Abstract. In this paper, the optimal fuel impulsive time-fixed rendezvous problem is considered. Under some simplifying assumptions, this problem may be recast as a non convex polynomial optimization problem. A numerical solving algorithm using a convex relaxation based on sum-of-squares representation of positive polynomials is proposed. Numerical results are evaluated on two examples and checked with a more classical numerical approach based on homotopy continuation for solving polynomial equation systems.

1. INTRODUCTION

Space rendezvous has been considered as a key technology since the first space missions. Nowadays, an increasing importance is given to autonomy issues of this orbital operation, mainly because of the growing number of supply missions to International Space Station (ISS), and the different formation flying missions (PRISMA).

Space rendezvous is an orbital transfer from and to known positions, over a fixed or a free duration. The rendezvous problem consists of finding the guidance scheme that leads to achieve the maneuver, using the minimum fuel cost. Under the impulsive thrust assumption, rendezvous problem can be rephrased into an optimal control problem over $N$ impulses. Each impulse needs to be determined in direction and modulus in addition to its time of application.

Two main solving approaches can be identified. A first methods class is known as "direct methods", where linear and nonlinear programming is used to determine directly the solutions. In this paper, our interest will be focused on the "indirect methods". When using this class of methods, the optimal conditions are derived using variation calculus through Pontryagin principle. Obtained optimality conditions are then used to find solutions of the optimal control problem.

Lawden in [4] derives the optimal conditions for a bounded continuous thrust minimum-fuel problem. Defining the limiting process, the results have been extended to the impulsive case. Several necessary optimality conditions on the relative speed costate, called primer vector, are obtained. Exploiting these results, Neustadt in [7] gives the upper bound of the necessary impulse number for optimal rendezvous. Prussing in [5] develops a geometric method to compute the primer vector trajectory. This method can be used in the coplanar rendezvous under circular assumptions. Getting degenerated solutions cannot be avoided when using Prussing approach. In this case the optimal times cannot be computed. Lion and Handelsman in [8] describe an iterative method which consists of adding new impulses to improve the
cost, using primer vector theory. This method implies tedious computations, in addition to giving many classes of solutions, where the choice is usually ambiguous.

The necessary and sufficient conditions presented by Carter (CNSC) in [1, 2], have been developed for an impulsive linear rendezvous problem. If relative motion is described using transition matrix, the CNSC set several constraints on the initial values and the trajectory of the primer vector. However these constraints cannot be used directly to compute the primer vector evolution, especially when the impulse times are optimized.

In this paper, our aim is to develop a solving method based on polynomial optimization, issued from Carter’s NSC. In the first part, we recall the optimal necessary and sufficient conditions as presented by Carter. The rendezvous problem will be rewritten as a polynomial optimization problem. We will present some basics about convex relaxation, which will be used to construct the rendezvous algorithm. A second approach using real algebraic geometry, through the homotopy continuation will be presented. Both methods will be compared through simulations, where two different rendezvous scenrii are studied.

2. TIME-FIXED OPTIMAL RENDEZVOUS PROBLEM AND POLYNOMIAL OPTIMIZATION

2.1. Carter’s necessary and sufficient conditions for optimal rendezvous

Carter in [2], assuming fixed-time and boundedness conditions on relative position and velocity, considers the following optimization problem:

$$\min_{N, \nu_i, \Delta v_i, \beta(\nu_i)} \sum_{i=1}^{N} \alpha(\nu_i) \Delta v_i$$

subject to

$$u_f = \sum_{i=1}^{N} \Phi(\nu_i) B(\nu_i) \Delta v_i \beta(\nu_i) = \sum_{i=1}^{N} R(\nu_i) \Delta v_i \beta(\nu_i)$$

$$\|\beta(\nu_i)\| = 1$$

$$\Delta v_i \geq 0$$

(1)

where

- $\Phi(\nu, \nu_1) = \phi^{-1}(\nu) \phi^{-1}(\nu_1)$ is the transition matrix of relative motion;
- $B(\nu)$ is the input matrix of relative motion dynamical model;
- $u_f = \phi^{-1}(\nu_f) X_f - \phi^{-1}(\nu_1) X_1 \neq 0$ includes the boundary conditions;
- $\alpha(\nu_i)$ is a weighting function.

Optimization decision variables are:

- $N$: number of velocity impulsive increments;
- $\nu_i$, $\forall i = 1, \cdots, N$: impulses application times;
- $\Delta v_i$: impulse modulus at $\nu_i$;
The primer vector is the costate vector of relative velocity. Its evolution matrix – issued from Pontryagin principle application– is given by \( R(\nu) = \phi^{\#}(\nu)B(\nu) = \phi^{-1}(\nu)B(\nu) \).

**Theorem 1** *(Carter’s NS conditions)*: \((\nu_1, \ldots, \nu_N, \Delta v_1, \ldots, \Delta v_N, \beta(\nu_1), \ldots, \beta(\nu_N))\) is the optimal solution of the problem (1) if and only if there exists a non-zero vector \( \lambda \in \mathbb{R}^m \), \( m = \dim(\phi) \) that verify all of the following conditions:

**NSC 1**: \( \Delta v_i = 0 \) or \( \beta(\nu_i) = -R'(\nu_i)\lambda, \forall i = 1, \ldots, N = 2 + r \)

**NSC 2**: \( \Delta v_i = 0 \) or \( \lambda'R(\nu_i)R(\nu_i)'\lambda = 1 \forall i = 1, \ldots, N \)

**NSC 3**: \( \Delta v_{ki} = 0 \) or \( \nu_{ki} = \nu_1 \) or \( \nu_{ki} = \nu_f \) or \( \lambda \frac{dR(\nu_{ki})}{d\nu}R(\nu_{ki})'\lambda = 0 \), \( \forall i = 2, \ldots, N - 1 \)

**NSC 4**: \( \sum_{i=1}^{N} [R(\nu_i)R'(\nu_i)] \lambda \Delta v_i = -u_f \)

**NSC 5**: \( \Delta v_i \geq 0, \forall i = 1, \ldots, N \)

**NSC 6**: \( \sum_{i=1}^{N} \Delta v_i = -u_f'\lambda > 0 \)

**NSC 7**: \( -u_f'\lambda \) is the minimum of the set defined as: \( \{ \lambda \in \mathbb{R}^n : \text{NSC1} - \text{NSC6 are verified} \} \)

**NSC 8**: \( \| \lambda_v(\nu) \| \leq 1 \) \( \forall \nu \in [\nu_0, \nu_f], \lambda_v(\nu) = R'(\nu)\lambda \)

Numerical solution of optimality conditions NSC1 to NSC8 in the unknowns \( \lambda \in \mathbb{R}^m, N, \{\nu_i\}_{i=1, \ldots, N}, \{\beta_i\}_{i=1, \ldots, N}, \{\Delta v_i\}_{i=1, \ldots, N} \) is a hard problem due to the non convex and transcendental nature of these conditions. A simpler version of problem (1) can be written as:

\[
\min_{\nu_i, \Delta v_i, \beta(\nu_i)} \sum_{i=1}^{N} \Delta v_i \\
\text{subject to} \\
u_f = \sum_{i=1}^{N} \phi^{\#}(\nu_i)B(\nu_i)\Delta v_i\beta(\nu_i) = \sum_{i=1}^{N} R(\nu_i)\Delta v_i\beta(\nu_i) \\
\| \beta(\nu_i) \| = 1 \\
\Delta v_i \geq 0
\]

where the number of impulses \( N \) is *a priori* fixed and chosen to be equal to the upper bound on the optimal number of impulses, according to Neustadt [7] (2 for out of plan rendezvous, 4 for coplanar, 6 for the complete rendezvous problem). We consider also that the first and last impulses are respectively applied at the beginning and at the end of the rendezvous. Furthermore, verifying NSC 8, implies to impose the positivity of the quadratic polynomial \( 1 - \lambda'R(\nu)R(\nu)\lambda \) during the whole rendezvous interval \([t_0, t_f]\). This is clearly a very hard requirement that will be dropped in the sequel. We propose instead to build possible candidates for the primer vector trajectory that will be checked *a posteriori*.
2.2. Relaxed fixed-scenario optimal rendezvous

Despite the previous simplifying assumptions, problem (2) is still a complex one, when impulse dates are part of the vector of decision variables. This complexity may be tackled via the numerical approximation consisting in gridding the time interval defined for the rendezvous. As far as the coplanar rendezvous problem is considered for near-circular orbits, the interior impulse dates should be chosen symmetrically with respect to the half of the duration of the rendezvous interval [6]. The time grid is therefore completely defined by a couple of interior dates \( \Theta = \{(\nu_{21}, \nu_{31}), \cdots, ((\nu_{2d}, \nu_{3d}))\} \) where \( d \) is the number of points of the grid. If \( d \) is sufficiently large, one may hope to get a close approximation of the genuine time-fixed optimal rendezvous. For each point (couple \( (\nu_{2i}, \nu_{3i}) \)) of the grid, a relaxed fixed-scenario of the optimal rendezvous problem (2) has to be solved.

For an \( a \ priori \) fixed-scenario, it is important to note that CNS 3 is not valid anymore and should be relaxed for every couple of interior impulse dates. Therefore, for each couple \( (\nu_{2i}, \nu_{3i}) \) of the grid, a residual will be associated to each candidate solution \( \lambda_j \).

\[
\text{Res}_{kj} = \lambda_j \frac{dR(\nu_{kj})}{d\nu} R(\nu_{k_j})' \lambda_j, \quad \forall \ k_j = 2, \cdots, (N - 1)
\]

Indeed, the best among the obtained \( d \) suboptimal solutions to (2) fulfilling the condition \(|\text{Res}_{kj}| < \epsilon\) should be a candidate for the optimal scenario.

For each point of the grid, we therefore get the following polynomial non convex optimization problem to solve:

\[
\begin{align*}
\min_{\lambda, \Delta \nu_i} & \quad -u_f' \lambda \\
\text{subject to} & \quad -u_f = \sum_{i=1}^{N} R(\nu_i) R'(\nu_i) \Delta \nu_i \lambda \\
& \quad \lambda' R(\nu_i) R'(\nu_i) \lambda = 1, \quad \forall \ i = 1, \cdots N \\
& \quad \Delta \nu_i \geq 0
\end{align*}
\]

In the next section, we give some basics about a hierarchy of convex relaxations for non convex polynomial optimization problem (4) borrowed from [9].

3. CONVEX RELAXATIONS FOR POLYNOMIAL OPTIMIZATION

The problem (4) is a polynomial non convex optimization problem with respect to the variables \( \lambda \in \mathbb{R}^m \) and \( \Delta \nu_i, \ i = 1, \ldots, N \) which can be written under the general form:

\[
P : \quad g^* = \min_{\lambda, \Delta \nu_i} \quad g_0(x) \\
\text{subject to} \quad g_i(x) \geq 0, \ i = 1, \ldots, l
\]

where \( g_i(x) \in \mathbb{R}[x_1, \cdots, x_n] : \mathbb{R}^n \rightarrow \mathbb{R} \). This class of optimization problems is known to be NP-hard. The quadratic non convex problems (0-1 problems) are particular cases of (5). The feasible set of (5) is noted:

\[
\mathbb{K} = \{ x \in \mathbb{R}^n \mid g_i \geq 0, \ i = 1, \ldots, m \}
\]

Computing the global optimum of \( P \) is done by finding \( g^* \), where \( g_0(x) - g^* \geq 0 \) is a globally positive polynomial, on the set \( \mathbb{K} \).
The convex cone of semi definite positive polynomials (SDP) in \( \mathbb{R}^D \) having degree \( \leq d \) is defined by:
\[
\mathcal{P}^d_n = \{ p \in \mathbb{R}[x_1, \ldots, x_n] \mid p(x) \geq 0 \quad \forall x \in \mathbb{R}^n \} \quad D = \left( \begin{array}{c} n + d \\ d \end{array} \right)
\] (7)

The sum of squares polynomials (SOS) in \( \mathbb{R}^D \) convex cone associated to the previous set is given by:
\[
\mathcal{S}^d_n = \left\{ p \in \mathbb{R}[x_1, \ldots, x_n] \mid p(x) = \sum_{i=1}^r q_i(x)^2 \right\}
\] (8)

Each element of this set can be characterized by a linear matrix inequality (LMI) formulation:
\[
p(x) = \sum_{\alpha} p_{\alpha} x^\alpha \in \mathcal{S}^d_n \iff \exists X : p(x) = z'Xz \quad X \succeq 0
\] (9)

where \( z \) is the monomials of degrees \( \leq d \) array. For a feasible matrix \( X \), Cholesky factorization gives:
\[
X = Q'Q = \begin{bmatrix} q_1 & \cdots & q_r \end{bmatrix}
\] (10)

and
\[
p(x) = z'Q'Qz = \|Qz\|_2^2 = \sum_{i=1}^r (q_i z)^2
\] (11)

where the number of squared terms is \( r = \text{rang}(X) \). By identifying the coefficients of \( p(x) = z'Xz = \sum_\alpha p_{\alpha} x^\alpha \geq 0 \), we obtain the following LMIs:
\[
\text{trace } H_\alpha X = p_\alpha \quad \forall \alpha \\
X \succeq 0
\] (12)

where \( H_\alpha \) is a Hankel matrix.

Proving if a polynomial belong to \( \mathcal{S}^d_n \) is equivalent to solve a convex problem of convex programming. Furthermore \( \mathcal{S}^d_n \subset \mathcal{P}^d_n \): we can define a lower bound to the polynomial optimization problem (5). Indeed, the last element is written as:
\[
\exists q_i(x) \in \mathcal{S}^d_n : p(x) = g_0(x) - g^* = g_0(x) + \sum_{i=1}^m g_i(x)q_i(x) \Rightarrow g_0(x) - g^* \geq 0 \quad \forall x \in \mathbb{K}
\] (13)

while the first suggests that finding the multipliers \( q_i(x) \in \mathcal{S}^d_n \) in (13) for a fixed \( \text{deg}(q_i(x)) \) is a semi-definite programming problem. For \( \text{deg}(p(x)) = 2k \), the \( k^{th} \) order convex LMI relation stating that \( p(x) = g_0(x) - g^* \in \mathcal{S}^d_n, \forall x \in \mathbb{K} \) is given by:
\[
p^*_k = \max_X \sum_\alpha \text{trace } A_0 X + \sum_i \text{trace } A_0^{g_i} X_i \\
\text{s.t. } \text{trace } A_\alpha X + \sum_i \text{trace } A_\alpha^{g_i} X_i = (g_0)_\alpha \quad \forall \alpha \neq 0
\] (14)

Note that the semi-definite positive programming dual problem is given by:
\[
d^*_k = \min_y \sum_\alpha (g_0)_\alpha y_\alpha \\
\text{s.t. } M_k(y) = \sum_\alpha A_\alpha y_\alpha \succeq 0 \\
M_{k-d_i}(g,y) = \sum_\alpha A_\alpha^{g_i} y_\alpha \succeq 0 \quad \forall i
\] (15)
where \( y_0 = 1 \), \( d_i = \text{deg}(g_i(x))/2 \) and \( M_k(y) \) is the moments matrix. \( M_{k-d_i}(g_iy) \) are called localization matrices. The SDP optimization problem (15) is the SDP relaxation of the moment problem [10] defined by:

\[
\begin{align*}
p^* &= \min_{\mu} \int g_0 d\mu \\
\text{s.t.} \quad &\mu(\mathbb{K}) = 1 \\
&\mu(\mathbb{K}^c) = 0
\end{align*}
\]  

(16)

where the minimum must be computed over the set of probabilities measures defined on the feasible set \( \mathbb{K} \). The cost function of (16) is a finite linear combination of the measure moments of probabilities \( \mu \) [9]:

\[
y_\alpha = \int x_1^{\alpha_1} \cdots x_n^{\alpha_n} d\mu
\]  

(17)

Under some assumptions, it was shown in [10] that it is possible to construct a hierarchy of monotone convex relaxations, that asymptotically converges to the global optimum of the problem (5).

**Theorem 2 ([10])** If \( \mathcal{P} \) is compact and if there exists \( u(x) \in \mathbb{R}[x_1, \cdots, x_n] \) such that:

\[
\begin{align*}
1 - \{u(x) \geq 0\} &\quad \text{is compact} \\
2 - u(x) = u_0(x) + \sum_{i=1}^{m} g_i(x) u_i(x) &\quad \forall \ x \in \mathbb{R}^n
\end{align*}
\]

where \( u_i(x) \in \mathcal{S}_n^l \), \( i = 0, \cdots, m \),

\[
p^*_k = d^*_k \leq g^*
\]  

(18)

with an asymptotical convergence guarantee:

\[
\lim_{k \to \infty} p^*_k = g^*
\]  

(19)

The convergence is generally fast. \( p^*_k \) is very close to \( g^* \) for a low relaxation order \( k \).

**4. RENDEZVOUS ALGORITHM USING POLYNOMIAL OPTIMIZATION**

**4.1. Polynomial-based algorithm**

In this section, the complete rendezvous algorithm is presented. This algorithm is based on the polynomial optimization as detailed in the previous section. We consider a coplanar rendezvous, using four \( (N = 4) \) impulses. Two of the impulses are a priori fixed at the beginning and at the end of the operation. Algorithm input arguments are:

- \( \nu_1 \) : first impulse time ;
- \( \nu_f \) : last impulse time ;
- \( X_1 \) : initial state (position/velocity);
- \( X_f \) : final state ;
- \( d \) : interior time grid parameter (number of points);
- \( n\_relax\_init \) : initial relaxation order for polynomial optimization problem ;
- \( \epsilon \) : error tolerance (used for testing the primer vector norm).

Algorithm outputs are the impulses (modulus, direction) and their times of application.
Polynomial optimization algorithm steps

1. Define a grid of interior dates \( \Theta = \{ (\nu_2^1, \nu_3^1), \ldots, (\nu_{2d}^1, \nu_{3d}^1) \} \)
2. Solve the \( n \)th order SDP relaxation for every grid point \((\nu_2^j, \nu_3^j), \ j = 1, \ldots, d: \)

\[
\begin{align*}
\min_{\lambda, \Delta v_j} & \quad -u_f'\lambda \\
\text{subject to} & \quad -u_f = \sum_{k_j=1}^{N_j} R(\nu_{k_j}) R'(\nu_{k_j}) \Delta v_{k_j} \lambda \\
& \quad \lambda' R(\nu_{k_j}) R(\nu_{k_j})' \lambda = 1, \ \forall \ k_j = 1_j, \ldots, N_j \\
& \quad \Delta v_{k_j} \geq 0 
\end{align*}
\]

\( n \) is initially equal to \( n_{\text{relax}_\text{init}} \)
3. If a result is obtained using \( n^{th} \) order go to step 4, otherwise \( n \leftarrow n + 1 \) and repeat step 2 ;
4. Keep the solutions verifying the residual condition \( \forall \lambda, \ \lambda = 1, \ldots, d : \)

\[
\text{Res}_{k_j} = \lambda_j \frac{d R(\nu_{k_j})}{d\nu} R(\nu_{k_j})' \lambda_j < \epsilon_1, \ \forall k_j = 2_j, \ldots, (N - 1)_j 
\]

5. Choose the best solution w.r. to the fuel consumption ;
6. Start a direct search method, with the initial conditions \((\lambda, (\Delta v_i)_{i=1,
\ldots,N}, (\nu_i)_{i=1,
\ldots,N}) \)
7. Check the primer vector with \((\lambda^*, (\Delta v_i^*)_{i=1,
\ldots,N}, (\nu_i^*)_{i=1,
\ldots,N}) \)

\[
\| R'(\nu) \lambda \|_2 - 1 \leq \epsilon_2 
\]
8. If the previous test is positive, compute the directions and amplitude of the impulses:

\[
\beta^*(\nu_i^*) = -R(\nu_i^*)' \lambda^* \ \Delta V^*(\nu_i^*) = \beta^*(\nu_i^*) \Delta v_i^* 
\]

Some additional comments are given for more clarity of each step of the algorithm:

- Polynomial optimization is achieved under MATLAB \textsuperscript{®}, using the free academical software Gloptipoly developed at LAAS [9]. This software requires the use of YALMIP [11] and SeDuMi [12].
- The 6\textsuperscript{th} step is realized by propagating the primer vector initial condition \( \lambda^* \) over the rendezvous time duration, using its transition matrix.
- Direct search (5\textsuperscript{th} step) used in this paper is done using FMINCON function of optimization toolbox under MATLAB \textsuperscript{®}.
- Relaxation order is generally less than 4, in order to keep a tractable computational burden.
4.2. An alternative solution via homotopy continuation

A more usual approach to polynomial optimization relies on algebraic tools for finding all the real solutions λ of the polynomial equations introduced by Carter as optimality conditions. Different methods, ranging from analytical methods to Gröbner basis formal approaches [17], [16]. In this paper, we will focus on homotopy continuation methods in order to check the results obtained with the polynomial optimization approach. The main advantage of this type of methods is that it is possible to keep tangency interior constraints in the solution process avoiding the need for computing any residual. However, including these hard geometrical constraints implies that the grid Θ must be more precise. This might, of course, increase the computational burden.

The main principles of the homotopy continuation methods are briefly recalled. Starting from a trivial equation system, whose solutions are known, or easy to determine, a homotopy scheme is used to continuously deform the system to approximate the original one and find its solutions. The major ingredients of the algorithm are:

- An equations system, noted \( f(x) = 0 \), and a trivial equations system \( g(x) = 0 \), which has the same number of isolated solutions than \( f(x) \). The number of solutions determination is done using root count techniques [13].

- A homotopy scheme:

\[
H(x, t) = \gamma t g(x) + (1 - t) f(x) = 0 \quad \gamma \in \mathbb{C}, \ t \in [0, 1] \tag{24}
\]

where \( \gamma \) is randomly chosen parameter.

- A prediction/correction method to compute the linking paths between trivial system roots, and original system solutions.

\[
\Delta x_{k+1} = x_k + \lambda \Delta x_k \tag{25}
\]

\[
\Delta x_{k+1} = H(x_k, t_k) \ x_{k+1} = x_k + \lambda \Delta x_k \tag{26}
\]

A classical prediction/correction scheme is known as the so-called Euler-Newton scheme. It is given by:

- Euler prediction:

\[
\Delta x_k = - \left[ \frac{\partial H(x_k, t_k)}{\partial x} \right]^{-1} \frac{\partial H(x_k, t_k)}{\partial t} \Delta t \quad x_{k+1} = x_k + \lambda \Delta x_k
\]

- Newton correction:

\[
\Delta x_k = - \left[ \frac{\partial H(x_k, t_k)}{\partial x} \right]^{-1} H(x_k, t_k) \quad x_{k+1} = x_k + \lambda \Delta x_k
\]
In this paper, the homotopy continuation computations are achieved using the free available software PHCpack, developed by Jan Verschelde [13]. The rendezvous algorithm using homotopy is summarized with the following steps:

**Homotopy continuation algorithm steps**

1. Define a grid of interior dates $\Theta = \{(\nu_{21}, \nu_{31}), \cdots, (\nu_{d2}, \nu_{3d})\}$
2. Find the real roots of polynomial 6-equations system for each $(\nu_{2j}, \nu_{3j})$, $j = 1, \cdots, d$:

   \[
   \lambda R(\nu_{kj}) R'(\nu_{kj}) \lambda = 1, \ \forall \ k_j = 1, \cdots, N \ \\
   \lambda \frac{dR(\nu_{kj})}{d\nu} R(\nu_{kj})' \lambda = 0, \ \forall \ k_j = 2, \cdots, (N - 1)j
   \] (27)

3. Compute amplitudes array:

   \[
   \Delta v_{sj} = -\left[ R(\nu_{1j}) R'(\nu_{1j}) \lambda_{rj} \cdots R(\nu_{Nj}) R'(\nu_{Nj}) \lambda_{sj} \right]^{-1} u_f \geq 0
   \] (28)

4. If $\Delta v_{sj} \geq 0$ then start a direct search with $(\lambda_{sj}, \Delta v_{sj}, (\nu_{1j}, \cdots, \nu_{Nj}))$.
5. Propagate the primer vector over the rendezvous time duration $(\lambda_{dj}, \Delta v_{dj}, (\nu_{1j}^*, \cdots, \nu_{Nj}^*))$.

   \[
   \lambda_{dj} = R'(\nu) \lambda_{dj}, \ \forall \ \nu \in [t_0, t_f]
   \] (29)

6. For a given $\epsilon$, keep the solutions verifying:

   \[
   \|\lambda_{dj}(\nu)\| < \epsilon, \ \forall \ \nu \in [t_0, t_f]
   \] (30)

7. Choose the best cost solution issued from the previous step:

   \[
   (\lambda^*, \{\Delta v_i^*\}_{i=1}, \cdots, \{\nu_i^*\}_{i=1}, \cdots, N)
   \] (31)

8. Compute the directions and amplitude of the impulses:

   \[
   \beta^*(\nu_i^*) = -R(\nu_i^*)' \lambda^* \ \Delta V(\nu_i^*) = \beta^*(\nu_i^*) \Delta v_i^*, \ i = 1, \cdots, N
   \] (32)

### 5. SIMULATIONS

This section is devoted to the illustration of the presented algorithms on two different examples. An academical example, borrowed from Carter in [3] is first studied. A more realistic scenario based on PRISMA mission [14] is also analyzed.
5.1. Example 1

5.1.1 Definition of the rendezvous problem

The simulation case presented in [3] is a coplanar rendezvous under circular assumptions. The rendezvous maneuver must be completed in one orbital period and requires 4 impulses.

- Time constraints:
  \[ \nu_0 = 0 \quad \nu_f = 2\pi \]

- Rendezvous initial and final conditions:
  \[
  X_1 = \begin{bmatrix} x_1 & x'_1 & z_1 & z'_1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}
  \]
  \[
  X_f = \begin{bmatrix} x_f & x'_f & z_f & z'_f \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0.427 \end{bmatrix}
  \]

  \[
  \phi(\nu) = \begin{bmatrix}
  -2 \cos \nu & -2 \sin \nu & -3\nu & 1 \\
  2 \sin \nu & -2 \cos \nu & -3 & 0 \\
  \sin \nu & -\cos \nu & -2 & 0 \\
  \cos \nu & \sin \nu & 0 & 0
  \end{bmatrix}
  \]
  \[
  \phi^{-1}(\nu) = \begin{bmatrix}
  0 & 2 \sin \nu & -3 \sin \nu & \cos \nu \\
  0 & -2 \cos \nu & 3 \cos \nu & \sin \nu \\
  0 & 1 & -2 & 0 \\
  1 & 3\nu & -6\nu & 2
  \end{bmatrix}
  \]

5.1.2 Results and comparisons

Obtained results with both algorithms are identical. The main difference concerns the time grid which has to be more precise when using homotopy since all the constraints are considered at the same time.

Here, we present 4 different results:

1. solution "1" is obtained using the two presented algorithms (optimized times, and impulses).
2. solution "2" is determined by inverting and minimizing the variance of the linear system defining the TPBVP:
   \[
   u_f = \begin{bmatrix} R(\nu_1) & \cdots & R(\nu_N) \end{bmatrix}
   \begin{bmatrix}
   \Delta V(\nu_1) \\
   \vdots \\
   \Delta V(\nu_N)
   \end{bmatrix} = A_N \Delta V
   \]
   and choosing a random vector \( \zeta \in \mathbb{R}^{2N} \);
   \[
   \Delta V = A_N^T u_f + (1_{2N} - A_N^T A_N) \zeta
   \]
   The impulse dates are identical to the ones obtained from solution 1.
3. solution "3" is the one presented by Carter in [3] \( \nu_2 = \pi/2, \nu_3 = \pi/3, \text{ a priori fixed by Carter} \).
4. solution "4" is computed using Carter’s results, for different fixed times: \( \nu_2 = \pi/4, \nu_3 = 7\pi/4 \).

The results are summarized in the table 1.
<table>
<thead>
<tr>
<th>Solution</th>
<th>Fuel cost</th>
<th>$\nu_2$</th>
<th>$\nu_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Solution 1</td>
<td>$J_1^* = 0.2670$</td>
<td>$\nu_2^* = 1.7001$</td>
<td>$\nu_3^* = 4.58308$</td>
</tr>
<tr>
<td>Solution 2</td>
<td>$J_1 = 5.4110$</td>
<td>$\nu_2 = 1.7001$</td>
<td>$\nu_3 = 4.58308$</td>
</tr>
<tr>
<td>Solution 3</td>
<td>$J_1 = 0.2686$</td>
<td>$\nu_2 = \pi/2$</td>
<td>$\nu_3 = 3\pi/2$</td>
</tr>
<tr>
<td>Solution 4</td>
<td>$J_1 = 0.3274$</td>
<td>$\nu_2 = \pi/4$</td>
<td>$\nu_3 = 7\pi/4$</td>
</tr>
</tbody>
</table>

Table 1: Results for example 1

In-plane trajectories are illustrated in figures 2 and 3. The figure 3 gives more details about solutions 1, 3 and 4. Solutions 1 and 3 lead to almost similar trajectories and consumption. This is mainly due to the clever choice of interior impulse dates by Carter, which is so close to the optimal solution. Solution 4 uses an arbitrary set of times showing a 20% increase of fuel cost.

Figure 2: $(x, z)$ In plane trajectories

Figure 3: Detailed trajectories for solutions 1 (blue), 4 (green)

Figure 4 shows the norm of the primer vector. Solutions 1 and 3 plots are very close while solution 4 plot clearly demonstrates its sub-optimality. The analysis of figure 4 allows to observe that Carter’s solution (3) is not optimal, since the primer vector norm is not equal to 1 when impulses 2, 3 are applied.
5.2. Example 2

5.2.1 Definition of the rendezvous problem

PRISMA program is a cooperative effort between SNSB, CNES, German aerospace research centre DLR and the Danish Technical University (DTU). It aims to test in orbit new guidance schemes, especially autonomous orbit control. The mission includes a rendezvous maneuver (formation acquiring). The parameters for this example are issued from [14]:

- Time constraints:
  \[ \nu_0 = 0 \quad \nu_f = 24\pi \text{ (12 orbits)} \]

- Rendezvous initial and final conditions:
  \[
  X_1 = \begin{bmatrix} x_1 & x'_1 & z_1 & z'_1 \end{bmatrix} = \begin{bmatrix} -10000 & 0 & 0 & 0 \end{bmatrix}
  \]
  \[
  X_f = \begin{bmatrix} x_f & x'_f & z_f & z'_f \end{bmatrix} = \begin{bmatrix} -100 & 0 & 0 & 0 \end{bmatrix}
  \]

The orbit is considered to be a quasi-circular one and Hill-Clohessy-Wiltshire transition matrix will be used.

5.2.2 Results and comparisons

Polynomial optimization algorithm has not been able to reach any result while homotopy continuation –using a 400 point time grid– gives the results given in table 3:

<table>
<thead>
<tr>
<th>[\Delta V] (cm/s)</th>
<th>-4.66</th>
<th>-0.05</th>
<th>0.04</th>
<th>4.66</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.02</td>
<td>0.00</td>
<td>0.00</td>
<td>0.02</td>
<td></td>
</tr>
</tbody>
</table>

| Times (rad) | 0 | 0.2173 | 75.1695 | 24\pi |

Table 3: Example 2 results
Indeed, the results show that the optimal configuration is almost a two-impulse maneuver. The interior maneuvers can be neglected due to their low amplitude. This is probably the main reason why polynomial optimization did not get a result. Indeed, the method is dedicated to consider only a 4-impulses maneuver while homotopy approach may more easily deal with degenerate solutions. Figure 5 shows the trajectory during the rendezvous. One can easily deduce that the interior impulses are so close to the initial and final ones, that they can be considered as coinciding.

![Figure 5: (x, z) In plan trajectory for PRISMA example](image)

The optimal primer vector initial value is

$$\lambda^* = \begin{bmatrix} 0.0134 & -0.0001 & 0.9999 & -0.0088 \end{bmatrix}$$

(35)

The norm of the primer vector is represented at figure 6. It confirms that the optimal solution is a 2-impulse scenario.
6. CONCLUSIONS

Two different numerical algorithms have been proposed to tackle the problem of time-fixed optimal rendezvous. First, a promising algorithm relying on convex relaxation of polynomial optimization problems gives interesting results on an academic example in the circular orbit context. It has been compared to a more classical algorithm based on homotopy continuation methods that is able to deal with more complex rendezvous. These preliminary results need to be reinforced by considering more complex evolution models taking into account ellipticity of the orbit as well as possible orbital perturbations (atmospheric drag for LEO missions and effects of the $J_2$ term for instance).

REFERENCES


