Impulsive, high-fidelity station keeping using semi-infinite programming: application to the geosynchronous case

Romain Serra[1], Eishi Kim[1], and Andrea Fiorentino[1]

[1] *Exotrail*
Toulouse, France
romain.serra@exotrail.com, eishi.kim@exotrail.com, andrea.fiorentino@exotrail.com

Abstract

This paper introduces a convex formulation to the problem of impulsive station keeping via a semi-infinite program with an infinite number of constraints and a finite number of optimization variables. The latter is achieved by fixing a grid of possible burn dates and linearizing the effects of the velocity discontinuities, which is a weaker assumption than a linearization of the differential equations. Furthermore, by relying on automatic differentiation, the proposed implementation is able to emulate all the usual orbital perturbations, making for rather high-fidelity guidance trajectories. The semi-infinite problem itself is solved iteratively by considering a sequence of linear programs, whose dimension remains tractable. The overall approach is illustrated on geosynchronous station-keeping scenarios and several perspectives are proposed for its extension.

I INTRODUCTION

In the context of a satellite mission, station keeping, also known as orbit maintenance, consists in actively remaining close to a reference trajectory, considered as nominal for its operation. It is required as many external forces can make a space object drift from its planned path, which is often calculated with simplified or idealized models, and maneuver plans must be regularly updated as new orbit determinations come in. This paper focuses on impulsive control, that is instantaneous changes of the velocity vectors, a common idealization for burns performed with high-thrust propulsion. Although nowadays the latter tends to be replaced by low-thrust, electric engines that consume less propellant, it is still widely utilized and thus remains relevant when simulating the global orbital environment. Moreover, impulsive maneuvers can in many cases be a good approximation or guess for their finite counterpart, making their fast and accurate computation a valuable tool in mission analysis and design.

Generally speaking, station keeping falls into the domain of optimal control [13]. The performance index usually represents fuel consumption, for its minimization maximizes the life expectancy of the satellite. With impulsive control, it reads as the sum of some norm of the velocity jumps, depending on the thrusters' configuration [11]. Station keeping conditions themselves translate the proximity to the nominal orbit and have multiple possible formulations depending on the choice of coordinates, typically in the form of inequalities. They ideally apply continuously along the whole trajectory and thus belong to the category of path constraints [2]. For local solving techniques, often the latter are too complex to be approached via so-called indirect methods, such as the ones based on the Pontryagin Maximum Principle, and are instead tackled directly, that is by discretization of both the variables and the time horizon before solving for optimality conditions [13]. This calls for a trade-off between dimension of the problem and relaxation of the original constraints. Moreover, in general, this finite set of inequalities does not have any particular properties, meaning that the problem falls into the generic realm of non-linear programming, with no guarantee of convergence or even of global optimality.

Historical approaches to station keeping, when calculations were very limited on board and had to be performed on the ground with not so powerful computers, often put aside numerical optimization altogether and used instead heuristics [3, 12], by nature simplified and lacking generality. More recently, low-thrust propulsion has offered a new paradigm for space trajectory design. As a consequence, high-thrust maneuvers have been less researched, especially for geostationary orbits, despite the fact that even the so-called impulsive model can still provide fast and insightful results. This paper introduces a tracktable approach to impulsive station keeping via convex Semi-Infinite Programming [10], that to the best of the authors' knowledge has not been previously explored in the literature. The dynamics is very generic as it does not make any particular assumption on the orbital perturbations considered, as long as curvilinear coordinates at any epoch can be linearized with respect to previously performed velocity jumps. The reference for linearization is important because unlike some approaches dealing with several perturbations [7, 5], it is not the equations of motion themselves that are approximated, but their solution around the non-maintained trajectory, thanks to automatic differentiation. In practise, with the SIP approach and the choice of the 1-norm as delta-V cost, the actual optimization is achieved by solving a sequence of linear programs, whose dimensions remain manageable. Moreover, the convex formulation comes with convergence and global optimality properties. Following the present introduction, Section II describes the methods involved. Section III explains the practical implementation, leveraging on open-source software such as the astrodynamics library Orekit, and showcases geostationary applications, before the conclusion.

II OPTIMIZATION

A Formulation with curvilinear coordinates

Let $[t_0, t_f]$ be the time interval on which Station Keeping (SK) maneuvers are to be computed. Let N be the fixed number of possible, instantaneous burns, at dates $t_0 \leq t_1 < t_2 \cdots < t_N \leq t_f$ and with jumps $\Delta\mathbf{v}_1, \ldots, \Delta\mathbf{v}_N \in \mathbb{R}^3$ written in some local frame, attached to the satellite. Considering six thrusters all mounted orthogonally to each other in this frame (rather than a single, gimbaled one), the total Delta-V, chosen as the cost function, is $\sum_i \|\Delta\mathbf{v}_i\|_1$ (the sum of their 1-norm rather than of their 2-norm) [11]. Although this does not necessarily represent truthfully every SK scenario, it still offers the perspective of a relatively low $\sum_i \|\Delta\mathbf{v}_i\|_p$, for any p as these norms are all equivalent from a topological point of view. The mathematical benefit of the choice $p = 1$ shall be made clearer later on. To express the orbital proximity with respect to (w.r.t.) the reference, let us use the so-called curvilinear coordinates $(\delta r, \delta\theta, \delta\phi)$ [4], which are interpretable as differences in spherical coordinates defined around the nominal orbital plane. Because the first two, in-plane components are strongly coupled from a dynamical point of view, the radial inequality can be dropped for near-circular references, although nothing dictates it from a mathematical point of view. In this case, the orbit maintenance conditions can be written as:

$$|\delta\theta(t)| \leq \bar{\Delta}_\theta,\ |\delta\phi(t)| \leq \bar{\Delta}_\phi \quad \forall t \in [t_0, t_f], \tag{1}$$

where $\bar{\Delta}_\theta, \bar{\Delta}_\phi > 0$ are the SK bounds. It is worth noticing that in Geostationnary Earth Orbit (GEO), the nominal orbit is a purely Keplerian, idealized one which is basically equatorial and circular, meaning that the aforementioned bounds are the dimension of the so-called dead-band box in longitude and latitude respectively [7]. That, and the good behaviour with linearized motion, make the curvilinear coordinates a very interesting choice of coordinates. Let us now write a tentative formulation of the SK problem:

$$\begin{aligned}
&\min_{t_i,\Delta\mathbf{v}_i} \sum_{i=1}^{N} \|\Delta\mathbf{v}_i\|_1 \\
\text{s.t.}\quad &\forall t \in [t_0, t_f], \\
&|\delta\theta(t)| \leq \bar{\Delta}_\theta, \\
&|\delta\phi(t)| \leq \bar{\Delta}_\phi,
\end{aligned} \tag{2}$$

where $\delta\theta$ and $\delta\phi$ depend on the optimization variables through a coordinate transformation on the result of the integrated equations of motion (that can be different for the reference and actual trajectories, in particular they can be idealized for the former). Note that one can consider additional constraints, such as a bound $\bar{\Delta}V_i$ of each component of a velocity jump i.e. $\|\Delta\mathbf{v}_i\|_\infty \leq \bar{\Delta}V_i$ for all $i = 1, \ldots, N$, or even a terminal constraint, for example imposing a specific angular position at t_f. Let us now rework Problem (2) to make it more tracktable.

B Towards convexity

Let us start by fixing the values of $t_1, \ldots, t_N$, for example uniformly on $[t_0, t_f]$. If N is large enough, it seems reasonable that this simplification does not introduce much loss. If some operational constraints, such as working hours, weight on the possible dates of burn, they can be incorporated indirectly in the design of this grid. As the next step, let us perform a linearization of the constraints. For that purpose, let us simply use the gradients $\nabla_i\theta(t) = \frac{\partial\delta\theta(t)}{\partial\Delta\mathbf{v}_i}$ and $\nabla_i\phi(t) = \frac{\partial\delta\phi(t)}{\partial\Delta\mathbf{v}_i}$ evaluated at $\mathbf{0}_3 = (0,0,0)$ for all $i = 1, \ldots, N$. In other words, the effects of the past velocity jumps on the curvilinear coordinates are approximated at first-order around the uncontrolled case of no discontinuities. Note that it does not require linearizing the equations of motion and more details will be given on the actual computation in Section III. Let us denote with a hat the curvilinear coordinates obtained when $\|\Delta\mathbf{v}_1\|_1 = \cdots = \|\Delta\mathbf{v}_N\|_1 = 0$. The new SK problem writes as follows:

$$\begin{aligned}
&\min_{\Delta\mathbf{v}_i} \sum_{i=1}^{N} \|\Delta\mathbf{v}_i\|_1 \\
\text{s.t.}\quad &\forall t \in [t_0, t_f], \\
&\left|\delta\hat{\theta}(t) + \sum_{i=1}^{N} \nabla_i\theta(t)(\mathbf{0}_3)\Delta\mathbf{v}_i\right| \leq \bar{\Delta}_\theta, \\
&\left|\delta\hat{\phi}(t) + \sum_{i=1}^{N} \nabla_i\phi(t)(\mathbf{0}_3)\Delta\mathbf{v}_i\right| \leq \bar{\Delta}_\phi.
\end{aligned} \tag{3}$$

Problem (3) is a so-called Semi-Infinite Program (SIP) as it has a discretely-defined cost function but continuously-defined (on an interval) constraints. Note that the latter can be rewritten as a single, scalar inequality as $\|c(t, \Delta\mathbf{v}_1, \ldots, \Delta\mathbf{v}_N)\|_\infty \leq 1$ for all $t \in [t_0, t_f]$, where:

$$\begin{aligned}
c = \Bigg(&\frac{1}{\bar{\Delta}\theta}\left\{\delta\hat{\theta}(t) + \sum_{i=1}^{N} \nabla_i\theta(t)(\mathbf{0}_3)\Delta\mathbf{v}_i\right\}, \\
&\frac{1}{\bar{\Delta}\phi}\left\{\delta\hat{\phi}(t) + \sum_{i=1}^{N} \nabla_i\phi(t)(\mathbf{0}_3)\Delta\mathbf{v}_i\right\}\Bigg).
\end{aligned} \tag{4}$$

Problem (3) is also convex, by virtue of the triangle inequalities satisfied by $\|.\|_\infty$ and $\|.\|_1$. The fact is that such a convex SIP can be solved at arbitrary precision via an iterative process described in [10] and reported in Algorithm 1. It has been already used in space trajectory optimization [1, 6], but apparently only in indirect optimal control i.e. involving the so-called adjoint variables. The theory of SIP in general seems little known or used in this domain.

In short, the principle of Algorithm 1 is that by working with a dynamical, finite time grid Ω where to impose the path constraints, it is possible to get arbitrarily close to the optimal solution. The rules defining the set update can vary: here, it is a simple addition of the time where the constraints are the least respected. In practice, one uses a thin grid to compute the argmax. As for the minimization problems to be solved in each iteration, they can be cast as Linear Programs (LPs) after the introduction of so-called slack variables: $\Delta\mathbf{v}_i^\pm \geq 0$ (component-wise inequality) such that $\Delta\mathbf{v}_i = \Delta\mathbf{v}_i^+ - \Delta\mathbf{v}_i^-$. Each minimization problem

Algorithm 1 Iterative solving process of convex SIP

```
Require: $\varepsilon > 0$
Require: $j_{\max} \in \mathbb{N}$
  $j \leftarrow 0$
  $\Omega \leftarrow \{t_0, t_f\}$
  $\Delta\mathbf{v}_1^*, \ldots, \Delta\mathbf{v}_N^* \leftarrow \mathrm{argmin}_{\Delta\mathbf{v}_i} \sum_i \|\Delta\mathbf{v}_i\|_1$
  s.t. $c(t, \Delta\mathbf{v}_1, \ldots, \Delta\mathbf{v}_N) \leq 1 \; \forall t \in \Omega$
  $\tau \leftarrow \mathrm{argmax}_{t\in[t_0,t_f]} c(t, \Delta\mathbf{v}_1^*, \ldots, \Delta\mathbf{v}_N^*)$
  while $c(\tau) \geq 1 + \varepsilon$ and $j \leq j_{\max}$ do
      $j \leftarrow j + 1$
      $\Omega \leftarrow \Omega \cup \{\tau\}$
      $\Delta\mathbf{v}_1^*, \ldots, \Delta\mathbf{v}_N^* \leftarrow \mathrm{argmin}_{\Delta\mathbf{v}_i} \sum_i \|\Delta\mathbf{v}_i\|_1$
  s.t. $c(t, \Delta\mathbf{v}_1, \ldots, \Delta\mathbf{v}_N) \leq 1 \; \forall t \in \Omega$
      $\tau \leftarrow \mathrm{argmax}_{t\in[t_0,t_f]} c(t, \Delta\mathbf{v}_1^*, \ldots, \Delta\mathbf{v}_N^*)$
  end while
```

in Algorithm 1 is then equivalent to:

$$
\begin{aligned}
\min_{\Delta\mathbf{v}_i^+, \Delta\mathbf{v}_i^-} \sum_{i=1}^{N} \; & \big(\Delta\mathbf{v}_{i,x}^+ + \Delta\mathbf{v}_{i,x}^- + \Delta\mathbf{v}_{i,y}^+ + \\
& \Delta\mathbf{v}_{i,y}^- + \Delta\mathbf{v}_{i,z}^+ + \Delta\mathbf{v}_{i,z}^-\big) \\
\text{s.t.} & \Delta\mathbf{v}_i^\pm \geq 0, \\
& \forall t \in \Omega, \\
& \delta\hat{\theta}(t) + \Sigma\nabla_i\theta(t)(\mathbf{0}_3)(\Delta\mathbf{v}_i^+ - \Delta\mathbf{v}_i^-) \in [-\bar{\Delta}_\theta, \bar{\Delta}_\theta], \\
& \delta\hat{\phi}(t) + \Sigma\nabla_i\phi(t)(\mathbf{0}_3)(\Delta\mathbf{v}_i^+ - \Delta\mathbf{v}_i^-) \in [-\bar{\Delta}_\phi, \bar{\Delta}_\phi].
\end{aligned}
\tag{5}
$$

One of the practical advantages of the dynamical grid is that is expected to remain sparse, meaning that the LPs are relatively small in size and thus efficiently solved via classical algorithms e.g. the simplex. Note that using the 2-norm instead of the 1-norm would lead to a cone program inplace of a linear one, via the introduction of different slack variables, but that case is left out of the scope of this paper.

C Extension without modelling gap

So far, the linearization of the constraints has been done around vanishing impulses. Depending on $t_f - t_0$, $\bar{\Delta}_\theta$, $\bar{\Delta}_\phi$ as well as the initial conditions of both actual and reference trajectories, the corresponding approximation can eventually become poor. The discrepancy is expected to lower by performing a sequence (indexed by $k \in \mathbb{N}$) of linearizations, around the previously found values for the velocity jumps $\Delta\mathbf{v}_i^{(k)}$, each time solving a SIP, until the differences $\mathbf{y}_i^{(k)} = \Delta\mathbf{v}_i^{(k+1)} - \Delta\mathbf{v}_i^{(k)}$ are deemed small enough. More precisely, the problems considered are (before the introduction of slack variables):

$$
\begin{aligned}
& \min_{\mathbf{y}_i^{(k)}} \sum_{i=1}^{N} \|\mathbf{y}_i^{(k)}\|_1 \\
\text{s.t.} \quad & \forall t \in [t_0, t_f], \\
& \left| \delta\hat{\theta}(t) + \sum_{i=1}^{N} \nabla_i\theta(t)(\Delta\mathbf{v}_i^{(k)})\mathbf{y}_i^{(k)} \right| \leq \bar{\Delta}_\theta, \\
& \left| \delta\hat{\phi}(t) + \sum_{i=1}^{N} \nabla_i\phi(t)(\Delta\mathbf{v}_i^{(k)})\mathbf{y}_i^{(k)} \right| \leq \bar{\Delta}_\phi.
\end{aligned}
\tag{6}
$$

Note that, as in general $\|\Delta\mathbf{v}_i^{(k+1)}\|_1 \leq \|\Delta\mathbf{v}_i^{(k)}\|_1 + \|\Delta\mathbf{v}_i^{(k+1)} - \Delta\mathbf{v}_i^{(k)}\|_1$, the cost function in Problem (6) is sub-optimal regarding the total Delta-V, but it is deemed satisfying as only small corrections are expected at this stage. It is worth mentioning that between the first and the second linearization, the grid for possible maneuvers can be pruned by only keeping the non-zero impulses obtained, thus making for smaller dimension in the subsequent LPs. As a matter of fact, the actual velocity increments taken as input for the first occurrence of Problem (6) could originate from any initial guess, not just the output of Algorithm 1.

III STUDY CASES

A Implementation

As is apparent from Section II, the timely computation of some first-order partial derivatives w.r.t. the control is paramount for the realization of the approach. For a given impulse $\Delta\mathbf{v}_i$ at time t_i, the vector function to be differentiated is the composition of two sub-functions: first a propagation block that maps the spacecraft position-velocity vector $\mathbf{x} = (\mathbf{p}, \mathbf{v})$ from initial conditions $(\mathbf{p}_0, \mathbf{v}_0)$ at t_0 to an epoch t and then a transformation module computing from $\mathbf{p}(t)$ the angular part $(\delta\theta, \delta\phi)(t)$ of the curvilinear coordinates w.r.t. the reference. The latter is a rather simple coordinate conversion and its Jacobian matrix can be obtained straightforwardly by hand. As for the former, its is identically null when $t < t_i$ as the maneuver has not happened yet. When $t \geq t_i$, assuming that there is no non-gravitational forces or simply neglecting the decreasing mass in the dynamics, it comes from the chain rule of derivation that:

$$\frac{\partial \mathbf{x}(t)}{\partial \Delta \mathbf{v}_i} = \frac{\partial \mathbf{x}(t)}{\partial \mathbf{x}(t_i)} . \frac{\partial \mathbf{x}}{\partial \mathbf{v}}(\mathbf{x}(t_i)) \tag{7}$$

On the right-hand side of Eq. (7), only the term $\frac{\partial \mathbf{x}(t)}{\partial \mathbf{x}(t_i)}$ is not trivial to evaluate. In order to calculate it, let us do the following manipulation (still based on the chain rule):

$$\begin{aligned} \frac{\partial \mathbf{x}(t)}{\partial \mathbf{x}(t_i)} &= \frac{\partial \mathbf{x}(t)}{\partial \mathbf{x}_0} . \frac{\partial \mathbf{x}_0}{\partial \mathbf{x}(t_i)} \\ &= \frac{\partial \mathbf{x}(t)}{\partial \mathbf{x}_0} . \left(\frac{\partial \mathbf{x}(t_i)}{\partial \mathbf{x}_0} \right)^{-1} \end{aligned} \tag{8}$$

Terms in the form of $\frac{\partial \mathbf{x}(.)}{\partial \mathbf{x}_0}$ are so-called state transition matrices associated to the propagation initialized at t_0. Here, they are computed via the combination of Automatic Differentiation (AD) and numerical integration with dense output, as implemented within the propagation routines of Orekit 12.0 [8], which also features a wide range of orbital perturbations. Although this open-source library offers via its dependency Hipparchus an LP solver, it was preferred here to use for that purpose a faster one, from Google's OR-tools suite [9].

As mentioned in Section II, there is no need to linearize the equations of motion themselves, as is often done (see for instance [7]). Although it would de facto lead to a linear propagation map easy to differentiate, it would also come with a dynamical model discrepancy that is correlated on the choice of dependant variables and that can potentially grow large, especially as it starts diverging right from t_0. Linearizing w.r.t. the impulses does not suffer these caveats and as such introduce fewer intrinsic errors. Note that the extension based on (6) uses repeated linearizations (hence propagations) to lift the modelling gap completely.

B Applications

In GEO, the in-plane disturbance essentially comes from the tesseral part of the geopotential, dominated by the C_{22} harmonics coefficient and motivating so-called East-West (EW) maneuvers. On the other hand, the main out-of-plane perturbations are due to the Sun and Moon gravity, and possibly the solar radiation pressure depending on the area-to-mass ratio. They create the need for so-called North-South (NS) SK. Note that in an operational context, maneuvers are recomputed regularly as orbit determination results come (within a timescale of a few days or so), while for mission design or situational awareness purposes it is useful to predict them over extended periods of time (months or years).

B.1 Example without solar radiation pressure

For this scenario, the assigned longitude is 50 deg East in the Greenwich True Of Date (GTOD) frame, with a dead-band of 0.1 deg. It is assumed that the satellite starts exactly at the center. SK is sought over 15 days, from 2021, March 3rd at 00:00:00.000 Coordinated Universal Time (UTC). The possible burns are spread uniformly every 12 hours, while the size of the dense grid for path constraint checks has 12 points per day. The orbital perturbations taken into account are purely gravitational in nature, namely a 6x6 geopotential and lunisolar effects. Propagation is fully numerical and performed with a Runge-Kutta 4 integration scheme, with a step of 864 seconds, using as dependent variables Cartesian coordinates in the Geocentric Celestial Reference Frame (GCRF).

A single LPs is enough to reach convergence with $\varepsilon = 0.1\%$ of the dead-band. The angular trajectory is depicted on Figure 1. The longitude and latitude over time are shown on Figure 2. Table 1 gives the following information about the maneuvers: elapsed time since t_0 and components in the Radial Transverse Normal (RTN) frame. There is a single, tangential burn: it is performed right at the start and its effects are such that the satellite is about to exit the box (by violating the upper bound on longitude) at the final date. It is also worth comparing the size of the LP solved here to a brute-force approach that would incorporate instead of just 2 dates the whole dense grid made of $15 \times 12 = 180$ points. As its size increases or the number of possible maneuvers does, the brute-force method becomes impractical well before Algorithm 1 does.

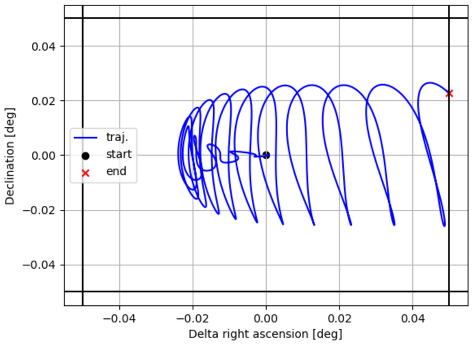

Figure 1: Angular trajectory within the constraint box for Example 1

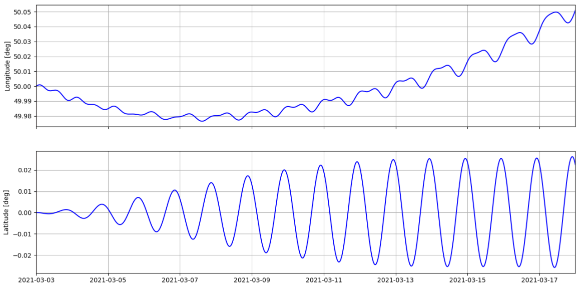

Figure 2: Longitude and latitude over time for Example 1

Time (days)	R (m/s)	T (m/s)	N (m/s)
0.0	0.000	0.130	0.000

Table 1: Burns' details for Example 1

B.2 Example with solar radiation pressure

This scenario is identical to the previous one, except for the consideration of Solar Radiation Pressure (SRP) with the satellite assumed to be a sphere, with an area-to-mass ratio of 0.04 m^2/kg and a reflectivity coefficient of 1.4.

Algorithm 1 requires 2 LPs and gives the impulse reported in Table 2, very similar to Example 1, yet with a slight improvement of almost 1 cm/s, showing how the optimization is able to leverage on the perturbations. The trajectory is depicted in Figure 3 and varies somewhat from the previous case, still touching the box, but not at final time.

Time (days)	R (m/s)	T (m/s)	N (m/s)
0.0	0.000	0.122	0.000

Table 2: Burns' details for Example 2

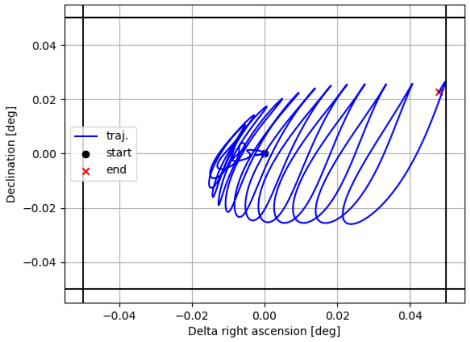

Figure 3: Angular trajectory within the constraint box for Example 2

B.3 Example over a month

This scenario is identical to the Example 1, except for the control duration which is now 30 days.

Algorithm 1 requires 11 LPs to converge, but the solution violates both the longitude and latitude constraints due to the approximation from linearizing around zero-normed impulses. So the extension proposed in Section C is used here instead and gives the control reported in Table 3 after the second linearization. The results only differ on the first Delta-V, by less than 1 mm/s. The overall strategy is a mix of EW and NS burns, costing respectively 0.190 and 0.148 m/s. Note that since the last two maneuvers are separated by only half a day and are in opposite, out-of-plane directions, they basically have the same effect on the orbit and could be merged into a single one.

Time (days)	R (m/s)	T (m/s)	N (m/s)
0.0	0.000	0.107	0.000
5.0	0.000	0.016	0.000
8.5	0.000	0.067	0.000
29.0	0.000	0.000	0.054
29.5	0.000	0.000	-0.094

Table 3: Burns' details for Example 3

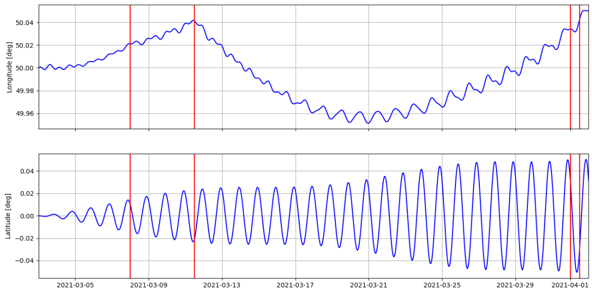

Figure 4: Longitude and latitude over time (red marks are burns' dates) for Example 3

B.4 Extrapolation over a year

As illustrated by the previous example, a single linearization can provide accurate estimation of fuel consumption, even if its solution does not satisfy the constraints anymore after a while. In this example, it is used to predict the Delta-V budget for 365 days. Initial conditions are taken from [7], witht the satellite at the center of a 0.1 or 0.01 deg dead-band box at longitude 60 deg East on January 1st, 2010. The object is assumed to have a mass of 4500 kg with a reflective coefficient of 1.3 and a cross section for SRP of 300 m^2. Orbital perturbations are the same as for Example 2, except for the geopotential which is now 3x3. Possible dates for maneuvers are evenly spread every 8 hours and the dense grid for constraint checks has 12 points per day. Algorithm 1 leads to extended computation time as points are added one by one and many LPs need to be solved. It has thus been adapted for this particular case, following lines from [6]. The modified version removes dates where constraints are respected with some marging whilst adding at once the ones where they are violated, decreasing drastically the number of LPs.

The optimal cost is reported in Table 4, along with values from [7]. The latter have been obtained for finite-thrust control, with a magnitude of 0.1 N. Moreover, the optimization is based on a quadratic cost function rather than the total Delta-V and uses a receding horizon approach, with the equations of motion being linearized several times. The consumptions only differ by a few meters per second, with coherent individual budget for both NS and EW station keeping.

Delta-V (m/s)	Solution	Reference [7]
EW	7.12 - 12.66	10.99 - 13.95
NS	49.53 - 57.64	58.40 - 63.42
Total	56.65 - 70.29	69.39 - 77.40

Table 4: Delta-V budget for Example 4 with deadband of 0.1 deg (left) versus 0.01 deg (right)

IV CONCLUSIONS

This paper has presented a novel approach to impulsive station keeping, carefully choosing the formulation of the mathematical constraints to benefit from convexity properties via semi-infinite programming and to limit model approximations. In practise, it requires two main capabilities: linearizing the propagation map w.r.t. impulses, which is done here with the open-source library Orekit, and solving linear optimization problems. The method has been applied to several scenarios of station keeping in geostationary orbit with high-fidelity propagation models.

Future work includes the addition of radial constraints, enabling direct control of altitude, preferable for station keeping in Low Earth Orbit. An interesting perspective would also be to adapt the formulation to non-impulsive burns, in order to consider low-thrust propulsion.

V References

[1] Denis Arzelier, Florent Bréhard, Norbert Deak, Mioara Joldes, Christophe Louembet, Aude Rondepierre, and Romain Serra. Linearized impulsive fixed-time fuel-optimal space rendezvous: A new numerical approach. *IFAC-PapersOnLine*, 49(17):373–378, 2016.

[2] John T Betts. Survey of numerical methods for trajectory optimization. *Journal of guidance, control, and dynamics*, 21(2):193–207, 1998.

[3] Jean-Pierre Carrou. *Spaceflight dynamics*. Cepadues-Editions, 1995.

[4] Ferdi deBruijn, Eberhard Gill, and Jonathan How. Comparative analysis of cartesian and curvilinear clohessy-wiltshire equations. *Journal of Aerospace Engineering*, 3(2):1, 2011.

[5] Clément Gazzino, Denis Arzelier, Damiana Losa, Christophe Louembet, Christelle Pittet, and Luca Cerri. Optimal control for minimum-fuel geostationary station keeping of satellites equipped with electric propulsion. *IFAC-PapersOnLine*, 49(17):379–384, 2016.

[6] Adam W Koenig and Simone D'Amico. Fast algorithm for fuel-optimal impulsive control of linear systems with time-varying cost. *IEEE Transactions on Automatic Control*, 66(9):4029–4042, 2020.

[7] Damiana Losa. High vs low thrust station keeping maneuver planning for geostationary satellite, 2007.

[8] maisonobe, Evan Ward, Maxime Journot, Bryan Cazabonne, Vincent Cucchietti, Romain Serra, Guilhem Bonnefille, Thomas Neidhart, Sébastien Dinot, Clément Jonglez, yjeand, Julio Hernanz, lirw1984, Andrew Goetz, Guiuux, gaetanpierre0, Lars Næsbye Christensen, Alberto Fossà, jvalet, gabb5, Alberto Ferrero, liscju, plan3d, Amir Allahvirdi-Zadeh, alugan, gprat31, Kevin222004, Luc Maisonobe, Mark Rutten, and Petrus Hyvönen. Cs-si/orekit: 12.0, November 2023.

[9] Laurent Perron and Vincent Furnon. Or-tools 9.6, 2023.

[10] Rembert Reemtsen and Jan-J Rückmann. *Semi-infinite programming*, volume 25. Springer Science & Business Media, 1998.

[11] I Michael Ross. Space trajectory optimization and L1-optimal control problems,". *Modern astrodynamics*, 1:155, 2006.

[12] Erik Mattias Soop. *Handbook of geostationary orbits*, volume 3. Springer Science & Business Media, 1994.

[13] Emmanuel Trélat. Optimal control and applications to aerospace: some results and challenges. *Journal of Optimization Theory and Applications*, 154:713–758, 2012.