Minimum-Fuel Fixed-Time Impulsive Elliptic Glideslope Guidance Algorithms Using Semidefinite Programming

Yassine Ariba*
LAAS-CNRS, Université de Toulouse; ICAM, 31031 Toulouse, France

Denis Arzelier†, Laura Sofia Urbina-Iglesias‡
LAAS-CNRS, Université de Toulouse, CNRS, 31031 Toulouse, France

A new minimum-fuel glideslope guidance algorithm for approaching autonomously a target evolving on an elliptic orbit is proposed in this paper. Assuming chemical propulsion, the present work aims at enhancing Hablani's seminal method. Although this reference method is efficient and easy to implement, there is no direct control on the fuel consumption and the guidance error. By identifying some relevant degrees of freedom, a new formulation of the glideslope guidance algorithm is proposed to address these issues. For a fixed-time rendezvous and a given number of maneuvers, the fuel-optimal multipulse glideslope is expressed as a semidefinite programming problem. A solution to this optimization problem provides an impulsive control sequence that guarantees a minimal consumption and makes sure the chaser trajectory remains inside a user-defined corridor. Besides, if trajectory constraints are removed, or if a specific direction is considered (V-bar or R-bar), then the formulation is reduced to a simple linear programming problem. Three realistic scenarios of rendezvous are simulated to illustrate the benefit of the proposed methodology.

Nomenclature

a = semi-major axis of the reference orbit

e = eccentricity of the reference orbit

θ = true anomaly of the reference orbit

μ = the standard gravitational parameter

T = orbital period of the reference orbit

$n = \sqrt{\dfrac{\mu}{a^3}} = 2\pi/T$ = the mean motion of the reference orbit

$\Phi(t_{k+1}, t_k) = \phi(t_{k+1})\phi^{-1}(t_k) = \Phi^{[k]}$ = transition matrix of relative motion

N = number of maneuvers

*Associate Professor, LAAS-CNRS, ICAM, yassine.ariba@icam.fr.
†Researcher CNRS, LAAS-CNRS, arzelier@laas.fr.
‡Ph.D. student CNRS, LAAS-CNRS, lsurbina@laas.fr.

$\Delta\vec{V}_i$ = velocity increment vector at t_i

Δv_i = velocity impulse modulus at t_i

$\mathbb{R}[x_1, \cdots, x_n]$ stands for the algebra of polynomials in variables $(x_1, \cdots, x_n)$ with coefficients in $\mathbb{R}$

$\mathbb{O}_{p\times m}$ = the null matrix of dimensions $p \times m$

$\mathcal{I}_m$ = the identity matrix of dimension m

$\mathbb{1}_m$ = the m-vector composed of ones

trace(A) = the trace of the matrix A (sum of its diagonal elements

I. Introduction

Rendezvous (RDV) between two spacecraft (a target and a chaser) has been one of the most salient operational technology since its first manual achievement in the sixties between a Gemini vehicle and an unmanned target vehicle. Recently, an increasing demand is witnessed to perform autonomous rendezvous so as to address various proximity operations between an active spacecraft and a passive one: on-orbit servicing, refueling, repairing or de-orbiting. The references [1], [2], [3] among many others give specific examples of missions involving rendezvous as a key technology.

Reference [4] defines the rendezvous process as a sequence of guided orbital maneuvers to drive the active spacecraft (chaser) into the vicinity of the passive spacecraft (target). This present work focuses on the *close range* rendezvous phase whose objective is to reduce the relative distance to the target before the entry into the final approach corridor. During this phase, relative navigation is used and the general goal of the control system is to ensure a safe approach to the target spacecraft for the next proximity operations. If the chaser is assumed to be actuated by six independent ungimbaled chemical thrusters, then the control system is able to produce an instantaneous incremental change of the spacecraft velocity vector in any direction while the relative position remains constant. This feature has led to several results based on impulsive maneuvers, and various strategies have been proposed to execute the closing flight as a series of hops in a corridor [4] [5], [6], either via the tangential direction, called *V-bar*, or via the radial direction, called *R-bar*. *Hops* refer to the chaser free trajectory between two successive impulsive maneuvers. Basically, each thrust is obtained by computing the velocity vector required to join the next specified location after some known duration of free motion. These calculations are a direct application of the transition matrix associated with the Clohessy-Wiltshire (HCW) equations. Then, the incremental velocity impulse to be applied equals the difference between the computed velocity and the current velocity resulting from the previous hop.

A more general scheme is proposed by the glideslope approach that guides the spacecraft toward any direction [7], [8]. The glideslope approach is a straight line approach, defined in the Local Vertical Local Horizontal (LVLH) frame, from the initial position of the chaser to a final position close to the target spacecraft. The suitable velocity increments are computed according to range and range rate profiles defined by the user with almost the same computational process as in the classical V-bar or R-bar approaches. Note that this generalization of the previous approaches is particularly

interesting in the case where the target has to keep a torque equilibrium attitude [9]. The off-axis glideslope approach has been first defined in the past for rendezvous and proximity operations involving the Space Shuttle [7]. This preliminary study, restricted to an in-plane motion with canted thrusters, has been extended for any direction in space in [8] (defining the so-called classical circular glideslope algorithm), reviewed in [10], and finally extended to elliptical orbits in [11] using the closed-form solution of the Tschauner-Hempel (TH) equations. However, a direct control on the fuel consumption imposed by a restricted propellant budget and on the guidance error inherent to hopping trajectories has not been explicitly considered in these references.

In this paper, the classical circular glideslope algorithm is revisited, and a new approach is proposed to include optimization features in the design. Indeed, the usual methodology generates an impulsive maneuver sequence according to a predefined velocity profile that sets all positions where impulsive maneuvers will occur. As a result, the trajectory is entirely determined, and the consumption and the excursion are by-products of the whole procedure. For that reason, we propose a new formulation of the glideslope guidance which aims to identify relevant degrees of freedom. Getting some degrees of freedom allows to derive an optimization algorithm to minimize the fuel consumption. Besides, in all impulsive approaches mentioned above, the chaser trajectory exhibits hops between maneuvers that are inherent to the natural relative motion of the spacecraft. Inspired from the result of [12], converting a constraint on the trajectory to a linear matrix inequality (using a suitable parametrization of the Yamanaka-Ankersen equations [13], a change of variable and a mathematical result on non-negative polynomials [14]), a new methodology is proposed to control the amplitude of hops. This feature is a desirable asset since it enforces the chaser to remain close enough to the commanded path. It becomes even essential if a *line-of-sight* (LoS) constraint is required to keep the spacecraft within the visibility cone of the target sensors. In addition, the optimization problem formulation also enables us to readily take the maximum thrust of the chaser actuators into account. It is worth noting that while the classical circular glideslope guidance of [8] is limited to circular orbits, the proposed version extends naturally to elliptical orbits. Moreover, the transfer time in the former approach is indirectly derived from the choice of the velocity profile, whereas it is a user-defined input in the present one and leads to a genuine *fixed-time* rendezvous approach. The contribution of this work is summarized as follows.

1) Contrary to the classical glideslope approach, the proposed guidance algorithm is designed with the TH equations, and thus is also workable for eccentric orbits.
2) The new glidelope algorithm results in a semidefinite optimization problem that minimizes the fuel consumption.
3) The trajectory hops of the chaser are enclosed into a user-defined corridor.

The resulting algorithm for off-axis glideslope is expressed as a SemiDefinite Programming (SDP) problem, which can be solved efficiently. This formulation can be further reduced into a Linear Programming (LP) problem when no constraints on the trajectory are considered or when the guidance is limited to a V-bar or R-bar approach. Three numerical examples illustrate the benefit of the proposed methodology and compare it with other approaches from

the literature. Two criteria are considered to assess the performance of algorithms: propellant consumption and the guidance error with respect to the glideslope reference.

II. The Classical Glideslope Approach Algorithm

A. Relative motion dynamics

The close range phase of the spacecraft rendezvous mission is characterized by the use of relative navigation since the separation between spacecraft is sufficiently small [4]. In that case, the relative motion of the chaser is expressed in the Local-Vertical-Local-Horizontal (LVLH) frame. The origin of the coordinate frame is located at the center of mass of the target and the space is spanned by $(\vec{X}_{ot},\ \vec{Y}_{ot},\ \vec{Z}_{ot})$ where the $\vec{Z}_{ot}$ axis is in the radial direction (R-bar) oriented towards the center of the Earth, the $\vec{Y}_{ot}$ axis is perpendicular to the target orbital plane and pointing in the opposite direction of the angular momentum (H-bar) while the $\vec{X}_{ot}$ axis completes the right-hand triad $\vec{X}_{ot} = \vec{Y}_{ot} \times \vec{Z}_{ot}$ (V-bar, see Figure 1). In Figure 1, the vector $\vec{\rho}(t)$ defines the relative position of the chaser with respect to the target at time t. Under Keplerian assumptions (no orbital perturbations are considered) and an elliptic reference orbit, the equations

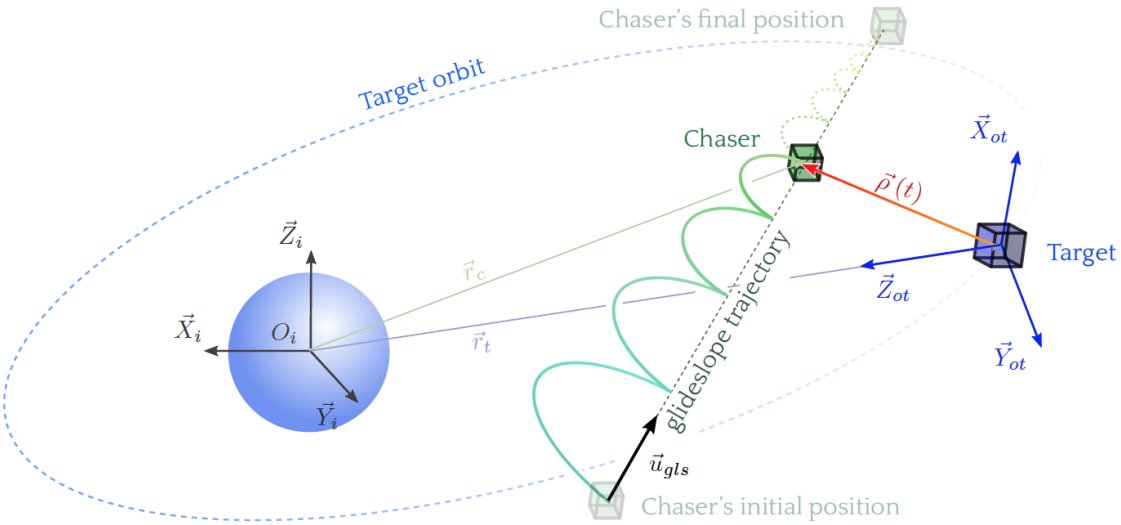


Figure 1 Glideslope approach (not to scale).

of motion for the relative motion in the LVLH frame may be linearized for close separation between the target and the chaser [15, Chapter 5, Section 5.6.1], leading to the state-space representation:

$$\dot{\vec{X}}(t) = A_0(t)\vec{X}(t) + \begin{bmatrix} 0_{3\times3} \\ I_3, \end{bmatrix} \frac{\vec{f}(t)}{m_F} \tag{1}$$

where state $\vec{X}(t) = \left[\vec{\rho}^T(t)\ \vec{v}^T(t)\right]^T$ represents the relative position and velocity of the chaser in the LVLH frame, $f(t) = \left[f_x(t)\ f_y(t)\ f_z(t)\right]^T$ is the thrust vector, m_F is the mass of the chaser and the dynamic matrix $A_0(t)$ is a periodic

matrix of time t given by :

$$A_0(t) = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ a_1(t) & 0 & \ddot{\theta} & 0 & 0 & 2\dot{\theta} \\ 0 & a_2(t) & 0 & 0 & 0 & 0 \\ -\ddot{\theta} & 0 & a_3(t) & -2\dot{\theta} & 0 & 0 \end{bmatrix} \tag{2}$$

Here,

$$a_1(t) = \dot{\theta}^2 - k^4\kappa^3(\theta) \quad a_2(t) = -k^4\kappa^3(\theta) \quad a_3(t) = \dot{\theta}^2 + 2k^4\kappa^3(\theta)$$

where θ is the true anomaly of the reference orbit and by noting that:

$$\frac{d\theta}{dt} = \frac{n}{(1-e^2)^{3/2}} \underbrace{(1 + e\cos\theta)^2}_{\kappa(\theta)} =: k^2\kappa^2(\theta) \tag{3}$$

where $n = \sqrt{\dfrac{\mu}{a^3}} = 2\pi/T$, $k^2 = \dfrac{n}{(1-e^2)^{3/2}}$ is the mean motion of the target orbit, satisfying for any fixed θ_0, t_0:

$$\text{If } \theta - \theta_0 = 2\pi \quad \text{then} \quad n(t - t_0) = 2\pi \tag{4}$$

It is assumed that only the chaser is active and actuated using 6 ungimbaled identical chemical thrusters. The use of chemical propulsion is idealized as impulsive maneuvers providing instantaneous velocity jumps in the three axes while the relative position remains relatively unchanged. The impulsive control input is thus defined as:

$$\Delta\vec{V}(t_k) := \Delta\vec{V}_k := \int_{t_k^-}^{t_k^+} \frac{1}{m_F} \begin{bmatrix} f_x(t) \\ f_y(t) \\ f_z(t) \end{bmatrix} \mathrm{d}t \tag{5}$$

where t_k is a generic firing time and $\Delta\vec{V}_k$ represents the applied impulsive thrust. In order to compute the transition matrix $\Phi(t, t_0)$ for the linearized Equations (1) and (2), classical derivations dating back to the seminal publications of Lawden [16, Chapter 5] and Tschauner-Hempel [17] apply a change of independent variable from time, t, to true

anomaly, θ, and a simplifying coordinate change leading to $\vec{X}_\theta(\theta) = P(\theta)\vec{X}(t)$ with:

$$P(\theta) := \begin{bmatrix} \kappa(\theta)\mathcal{I}_3 & \mathbb{O}_{3\times3} \\ \kappa(\theta)'\mathcal{I}_3 & \frac{1}{k^2\kappa(\theta)}\mathcal{I}_3 \end{bmatrix} \tag{6}$$

A simplified autonomous state space representation $\vec{X}_\theta{}'(\theta) = \tilde{A}(\theta)\vec{X}_\theta(\theta)$ is obtained with:

$$\tilde{A}(\theta) = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 2 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{3}{\kappa(\theta)} & -2 & 0 & 0 \end{bmatrix} \tag{7}$$

Based on a particular fundamental solution $\tilde{\varphi}_{\theta_0}(\theta)$, the so-called Yamanaka-Ankersen form of the transition matrix $\tilde{\Phi}(\theta, \theta_0)$ has been proposed in the reference [13]. This form is particularly interesting regarding the computing efficiency, and the transition matrix $\Phi(t, t_0)$ may be easily computed by:

$$\Phi(t, t_0) = T(\theta)^{-1}\tilde{\varphi}_{\theta_0}(\theta)\tilde{\varphi}_{\theta_0}(\theta_0)T(\theta_0) \tag{8}$$

Thus, a controlled trajectory composed of $N + 1$ impulses is described by the following equation:

$$\vec{X}(t) = \Phi(t, t_0)\vec{X}(t_0) + \sum_{k=0}^{N} \Phi(t, t_k)B\Delta\vec{V}_k \tag{9}$$

where $t_1 < t_2 < \cdots < t_N \leq t$ and $\Delta\vec{V}_k$ is the impulsive control applied at t_k. $B = [\mathbb{O}_{3\times3}\ \mathcal{I}_3]^T$ is the input matrix. Hereafter, the following notation describing the free motion with a block partitioned transition matrix is adopted:

$$\vec{X}(t) = \left[\begin{array}{c} \vec{\rho}(t) \\ \hline \vec{v}(t) \end{array}\right] = \left[\begin{array}{c|c} \Phi_{\rho\rho}(t, t_0) & \Phi_{\rho v}(t, t_0) \\ \hline \Phi_{v\rho}(t, t_0) & \Phi_{vv}(t, t_0) \end{array}\right] \left[\begin{array}{c} \vec{\rho}_0 \\ \hline \vec{v}_0 \end{array}\right] \tag{10}$$

The analytical expressions of each partition of the state transition matrix $\Phi(t, t_0)$ in the context of an elliptical orbit are not given here, since they have a convoluted analytical form and all we need is to know how to compute them efficiently.

However, in the case of a circular orbit, the relative dynamics are governed by the *Hill-Clohessy-Wiltshire* equations

[18], leading to a simpler expression of the four partitions of $\Phi(t, t_0) \in \mathbb{R}^{6\times6}$.

B. Hablani's classical circular glideslope approach for rendezvous

When considering design of impulsive maneuvers for a glideslope rendezvous, the most cited reference is the paper [8], in which the so-called classical inbound and outbound glideslope approaches for a circular reference orbit are presented in a general setup. When the chaser must approach the target satellite, an inbound glideslope guidance algorithm is used. On the other hand, an outbound algorithm is used for receding away from the target. In both cases, the chaser relative velocity must be low when it is close to the target, in order to comply with safety constraints. The real trajectory of the chaser will not actually be a straight line, but will be composed of a sequence of *humps* between the different thrust locations on the glideslope line, as shown in Figure 1. These humps appear because of the effect of the gravitational forces acting over the chaser and of the applied impulses, and the humps will vary according to the type of orbit. When performing a glideslope approach, the chaser will depart from an initial position $\vec{\rho}(t_0) = \vec{\rho}_0 = [x_0 \;\; y_0 \;\; z_0]^T$ and will arrive at a final position $\vec{\rho}(t_f) = \vec{\rho}_f = [x_f \;\; y_f \;\; z_f]^T$ at a fixed final time t_f. This final position can be a station keeping position, a point that belongs to a given periodic reference trajectory or a docking point. In order to parameterize the rectilinear trajectory to be followed by the chaser — also known as commanded path [8] — from $\vec{\rho}_0$ to $\vec{\rho}_f$, the vector $\vec{\lambda}(t)$ is introduced, whose origin is located at the chaser initial position and pointing to the chaser current position, as shown in Figure 2. It thus verifies:

$$\vec{\lambda}(t) = \vec{\rho}_{gls}(t) - \vec{\rho}_0 \tag{11}$$

where $\vec{\rho}_{gls}(t)$ is the relative position vector of the orthogonal projection of the chaser on the glideslope line with respect to the target. The unit vector of the rectilinear trajectory direction is expressed as:

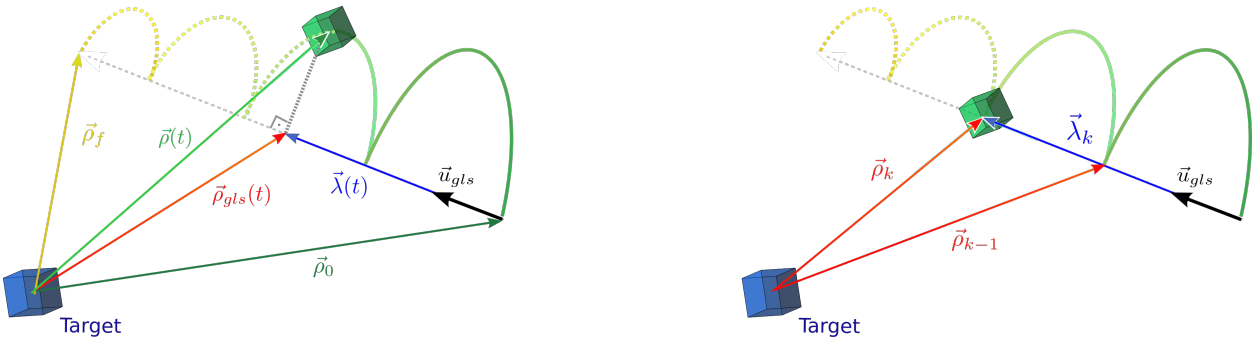


Figure 2 General position.

$$\vec{u}_{gls} = \left[\frac{x_f - x_0}{||\vec{\rho}_f - \vec{\rho}_0||} \;\; \frac{y_f - y_0}{||\vec{\rho}_f - \vec{\rho}_0||} \;\; \frac{z_f - z_0}{||\vec{\rho}_f - \vec{\rho}_0||} \right] \tag{12}$$

Note that the commanded path corresponds to the vector $\vec{\lambda}_f = \vec{\rho}_f - \vec{\rho}_0$. At every instant t, the *travelled distance* is a function of time, and is given by $\lambda(t) = ||\vec{\lambda}(t)||$ in the direction of the unit vector $\vec{u}_{gls}$:

$$\vec{\lambda}(t) = \lambda(t)\, \vec{u}_{gls} \tag{13}$$

The Authors of [8] define *the distance to go* as $\sigma(t) = \|\vec{\lambda}_f\| - \lambda(t)$. The main features of the classical circular glideslope algorithm are now recalled. The chaser is commanded to reach $\vec{\rho}_f$ from $\vec{\rho}_0$ following a specific linear commanded velocity profile:

$$\dot{\sigma} = \alpha\, \sigma - \dot{\lambda}_f \tag{14}$$

where α is the slope of $\dot{\sigma}$ vs σ. The initial distance to go $\sigma_0 = \|\vec{\lambda}_f\|$, the initial commanded velocity $\dot{\sigma}_0 = -\dot{\lambda}_0 < 0$, and the final arrival velocity $\dot{\sigma}_f = -\dot{\lambda}_f < 0$ are quantities specified by the designer, required to define the slope α:

$$\alpha = \frac{\dot{\lambda}_f - \dot{\lambda}_0}{\|\vec{\lambda}_f\|} < 0 \tag{15}$$

The profile for the distance to go $\sigma(t)$ is therefore readily deduced as:

$$\sigma(t) = \sigma_0\, e^{\alpha t} - \frac{\dot{\lambda}_f}{\alpha}(e^{\alpha t} - 1) \tag{16}$$

Note that $\dot{\lambda}_0 > \dot{\lambda}_f$ since a decelerating glideslope trajectory is sought. For a given set of these parameters, the basic principle of the classical classical algorithm is then to analytically compute a fixed number of impulses equally spaced in time over the transfer duration t_f, leading to a constant interval between two consecutive thrusts:

$$\Delta t = \frac{t_f}{N} \tag{17}$$

The thrusters will be then fired at times:

$$t_k = k\, \Delta t\,, \quad k = 0, 1, \cdots, N-1 \tag{18}$$

Each incremental velocity control at $\vec{\rho}_k$ is computed as $\Delta\vec{V}_k = \vec{v}_{k_+} - \vec{v}_{k_-}$ where $\vec{v}_{k_+}$ is the departure velocity needed to go from $\vec{\rho}_k$ to $\vec{\rho}_{k+1}$ and $\vec{v}_{k_-}$ is the arrival velocity at $\vec{\rho}_k$. In order to obtain the maneuver plan composed by the N impulses $\left\{\Delta\vec{V}_0, \Delta\vec{V}_1, \cdots, \Delta\vec{V}_k, \cdots, \Delta\vec{V}_{N-1}\right\}$, we need the relative velocities right before and just after the thrust. Both quantities are simply obtained by solving the autonomous equations (10) at each step k, since:

$$\vec{\rho}_{k+1} = \Phi_{\rho\rho}(t_{k+1}, t_k)\, \vec{\rho}_k + \Phi_{\rho v}(t_{k+1}, t_k)\, \vec{v}_{k_+} \tag{19}$$

$$\vec{v}_{k+1_-} = \Phi_{v\rho}(t_{k+1}, t_k)\,\vec{\rho}_k + \Phi_{vv}(t_{k+1}, t_k)\,\vec{v}_{k+} \tag{20}$$

and:

$$\vec{v}_{k+} = \Phi_{\rho v}(t_{k+1}, t_k)^{-1}\left[\vec{\rho}_{k+1} - \Phi_{\rho\rho}(t_{k+1}, t_k)\,\vec{\rho}_k\right] \tag{21}$$

which allows the chaser to arrive at the next position $\vec{\rho}_{k+1}$ with relative velocity $\vec{v}_{k+1_-}$ at time t_{k+1}. Recall that in the classical circular approach, transition matrices are actually based on the HCW equations, restricted to circular orbits, and depend only on the difference $t_{k+1} - t_k$.

The classical circular glideslope algorithm is straightforward and easy to implement but suffers from key shortcomings. First, it is limited to circular reference orbits. Second, it is important to note that the actual trajectory of the chaser will not be strictly along the commanded straight line path but will exhibit humps between the N points where an impulsive maneuver is performed and located on the commanded path (Figure 2). In addition, if the initial and final commanded velocities of the glideslope profile are *a priori* given, there is no degree of freedom left to control the transfer time and the consumption. Indeed, the transfer time t_f is not fixed *a priori* but deduced from the initial and final commanded velocities $(\dot{\lambda}_0, \dot{\lambda}_f)$ and from the initial distance to go σ_0:

$$t_f = \frac{1}{\alpha}\ln\left[\frac{\dot{\lambda}_f}{\dot{\lambda}_0}\right] = \frac{\sigma_0}{\dot{\lambda}_f - \dot{\lambda}_0}\ln\left[\frac{\dot{\lambda}_f}{\dot{\lambda}_0}\right] \tag{22}$$

The consumption itself is computed *a posteriori* from the velocity increments without any possibility to optimize it for given side conditions of the rendezvous. The objective of the present paper is therefore to propose a new optimization algorithm for the general glideslope framework and extend the results to elliptic reference orbits, taking these two important features into account:

- Minimize the fuel-consumption for a given set of initial and final rendezvous conditions and an *a priori* fixed time of transfer;
- Control the maximum guidance error by defining constraints on the hump profile.

III. Minimum-Fuel Glideslope Approach with Controlled Humps

The main result of the paper is now presented. It mainly consists in identifying some degrees of freedom, then deriving a numerically tractable expression for the different constraints on the chaser trajectory and finally to minimize the fuel consumption during the glideslope transfer. First, the glideslope line tracking constraints are defined according to the transition matrix of the relative linearized Keplerian elliptic dynamics. Contrasting with [8] and [11], intermediate positions are free variables and a constraint is added to control the final relative velocity of the chaser. Next, the constraints on the hump profile are dealt with using a parametrization of the relative trajectory defined in [12] and results from polynomial optimization from the reference [14]. Finally, a general minimum-fuel multipulse

glideslope guidance algorithm relying on the solution of a semidefinite programming problem is proposed.

A. Glideslope line tracking

In order to perform the transfer from $\vec{\rho}_0$ to $\vec{\rho}_f$ in a given duration t_f, the number of thruster firings is fixed and equal to $N+1$. Our approach does not actually require maneuvers to be equally spaced in time since only the time grid, t_k for $k = 0, 1, \cdots, N$, needs to be specified to evaluate the transition matrices as will be seen in the sequel. However, for the sake of simplicity of exposition, it will be assumed throughout the paper an equally spaced time grid and that two successive impulsive maneuvers will be separated by $\Delta t = t_f/N$, i.e. impulsive thrusts are applied at times $t_k = k\Delta t$, $k = 0, 1, \cdots, N$. Throughout the transfer, the spacecraft must follow the commanded path. After each maneuver, the chaser must be back on the glideslope line. The initial $\vec{\rho}_0$ and final $\vec{\rho}_N = \vec{\rho}_f$ positions are fixed by specifications. Intermediate positions are set free and are parameterized as:

$$\vec{\rho}_k = \vec{\rho}_0 + \lambda_k \vec{u}_{gls},\ k = 0, \ldots, N \tag{23}$$

The scalars λ_k are free and denote here the traveled distance from $\vec{\rho}_0$ to $\vec{\rho}_k$. Note that $\lambda_0 = 0$ and $\lambda_N = \lambda_f = \|\vec{\rho}_f - \vec{\rho}_0\|$. A set of N equations of the form:

$$\vec{\rho}_{k+1} = \Phi_{\rho\rho}^{[k]}\vec{\rho}_k + \Phi_{\rho v}^{[k]}\vec{v}_{k_+},\ k = 0, \ldots, N-1 \tag{24}$$

where the position vector $\vec{\rho}_{k+1}$ describes the transfer of the chaser after the impulsive actuation at t_k. $\vec{v}_{k_+}$ is the velocity vector right after the impulse is applied. Combining Equations (23) and (24) enforces the requirement for the chaser to come back to the path after each maneuver period, leading to the set of equations:

$$\begin{aligned} \lambda_{k+1}\vec{u}_{gls} - \lambda_k \Phi_{\rho\rho}^{[k]}\vec{u}_{gls} - \Phi_{\rho v}^{[k]}\vec{v}_{k_+} &= (\Phi_{\rho\rho}^{[k]} - \mathcal{I}_3)\vec{\rho}_0 \\ \lambda_1 \vec{u}_{gls} - \Phi_{\rho v}^{[0]}\vec{v}_{0_+} &= (\Phi_{\rho\rho}^{[0]} - \mathcal{I}_3)\vec{\rho}_0 \end{aligned} \tag{25}$$

for $k = 1, \cdots, N-1$. Because the reference orbit is considered to be elliptic, the transition matrix is not constant all over the orbit and needs to be updated for each maneuver. All transition matrices $\Phi^{[k]}$ can be computed *a priori* from the input data N and t_f with $t_k = kt_f/N$, $k = 0, 1, \cdots, N$. The decision variables in (25) are composed by the sequence of scalar variables λ_k for $k = 1, \cdots, N-1$ and by the sequence of vectors $\vec{v}_{k_+}$ for $k = 0, 1, \cdots, N-1$. The sequence of impulses is deduced afterwards, computing the difference between the design variable $\vec{v}_{k_+}$ and the velocity vector $\vec{v}_{k_-}$ resulting from the previous maneuver and from the relative dynamics of the chaser:

$$\vec{v}_{k+1_-} = \Phi_{v\rho}^{[k]}\vec{\rho}_k + \Phi_{vv}^{[k]}\vec{v}_{k_+},\ k = 0, 1, \cdots, N-1 \tag{26}$$

Therefore, we have

$$\Delta\vec{V}_k = \vec{v}_{k_+} - \vec{v}_{k_-} \tag{27}$$

B. Final velocity constraint

Since a last impulse is needed to control the final velocity of the spacecraft, an additional equality constraint is defined. This $(N+1)^{th}$ impulse maneuver is given by:

$$\begin{aligned} \Delta\vec{V}_N &= \vec{v}_{N_+} - \vec{v}_{N_-} \\ \Delta\vec{V}_N &= \vec{v}_{N_+} - \Phi_{v\rho}^{[N-1]}\vec{\rho}_{N-1} - \Phi_{vv}^{[N-1]}\vec{v}_{N-1_+} \end{aligned} \tag{28}$$

Setting the vector $\vec{v}_{N_+} = \vec{v}_f$ as the desired final velocity and $\Delta\vec{V}_N$ being a free variable, an extra equality constraint is appended:

$$\vec{v}_f - \Phi_{v\rho}^{[N-1]}\vec{\rho}_0 = \Phi_{v\rho}^{[N-1]}\vec{u}_{gls}\lambda_{N-1} + \Phi_{vv}^{[N-1]}\vec{v}_{N-1_+} + \Delta\vec{V}_N \tag{29}$$

As mentioned earlier, $\Delta\vec{V}_N$, λ_{N-1} and $\vec{v}_{N-1_+}$ are the only free variables of (29).

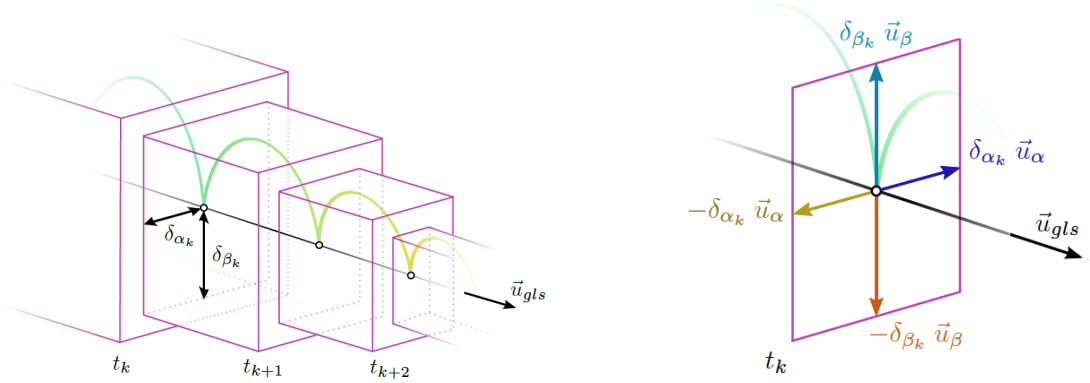
C. Constraints on guidance error

A set of constraints on the spacecraft relative trajectory is now specified in order to bound the guidance error inherent to the impulsive glideslope approach. These constraints, formulated as inequalities on the trajectory $\vec{\rho}$, must be enforced all along the mission, for each hump, $\forall t \in [t_k\ ,\ t_{k+1}]$, for $k = 0, 1, \ldots, N-1$. It thus results in continuous constraints, i.e. infinitely many inequalities to be satisfied and for which a numerically tractable formulation is given in this subsection. In the spirit of the method developed in [12], the idea is to look for an equivalent finite description of the admissible relative trajectories using various tools from algebraic geometry and in particular, properties of non negative polynomials. The main steps of the method are: 1) define a piecewise linear envelope enclosing the admissible trajectory; 2) use a rational parametrization of the trajectory between each pulse to transform the previous continuous linear constraints into polynomial non negativity constraints; 3) apply representation theorems of cones of nonnegative polynomials from [14] to get a final semidefinite formulation of the constraints on guidance error.

1. Admissible trajectories envelope

First, a set of linear constraints on the chaser relative trajectory is defined for each maneuver interval $[t_k\ t_{k+1}]$, defined as the elapsed time between the k^{th} impulse to the instant when the spacecraft is back on the glideslope line.

The input specifications regarding the maximal allowable excursion will define a rectangular corridor with four planes parallel to the glideslope direction $\vec{u}_{gls}$ bounding the trajectory at each step k. We define the parameters δ_{α_k} and δ_{β_k} to specify the distance from the glideslope line to each pair of planes, as shown in Figure 3a.



(a) Rectangular corridor bounding the trajectory.

(b) Orthonormal basis to vector $\vec{u}_{gls}$.

Figure 3 Glideslope corridor constraints.

A set of four linear constraints (upper and lower bounds for both parameters δ_{α_k} and δ_{β_k}) will be therefore necessary to define the maximum allowable excursion at each maneuver, represented by the following inequalities:

$$\begin{aligned} -\vec{u}_\alpha^{\,T}(\vec{\rho}_k - \delta_{\alpha_k}\ \vec{u}_\alpha) \le \vec{u}_\alpha^{\,T}\ \vec{\rho}(t) \le \vec{u}_\alpha^{\,T}(\vec{\rho}_k + \delta_{\alpha_k}\ \vec{u}_\alpha) \\ -\vec{u}_\beta^{\,T}(\vec{\rho}_k - \delta_{\beta_k}\ \vec{u}_\beta) \le \vec{u}_\beta^{\,T}\ \vec{\rho}(t) \le \vec{u}_\beta^{\,T}(\vec{\rho}_k + \delta_{\beta_k}\ \vec{u}_\beta) \end{aligned} \tag{30}$$

where $\vec{u}_\alpha$ and $\vec{u}_\beta$ are unit vectors, chosen as an orthonormal basis for the null space of $\vec{u}_{gls}^{\,T}$, as shown in Figure 3b. The inequalities in Expression (30) can be recast in a matrix form as:

$$\underbrace{\begin{bmatrix} \vec{u}_\alpha^{\,T} \\ -\vec{u}_\alpha^{\,T} \\ \vec{u}_\beta^{\,T} \\ -\vec{u}_\beta^{\,T} \end{bmatrix}}_{A_k} \vec{\rho}(t) \le \underbrace{\begin{bmatrix} \vec{u}_\alpha^{\,T}(\vec{\rho}_k + \delta_{\alpha_k}\ \vec{u}_\alpha) \\ -\vec{u}_\alpha^{\,T}(\vec{\rho}_k - \delta_{\alpha_k}\ \vec{u}_\alpha) \\ \vec{u}_\beta^{\,T}(\vec{\rho}_k + \delta_{\beta_k}\ \vec{u}_\beta) \\ -\vec{u}_\beta^{\,T}(\vec{\rho}_k - \delta_{\beta_k}\ \vec{u}_\beta) \end{bmatrix}}_{\vec{b}_k}, \ \forall t \in [t_k\ t_{k+1}],\ \forall k = 0, 1, \ldots, N-1 \tag{31}$$

where the inequalities are componentwise inequalities between vectors. $A_k \in \mathbb{R}^{n_c \times 3}$ is a constant matrix $\forall k = 0, 1, \ldots, N-1$, since all boxes are oriented along the glideslope line. n_c denotes the number of scalar inequalities, each of which defines a plane bounding the trajectory ($n_c = 4$ in this case) and $\vec{b}_k \in \mathbb{R}^{n_c}$ is a vector that depends on the user-defined parameters and might change according to the different distance specifications $\{\delta_{\alpha_k}, \delta_{\beta_k}\}$ associated to each maneuver k, as shown in Figure 3a. Note also that a specific hops plane rotation around the glideslope line may be specified by choosing appropriate matrices A_k and vectors $\vec{b}_k$.

Applying the change of variable (6), $\vec{\rho}_\theta(\theta) = \kappa(\theta)\vec{\rho}(\theta)$, to the general constraint (31), we get:

$$A_k \vec{\rho}_\theta(\theta) \quad \leq \quad \kappa(\theta)\vec{b}_k, \quad \forall\theta \in [\theta_k \ \theta_{k+1}], \ \forall k = \{0, \ldots, N-1\} \tag{32}$$

Considering that the target spacecraft is evolving on an elliptic Keplerian orbit, the free relative motion is modelled by the Tschauner-Hempel equations which are linear homogeneous differential equations with the chief's true anomaly as the independent variable [17]. Since their first occurrence, numerous analytical solutions based on different sets of fundamental solutions have been proposed. The well-known Yamanaka-Ankersen fundamental solutions are adopted here [13]. Hence, the components of the relative position vector $\vec{\rho}_\theta = [\tilde{x} \ \tilde{y} \ \tilde{z}]^T$ can be obtained as the first three lines of a linear combination of the Yamanaka-Ankersen fundamental solutions:

$$\begin{cases} \tilde{x}(\theta) &= (2 + e\cos\theta)(d_1 \sin\theta - d_2 \cos\theta) + d_3 + 3d_4 J(\theta, \theta_k)(1 + e\cos\theta)^2 \\ \tilde{y}(\theta) &= d_5 \cos\theta + d_6 \sin\theta \\ \tilde{z}(\theta) &= (1 + e\cos\theta)(d_2 \sin\theta + d_1 \cos\theta) - 3\, e\, d_4 J(\theta, \theta_k) \sin\theta(1 + e\cos\theta) + 2d_4 \end{cases} \tag{33}$$

for $\theta \in [\theta_k, \theta_{k+1}]$, where the vector of parameters $\vec{D}$ is defined by (35) and depends linearly on the initial state [12]. The integral term $J(\theta, \theta_k)$ is given by:

$$J(\theta, \theta_k) = \int_{\theta_k}^{\theta} \frac{1}{\kappa(u)^2} du = \frac{n}{(1-e^2)^{3/2}}(t - t_k) \tag{34}$$

$$\underbrace{\begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ d_4 \\ d_5 \\ d_6 \end{bmatrix}}_{\vec{D}} = \underbrace{\left[\begin{array}{ccc|ccc} 0 & 0 & \frac{3(e+\cos\theta_k)}{e^2-1} & -\frac{2\cos\theta_k + e\cos^2\theta_k + e}{e^2-1} & 0 & \frac{\sin\theta_k(1+e\cos\theta_k)}{e^2-1} \\ 0 & 0 & \frac{3\sin\theta_k(1+e\cos\theta_k+e^2)}{(e^2-1)(1+e\cos\theta_k)} & -\frac{\sin\theta_k(2+e\cos\theta_k)}{e^2-1} & 0 & -\frac{\cos\theta_k + e\cos^2\theta_k - 2e}{e^2-1} \\ 1 & 0 & -\frac{3e\sin\theta_k(2+e\cos\theta_k)}{(e^2-1)(1+e\cos\theta_k)} & \frac{e\sin\theta_k(2+e\cos\theta_k)}{e^2-1} & 0 & \frac{e^2\cos^2\theta_k + e\cos\theta_k - 2}{e^2-1} \\ 0 & 0 & -\frac{3e\cos\theta_k + e^2 + 2}{e^2-1} & \frac{(1+e\cos\theta_k)^2}{e^2-1} & 0 & -\frac{e\sin\theta_k(1+e\cos\theta_k)}{e^2-1} \\ 0 & \cos\theta_k & 0 & 0 & -\sin\theta_k & 0 \\ 0 & \sin\theta_k & 0 & 0 & \cos\theta_k & 0 \end{array}\right]}_{C(\theta_k)} \underbrace{\begin{bmatrix} \vec{\rho}_{\theta k} \\ \vec{v}_{\theta k} \end{bmatrix}}_{\vec{X}_\theta(\theta_k)} \tag{35}$$

2. Rational parametrization of the trajectory constraints via polynomial non-negativity constraints

Expression (33) shows several trigonometric terms. The following change of variable is used in order to transform the trigonometric functions into rational functions:

$$\tan\frac{\theta}{2} = w,\ \cos\theta = \frac{1-w^2}{1+w^2},\ \sin\theta = \frac{2w}{1+w^2} \tag{36}$$

The propagation of the spacecraft relative motion can then be expressed as a function of w:

$$\begin{cases} \tilde{x}(w) &= \dfrac{1}{(1+w^2)^2}\left[P_x(w) + 3d_4 P_{Jx}(w)J(w)\right] \\ \tilde{y}(w) &= \dfrac{1}{(1+w^2)}P_y(w) \\ \tilde{z}(w) &= \dfrac{1}{(1+w^2)^2}\left[P_z(w) + 2d_4 P_{Jz}(w)J(w)\right] \end{cases} \tag{37}$$

for $w \in [w_k, w_{k+1}]$. Only the term with $J(w)$ is non-rational and requires to be dealt with. The closed form of $J(w)$ is given by:

$$J(w) = \left[\frac{2ew}{(e^2-1)(e+1+(1-e)w^2)} - \frac{2\arctan h\left(\dfrac{\sqrt{e-1}}{\sqrt{e+1}}w\right)}{\sqrt{(e^2-1)^3}}\right]_{w_k}^{w} \tag{38}$$

All P_* functions are polynomials:

$$\begin{aligned} P_{Jx}(w) &= \Big((1+e)+(1-e)w^2\Big)^2 \\ P_{Jz}(w) &= -3e\Big((1+e)w+(1-e)w^3\Big) \\ P_x(w) &= \sum_{i=0}^{4} p_{xi}w^i,\ P_y(w) = \sum_{i=0}^{2} p_{yi}w^i,\ P_z(w) = \sum_{i=0}^{4} p_{zi}w^i \end{aligned} \tag{39}$$

where coefficients of P_x, P_y and P_z depend linearly on the vector $\vec{D}$, and therefore depend linearly on the state $\vec{X}_\theta(\theta_k)$

at the firing location θ_k.

$$\underbrace{\begin{bmatrix} p_{x0} \\ p_{x1} \\ p_{x2} \\ p_{x3} \\ p_{x4} \end{bmatrix}}_{p_x} = \underbrace{\begin{bmatrix} 0 & -2-e & 1 & 0 & 0 & 0 \\ 4+2e & 0 & 0 & 0 & 0 & 0 \\ 0 & 2e & 2 & 0 & 0 & 0 \\ 4-2e & 0 & 0 & 0 & 0 & 0 \\ 0 & 2-e & 1 & 0 & 0 & 0 \end{bmatrix}}_{C_x} \vec{D} \tag{40}$$

$$\underbrace{\begin{bmatrix} p_{y0} \\ p_{y1} \\ p_{y2} \end{bmatrix}}_{p_y} = \underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 & -1 & 0 \end{bmatrix}}_{C_y} \vec{D} \tag{41}$$

$$\underbrace{\begin{bmatrix} p_{z0} \\ p_{z1} \\ p_{z2} \\ p_{z3} \\ p_{z4} \end{bmatrix}}_{p_z} = \underbrace{\begin{bmatrix} e+1 & 0 & 0 & 2 & 0 & 0 \\ 0 & 2e+2 & 0 & 0 & 0 & 0 \\ -2e & 0 & 0 & 4 & 0 & 0 \\ 0 & 2-2e & 0 & 0 & 0 & 0 \\ e-1 & 0 & 0 & 2 & 0 & 0 \end{bmatrix}}_{C_z} \vec{D} \tag{42}$$

Before defining trajectory constraints, we first need to deal with the integral term $J(w)$ in the spacecraft relative motion equation (37). In order to have a rational expression for the motion, a polynomial approximation is derived to bound J over $w \in [w_k, w_{k+1}]$:

$$J(w) = \Theta_r(w) + \varepsilon(w) \Rightarrow \underbrace{\Theta_r(w) - \bar{\varepsilon}}_{\Theta_l(w)} \leq J(w) \leq \underbrace{\Theta_r(w) + \bar{\varepsilon}}_{\Theta_u(w)} \tag{43}$$

where $\Theta_r(w)$ is a polynomial of degree r derived with the least-squares method. Basically, the larger the polynomial order r is, the smaller the approximation error will be. The scalar $\bar{\varepsilon}$ must be chosen large enough such that the interval $\Theta_r(w) \pm \bar{\varepsilon}$ covers the approximation, that is, $|\varepsilon(w)| \leq \bar{\varepsilon}$, $\forall w \in [w_k, w_{k+1}]$. On the other hand, a larger $\bar{\varepsilon}$ will lead to a more conservative inequality (43) and to a higher computational complexity. In the numerical examples of Section V, a second order polynomial ($r = 2$) is considered and $\bar{\varepsilon}$ is set to 10^{-4}.

The linear constraints (32) are transformed by the change of variables (36) into:

$$A_k \vec{\rho}_\theta(w) \quad \leq \quad \left(\frac{1+e+(1-e)w^2}{1+w^2}\right)\vec{b}_k, \ \forall w \in [w_k \ w_{k+1}], \ \forall k = \{0, \ldots, N-1\} \tag{44}$$

with $w_k = \tan(\theta_k/2)$. The expanded form of the above equation is written as:

$$\begin{bmatrix} A_{k_{1,1}} & A_{k_{1,2}} & A_{k_{1,3}} \\ A_{k_{2,1}} & A_{k_{2,2}} & A_{k_{2,3}} \\ A_{k_{3,1}} & A_{k_{3,2}} & A_{k_{3,3}} \\ A_{k_{4,1}} & A_{k_{4,2}} & A_{k_{4,3}} \end{bmatrix} \begin{bmatrix} \tilde{x}(w) \\ \tilde{y}(w) \\ \tilde{z}(w) \end{bmatrix} \leq \left(\frac{1+e+(1-e)w^2}{1+w^2}\right) \begin{bmatrix} b_{k_1} \\ b_{k_2} \\ b_{k_3} \\ b_{k_4} \end{bmatrix} \tag{45}$$

where the i^{th} row is of the form:

$$A_{k_{i1}}\tilde{x}(w) + A_{k_{i2}}\tilde{y}(w) + A_{k_{i3}}\tilde{z}(w) \leq \left(\frac{1+e+(1-e)w^2}{1+w^2}\right) b_{k_i}, \tag{46}$$

Replacing $\tilde{x}(w)$, $\tilde{y}(w)$ and $\tilde{z}(w)$ by their expressions from (37), the previous inequality is transformed into:

$$\frac{\Gamma_i^k(w)}{(1+w^2)^2} \geq 0 \tag{47}$$

with:

$$\begin{aligned} \Gamma_i^k(w) = \quad & b_{k_i}\left(\left(1+w^2\right)(1+e+(1-e)w^2\right)\right) - A_{k_{i1}}\left[P_x(w) + 3d_4 P_{Jx}(w)J(w)\right] \\ & -A_{k_{i2}}(1+w^2)P_y(w) - A_{k_{i3}}\left[P_z(w) + 2d_4 P_{Jz}(w)J(w)\right]. \end{aligned} \tag{48}$$

We aim to get polynomial inequality constraints, for which some strong results are known in the optimization literature. The function $J(w)$ being non-rational, it is replaced by the two bounding polynomials Θ_l and Θ_u. Two polynomials, Γ_{il}^k and Γ_{iu}^k, are then obtained. Hence, the inequality (47) becomes a pair of inequalities with Γ_{il}^k and Γ_{iu}^k, that must be repeated for each constraint i (rows of A_k) and for each maneuver k. Finally, the whole constraint on the guidance

error is formulated as the polynomial non negativity constraints:

$$\begin{cases} 0 \leq \Gamma_{il}^k(w) &= b_{k_i}\left(\left(1+w^2\right)(1+e+(1-e)w^2\right)\right) - A_{k_{i1}}\left[P_x(w) + 3d_4 P_{Jx}(w)\Theta_l(w)\right] \\ & \quad -A_{k_{i2}}(1+w^2)P_y(w) - A_{k_{i3}}\left[P_z(w) + 2d_4 P_{Jz}(w)\Theta_l(w)\right] \\ 0 \leq \Gamma_{iu}^k(w) &= b_{k_i}\left(\left(1+w^2\right)(1+e+(1-e)w^2\right)\right) - A_{k_{i1}}\left[P_x(w) + 3d_4 P_{Jx}(w)\Theta_u(w)\right] \\ & \quad -A_{k_{i2}}(1+w^2)P_y(w) - A_{k_{i3}}\left[P_z(w) + 2d_4 P_{Jz}(w)\Theta_u(w)\right] \end{cases} \tag{49}$$

for $i = \{1, \ldots, n_c\}$, for $k = \{0, \ldots, N-1\}$, $\forall w \in [w_k \ w_{k+1}]$.

3. SDP formulation for the guidance error

Representation theorems of the cone of non negative polynomials given in [14] are now used to translate these inequalities defined on an interval into a semidefinite programming problem. Indeed, non negativity of a univariate polynomial over finite or infinite intervals can be characterized by the existence of some symmetric positive semidefinite matrices linearly related to the vector of coefficients of the polynomial. For instance, the univariate polynomial π is non negative on $\mathbb{R}$ if and only if there exists a symmetric semidefinite matrix $Y \succeq 0$ of appropriate dimensions such that $\pi(w) = w^T Y w$. In our case, non negativity of polynomials Γ_{il}^k and Γ_{iu}^k are enforced on the intervals $[w_k \ w_{k+1}]$ for $k = \{0, \ldots, N-1\}$ and is equivalent to the following SDP conditions:

$$\begin{cases} \exists\, Y_{1il}^k, Y_{2il}^k \succeq 0 \quad \text{s.t.} \quad \gamma_{il}^k = \Lambda^*(Y_{1il}^k, Y_{2il}^k) \\ \exists\, Y_{1iu}^k, Y_{2iu}^k \succeq 0 \quad \text{s.t.} \quad \gamma_{iu}^k = \Lambda^*(Y_{1iu}^k, Y_{2iu}^k) \end{cases} \tag{50}$$

for $i = \{1, \ldots, n_c\}$, for $k = \{0, \ldots, N-1\}$. Γ_{il}^k and Γ_{iu}^k are represented by their vector of coefficients γ_{il}^k and γ_{iu}^k, respectively. Let d be the degree of polynomial Γ_*^k, then the linear operator Λ^* and the square matrices Y_*^k are defined by:

- if d is odd, define $\overline{d} = (d-1)/2$, then $Y_*^k \in \mathbb{R}^{(\overline{d}+1)\times(\overline{d}+1)} \succeq 0$ and the linear operator $\Lambda^* : \mathbb{R}^{(\overline{d}+1)\times(\overline{d}+1)} \times$

$\mathbb{R}^{(\overline{d}+1)\times(\overline{d}+1)} \to \mathbb{R}^{(\overline{d}+1)}$ is given by:

$$\Lambda^*(Y_{1*}^k, Y_{2*}^k) = \begin{bmatrix} \text{trace}\left(Y_{1*}^k(-w_k H_{\overline{d},1}) + (Y_{2*}^k w_{k+1} H_{\overline{d},1})\right) \\ \text{trace}\left(Y_{1*}^k(H_{\overline{d},1} - w_k H_{\overline{d},2}) + Y_{2*}^k(w_{k+1} H_{\overline{d},2} - H_{\overline{d},1})\right) \\ \vdots \\ \text{trace}\left(Y_{1*}^k(H_{\overline{d},i-1} - w_k H_{\overline{d},i}) + Y_{2*}^k(w_{k+1} H_{\overline{d},i} - H_{\overline{d},i-1})\right) \\ \vdots \\ \text{trace}\left(Y_{1*}^k H_{\overline{d},2\overline{d}+1} - Y_{2*}^k H_{\overline{d},2\overline{d}+1}\right) \end{bmatrix}$$

- if d is even, define $\overline{d} = d/2$, then $Y_{1*}^k \in \mathbb{R}^{(\overline{d}+1)\times((\overline{d}+1)} \geq 0$, $Y_{2*}^k \in \mathbb{R}^{(\overline{d})\times((\overline{d})} \geq 0$ and the linear operator $\Lambda^* : \mathbb{R}^{(\overline{d}+1)\times((\overline{d}+1)} \times \mathbb{R}^{(\overline{d})\times((\overline{d})} \to \mathbb{R}^{(\overline{d}+1)}$ is given by:

$$\Lambda^*(Y_{1*}^k, Y_{2*}^k) = \begin{bmatrix} \text{trace}\left(Y_{1*}^k H_{\overline{d},1} - Y_{2*}^k w_k w_{k+1} H_{\overline{d}-1,1}\right) \\ \text{trace}\left(Y_{1*}^k H_{\overline{d},2} + Y_{2*}^k((w_{k+1} + w_k) H_{\overline{d}-1,1} - w_k w_{k+1} H_{\overline{d}-1,2})\right) \\ \text{trace}\left(Y_{1*}^k H_{\overline{d},3} + Y_{2*}^k((w_{k+1} + w_k) H_{\overline{d}-1,2} - H_{\overline{d}-1,1} - w_k w_{k+1} H_{\overline{d}-1,3})\right) \\ \vdots \\ \text{trace}\left(Y_{1*}^k H_{\overline{d},i} + Y_{2*}^k((w_{k+1} + w_k) H_{\overline{d}-1,i-1} - H_{\overline{d}-1,i-2} - w_k w_{k+1} H_{\overline{d}-1,i})\right) \\ \vdots \\ \text{trace}\left(Y_{1*}^k H_{\overline{d},2\overline{d}} + Y_{2*}^k((w_{k+1} + w_k) H_{\overline{d}-1,2\overline{d}-1} - w_k w_{k+1} H_{\overline{d}-1,2\overline{d}-2})\right) \\ \text{trace}\left(Y_{1*}^k H_{\overline{d},2\overline{d}+1} - Y_{2*}^k H_{\overline{d}-1,2\overline{d}-1}\right) \end{bmatrix}$$

where $H_{j,i} \in \mathbb{R}^{(j+1)\times((j+1)}$ are Hankel matrices with 1 on the ith antidiagonal and 0 elsewhere.

D. Definition of the Cost Function

Apart from the control of each hump height during the glideslope, the other main objective of the approach is to minimize the fuel consumption during the transfer. As 6 ungimbaled identical chemical thrusters are used, the cost function may be naturally defined as the 1-norm of the $N + 1$ impulsive thrusts:

$$\Upsilon(N) = \sum_{k=0}^{N} \|\Delta\vec{V}_k\|_1 \tag{51}$$

The formulation (51) is transformed in order to express the above criterion with respect to the decision variables $\vec{v}_{k_+}$, $k = 0, \cdots, N-1$, λ_j, $j = 1, \cdots, N-2$ and $\Delta\vec{V}_N$:

$$\Upsilon(N) \quad = \quad \|\vec{v}_{0_+} - \vec{v}_{0_-}\|_1 + \sum_{k=1}^{N-1} \|\vec{v}_{k_+} - \Phi_{v\rho}^{[k-1]}(\vec{\rho}_0 + \lambda_{k-1}\vec{u}_{gls}) - \Phi_{vv}^{[k-1]}\vec{v}_{k-1_+}\|_1 + \|\Delta\vec{V}_N\|_1 \tag{52}$$

where $\vec{v}_{0_-}$ is the initial velocity vector. This cost function involving absolute values can be transformed into a linear function with the introduction of new slack variables and inequality constraints (see the reference [19, Section 1.3] for technical details about this usual trick used in linear programming):

$$\begin{aligned}
\vec{v}_{0_+} - \vec{v}_{0_-} &\leq \alpha_0 \\
-\left(\vec{v}_{0_+} - \vec{v}_{0_-}\right) &\leq \alpha_0 \\
\vec{v}_{k_+} - \Phi_{v\rho}^{[k-1]}(\vec{\rho}_0 + \lambda_{k-1}\vec{u}_{gls}) - \Phi_{vv}^{[k-1]}\vec{v}_{k-1_+} &\leq \alpha_k \\
-\left(\vec{v}_{k_+} - \Phi_{v\rho}^{[k-1]}(\vec{\rho}_0 + \lambda_{k-1}\vec{u}_{gls}) - \Phi_{vv}^{[k-1]}\vec{v}_{k-1_+}\right) &\leq \alpha_k \\
\Delta\vec{v}_N \leq \alpha_N,\ -\Delta\vec{v}_N &\leq \alpha_N
\end{aligned} \tag{53}$$

where α_k are extra decision variables, and the cost function becomes $\Upsilon(N) = \sum_{k=0}^{N} \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \alpha_k$.

E. A semidefinite programming problem

After having defined all the different ingredients in the previous subsections, the last step consists in gathering them in a compact formulation. Therefore, a solution to the initial minimum-fuel glideslope guidance problem may be obtained via the solution of the semidefinite programming problem where the decision variables are:

- the auxiliary variables α_k for $k = \{0, \ldots, N\}$;
- the position parameters λ_k for $k = \{1, \ldots, N-1\}$;
- the velocity vectors $\vec{v}_{k_+}$ for $k = \{0, \ldots, N-1\}$;
- the last increment vector $\Delta\vec{V}_N$;
- the semidefinite matrices Y_{1il}^k, Y_{2il}^k, Y_{1iu}^k, Y_{2iu}^k for $i = \{1, \ldots, n_c\}$, for $k = \{0, \ldots, N-1\}$.

$$
\begin{aligned}
&\min\ c^T\alpha\\
&\text{s.t.}\\
&\lambda_{k+1}\vec{u}_{gls} - \lambda_k\Phi_{\rho\rho}^{[k]}\vec{u}_{gls} - \Phi_{\rho v}^{[k]}\vec{v}_{k_+} = (\Phi_{\rho\rho}^{[k]} - \mathcal{I}_3)\vec{\rho}_0,\ k = 1,\ldots,N-1\\
&\lambda_1\vec{u}_{gls} - \Phi_{\rho v}^{[0]}\vec{v}_{0_+} = (\Phi_{\rho\rho}^{[0]} - \mathcal{I}_3)\vec{\rho}_0\\
&\vec{v}_f - \Phi_{v\rho}^{[N-1]}\vec{\rho}_0 = \Phi_{v\rho}^{[N-1]}\vec{u}_{gls}\lambda_{N-1} + \Phi_{vv}^{[N-1]}\vec{v}_{N-1_+} + \Delta\vec{V}_N\\
&\\
&\alpha_0 \geq \vec{v}_{0_+} - \vec{v}_{0_-}\\
&\alpha_0 \geq -(\vec{v}_{0_+} - \vec{v}_{0_-})\\
&\alpha_k \geq \Phi_{v\rho}^{[k-1]}(\lambda_{k-1}\vec{u}_{gls} + \rho_0) + \Phi_{vv}^{[k-1]}\vec{v}_{k-1_+} - \vec{v}_{k_+}\\
&\alpha_k \geq \vec{v}_{k_+} - \Phi_{v\rho}^{[k-1]}(\lambda_{k-1}\vec{u}_{gls} + \rho_0) - \Phi_{vv}^{[k-1]}\vec{v}_{k-1_+},\ k = 1,\ldots,N-1\\
&\alpha_N \geq \Delta\vec{V}_N\\
&\alpha_N \geq -\Delta\vec{V}_N\\
&\\
&\gamma_{il}^k = \Lambda^*(Y_{1il}^k, Y_{2il}^k),\ i = \{1,\ldots,n_c\},\ k = \{0,\ldots,N-1\}\\
&\gamma_{iu}^k = \Lambda^*(Y_{1iu}^k, Y_{2iu}^k)\\
&Y_{1il}^k \succeq 0,\ Y_{2il}^k \succeq 0\\
&Y_{1iu}^k \succeq 0,\ Y_{2iu}^k \succeq 0
\end{aligned}
\tag{54}
$$

with $c = \mathbb{1}_{3(N+1)}$ and $\alpha = \begin{bmatrix} \alpha_0^T & \ldots & \alpha_N^T \end{bmatrix}^T$. When considering a second-order polynomial approximation, we have that $Y_{1il}^k \in \mathbb{R}^{4\times4}$, $Y_{2il}^k \in \mathbb{R}^{3\times3}$, $Y_{1iu}^k \in \mathbb{R}^{4\times4}$ and $Y_{2iu}^k \in \mathbb{R}^{3\times3}$. Vectors γ_{il}^k and γ_{iu}^k are the given vectors of coefficients of polynomials Γ_{il}^k and Γ_{iu}^k. n_c is the number of rows of matrix A_k. This problem can hence be solved efficiently with an SDP solver. It is worthy to note that if constraints on the guidance error are removed, the optimization problem is reduced to an LP problem.

Remark 1 *As noted in the introduction of this section, an unequally spaced time grid could have been used without changing much the proposed algorithm. In* (54)*, one needs to evaluate the transition submatrices* $\Phi_{**}^{[k]}$ *at* t_k *which is an input data even in the case of an unequally spaced time grid.*

IV. Simplified Linear Programming Solutions for V-bar and R-bar Approaches

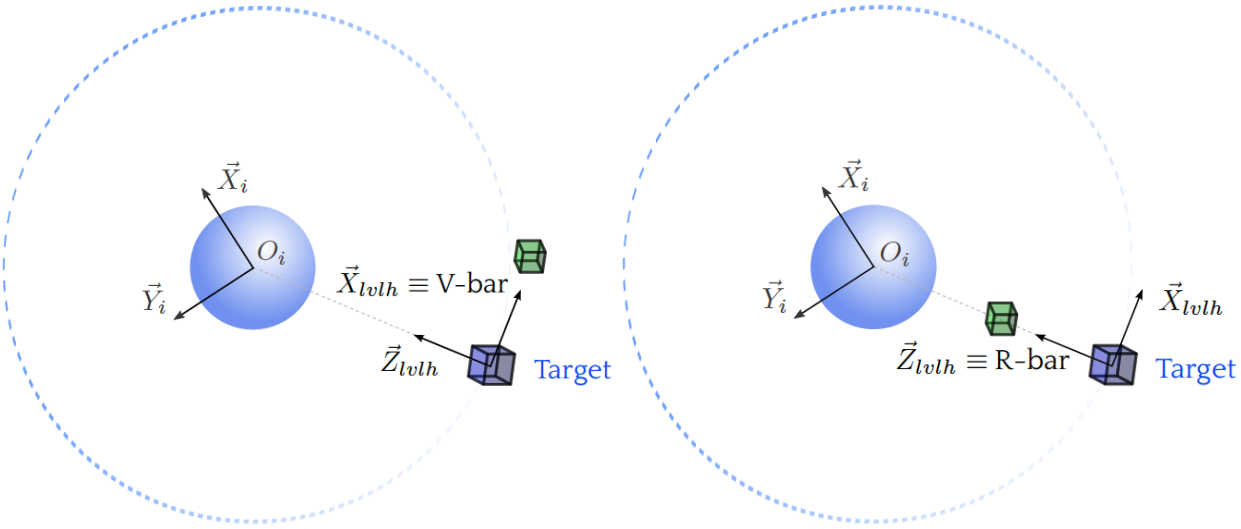
The goal here is to show that when the direction of the glideslope is defined to be the V-bar or the R-bar directions and for a circular reference orbit, the previous semidefinite programming problem may be simplified by using analytical developments to characterize the bounding corridor. Indeed, the main development is the same as for the previous

section: we first define the glideslope line tracking constraints according to the classical approach. The maximum guidance error is then addressed by defining constraints on the hump profile and bounding them for both V-bar and R-bar cases. Finally, we propose a linear programming formulation allowing to obtain a minimum-fuel solution for the glideslope guidance problem while controlling the guidance error for both V-bar and R-bar approaches.

A. V-bar and R-bar approaches

In the literature, we find several missions that were performed following the V-bar classical approach, such as the ESA's ATV program [1] or the NASA's Space Shuttle mission [7]. Among the R-bar approach, Japan's HTV [5] or NASA's Cygnus program [3] may be distinguished.

These two common approach strategies are based on the directions of approaching a target in the close range phase of the rendezvous mission in a circular reference orbit, as shown in Figure 4. This is mainly due to observability — LoS — constraints and safety reasons imposing the requirement of a trajectory belonging to a cone-shaped approach corridor.



(a) V-bar approach.

(b) R-bar approach.

Figure 4 Chaser satellite following a V-bar and R-bar glideslope approach.

V-bar and R-bar approaches are specific motions by which the chaser approaches the target in the tangential direction (V-bar) or radial direction (R-bar), meaning that the motion is restricted to the target orbital plane. In the case of a circular orbit, the relative dynamics can be simplified by taking advantage of the fact that the dynamic matrix is Linear Time Invariant (LTI). By defining the relative position vector as $\vec{\rho}(t) = [x \;\; z]^T$ and the relative velocity as $\vec{v}(t) = [\dot{x} \;\; \dot{z}]^T$, the four partitions of $\Phi(t, t_0)$ have a simpler expression by just selecting the appropriate rows and columns. Having set the interval Δt, equal to T/N, the transition matrix is constant. We have therefore $\Phi_{\rho\rho}^{[k]} = \Phi_{\rho\rho}$, $\Phi_{\rho v}^{[k]} = \Phi_{\rho v}$, $\Phi_{v\rho}^{[k]} = \Phi_{v\rho}$

and $\Phi_{vv}^{[k]} = \Phi_{vv}$, as:

$$\Phi_{\rho\rho} = \begin{bmatrix} 1 & 6\,(n\Delta t - \sin(n\Delta t)) \\ 0 & 4 - 3\cos(n\Delta t) \end{bmatrix}, \qquad \Phi_{\rho v} = \begin{bmatrix} (4/n)\sin(n\Delta t) - 3\Delta t & (2/n)(1-\cos(n\Delta t)) \\ -(2/n)(1-\cos(n\Delta t)) & \sin(n\Delta t)/n \end{bmatrix}$$
$$\Phi_{v\rho} = \begin{bmatrix} 0 & 6\,n(1-\cos(n\Delta t)) \\ 0 & 3n\sin(n\Delta t) \end{bmatrix}, \qquad \Phi_{vv} = \begin{bmatrix} -3 + 4\cos(n\Delta t) & 2\sin(n\Delta t) \\ -2\sin(n\Delta t) & \cos(n\Delta t) \end{bmatrix} \tag{55}$$

for all $k = 0, 1, \ldots, N-1$.

Depending on the choice for a V-bar approach or an R-bar approach, the relative position vector to be considered will be different. For a V-bar approach, the z component along the $\vec{Z}_{ot}$ axis remains constant and equals its initial value, z_0, leading to:

$$\vec{\rho}_0 = \begin{bmatrix} x_0 \\ z_0 \end{bmatrix}, \qquad \vec{\rho}_k = \begin{bmatrix} x_k \\ z_0 \end{bmatrix}, \qquad \vec{\rho}_f = \begin{bmatrix} x_f \\ z_0 \end{bmatrix} \tag{56}$$

On the other hand, for an R-bar approach, the x component along the $\vec{X}_{ot}$ axis remains constant and equals x_0:

$$\vec{\rho}_0 = \begin{bmatrix} x_0 \\ z_0 \end{bmatrix}, \qquad \vec{\rho}_k = \begin{bmatrix} x_0 \\ z_k \end{bmatrix}, \qquad \vec{\rho}_f = \begin{bmatrix} x_0 \\ z_f \end{bmatrix} \tag{57}$$

Hereafter, considering a specific direction (V-bar or R-bar) and a circular reference orbit will significantly simplify the constraints on guidance error. On the other hand, the other features (glideslope line tracking, final velocity, cost function) remain basically of the same form.

B. Constraints on guidance error

In order to control the maximum guidance error, we first need to define the error in the orbital plane $\vec{\epsilon}$ in a general framework for both V-bar and R-bar approaches.

Let us define the point A in the $x - z$ plane (see Figure 5) as the orthogonal projection of the chaser position $\vec{\rho}$ on the glideslope straight line. The excursion $\vec{\epsilon}$ is given by the difference between vectors $\vec{\rho}$ and $\vec{\rho}_{gls}$: $\vec{\epsilon}(t) = \vec{\rho}(t) - \vec{\rho}_{gls}(t)$. The auxiliary vector $\vec{\rho}_{gls}$ can be itself expressed as:

$$\vec{\rho}_{gls}(t) = \vec{\rho}_0 + (\Delta\vec{\rho}^{\,T}\ \vec{u}_{gls})\ \vec{u}_{gls} \tag{58}$$

where $\Delta\vec{\rho} = \vec{\rho}(t) - \vec{\rho}_0$ is also an auxiliary vector and $\vec{u}_{gls}$ is the unit vector defining the glideslope straight line, defined

in Equation (12), but for the in-plane case, it becomes:

$$\vec{u}_{gls} = \begin{bmatrix} \dfrac{x_f - x_0}{\lambda_f} & 0 & \dfrac{z_f - z_0}{\lambda_f} \end{bmatrix} \tag{59}$$

Hence, the excursion can be written as:

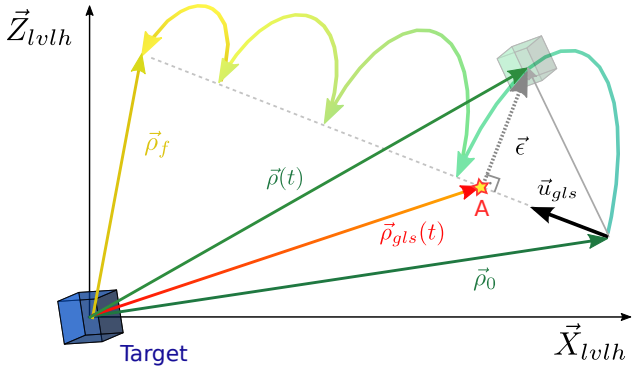


Figure 5 In-plane glideslope approach.

$$\vec{\epsilon}(t) = \Delta\vec{\rho} - (\Delta\vec{\rho}^{\,T}\,\vec{u}_{gls})\,\vec{u}_{gls} \tag{60}$$

and the guidance error norm is then deduced as:

$$\begin{aligned} \| \vec{\epsilon} \|^2 &= \left[\Delta\vec{\rho} - (\Delta\vec{\rho}^{\,T}\vec{u}_{gls})\,\vec{u}_{gls}\right]^T \left[\Delta\vec{\rho} - (\Delta\vec{\rho}^{\,T}\vec{u}_{gls})\,\vec{u}_{gls}\right] \\ &= \Delta\rho_x^2(1 - u_{gls_x}^2) + \Delta\rho_z^2(1 - u_{gls_z}^2) - 2u_{gls_x}u_{gls_z}\Delta\rho_x\Delta r_z \end{aligned} \tag{61}$$

with $\Delta\rho_x = x - x_0$ and $\Delta\rho_z = z - z_0$.

1. Constraints on guidance error for V-bar

If the approach is performed along a line parallel to the x-axis, the unit direction vector is:

$$\vec{u}_{gls} = [1 \quad 0 \quad 0]^T \tag{62}$$

Hence, the guidance error norm in Expression (61) is reduced to:

$$\| \vec{\epsilon} \|^2 = \Delta\rho_z^2 \tag{63}$$

Using the HCW $z-$equation, we get:

$$\| \vec{\epsilon} \|^2 = \left(3\Big(1 - \cos(n(t-t_0))\Big)\, z_0 + \frac{2}{n}\Big(\cos(n(t-t_0)) - 1\Big)\, \dot{x}_0 + \frac{1}{n}\sin(n(t-t_0))\, \dot{z}_0\right)^2 \tag{64}$$

When the guidance error distance is maximal, the velocity vector $\vec{v}$ is parallel to the glideslope line. A maximum condition is then given by:

$$\dot{z} = 0 \;\Leftrightarrow\; 3\,n\;\sin(n(t-t_0))\, z_0 - 2\;\sin(n(t-t_0))\, \dot{x}_0 + \cos(n(t-t_0))\, \dot{z}_0 = 0 \tag{65}$$

Lemma 1 *Let us define Δt_m the value of the time at which the guidance error is maximal and Δt the value of the time when guidance error is zero, that is, when the satellite is back on the glideslope commanded path. Considering a single hump, we have the following relationship:*

$$\Delta t = 2\,\Delta t_m \tag{66}$$

meaning that the maximum occurs at the middle (in terms of time) of the hump. Proof. The proof is presented only in cases for which $n\Delta t \neq 2k\pi,\ k = 1, 2, \ldots$. Consider Equation (65) *on one hand and Equation $\|\vec{\epsilon}\| = 0$ on the other hand. In both cases, z_0 can be expressed as:*

$$\begin{aligned} z_0 =& \frac{2}{3\,n}\dot{x}_0 - \frac{\cos(n\Delta t_m)}{3\,n\;\sin(n\Delta t_m)}\;\dot{z}_0 \\ z_0 =& \frac{2}{3\,n}\dot{x}_0 + \frac{\sin(n\Delta t)}{3n(\cos(n\Delta t) - 1)}\;\dot{z}_0 \end{aligned} \tag{67}$$

Equalizing the two equations leads to:

$$\sin(n\Delta t)\;\sin(n\Delta t_m) = -\cos(n\Delta t_m)\;\cos(n\Delta t) + \cos(n\Delta t_m), \Leftrightarrow\; \cos(n\Delta t - n\Delta t_m) = \cos(n\Delta t_m) \tag{68}$$

Consequently, considering the time range only over a single hump, solutions of the above equation are $\Delta t = 2\,\Delta t_m$ and $\Delta t = 0$. □

From Equation (63), the guidance error distance can be readily upper-bounded as:

$$\begin{aligned} |\,\vec{\epsilon}\,| =& \left|(3 - 3\cos(n\Delta t))\, z_0 + \frac{2}{n}(\cos(n\Delta t) - 1)\, \dot{x}_0 + \frac{1}{n}\sin(n\Delta t)\, \dot{z}_0\right| \\ \leq& \left|(3 - 3\cos(n\Delta t_m))\, z_0 + \frac{2}{n}(\cos(n\Delta t_m) - 1)\, \dot{x}_0 + \frac{1}{n}\sin(n\Delta t_m)\, \dot{z}_0\right| \end{aligned} \tag{69}$$

It provides an inequality that will be used to specify maximum conditions on guidance error for each hump, $k = 0, 1, \ldots, N-1$:

$$\left|(3 - 3\cos(n\Delta t_m))\, z_0 + \frac{2}{n}(\cos(n\Delta t_m) - 1)\, \dot{x}_k + \frac{1}{n}\sin(n\Delta t_m)\, \dot{z}_k\right| \leq \delta_k \tag{70}$$

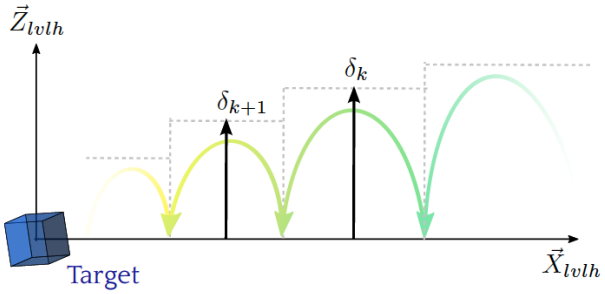


Figure 6 In-plane glideslope approach.

where δ_k is given and specifies the maximal allowable guidance error during the $(k+1)^{\text{th}}$ maneuver, as shown in Figure 6. Expression (70) is equivalently written as:

$$\begin{cases} (3 - 3\cos(n\Delta t_m))\, z_0 + \dfrac{2}{n}(\cos(n\Delta t_m) - 1)\, \dot{x}_k + \dfrac{1}{n}\sin(n\Delta t_m)\, \dot{z}_k \leq \delta_k \\ -(3 - 3\cos(n\Delta t_m))\, z_0 - \dfrac{2}{n}(\cos(n\Delta t_m) - 1)\, \dot{x}_k - \dfrac{1}{n}\sin(n\Delta t_m)\, \dot{z}_k \leq \delta_k \end{cases} \tag{71}$$

We recall that the interval Δt being constant, the parameter Δt_m is constant too. Being aware of the fact that $\vec{v}_{k_+} = [\dot{x}_k \quad \dot{z}_k]^T$, the constraints (71) for the V-bar approach can be written in a matrix form as:

$$M_V\, \vec{v}_{k_+} \leq \vec{M}_{V_k} \tag{72}$$

where:

$$M_V = \begin{bmatrix} \dfrac{2}{n}(\cos(n\Delta t_m) - 1) & \dfrac{1}{n}\sin(n\Delta t_m) \\ -\dfrac{2}{n}(\cos(n\Delta t_m) - 1) & -\dfrac{1}{n}\sin(n\Delta t_m) \end{bmatrix}, \qquad \vec{M}_{V_k} = \begin{bmatrix} \delta_k - (3 - 3\cos(n\Delta t_m))\, z_0 \\ \delta_k + (3 - 3\cos(n\Delta t_m))\, z_0 \end{bmatrix} \tag{73}$$

2. Constraints on guidance error for R-bar

If the approach is performed along a line parallel to the z-axis, the unit direction vector becomes:

$$\vec{u}_{gls} = [0 \quad 0 \quad 1]^T \tag{74}$$

Hence, the guidance error norm in Expression (61) is reduced to:

$$\| \vec{\epsilon} \|^2 = \Delta\rho_x^2 \tag{75}$$

Using the HCW $x-$equation, we get:

$$\| \vec{\epsilon} \|^2 = \left(6\Big(n(t-t_0) - \sin(n(t-t_0))\Big)\, z_0 + \frac{1}{n}\Big(4\sin(n(t-t_0)) - 3n(t-t_0)\Big)\, \dot{x}_0 + \frac{2}{n}\Big(1-\cos(n(t-t_0))\Big)\, \dot{z}_0\right)^2 \tag{76}$$

Unlike the V-bar case, there is not a simple relationship between the time Δt_m when the error is maximal and the maneuver interval Δt. Hence, a conservative bound is introduced.

Lemma 2 *Let us define Δt the value of the time when the guidance error is 0, that is, when the satellite is back on the glideslope commanded path. Considering a single hump such that $n\Delta t \in [0 \ \arccos\frac{3}{4}]$, a conservative upper-bound of the guidance error is given by:*

$$| \vec{\epsilon} | \leq \bar{\tau}_1 |z_0| + \bar{\tau}_2 |\dot{x}_0| + \leq \bar{\tau}_3 |\dot{z}_0| \tag{77}$$

with:

$$\begin{aligned} \bar{\tau}_1 &= \left|6(n\Delta t - \sin(n\Delta t))\right| \\ \bar{\tau}_2 &= \left|\frac{1}{n}(4\sin(n\Delta t) - 3n\Delta t)\right| \\ \bar{\tau}_3 &= \left|\frac{2}{n}(1-\cos(n\Delta t))\right| \end{aligned} \tag{78}$$

Proof. Let us first define the interval of duration of a generic hump by $[0 \ \Delta t]$ and apply the triangle inequality to Expression (76). The guidance error during a hump is bounded as the following:

$$| \vec{\epsilon} | \leq \overbrace{\left|6\Big(nt - \sin(nt)\Big)\right|}^{\tau_1(nt)} |z_0| + \overbrace{\left|\frac{1}{n}\Big(4\sin(nt) - 3nt\Big)\right|}^{\tau_2(nt)} |\dot{x}_0| + \underbrace{\left|\frac{2}{n}\Big(1-\cos(nt)\Big)\right|}_{\tau_3(nt)} |\dot{z}_0| \tag{79}$$

An upper-bound for the guidance error $| \vec{\epsilon} |$ may be obtained by bounding the functions $\tau_i(nt)$, $i = 1, 2, 3$ on the interval $[\ 0 \quad \Delta t\ ]$:

- For the first term, the derivative of the function $\tau_1(nt)$ with respect to its argument is equal to $1 - \cos(nt) \geq 0$. Hence, τ_1 is a monotone non decreasing function which will reach its maximum at $t = \Delta t$ on the interval $[\ 0 \quad \Delta t\ ]$. Therefore, we have that $\tau_1(nt) \leq \left|6(n\Delta t - \sin(n\Delta t))\right| = \bar{\tau}_1$ on $[\ 0 \quad \Delta t\ ]$.
- The function $\tau_2(nt)$ is more difficult to handle since it is nor differentiable nor monotone. The derivative with respect to its argument is equal to $4\cos(nt) - 3$ and its second derivative is equal to $-4\sin(nt)$. It may be easily deduced that the function τ_2 is differentiable and monotone non decreasing on $[0 \ \arccos\frac{3}{4}]$ and the maximum $\bar{\tau}_2$ is reached at $n\Delta t \leq \arccos\frac{3}{4}$ in this case.

- The function τ_3 is non negative and periodic with its derivative given by $\sin(nt) \geq 0$, if $nt \leq \pi$. This condition holds if the condition $n\Delta t \leq \arccos\frac{3}{4}$ is enforced. The upper-bound $\bar{\tau}_3$ is then deduced as the maximum of τ_3 on $[0 \quad \Delta t]$.

Gathering all these inequalities shows that the right-hand side of (77) is greater or equal to the right-hand side of (79), which in turn is greater than $|\vec{\epsilon}|$. □

REMARK 2 *It is important to notice that the condition* $n\Delta t \in [0 \ \arccos\frac{3}{4}]$ *induces a constraint on the glideslope transfer. For instance, this implies a minimum number of maneuvers for equally spaced maneuvers since* $\Delta t = t_f/N$. *Nevertheless, as shown in the numerical examples, this restriction is not too stringent. In any case, it is always possible to get rid of this constraint by deriving another upper-bound of the function* τ_2 *for* $\delta t \notin [0 \ \arccos\frac{3}{4}]$.

At each stage k, the constraints $|\vec{\epsilon}| \leq \bar{\tau}_1|z_k| + \bar{\tau}_2|\dot{x}_k| + \leq \bar{\tau}_3|\dot{z}_k| \leq \delta_k$ are then imposed for the R-bar approach to get a bounded guidance error. Keeping in mind that $\vec{v}_{k_+} = [\dot{x}_k \quad \dot{z}_k]^T$, these constraints can be rewritten in a matrix form as:

$$M_R \, \vec{v}_{k_+} \leq \vec{M}_{R_k} \tag{80}$$

where:

$$M_R = \begin{bmatrix} \bar{\tau}_2 & \bar{\tau}_3 \\ -\bar{\tau}_2 & -\bar{\tau}_3 \end{bmatrix}, \qquad \vec{M}_{R_k} = \begin{bmatrix} \delta_k - \bar{\tau}_1 \, z_k \\ \delta_k + \bar{\tau}_1 \, z_k \end{bmatrix} \tag{81}$$

Note that the maximal allowable excursion at each step is denoted by δ_k, which can be seen in Figure 6 with the corresponding permutation between the X_{lvlh} and Z_{lvlh} axes.

C. A linear programming problem

The optimization problem is now formulated. The glideslope line tracking constraint (24) and the final velocity constraint (28) require a slight adjustment. Positions on the glideslope $\vec{\rho}_k$ are now defined as (56) for V-bar, and (57) for R-bar (instead of (23)). Blocks of the transition matrices $\Phi_{**}^{[k]}$ reduce to 2×2 blocks (55), constant w.r.t. k. The velocity decision variables are obviously restricted to the plane : $\vec{v}_{k_+} = [\dot{x}_k \quad \dot{z}_k]^T$. Regarding the cost-related constraints (53), the same adjustments hold, and variables $\alpha_k \in \mathbb{R}^2$ (instead of $\mathbb{R}^3$).

Gathering all the constraints with the appropriate decision variables, an LP problem is built for which a solution provides a minimum-fuel glideslope guidance along V-bar or R-bar:

$$
\begin{aligned}
&\min\ c^T\alpha \\
&\text{s.t.} \\
&\vec{\rho}_{k+1} = \Phi_{\rho\rho}\vec{\rho}_k + \Phi_{\rho v}\vec{v}_{k_+},\ k = 0,\ldots,N-1 \\
&\Delta\vec{V}_N = \vec{v}_f - \Phi_{v\rho}\vec{\rho}_{N-1} - \Phi_{vv}\vec{v}_{N-1_+} \\
&\alpha_0 \geq \vec{v}_{0_+} - \vec{v}_{0_-},\ \alpha_0 \geq -\left(\vec{v}_{0_+} - \vec{v}_{0_-}\right) \\
&\alpha_k \geq \Phi_{v\rho}\vec{\rho}_{k-1} + \Phi_{vv}\vec{v}_{k-1_+} - \vec{v}_{k_+} \\
&\alpha_k \geq \vec{v}_{k_+} - \Phi_{v\rho}\vec{\rho}_{k-1} - \Phi_{vv}\vec{v}_{k-1_+},\ k = 1,\ldots,N-1 \\
&\alpha_N \geq \Delta\vec{V}_N,\ \alpha_N \geq -\Delta\vec{V}_N \\
&M_*\vec{v}_{k_+} \leq M_{*_k},\ k = 0,\ldots,N-1
\end{aligned}
\tag{82}
$$

with $c = \mathbb{1}_{2(N+1)}$ and $\alpha = \begin{bmatrix} \alpha_0^T & \ldots & \alpha_N^T \end{bmatrix}^T$. In the last inequality, $* = V$ or $* = R$ depending on the approach, and the corresponding matrices are defined in (73) and (81). $\vec{\rho}_k$ is defined as (56) for V-bar, and (57) for R-bar. Let us recall that scalars δ_k in M_{*_k} are user-defined parameters that set the maximum allowable magnitude for each hump.

Our objective in this paper is not to address specifically the feasibility of real-time implementation of the proposed (SDP or LP) algorithms on-board the chaser but some recent works and in-flight experiments have shown that involving linear programming on-board is now possible [2] even with the limited capability of typical flight processors (LEON2 or LEON3 family for instance). The technological implications of using optimization in autonomous space rendezvous have also been investigated later in [20], [21] for linear and quadratic programming, and in [22] for semidefinite programming, in a model predictive control framework.

V. Numerical examples

Three different examples are now used to illustrate the proposed results and compare them to existing ones. Two are within the PRISMA framework [23] and the last one is dedicated to a glideslope approach for a highly elliptic reference orbit. We want to call attention of the reader to the fact that the boxes containing hops and presented on the different figures have only two dimensions fixed *a priori* by the designer while the last one parallel to the glideslope direction is optimized by the algorithm. Note also that all the algorithms used in the paper have been run with MATLAB© on a 64-bit Linux OS (8GB RAM and intel core i7 processor).

A. Example 1

First, an illustration based on PRISMA is presented. The PRISMA program is a cooperative effort between the Swedish National Space Board (SNSB), the French Centre National d'Etudes Spatiales (CNES), the German Deutsche Zentrum für Luft- und Raumfahrt (DLR) and the Danish Danmarks Tekniske Universitet (DTU) [2]. Launched on

June 15, 2010 from Yasny (Russia), it was intended to test in-orbit new guidance schemes (particularly autonomous orbit control) for formation flying and rendezvous technologies. The orbital elements of the target orbit, as well as initial and final rendezvous conditions, are listed in Table 1.

Semi-major axis	a = 7011 km
Inclination	i = 98 deg
Argument of Perigee	ω = 0 deg
RAAN	Ω = 190 deg
Eccentricity	e = 0.004
Initial True Anomaly	θ_0 = 0 rad
t_0	0 s
$X_0^T = [\rho_0^T \; v_{0\_}^T]$	[−400 40 −50 −0.5 0 0] m -m/s
T	1500 s
$X_f^T = [\rho_f^T \; v_f^T]$	[−40 0 −10 0 0 0] m -m/s
N	6

Table 1 PRISMA rendezvous characteristics.

In this first example, the target evolves on a quasi-circular orbit and the chaser must be transferred from $\vec{\rho}_0$ to $\vec{\rho}_f$ in 6 maneuvers (N) during 25 min (t_f = 1500 s) following a glideslope line. Three algorithms are compared: the classical circular glideslope method, the classical elliptic glideslope method as it may be built from the information obtained in [11] and the proposed optimal glideslope algorithm. In all cases, the resulting impulsive control sequence $\Delta\vec{V}_i$ is applied and propagated via the Tschauner-Hempel equations. The final position of the chaser resulting from the the classical circular glideslope method is $[-49.2, -0.2, -16]$ m, that is 11 m away from the planed final position. We aim at emphasizing this significant drift induced when using the glideslope control computed with the Hill-Clohessy-Wiltshire equations even when the reference orbit is quasi-circular. Figure 7 shows the chaser trajectories for the different approaches. Regarding the velocity profile along the glideslope, the relevant parameters are set such that $\dot{\lambda}_0 = 1$ and $\dot{\lambda}_{t_f}$ is imposed by the maneuver duration t_f. The global consumptions of the the classical circular glideslope and classical elliptic glideslope algorithms are similar and given respectively by 2.9705 m/s and 2.977 m/s. The specifications for the proposed optimal algorithm are defined by a final velocity $\vec{v}_f = \vec{0}$ m/s and a constraint on the trajectory characterized by a 10 m ×10 m corridor. The optimal algorithm produces an optimal solution in 1.5 s with a consumption given by 1.9698 m/s which is much lower than the two classical approaches.

The sequences of impulsive maneuvers for the classical elliptic glideslope and the minimum-fuel algorithm are detailed below in Table 2 and depicted in Figure 8. Note also that the classical trajectories do not respect the admissible corridor in green.

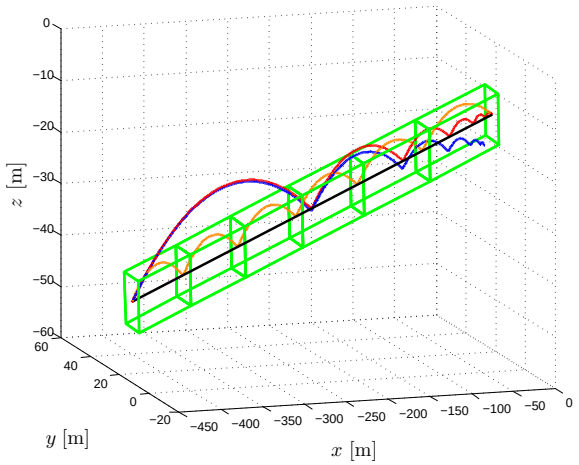


Figure 7 *Chaser relative trajectories, classical circular glideslope (blue) vs. classical elliptic glideslope (red) vs. proposed optimal algorithm (orange).*

time (s)	0	250	500	750	1000	1250	1500
standard glideslope							
ΔV_x (m/s)	1.1784	-0.3757	-0.1915	-0.0976	-0.0498	-0.0254	-0.0245
ΔV_y (m/s)	-0.0749	0.0455	0.0231	0.0117	0.0058	0.0029	0.0028
ΔV_z (m/s)	0.2879	0.2779	0.1455	0.0780	0.0436	0.0261	0.0083
optimal glideslope							
ΔV_x (m/s)	0.6930	0	0	0	0	0	-0.2785
ΔV_y (m/s)	-0.0173	0.0087	0.0068	0.0048	0.0027	0.0004	0.0312
ΔV_z (m/s)	0.0979	0.1525	0.1543	0.1560	0.1577	0.1595	0.0486

Table 2 *Impulsive control sequences for Example 1.*

B. Example 2

The PRISMA framework is now used to illustrate the results obtained for a V-bar approach in a circular orbit. Scenario parameters from Table 1 are identical, except for the eccentricity which is set to 0 to define a circular reference orbit. The out-of-plane component of the initial and final positions (ρ_{0_y} and ρ_{f_y}) are set to 0, and the initial and final $z-$components must be equal: $\rho_{0_z} = \rho_{f_z} = -20\ m$. The transfer time is set to 540 s as in the scenario 1 in [8] which consists in an Inbound/Outbound V-bar glideslope and In-plane circumnavigation with different starting and final relative positions and velocities. Here, only the first part of this scenario is relevant to our numerical application. The proposed optimal V-bar glideslope approach, the classical circular glideslope applied to our particular case and the two-impulse method from [4] are compared on Figure 9. Regarding the proposed algorithm, three different sets of constraints on the maximum allowable excursion have been enforced (green lines). Results for each method are summarized in Table 3. The new algorithm gives a solution in $0.3 - 0.5$ s which reduces significantly the consumption

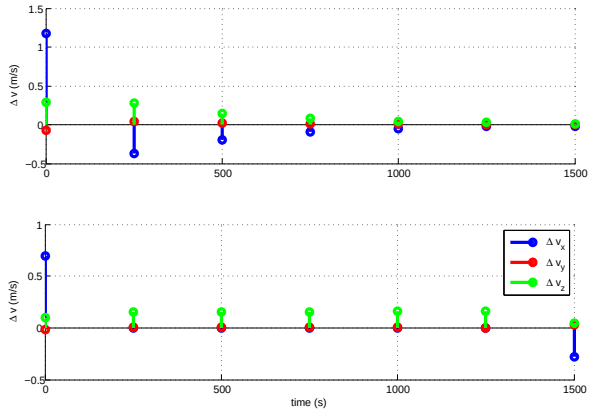
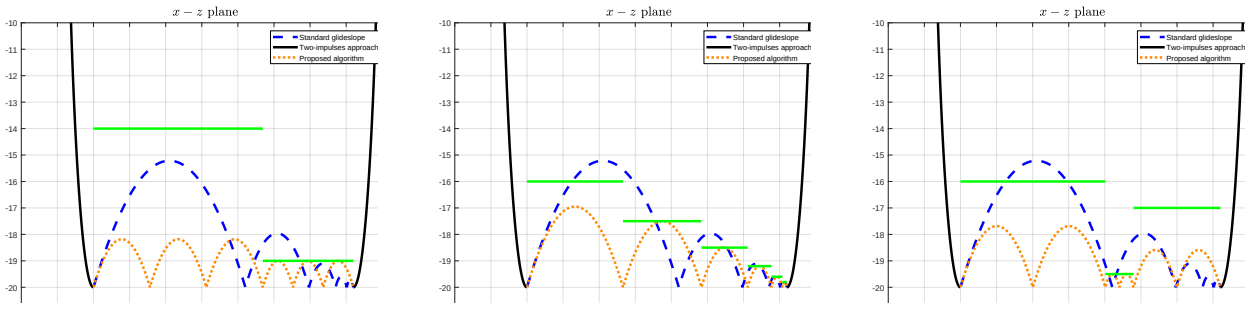


Figure 8 *Impulsive control sequences for the classical elliptic glideslope (top) and for the optimal glideslope (bottom) in Example 1.*

while keeping the chaser close to the commanded path. As expected, the two-impulse method is very efficient from the consumption point of view but is less successful regarding the guidance error.



(a) (b) (c)

Figure 9 V-bar circular glideslope trajectories for different constraint sequences.

	Consumption [m/s]			Hop height [m]		
Simulations	(a)	(b)	(c)	(a)	(b)	(c)
Classical circular glideslope [8]	5.87			4.78		
Two-impulse method [4]	0.5			117		
optimal V-bar glideslope	2.98	4.19	4.37	1.81	3.04	2.31

Table 3 *Simulation results for Example 2, (cf. Figure 9).*

In Figure 9a, the sequence of bounds on the trajectory is given by the vector [6 6 6 1 1 1]. While there is no constraint on the location of intermediate positions along the commanded path (order or distance, x_k totally free), it is interesting to see that, for identical excursion constraints, the optimization naturally spaces them evenly. Besides, minimizing the consumption does not necessarily lead to a trade-off with guidance error. The first three humps do not reach the allowable limit. However, when the trajectory constraints become tighter on some maneuvers, humps

become larger on others. For instance, the vector of bounds, given by [4 2.5 1.5 0.8 0.4 0.2] and illustrated in Figure 9b may typically arise from a LoS requirement. Demanding smaller guidance errors in the last maneuvers induces larger humps in the first ones. One can thus observe a waterbed effect where, for a given distance λ_f to travel, short humps need to be offset by larger ones, the number of maneuvers N being fixed. It thus appears that the optimal algorithm provides velocity increments such that the range of the excursion is related to the travelled distance along the glideslope (for each hump). Obviously, this is not true for the classical circular glideslope algorithm where large humps with close intermediate positions may occur. This observation can also be made in the third case in Figure 9c. Indeed, the guidance error constraints make the "shaping" of the trajectory profile possible.

Some additional numerical experiments are presented to see how the number of waypoints (N) may affect the fuel consumption and the guidance error. Results for different number of waypoints and slack guidance error constraints ($\delta_k \leq 6$m) are given in Table 4. It can be noticed that the consumption remains similar for all N while the maximum trajectory excursion changes with the number of maneuvers.

N	6	7	8	12	20
Actual guidance error [m]	1.40	1.04	0.79	0.35	0.12
Consumption [m/s]	2.58	2.58	2.58	2.58	2.58

Table 4 Consumption and guidance error for different numbers of waypoints

C. Example 3

In this third example, the classical elliptic glideslope approach and the proposed optimal algorithm are compared in a case for which the eccentricity of the reference orbit is high. The parameters defining the conditions of the rendezvous are set as follows:

$$t_f = 540 \text{ s},\ N = 4,\ n = 0.001 \text{ rad/s},\ e = 0.7,\ \vec{v}_f = \vec{0},$$

$$\vec{\rho}_0 = \begin{bmatrix} -500 \\ 10 \\ 30 \end{bmatrix} \text{m},\ \vec{\rho}_f = \begin{bmatrix} -100 \\ 0 \\ 20 \end{bmatrix} \text{m},\ \vec{v}_{0_-} = \begin{bmatrix} 0 \\ 0 \\ 0.5 \end{bmatrix} \text{m/s}.$$

Regarding the constraints on the trajectory guidance error, we define for each maneuver a box (or a corridor) characterized by 4 planes with (31). Thus, 4 boxes centered around the glideslope line are defined:

maneuver	1	2	3	4
height [m]	90	50	30	20
width [m]	10	10	10	10

Figure 10 depicts the chaser trajectories for the two methods and where the four boxes corresponding to trajectory constraints are represented in green. The consumption of the classical elliptic glideslope algorithm is 7.74 m/s whereas the consumption of our optimal algorithm, giving a solution in less than 1.5 s, is 4.53 m/s. The respective sequences of impulsive maneuvers are presented in Table 5 and depicted in Figure 11.

time (s)	0	135	270	405	540
classical elliptic glideslope					
ΔV_x (m/s)	2.2830	-1.5736	-0.3350	-0.0669	-0.0165
ΔV_y (m/s)	-0.0512	0.0544	0.0095	0.0017	0.0004
ΔV_z (m/s)	1.3608	1.6940	0.2323	0.0486	0.0114
optimal glideslope					
ΔV_x (m/s)	0.9751	0	0	0	-0.6517
ΔV_y (m/s)	-0.0129	0.0168	0.0039	0.0009	0.0165
ΔV_z (m/s)	0.5938	1.2648	0.5613	0.3079	0.1267

Table 5 *Impulsive control sequences for Example 3.*

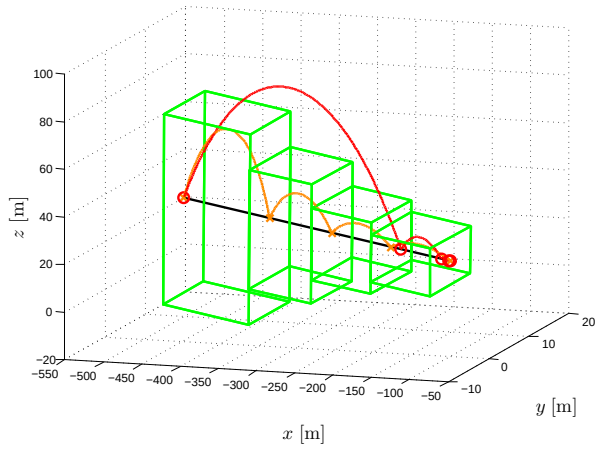


Figure 10 *Chaser relative trajectories: classical elliptic glideslope algorithm (red) vs. proposed optimal algorithm (orange).*

VI. Conclusions

A revisited solution to the problem of the impulsive close range rendezvous along a glideslope line is proposed, first for any type of approach, and later on, in the specific cases of the V-bar and R-bar approaches. The motivations for the development of a new solution to the glideslope method come from two main issues regarding the standard algorithm: the uncontrolled humps, inherent to the impulsive method and the relative motion dynamics, and the fuel consumption that is assessed a posteriori. These shortcomings are addressed via a reformulation of the problem in an optimization framework. To this end, some degrees of freedom are introduced by setting free suitable variables.

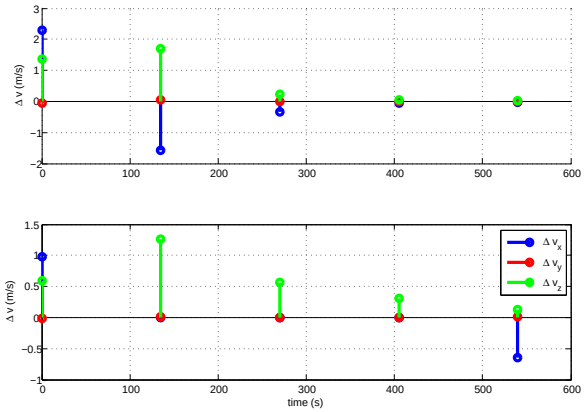


Figure 11 *Impulsive control sequences for the classical elliptic glideslope (top) and for the optimal glideslope (bottom) in example 3.*

Then, combining an appropriate parametrization of the Tschauner-Hempel equations and a well-known result on non negative polynomials, the design of the glideslope guidance algorithm becomes a semidefinite programming problem in the most general case.

The main design features included in the new proposed glideslope algorithm are on one hand the minimization of the consumption and, on the other hand, the possibility to specify an admissible volume for each hump of the relative trajectory and therefore to control the guidance error all along the rectilinear path. From this general case using an SDP formulation, and going further into the analysis with analytical expressions, we derived the solution via an LP problem for the specific cases of V-bar and R-bar approaches. This solution yields optimal consumption while respecting a user-defined maximum bound profile on the trajectory, which turns out to be very useful when dealing with visibility constraints while keeping a reasonable numerical computation complexity.

Different numerical examples illustrate the usefulness of the new methods compared to the classical ones, especially when the approach corridor has to verify stringent geometrical restrictions such as LoS constraints. The consumption is also significantly reduced.

Acknowledgments The authors would like to thank J.C. Berges from CNES Toulouse for the grants REF. ACTION R & T- R-S-13-BS0005-014, DAJ/AR/EO 2013-0008153 that partly supports this activity.

References

[1] Fabrega, J., Frezet, M., and Gonnaud, J.-L., "ATV GNC during rendezvous," *Proceedings of the Third International Conference on Spacecraft Guidance, Navigation and Control Systems*, Noordwijk The Netherlands, 1996.

[2] Larsson, R., Berge, S., Bodin, P., and Jönsson, U., "Fuel efficient relative orbit control stategies for formation flying and rendezvous within Prisma," 29^{th} *Annual AAS Guidance and Control Conference*, Breckenridge, Colorado, USA, 2006.

[3] Miotto, P., Breger, L., Mitchell, I., Keller, B., and Rishikof, B., "Designing and validating proximity operations rendezvous and approach trajectories for the Cygnus mission," *AIAA Guidance, Navigation, and Control Conference*, AIAA-2010-8446, Toronto, Ontario, Canada, 2009. doi:10.2514/6.2010-8446.

[4] Fehse, W. (ed.), *Automated rendezvous and docking of spacecraft*, Cambridge Aerospace Series, Cambridge University Press, Cambridge, UK, 2003. doi:10.1017/CBO9780511543388.

[5] Ueda, S., Kasai, T., and Uematsu, H., "HTV Rendezvous Technique And GNC Design Evaluation Based on 1st Flight On-Orbit Operation Result," *AIAA/AAS Astrodynamics Specialist Conference, AIAA 2010-7664*, Toronto, Ontario, Canada, 2010. doi:10.2514/6.2010-7664.

[6] Wen, C., and Gurfil, P., "Guidance, Navigation and Control for Autonomous R-bar Proximity Operations for Geostationary Satellites," *Proceedings of Institution of Mechanical Engineers Part G: Journal of Aerospace Engineering*, 2016.

[7] Pearson, D., "The glideslope approach," *Advances in Astronautical Sciences*, Vol. 69, 1989, pp. 109–123.

[8] Hablani, H., Tapper, M., and David J. Dana-Bashian, D., "Guidance and relative navigation for autonomous rendezvous in a circular orbit," *Journal of Guidance, Control and Dynamics*, Vol. Vol. 25, No. No. 3, 2002.

[9] Goodman, J., "Approaches and Fly-Arounds for Spacecraft Proximity Operations," *Proceedings of the 27th Space Flight Mechanics Meeting, AAS 17-272*, San Antonio, TX, USA, 2017.

[10] Wang, F., Cao, X., and Chen, X., "Guidance Algorithms for the Near-Distance Rendezvous of On-Orbit-Servicing Spacecraft," *Transactions of Japanese Society for Aeronautical and Space Sciences*, Vol. 50, No. 167, 2007, pp. 9–17.

[11] Okasha, M., and Newman, B., "Guidance, Navigation and Control for Satellite Proximity Operations using Tschauner-Hempel Equations," *AIAA Guidance, Navigation and Control Conference*, Portland, Oregon, USA, 2011.

[12] Deaconu, G., Louembet, C., and Théron, A., "Designing Continuously Constrained Spacecraft Relative Trajectories for Proximity Operations," *Journal of Guidance, Control, and Dynamics*, Vol. 38, No. 7, 2015, pp. 1208–1217.

[13] Yamanaka, K., and Ankersen, F., "New State Transition Matrix for Relative Motion on an Arbitrary Elliptical Orbit," *Journal of Guidance, Control, and Dynamics*, Vol. 25, 2002, pp. 60–66.

[14] Nesterov, Y., "Squared Functionals Systems and Optimization Problems," *High Performance Optimization*, edited by H. Frenck, K. Roos, T. Terlaky, and S. Zhang, Springer US, Boston, MA, USA, 2000, Chap. 17, pp. 405–440.

[15] Alfriend, K., Vadali, S., Gurfil, P., How, J., and Breger, L., *Spacecraft Formation Flying: Dynamics, Control and Navigation*, Astrodynamics Series, Elsevier, Burlington, MA, USA, 2010.

[16] Lawden, D., *Optimal trajectories for space navigation*, Butterworth, London, England, 1963.

[17] Tschauner, J., "Elliptic orbit rendezvous," *AIAA Journal*, Vol. 5, No. 6, 1967, pp. 1110–1113.

[18] Clohessy, W., and Wiltshire, R., "Terminal Guidance System for Satellite RendezVous," *Journal of the Astronautical Sciences*, Vol. 27, No. 9, 1960, pp. 653–658.

[19] Bertsimas, D., and Tsitsiklis, J., *Introduction to Linear Optimization*, 1st ed., Athena Scientific, 1997.

[20] Hartley, E., and Maciejowski, J., "Predictive control for spacecraft rendezvous in an elliptical orbit using an FPGA. In Control Conference," *Proceedings of the European Control Conference (ECC)*, Zürich, Switzerland, 2013, p. 1359–1364.

[21] Hartley, E., and Maciejowski, J., "Field programmable gate array based predictive control system for spacecraft rendezvous in elliptical orbits," *Optimal Control Applications and Methods*, Vol. Vol. 36, No. No. 5, 2015. doi:10.1002/oca.2117.

[22] Camps, F., Arantes Gilz, P. R., Joldes, M., and Louembet, C., "Embedding a Semidefinite Programming-Based Control Algorithm for the Impulsive Spacecraft Orbital Rendezvous Hovering Phases," *IEEE 25th International Conference on Integrated Navigation Systems*, IEEE, 2018.

[23] Berges, J.-C., Cayeux, P., Gaudel-Vacaresse, A., and Meyssignac, B., "CNES approaching guidance experiment on FFIORD," *20th. International Symposium on Space Flight Dynamics*, Annapolis, Maryland, USA, 2007.