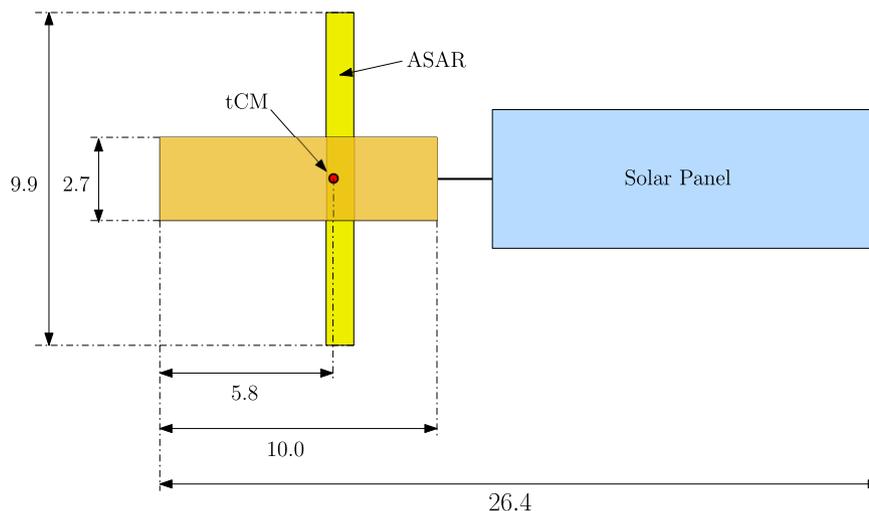
Fig. 9   EnviSat spacecraft layout. The length of the various components is reported in meters.

docking (see Fig. 1). To account for the geometrical configuration of the servicer, a clearance of 1.5 m is left between the docking points and the nearby EnviSat surfaces, similarly to what done in [37]. Another important modeling issue is the characterization of the spin axis. It is generally agreed (see, e.g., [39]) that the EnviSat spin axis is approximately fixed with respect to the body frame, and aligned with the vector joining P1 and the tCM, as depicted in Fig. 10. Moreover, over the typical RVD maneuver time scales, one can safely assume that the target angular velocity remains approximately constant in the inertial frame. Notice that, under this assumption, the angular velocity vector does still vary when seen from the RTN frame, due the rotation of the RTN frame relative to the inertial frame. More specifically, the vector $\boldsymbol{\omega}(t)$ in

Eq. (4) (which is expressed in RTN coordinates) evolves according to

$$\boldsymbol{\omega}(t) = \begin{bmatrix} \cos(\eta(t-t_0)) & \sin(\eta(t-t_0)) & 0 \\ -\sin(\eta(t-t_0)) & \cos(\eta(t-t_0)) & 0 \\ 0 & 0 & 1 \end{bmatrix} \boldsymbol{\omega}(t_0) \quad (23)$$

where $\boldsymbol{\omega}(t_0)$ is the angular velocity of the target body frame relative to the RTN frame at the initial time t_0. In this work, the docking-point trajectory profile is generated by propagating equations (4) and (23). Based on the results in [39], the EnviSat spin period is taken as 220 s,
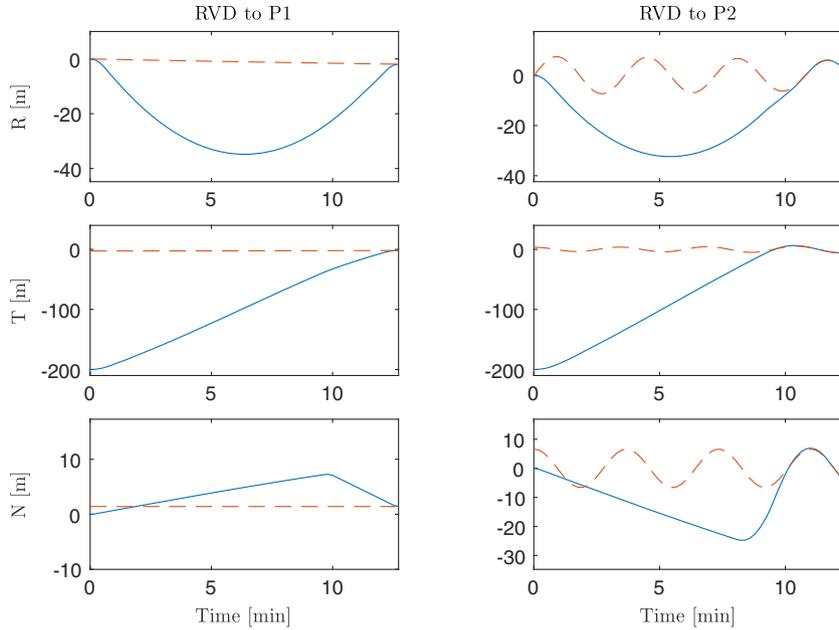
Fig. 11 Radial (R), transverse (T), and normal (N) components of the trajectories of the target docking-point (dashed) and of the servicer (solid) for RVD to P1 (left) and P2 (right).
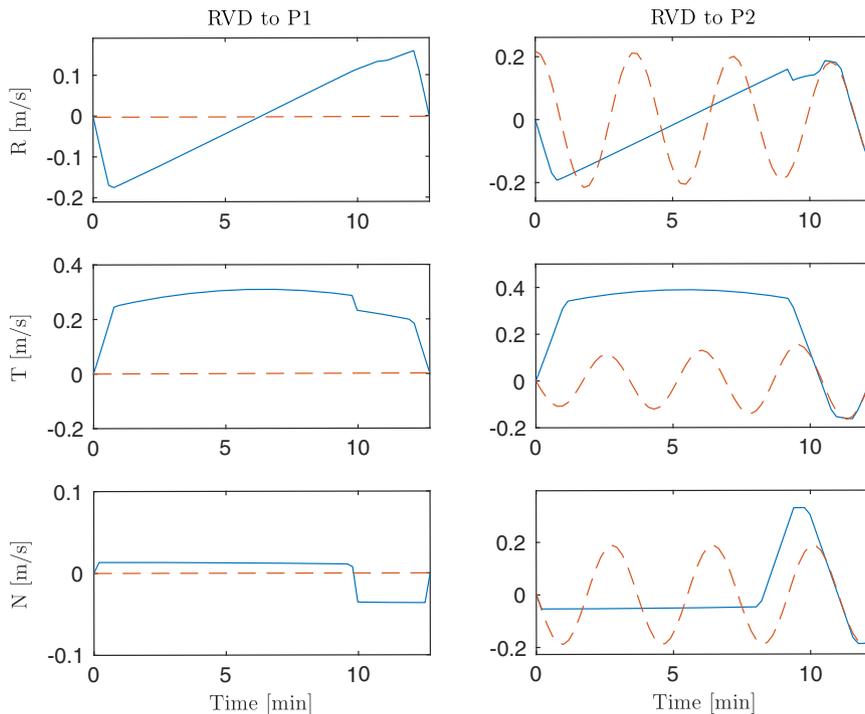
Fig. 12 Radial (R), transverse (T), and normal (N) components of the velocities of the target docking-point (dashed) and of the servicer (solid) for RVD to P1 (left) and P2 (right).

corresponding to a constant angular rate of $\|\boldsymbol{\omega}(t)\|_2 = \|\boldsymbol{\omega}(t_0)\|_2 = 0.029$ rad/s. Reasonable values of a_{max} for the problem at hand may range from 10^{-3} to 10^{-1} m/s^2, on a rough estimate. A relatively low value of $a_{max} = 0.005$ m/s^2 is selected. This choice is made in order to showcase the proposed method on a challenging optimization problem involving a low control authority. Moreover, the resulting maneuver plan may be realized by using small thrusters, which are lighter and more accurate than larger ones. The half-angle amplitude of the docking cone is set as $\alpha = 20$ deg, which is in line with the values in [8,29]. The maneuver parameters are summarized in Table 3.

Problem (6) has been solved using Algorithm 1 for RVD maneuvers aimed at reaching the docking points P1 and P2. The resulting open-loop state and input trajectories are reported in Figs. 11–13. In both cases, the servicer is successfully steered from a hold point located 200 m behind the target (i.e., EnviSat) to the selected docking point, in a time interval of approximately 12 minutes. Notice from Figs. 11 and 12 that the docking-point trajectory (dashed) of P2 displays a much faster variation compared with that of P1. This is

not surprising, because the point P1 lies along the spin axis (see Fig. 10). Consequently, the motion of P1 (as seen from the RTN frame) is only due to the precession of this axis, which occurs at the orbital rate η [see Eq. (23)]. Conversely, the point P2 is orthogonal to the spin axis. Therefore, its evolution in terms of RTN coordinates stems from both the spin axis precession and the rotation of P2 about the spin axis, the latter of which occurs at a frequency much higher (approximately 30 times) than η. As a result, reaching P2 is far more challenging than reaching P1. In fact, the fuel consumption corresponding to the control input trajectories in Fig. 13 is about three times higher in the P2 case than in the P1 test. In Fig. 13, it can also be seen that the magnitude of the acceleration components stays within the assigned bound a_{max} (reported in Table 3), over the entire maneuvering interval. Moreover, the obtained trajectories satisfy by construction the RVD constraints described in Sec. III. Figure 14 depicts the transition from the rendezvous constraints to the docking constraints, showing that feasibility is retained during this event. Figure 15 illustrates how the docking corridor rotates during the final
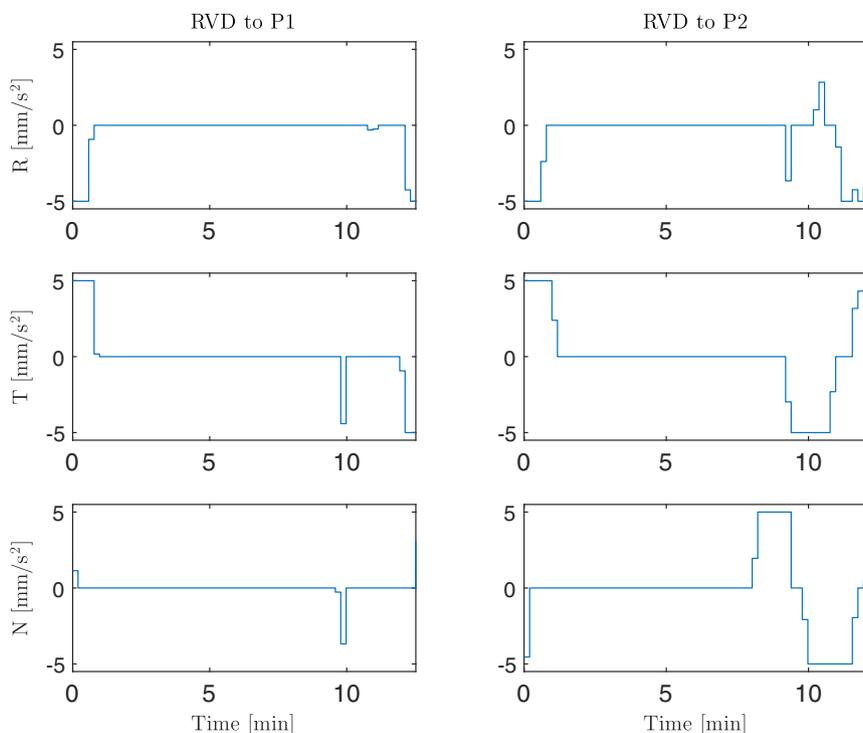
Fig. 13   Radial (R), transverse (T), and normal (N) components of the servicer acceleration vector for RVD to P1 (left) and P2 (right).
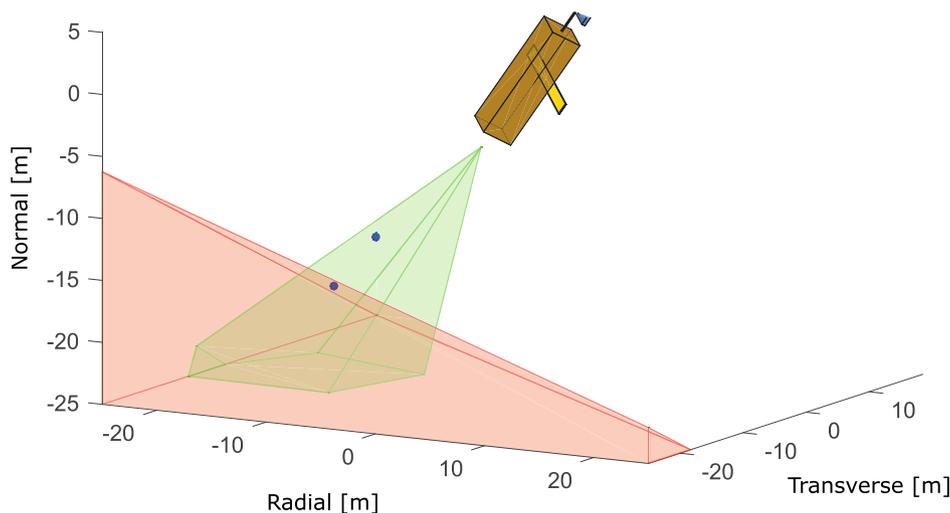
Fig. 14   Illustration of the transition from the safe rendezvous region (red) to the docking corridor (green), for RVD to P2. The servicer center of mass (sCM) is marked by a blue point.
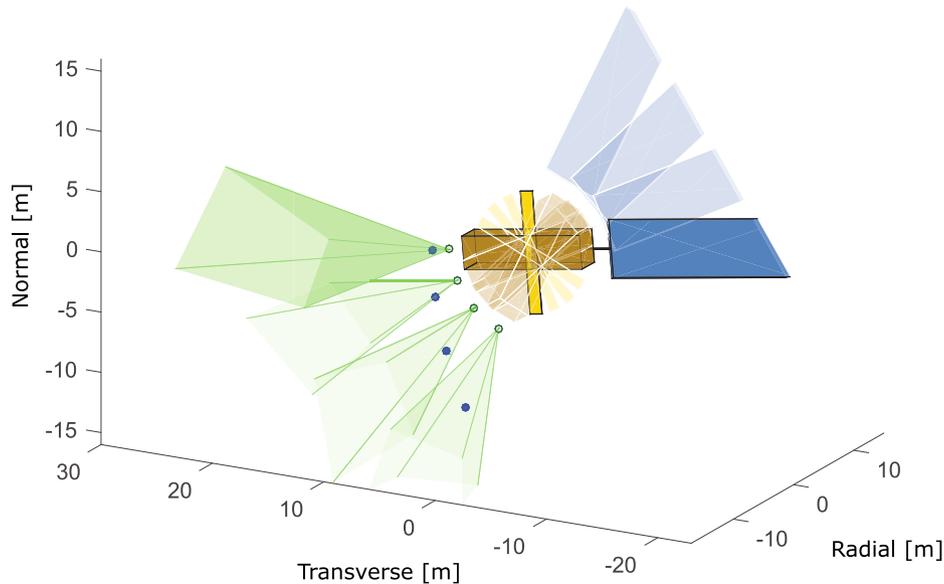
Fig. 15   Illustration of the docking corridor rotation during the final approach to P2 (green point). The servicer center of mass (sCM) is marked by a blue point.

part of the maneuver. It can be seen that the sCM always lies inside the set defined by Eq. (16).

The obtained trajectories have been compared with those resulting from the solution of a continuous-time version of problem (6), in which the final time is free and the state constraints are nonlinear. In particular, the keep-out zone is enforced as in Eq. (9), whereas the docking corridor is modeled as in Eq. (14). The transition between the RVD phases is accounted for by formulating a two-phase optimal control problem, which is solved by using the commercial package GPOPS-II [40]. Similarly to what has been done in Sec. IV, the initial guess for the nonlinear solver is constructed from the unconstrained minimum-energy solution. The results of the comparison are summarized in Table 4. It can be seen that the Algorithm 1 solution is about 40 times faster than that based on GPOPS-II, whereas the maneuver cost is approximately the same for the two methods, for RVD to either P1 or P2. In Fig. 16, the cost incurred by the two solutions is compared with the profile of $J_N^* \tau_s$ for case P2. It can be seen that the nonlinear solution is close to a local optimum of problem (6). This indicates that the constraint approximation scheme described in Sec. III is not overly conservative.

On the whole, the obtained results clearly demonstrate the suitability of the proposed approach for autonomous RVD applications. In particular, in all our tests Algorithm 1 returned a solution in a fraction of a second, whereas the CPU time of the full enumeration, MILP, and GPOPS-II approaches was always greater than 3 s. Considering that the sampling time of the guidance scheme is in the order of 10 s, and that spacecraft onboard computers are far less powerful
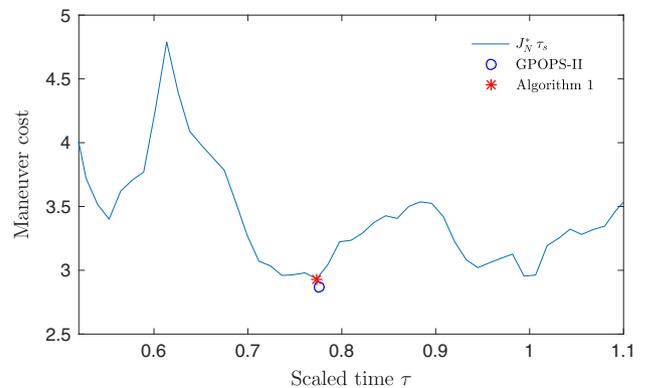
Fig. 16   Profile of $J_N^* \tau_s$, together with the cost incurred by GPOPS-II and Algorithm 1, for RVD to P2.

than the employed hardware, the proposed method appears to be the right candidate for real-time implementation.

VII.   Conclusions

A variable-horizon guidance scheme has been presented for spacecraft RVD applications featuring a tumbling target. As opposed to approaches employing a fixed planning horizon, the proposed formulation provides the ability to identify favorable docking opportunities, which are singled out as local minima of a suitably chosen horizon-dependent cost function. A local optimization strategy has been developed for this new formulation, which is capable of finding high-performance solutions, while incurring a modest computational demand. The method also inherits other advantages of convex formulations, such as guaranteed convergence and ease of implementation. In view of these features, the proposed guidance scheme may be employed either as a standalone module for the autonomous planning (and replanning) of optimized trajectories to be tracked by the control system, or as a baseline for the development of variable-horizon model predictive control strategies, tailored to uncooperative mission scenarios. The method has been demonstrated on a real-world scenario involving docking with the defunct EnviSat spacecraft, and found to provide remarkable improvements in terms of computational efficiency with respect to a nonlinear solver, while incurring only a negligible performance loss. Extensions of this work include

Table 4   GPOPS-II and Algorithm 1 performance for EnviSat RVD

RVD to P1	GPOPS-II	Algorithm 1
CPU time, s	5.767	0.156
Maneuver cost	1.5697	$J_N^* \tau_s = 1.5774$
Normalized time of flight	0.7852	$\hat{N} \tau_s = 0.7977$
Normalized fuel consumption	0.1961	$\|u_N^*\|_1 \tau_s = 0.1949$
RVD to P2	GPOPS-II	Algorithm 1
CPU time, s	9.319	0.205
Maneuver cost	2.8721	$J_N^* \tau_s = 2.9328$
Normalized time of flight	0.7757	$\hat{N} \tau_s = 0.7731$
Normalized fuel consumption	0.5241	$\|u_N^*\|_1 \tau_s = 0.5399$

the incorporation of the guidance scheme into a suitable feedback loop for noise attenuation and disturbance rejection.

Acknowledgment

References

[1] Woffinden, D. C., and Geller, D. K., "Navigating the Road to Autonomous Orbital Rendezvous," *Journal of Spacecraft and Rockets*, Vol. 44, No. 4, 2007, pp. 898–909.
https://doi.org/10.2514/1.30734

[2] Kawano, I., Mokuno, M., Kasai, T., and Suzuki, T., "Result of Autonomous Rendezvous Docking Experiment of Engineering Test Satellite-VII," *Journal of Spacecraft and Rockets*, Vol. 38, No. 1, 2001, pp. 105–111.
https://doi.org/10.2514/2.3661

[3] Rumford, T. E., "Demonstration of Autonomous Rendezvous Technology (DART) Project Summary," *Proceedings of the Society of Photo-Optical Instrumentation Engineers: Space Systems Technology and Operations*, Vol. 5088, International Soc. for Optics and Photonics, Bellingham, WA, 2003, pp. 10–19.
https://doi.org/10.1117/12.498811

[4] Mitchell, I, Gordon, T., Taskov, K., Drews, M., Luckey, D., Osborne, M., Page, L., Norris, H., and Shepperd, S., "GNC Development of the XSS-11 Micro-Satellite for Autonomous Rendezvous and Proximity Operations," *29th AAS Guidance and Control Conference*, AAS Paper 06-014, 2006.

[5] Weismuller, T., and Leinz, M., "GN&C Technology Demonstrated by the Orbital Express Autonomous Rendezvous and Capture Sensor System," *29th AAS Guidance and Control Conference*, AAS Paper 06-016, 2006.

[6] Boyarko, G., Yakimenko, O., and Romano, M., "Optimal Rendezvous Trajectories of a Controlled Spacecraft and a Tumbling Object," *Journal of Guidance, Control, and Dynamics*, Vol. 34, No. 4, 2011, pp. 1239–1252.
https://doi.org/10.2514/1.47645

[7] Ventura, J., Ciarcía, M., Romano, M., and Walter, U., "Fast and Near-Optimal Guidance for Docking to Uncontrolled Spacecraft," *Journal of Guidance, Control, and Dynamics*, Vol. 40, No. 12, 2017, pp. 3138–3154.
https://doi.org/10.2514/1.G001843

[8] Park, H., Zappulla, R., Zagaris, C., Virgili-Llop, J., and Romano, M., "Nonlinear Model Predictive Control for Spacecraft Rendezvous and Docking with a Rotating Target," *27th AAS/AIAA Spaceflight Mechanics Meeting*, Vol. 2, AAS Paper :17-496, 2017.

[9] Lu, P., and Liu, X., "Autonomous Trajectory Planning for Rendezvous and Proximity Operations by Conic Optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 36, No. 2, 2013, pp. 375–389.
https://doi.org/10.2514/1.58436

[10] Liu, X., and Lu, P., "Robust Trajectory Optimization for Highly Constrained Rendezvous and Proximity Operations," *AIAA Guidance, Navigation, and Control (GNC) Conference*, AIAA Paper 2013-4720, 2013.
https://doi.org/10.2514/6.2013-4720

[11] Morgan, D., Chung, S.-J., and Hadaegh, F. Y., "Model Predictive Control of Swarms of Spacecraft Using Sequential Convex Programming," *Journal of Guidance, Control, and Dynamics*, Vol. 37, No. 6, 2014, pp. 1725–1740.
https://doi.org/10.2514/1.G000218

[12] Mao, Y., Szmuk, M., Xu, X., and Acikmese, B., "Successive Convexification: A Superlinearly Convergent Algorithm for Non-Convex Optimal Control Problems," arXiv:1804.06539v2 [math.OC], 2019.

[13] Bonalli, R., Cauligi, A., Bylard, A., and Pavone, M., "GuSTO: Guaranteed Sequential Trajectory Optimization via Sequential Convex Programming," *2019 International Conference on Robotics and Automation (ICRA)*, IEEE Publ., Piscataway, NJ, 2019, pp. 6741–6747.
https://doi.org/10.1109/ICRA.2019.8794205

[14] Liu, X., and Lu, P., "Solving Nonconvex Optimal Control Problems by Convex Optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 37, No. 3, 2014, pp. 750–765.
https://doi.org/10.2514/1.62110

[15] Malyuta, D., Reynolds, T. P., Szmuk, M., Lew, T., Bonalli, R., Pavone, M., and Acikmese, B., "Convex Optimization for Trajectory Generation," arXiv:2106.09125v1 [math.OC], 2021.

[16] Foust, R., Chung, S.-J., and Hadaegh, F. Y., "Optimal Guidance and Control with Nonlinear Dynamics Using Sequential Convex Programming," *Journal of Guidance, Control, and Dynamics*, Vol. 43, No. 4, 2020, pp. 633–644.
https://doi.org/10.2514/1.G004590

[17] Lu, P., "Convex-Concave Decomposition of Nonlinear Equality Constraints in Optimal Control," *Journal of Guidance, Control, and Dynamics*, Vol. 44, No. 1, 2021, pp. 4–14.
https://doi.org/10.2514/1.G005443

[18] Malyuta, D., Reynolds, T., Szmuk, M., Acikmese, B., and Mesbahi, M., "Fast Trajectory Optimization via Successive Convexification for Spacecraft Rendezvous with Integer Constraints," *AIAA Scitech 2020 Forum*, AIAA Paper 2020-0616, 2020.
https://doi.org/10.2514/6.2020-0616

[19] Weiss, A., Baldwin, M., Erwin, R. S., and Kolmanovsky, I., "Model Predictive Control for Spacecraft Rendezvous and Docking: Strategies for Handling Constraints and Case Studies," *IEEE Transactions on Control Systems Technology*, Vol. 23, No. 4, 2015, pp. 1638–1647.
https://doi.org/10.1109/TCST.2014.2379639

[20] Zagaris, C., Park, H., Virgili-Llop, J., Zappulla, R., Romano, M., and Kolmanovsky, I., "Model Predictive Control of Spacecraft Relative Motion with Convexified Keep-Out-Zone Constraints," *Journal of Guidance, Control, and Dynamics*, Vol. 41, No. 9, 2018, pp. 2054–2062.
https://doi.org/10.2514/1.G003549

[21] Hartley, E. N., Gallieri, M., and Maciejowski, J. M., "Terminal Spacecraft Rendezvous and Capture with LASSO Model Predictive Control," *International Journal of Control*, Vol. 86, No. 11, 2013, pp. 2104–2113.
https://doi.org/10.1080/00207179.2013.789608

[22] Leomanni, M., Bianchini, G., Garulli, A., Giannitrapani, A., and Quartullo, R., "Orbit Control Techniques for Space Debris Removal Missions Using Electric Propulsion," *Journal of Guidance, Control, and Dynamics*, Vol. 43, No. 7, 2020, pp. 1259–1268.
https://doi.org/10.2514/1.G004735

[23] Mammarella, M., Lorenzen, M., Capello, E., Park, H., Dabbene, F., Guglieri, G., Romano, M., and Allgöwer, F., "An Offline-Sampling SMPC Framework with Application to Autonomous Space Maneuvers," *IEEE Transactions on Control Systems Technology*, Vol. 28, No. 2, 2020, pp. 388–402.
https://doi.org/10.1109/TCST.2018.2879938

[24] Breger, L., and How, J. P., "Safe Trajectories for Autonomous Rendezvous of Spacecraft," *Journal of Guidance, Control, and Dynamics*, Vol. 31, No. 5, 2008, pp. 1478–1489.
https://doi.org/10.2514/1.29590

[25] Richards, A., Schouwenaars, T., How, J. P., and Feron, E., "Spacecraft Trajectory Planning with Avoidance Constraints Using Mixed-Integer Linear Programming," *Journal of Guidance, Control, and Dynamics*, Vol. 25, No. 4, 2002, pp. 755–764.
https://doi.org/10.2514/2.4943

[26] Di Cairano, S., Park, H., and Kolmanovsky, I., "Model Predictive Control Approach for Guidance of Spacecraft Rendezvous and Proximity Maneuvering," *International Journal of Robust and Nonlinear Control*, Vol. 22, No. 12, 2012, pp. 1398–1427.
https://doi.org/10.1002/rnc.2827

[27] Li, Q., Yuan, J., Zhang, B., and Gao, C., "Model Predictive Control for Autonomous Rendezvous and Docking with a Tumbling Target," *Aerospace Science and Technology*, Vol. 69, Oct. 2017, pp. 700–711.
https://doi.org/10.1016/j.ast.2017.07.022

[28] Dong, K., Luo, J., Dang, Z., and Wei, L., "Tube-Based Robust Output Feedback Model Predictive Control for Autonomous Rendezvous and Docking with a Tumbling Target," *Advances in Space Research*, Vol. 65, No. 4, 2020, pp. 1158–1181.
https://doi.org/10.1016/j.asr.2019.11.014

[29] Richards, A., and How, J. P., "Robust Variable Horizon Model Predictive Control for Vehicle Maneuvering," *International Journal of Robust and Nonlinear Control*, Vol. 16, No. 7, 2006, pp. 333–351.
https://doi.org/10.1002/rnc.1059

[30] Hartley, E. N., Trodden, P. A., Richards, A. G., and Maciejowski, J. M., "Model Predictive Control System Design and Implementation for Spacecraft Rendezvous," *Control Engineering Practice*, Vol. 20, No. 7, 2012, pp. 695–713.
https://doi.org/10.1016/j.conengprac.2012.03.009

[31] Louet, J., and Bruzzi, S., "ENVISAT Mission and System," *IEEE 1999 International Geoscience and Remote Sensing Symposium*, Vol. 3, IEEE Publ., Piscataway, NJ, 1999, pp. 1680–1682.
https://doi.org/10.1109/IGARSS.1999.772059

[32] Fehse, W., *Automated Rendezvous and Docking of Spacecraft*, Cambridge Aerospace Series, Cambridge Univ. Press, Cambridge, England, U.K., 2003, pp. 173–174.
https://doi.org/10.1017/CBO9780511543388

[33] Clohessy, W. H., and Wiltshire, R. S., "Terminal Guidance System for Satellite Rendezvous," *Journal of the Aerospace Sciences*, Vol. 27, No. 9, 1960, pp. 653–658.
https://doi.org/10.2514/8.8704

[34] Leomanni, M., Bianchini, G., Garulli, A., and Quartullo, R., "Sum-of-Norms Periodic Model Predictive Control for Space Rendezvous," *IEEE Transactions on Control Systems Technology*, 2021 (Early Access).
https://doi.org/10.1109/TCST.2021.3095390

[35] *Gurobi Optimizer Reference Manual*, Gurobi Optimization LLC, 2021.

[36] Estable, S., Pruvost, C., Ferreira, E., Telaar, J., Fruhnert, M., Imhof, C., Rybus, T., Peckover, G., Lucas, R., Ahmed, R., Oki, T., Wygachiewicz, M., Kicman, P., Lukasik, A., Santos, N., Milhano, T., Arroz, P., Biesbroek, R., and Wolahan, A., "Capturing and Deorbiting Envisat with an Airbus Spacetug. Results from the ESA e. Deorbit Consolidation Phase Study," *Journal of Space Safety Engineering*, Vol. 7, No. 1, 2020, pp. 52–66.
https://doi.org/10.1016/j.jsse.2020.01.003

[37] Deloo, J., and Mooij, E., "Active Debris Removal: Aspects of Trajectories, Communication and Illumination During Final Approach," *Acta Astronautica*, Vol. 117, Dec. 2015, pp. 277–295.
https://doi.org/10.1016/j.actaastro.2015.08.001

[38] Li, H., Dong, Y., and Li, P., "Real-Time Optimal Approach and Capture of ENVISAT Based on Neural Networks," *International Journal of Aerospace Engineering*, Vol. 2020, Jan. 2020.
https://doi.org/10.1155/2020/8165147

[39] Kucharski, D., Kirchner, G., Koidl, F., Fan, C., Carman, R., Moore, C., Dmytrotsa, A., Ploner, M., Bianco, G., Medvedskij, M., Makeyev, A., Appleby, G., Suzuki, M., Torre, J.-M., Zhongping, Z., Grunwaldt, L., and Feng, Q., "Attitude and Spin Period of Space Debris Envisat Measured by Satellite Laser Ranging," *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 52, No. 12, 2014, pp. 7651–7657.
https://doi.org/10.1109/TGRS.2014.2316138

[40] Patterson, M. A., and Rao, A. V., "GPOPS-II: A MATLAB Software for Solving Multiple-Phase Optimal Control Problems Using Hp-Adaptive Gaussian Quadrature Collocation Methods and Sparse Nonlinear Programming," *ACM Transactions on Mathematical Software (TOMS)*, Vol. 41, No. 1, 2014, pp. 1–37.
https://doi.org/10.1145/2558904