MITSUBISHI ELECTRIC RESEARCH LABORATORIES
https://www.merl.com

# Optimization-Based Phase-Constrained x-Axis Crossing Control for Station-Keeping on Libration Point Orbits

Shimane, Yuri; Ho, Koki; Weiss, Avishai

TR2026-004 January 07, 2026

**Abstract**

Libration Point Orbits (LPO) play a central role in lunar exploration of the 21st century. Long-term operations of both crewed and robotic assets on LPOs necessitate propellant-efficient and reliable station-keeping schemes. Recent developments highlight the importance of station-keeping schemes that not only maintain the geometric orbital regime but also avoid leading or lagging along the orbit as a result of a drifting phase. In this work, an optimization-based station-keeping algorithm for colinear LPO based on the x-axis crossing control, an event-based targeting scheme for impulsive maneuver design, is conceived. The optimization problem is cast as a sequential second-order cone program and incorporates an explicit constraint on the perilune pass epoch to ensure the steered trajectory follows the reference baseline without deviating in phase. The resulting formulation has easily interpretable tuning parameters that may be obtained directly from mission requirements. The algorithm is demonstrated through Monte-Carlo simulations on Gateway's Near Rectilinear Halo Orbit (NRHO) in the high-fidelity ephemeris model with realistic error models

*Journal of the Astronautical Sciences 2025*



Mitsubishi Electric Research Laboratories, Inc.
201 Broadway, Cambridge, Massachusetts 02139

# Optimization-Based Phase-Constrained $x$-Axis Crossing Control for Station-Keeping on Libration Point Orbits *

**Yuri Shimane†, Koki Ho‡, and Avishai Weiss§**

## ABSTRACT

Libration Point Orbits (LPO) play a central role in lunar exploration of the $21^{\text{st}}$ century. Long-term operations of both crewed and robotic assets on LPOs necessitate propellant-efficient and reliable station-keeping schemes. Recent developments highlight the importance of station-keeping schemes that not only maintain the geometric orbital regime but also avoid leading or lagging along the orbit as a result of a drifting phase. In this work, an optimization-based station-keeping algorithm for colinear LPO based on the $x$-axis crossing control, an event-based targeting scheme for impulsive maneuver design, is conceived. The optimization problem is cast as a sequential second-order cone program and incorporates an explicit constraint on the perilune pass epoch to ensure the steered trajectory follows the reference baseline without deviating in phase. The resulting formulation has easily interpretable tuning parameters that may be obtained directly from mission requirements. The algorithm is demonstrated through Monte-Carlo simulations on Gateway's Near Rectilinear Halo Orbit (NRHO) in the high-fidelity ephemeris model with realistic error models.

## 1 INTRODUCTION

Libration point orbits (LPOs) will play a central role in humanity's cislunar presence, hosting both crewed and robotic spacecraft for various purposes. For example, Gateway is planned to be hosted on the 9:2 resonant L2 near-rectilinear halo orbit (NRHO). Even though some LPOs, such as the NRHO, exhibit favorable stability properties, station-keeping is still required; in the absence of appropriate correction maneuvers, the spacecraft may diverge into nearby LPO regimes, or in worst cases, diverge and crash on the Moon or escape from the lunar vicinity.

The performance of station-keeping controllers may be evaluated with regard to two considerations. First and foremost, the controller must be able to maintain the spacecraft in the vicinity of the intended baseline trajectory, rejecting any noise or uncertainty that arises from the dynamics, state estimation error, and control execution error. The proximity to the baseline is defined not only in terms of physical vicinity from the intended LPO (for example, the

*A preliminary version of this work has been presented as manuscript 24-499 at the AS/AIAA Astrodynamics Specialist Conference, Broomfield, CO, in Aug. 2024.

†PhD Candidate, School of Aerospace Engineering, Georgia Institute of Technology, GA 30332, USA

‡Associate Professor, School of Aerospace Engineering, Georgia Institute of Technology, GA 30332, USA

§Senior Principal Research Scientist, Mitsubishi Electric Research Laboratories (MERL), Cambridge, MA 02139, USA

9:2 resonant NRHO), but also in terms of its temporal location along the LPO, referred to as its phase;[1,2] maintaining the correct phase is crucial for applications such as Gateway, where the baseline location of the space station is carefully chosen to minimize the impact of eclipse,[1] or for cislunar space domain awareness problems, where observer spacecraft are located on phases that result in favorable illumination conditions.[3–5] The second consideration is the resulting cumulative $\Delta V$ cost incurred with maintaining the orbit over an extended amount of time. We note that minimizing the maneuver cost at each maneuver opportunity is not necessarily equivalent to minimizing the cumulative cost, thus complicating the task of designing an effective station-keeping algorithm.

Over the past few decades, there have been multiple works on the topic of station-keeping on LPOs. Folta and Vaughn,[6] and more recently Shirobokov et al.,[7] provide reviews of the various approaches that have been studied to date. In accordance with the categorization by Shirobokov et al.,[7] the station-keeping problem may be tackled either from a dynamical system theoretic perspective or a control theoretic perspective. The former includes leveraging information of the local flow, such as Floquet mode-based,[8–12] eigenmotion,[13] and Cauchy Green Tensor-based approaches,[2,14] while the latter includes approaches such as targeting control,[15,16] sliding mode,[17] model predictive control (MPC),[18,19] and, more recently, robust control strategies.[20,21]

In the context of impulsive station-keeping on the NRHO in high-fidelity dynamics, the $x$-axis crossing control, belonging to the class of targeting-based approach, has been shown to provide exceptional performance by multiple authors,[1,14,22,23] and has also demonstrated in flight on the recent CAPSTONE mission.[24] In its classical formulation, the $x$-axis crossing control targets a subset of the state, typically the $x$-component velocity in the Earth-Moon rotating frame. The subset of the state components is targeted at the event where the predicted state of the spacecraft crosses the $xz$-plane near the Moon, coinciding approximately with its perilune. The classical $x$-axis crossing control, despite yielding low station-keeping cost, is susceptible to drift in phase due to the way in which the algorithm is designed; as a result, one recent research focus with this approach has been to simultaneously ensure the spacecraft does not drift in phase.[1,2] To date, phase control has been done with a two-stage differential correction process, which involves a tuning weight that must balance the tolerance on the targeted state components (e.g. $x$-component velocity) and time, which is non-intuitive.

The station-keeping problem on LPOs, due to its sparse control opportunities involving one maneuver every few days, may be seen as a recursive application of a finite-horizon trajectory design problem. In the context of trajectory design, one popular approach is sequential convex programming (SCP),[25] adopted in a number of recent astrodynamics applications.[26–32] Within an SCP, a convex subproblem with linearized dynamics about a reference solution is iteratively solved, each time updating the reference solution with the solution to the convex subproblem based on step-acceptance criteria. One similarity between SCP and differential correction for the purpose of constructing trajectories is their use of linearized dynamics to iteratively correct the controlled trajectory. In addition, since the sought optimal solution in station-keeping scenarios typically lies close to the uncontrolled trajectory, only a small number of convex

subproblem iterations are expected to be necessary.

In this work, we propose a novel, optimization-based approach to achieve phase-constrained $x$-axis crossing control. We leverage the sequential linearization-based formulation previously proposed by Elango et al.[33] and extend it to include a phase maintenance constraint. In our approach, the linearization is conducted in terms of both velocity and time increments, thereby allowing the minimization problem to explore both the control maneuver and the time until the targeted $xz$-plane crossing event simultaneously. To ensure the scheme produces propellant-minimizing results, a two-norm objective, resolved as a second-order cone (SOC) constraint, is adopted. With the use of successive solutions to conic subproblems with linearized dynamics, the proposed algorithm may be understood as an SCP-based approach to the $x$-axis crossing control framework. Due to the use of an explicit optimization problem formulation, our approach replaces the non-intuitive weight in Davis et al.[1] with explicit thresholds on targeting the state and the phase, respectively. The proposed approach, named the phase-constrained sequential cone program (PC-SCoP), is demonstrated via Monte-Carlo simulations for a spacecraft on the NRHO with high-fidelity dynamics.

The remainder of this paper is organized as follows: first, in Section 2, the high-fidelity ephemeris model dynamics and LPOs are briefly discussed. This is followed by Section 3, where the $x$-axis crossing control scheme is introduced; first, the conventional, differential correction-based schemes are described. Then, the proposed PC-SCoP, starting from its mathematical formulation and summarizing it through a pseudo-algorithm, is presented. In Section 4, we present the recursive simulation setup, including realistic uncertainty models, adopted to conduct Monte-Carlo experiments with the studied controllers. Section 5 presents experimental results using the PC-SCoP with a few different choices of controller parameters. Finally, a conclusion is provided in Section 6.

## 2 BACKGROUND

We consider the translational motion of a spacecraft in the high-fidelity ephemeris model dynamics in the vicinity of the Moon. In addition to the two-body gravitational acceleration of the Moon, we consider perturbations due to the Moon's J2 term, third-body perturbations from the Earth and the Sun, and solar radiation pressure (SRP). In this section, we present the equations of motion and provide a brief overview of LPOs.

### 2.1 Equations of Motion in the High-Fidelity Ephemeris Model

We consider the motion of the spacecraft in the inertial frame, $\mathcal{F}_{\mathrm{Inr}}$ centered at the Moon. In this work, $\mathcal{F}_{\mathrm{Inr}}$ is set to the J2000 frame defined using the DE440 ephemerides.[34] Let $\boldsymbol{\theta} \in \mathbb{R}^6$ denote the state of the spacecraft in the inertial frame centered at the Moon, composed of its position $\boldsymbol{r} \in \mathbb{R}^3$ in $\mathcal{F}_{\mathrm{Inr}}$ and the derivative of $\boldsymbol{r}$ in $\mathcal{F}_{\mathrm{Inr}}$, denoted by $\boldsymbol{v} \triangleq \dot{\boldsymbol{r}} \in \mathbb{R}^3$. The equations of motion $\boldsymbol{f}[t, \boldsymbol{\theta}(t)]$ is given by

$$\dot{\boldsymbol{\theta}} = \boldsymbol{f}[t, \boldsymbol{\theta}(t)] = \begin{bmatrix} \boldsymbol{v} \\ -\frac{\mu}{r^3}\boldsymbol{r} + \boldsymbol{a}_{\mathrm{J2}} + \boldsymbol{a}_{\mathrm{SRP}}(t) + \sum_i \boldsymbol{a}_{N_i}(t) \end{bmatrix}, \tag{1}$$

where $r = \|\boldsymbol{r}\|_2$, $\mu$ is the gravitational parameter of the central body; $\boldsymbol{a}_{\mathrm{J2}}$ is the J2 acceleration of the central body, given by

$$\boldsymbol{a}_{\mathrm{J2}} = -\frac{3\mu J_2}{2r^3}\left(\frac{R_e}{r}\right)^2 \boldsymbol{T}_{\mathrm{Inr}}^{\mathrm{PA}} \begin{bmatrix} \left(1 - 5\frac{z_{\mathrm{PA}}^2}{r^2}\right) x_{\mathrm{PA}} \\ \left(1 - 5\frac{z_{\mathrm{PA}}^2}{r^2}\right) y_{\mathrm{PA}} \\ \left(3 - 5\frac{z_{\mathrm{PA}}^2}{r^2}\right) z_{\mathrm{PA}} \end{bmatrix}, \tag{2}$$

where $\boldsymbol{T}_{\mathrm{Inr}}^{\mathrm{PA}} \in \mathbb{R}^{3\times 3}$ is the transformation matrix from the Moon's principal axes frame $\mathcal{F}_{\mathrm{PA}}$, aligned with the Moon's principal axes whose orientations are defined with the DE440 ephemerides, to $\mathcal{F}_{\mathrm{Inr}}$, $(\cdot)_{\mathrm{PA}}$ is the position vector component in $\mathcal{F}_{\mathrm{PA}}$, $R_e$ is the equatorial radius, $J_2$ is the J2 coefficient, $\boldsymbol{a}_{\mathrm{SRP}}$ is the SRP acceleration, given by

$$\boldsymbol{a}_{\mathrm{SRP}}(t) = P_\odot \left(\frac{1\mathrm{AU}}{r_\odot}\right)^2 \frac{C_r A}{m}\frac{\boldsymbol{r}_\odot}{r_\odot}, \tag{3}$$

where $P_\odot$ is the radiation pressure at 1AU, $\boldsymbol{r}_\odot$ denotes the position vector of the spacecraft with respect to the Sun, $r_\odot = \|\boldsymbol{r}_\odot\|_2$, $C_r$ is the reflection pressure coefficient, $A$ is the spacecraft's cross-sectional area, and $m$ is its mass, and finally, $\boldsymbol{a}_{N_i}$ is the third-body perturbation due to the $i^{\mathrm{th}}$ body, given by

$$\boldsymbol{a}_{N_i}(t) = -\mu_i \left[\frac{\boldsymbol{d}_i}{d_i^3} + \frac{\boldsymbol{r}_i}{r_i^3}\right], \tag{4}$$

where $\boldsymbol{r}_i$ is the position of the $i^{\mathrm{th}}$ perturbing body with respect to the Moon, $r_i = \|\boldsymbol{r}_i\|_2$, $\boldsymbol{d}_i = \boldsymbol{r} - \boldsymbol{r}_i$ is the position of the spacecraft with respect to the $i^{\mathrm{th}}$ body, and $d_i = \|\boldsymbol{d}_i\|_2$.

The station-keeping scheme will make use of the linearized flow in the vicinity of a nominal trajectory. An initial linear perturbation $\delta\boldsymbol{\theta}(t_0)$ may be mapped to some future time $t > t_0$ via the state-transition matrix (STM) $\Phi(t, t_0) \in \mathbb{R}^{6\times 6}$,

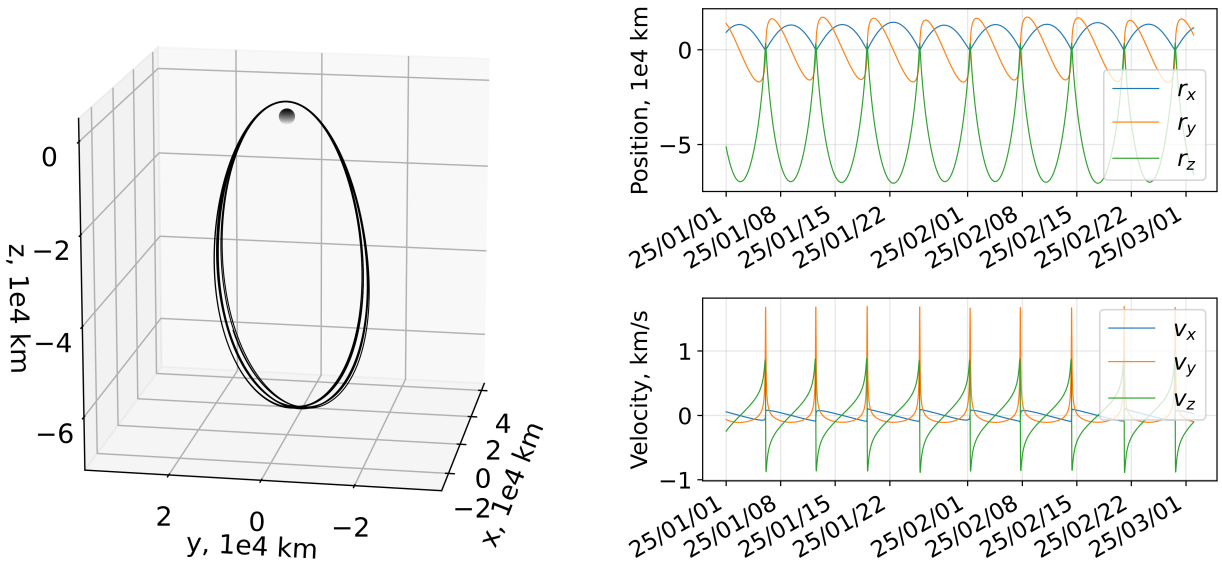
$$\delta\boldsymbol{\theta}(t) = \Phi(t, t_0)\delta\boldsymbol{\theta}(t_0), \tag{5}$$

where $\Phi(t, t_0)$ is obtained by solving the nonlinear initial value problem (IVP)

$$\dot{\boldsymbol{\theta}}(t) = \boldsymbol{f}(t, \boldsymbol{\theta}), \tag{6a}$$

$$\dot{\Phi}(t, t_0) = \frac{\partial \boldsymbol{f}(t, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\Phi(t, t_0), \tag{6b}$$

with initial conditions $\boldsymbol{\theta}(t_0) = \boldsymbol{\theta}_0$ and $\Phi_0 = \boldsymbol{I}_6$.




a Position space

b State history

**Figure 1**: Reference 9:2-resonant NRHO in the Moon-centered Earth-Moon rotating frame

### 2.2 Libration Point Orbits

LPOs, as the name suggests, designate orbit-like motions that revolve around the libration points. While exactly periodic motions may be constructed in simplified models such as the various restricted three-body problems, the motion can only be quasi-periodic in higher-fidelity models such as the one considered in this work. Nevertheless, it is possible to construct a ballistic or nearly ballistic quasi-periodic motion of a significant duration, consisting of multiple revolutions. In this work, we make use of the 15-year 9:2-resonant NRHO baseline from NASA.[35] Figure 1 shows a portion of the baseline NRHO trajectory for 60 days in the Moon-centered Earth-Moon rotating frame, defined by its $x$-axis aligned with the Earth-Moon direction, its $z$-axis aligned with the cross-product between the $x$-axis and the Moon's velocity vector with respect to the Earth, and its $y$-axis completing the triad.

## 3 STATION-KEEPING CONTROL WITH $x$-AXIS CROSSING CONTROL

The basis of this work's control framework is $x$-axis crossing control. In this section, we begin by providing an overview of the differential correction (DC)-based $x$-axis crossing control[1,2,14,22–24]; the original algorithm[14,22–24] does not consider tracking with respect to the phase but is presented for context. We then provide a brief discussion on the phase-constrained $xz$-plane crossing control with DC, as reported in recent works.[1,2] This DC-based approach necessitates tuning a non-physical weight in order to scale residuals of targeted state components and time deviation within a single residual vector. The lack of a physical interpretation for this weight parameter makes the tuning process non-intuitive. Instead, we propose an optimization-based formulation for the phase-constrained $xz$-plane crossing

control, replacing the residual vector with explicit constraints on each residual, which can make use of tolerances derived directly from mission requirements.

### 3.1 Preliminaries on $x$-axis crossing control

Background on $x$-axis crossing control is laid out in terms of three building-blocks: first, we define the targeted event where the state components are to be matched; then, we discuss the choice of state components to be targeted; finally, we discuss the choice of location(s) along the LPO where the maneuver is to be executed.

*3.1.1 Targeted Event* The controller aims to maintain the steered motion of the spacecraft in the vicinity of a pre-computed, ballistic reference path, which will be referred to as the *baseline* hereafter. Consider the Earth-Moon rotating frame $\mathcal{F}_{\mathrm{EM}}$, defined with the first $x$-axis aligned with the vector from the Earth to the Moon, and the $z$-axis aligned with the angular momentum vector of the Earth-Moon system's co-rotation about their barycenter. In the $x$-axis crossing controller, this is achieved by matching certain conditions of the spacecraft's motion along the baseline and the steered path at their respective intersection with the $xz$-plane in $\mathcal{F}_{\mathrm{EM}}$ around perilune. The intersection with the $xz$-plane crossing happens *exactly* at the perilune in simplified dynamics models such as the circular restricted three-body problem; in contrast, in the high-fidelity ephemeris model, the perilune and the intersection do not occur at the same epoch. However, these two conditions may be used interchangeably without noticeable differences in the controller performance. Without loss of generality in describing the classic $x$-axis crossing control, in this work, we use the perilune-based condition as a proxy for the $xz$-plane crossing event.

We note that the challenge of the classical $x$-axis crossing controller with regards to deviation in phase is a consequence of this event-based targeting scheme, which is unusual compared to typical tracking controllers. To track a reference baseline path, a generic tracking controller would attempt to match the predicted path at some chosen future time to the corresponding baseline state; this would for example be to match the expected state along the steered path to the baseline's perilune state at the *baseline perilune time*, as illustrated in case (a) from Figure 2. Instead, the event-based targeting attempts to match the perilune state of the steered path to the baseline's perilune state, even though these occur at different times, as illustrated in case (b) in the same Figure. Through preliminary experiments, we find the application of approach (a) with the $x$-axis crossing controller, by setting $t_{p,\mathrm{ref}}$ to either the baseline or the unsteered path's epoch at perilune, to result in the steered path to diverge within a few controlled revolution.

*3.1.2 Targeted State Components* When considering a station-keeping task with a single maneuver, the problem is limited to at most 3 degrees of freedom corresponding to each $\Delta\boldsymbol{v} \in \mathbb{R}^3$ component. It is thus sensible to limit the number of targeted state components to $m \leq 3$. Let $\boldsymbol{\vartheta} \in \mathbb{R}^m$ denote a subset of state components from $\boldsymbol{\theta} \in \mathbb{R}^6$, transformed from the inertial frame to the Earth-Moon rotating frame,

$$\boldsymbol{\vartheta}(t) = \left(\boldsymbol{T}^{\mathrm{Inr}}_{\mathrm{EM}}(t)\boldsymbol{\theta}(t)\right)_{\mathcal{M}}, \tag{7}$$

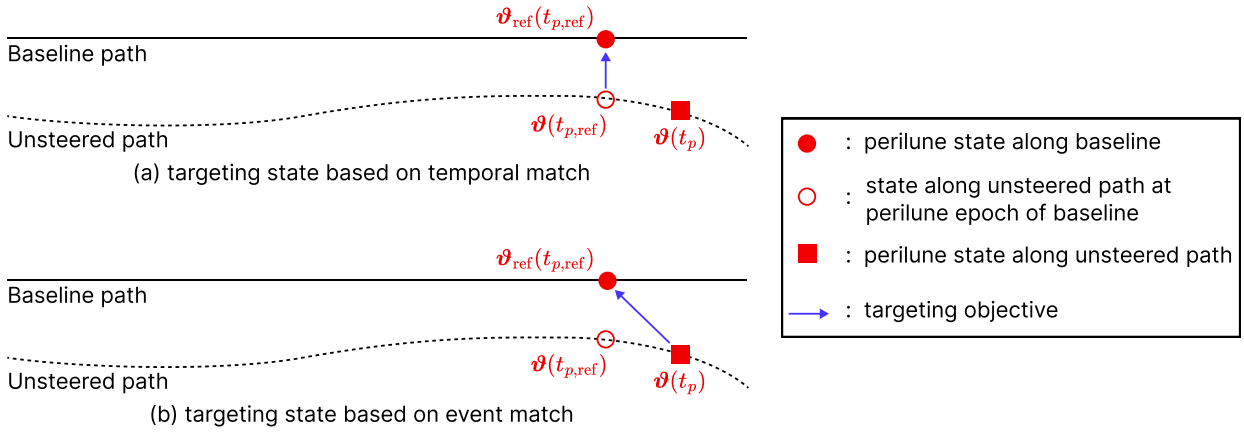



**Figure 2**: Targeting paradigms in station-keeping control for liberation point orbits

where $\boldsymbol{T}_{\text{EM}}^{\text{Inr}}(t) \in \mathbb{R}^{6\times 6}$ is the transformation matrix from $\mathcal{F}_{\text{Inr}}$ to $\mathcal{F}_{\text{EM}}$, and $(\cdot)_{\mathcal{M}}$ denotes the vector constructed from the rows corresponding to the $m$ targeted states. For example, if $\mathcal{M} = \{v_x, v_z\}$, then $\boldsymbol{\vartheta}(t)$ is constructed by selecting the $4^{\text{th}}$ and $6^{\text{th}}$ components of the vector $\boldsymbol{T}_{\text{EM}}^{\text{Inr}}(t)\boldsymbol{\theta}(t)$. By selecting $\mathcal{M}$ to consist purely of velocity components, the $x$-axis crossing control may be effectively understood as a perilune velocity targeting method. Note that the transformation matrix $\boldsymbol{T}_{\text{EM}}^{\text{Inr}}$ is time-dependent, as the Earth-Moon rotating frame is non-inertial.

*3.1.3 Control Location* The location and frequency of maneuvers must be carefully planned as these directly impact both the stability and the cost of station-keeping algorithms. Due to the unstable characteristic of the dynamics along with the existence of uncertainties due to navigation, modeling, and control execution error, it is preferable to execute the control where the dynamics has reduced sensitivity to maneuvers, which corresponds to regions around apolune.[14,36] While the exact location of the maneuver in the vicinity of the apolune may consider other operational aspects,[1] we limit the scope of this work to controls that occur exactly on the osculating apolune, where the radial velocity is null. We also fix the scheme to make use of a single maneuver per revolution, as is typically considered for realistic impulsive station-keeping scenarios on the NRHO.[1,14,22] Along the NRHO, this corresponds to executing a maneuver approximately every $6.5$ days.

### 3.2 Phase-Free Differential Correction

The phase-free DC formulation of $x$-axis crossing control is, in essence, shooting-based trajectory patching. The targeted residual vector $F \in \mathbb{R}^m$ for the DC is given by

$$F = \boldsymbol{\vartheta}_f - \boldsymbol{\vartheta}_{\text{ref}}, \tag{8}$$

where $\boldsymbol{\vartheta}_{\text{ref}}$ is the baseline state at $t_f$, and $\boldsymbol{\vartheta}_f$ is obtained from the IVP

$$\boldsymbol{\vartheta}_f = \left( \boldsymbol{T}_{\text{EM}}^{\text{Inr}} \int_{t_0}^{t_f} \boldsymbol{f}(t, \boldsymbol{\theta}_0) \mathrm{d}t \right)_{\mathcal{M}}, \tag{9}$$

where $\boldsymbol{\theta}_0$ is the state at which the maneuver is computed. Note that due to navigation errors, $\boldsymbol{\theta}_0$ cannot be known precisely, and, in practice, $\boldsymbol{\vartheta}_f$ in equation (9) is computed based on state estimates. The free variables vector $X \in \mathbb{R}^3$ is the initial variation in velocity,

$$X = \begin{bmatrix} \Delta v_{x,0} \\ \Delta v_{y,0} \\ \Delta v_{z,0} \end{bmatrix}. \tag{10}$$

The corresponding Jacobian $DF$ is given by

$$DF = \frac{\partial F}{\partial X} = \frac{\partial \boldsymbol{\vartheta}_f}{\partial \boldsymbol{v}_0}. \tag{11}$$

The DC problem involves iteratively solving for $F = \mathbf{0}_{m\times 1}$, each time updating $X$ in a Newton-Raphson fashion. For $m < 3$, it is common in the literature to adopt the minimum-norm update given by

$$X^{(k+1)} = X^{(k)} - \left(DF^{(k)}\right)^T \left[(DF^{(k)})(DF^{(k)})^T\right]^{-1} F(X^{(k)}). \tag{12}$$

Note that even though DC is not explicitly minimizing the maneuver magnitude, the minimum-norm update gives a new solution $X^{(k+1)}$ that minimizes the norm of $X^{(k+1)} - X^{(k)}$, which translates to the smallest $\Delta \boldsymbol{v}$ update that reduces $\|F\|$ in a quadratic scheme assuming an initial guess $X^{(0)} = \mathbf{0}_{3\times 1}$, i.e. no maneuver. The iteration is stopped when components of $F$ is smaller than corresponding components in a tolerance vector $\boldsymbol{\varepsilon}_{\boldsymbol{\vartheta},\text{targ}} \in \mathbb{R}^m$,

$$\left| \left(\boldsymbol{\vartheta}_f - \boldsymbol{\vartheta}_{\text{ref}}\right)_i \right| \leq \varepsilon_{\boldsymbol{\vartheta}_i,\text{targ}} \quad i = 1, \ldots, m. \tag{13}$$

### 3.3 Phase-Constrained Differential Correction

Davis et al.[1] presents a two-stage DC process that introduces a constraint on the phase. The first-stage DC is the phase-free problem (8)-(11), which provides an initial guess $\Delta \boldsymbol{v}$ to the second stage. The second-stage DC uses an

extended residual function $F \in \mathbb{R}^{m+1}$ given by

$$F = \begin{bmatrix} \boldsymbol{\vartheta}_f - \boldsymbol{\vartheta}_{\text{ref}} \\ W_{t_f} \left( t_f - t_{p,\text{ref}} \right) \end{bmatrix}, \tag{14}$$

where $W_{t_f}$ is a heuristic scalar weight that must be tuned to achieve the tolerance on both the state component(s) within tolerance $\boldsymbol{\varepsilon}_{\boldsymbol{\vartheta},\text{targ}}$ according to (13) and the phase within tolerance $\varepsilon_{t_f,\text{targ}}$ given by

$$|t_f - t_{p,\text{ref}}| \leq \varepsilon_{t_f,\text{targ}}. \tag{15}$$

This algorithm has been used in the recently launched CAPSTONE mission,[24] and is the proposed station-keeping scheme for Gateway.[1]

### 3.4 Phase-Constrained Sequential Cone Program

A potential drawback of the differential correction-based approach is driven by (1) the lack of a formal minimization scheme for the $\Delta V$ cost besides the fact that the Newton-like update is minimum-norm, and (2) the unintuitive nature of the scaling parameter $W_{t_f}$ due to its lack of physical meaning. To overcome these challenges, we propose an alternative approach to achieve phase-constrained $x$-axis crossing control by formulating an explicit nonlinear program (NLP). We begin by providing the general NLP formulation, followed by a description of the sequential linearization process that recasts the NLP to a sequentially solved second-order cone program (SOCP).

*3.4.1 Formulation* The general NLP is given by

$$\min_{\Delta \boldsymbol{v}, t_f} \quad \|\Delta \boldsymbol{v}\|_2, \tag{16a}$$

$$\text{such that} \quad \left| \left( \boldsymbol{\vartheta}_f - \boldsymbol{\vartheta}_{\text{ref}} \right)_i \right| \leq \boldsymbol{\varepsilon}_{\boldsymbol{\vartheta}_i,\text{targ}} \quad i = 1, \ldots, m, \tag{16b}$$

$$|t_f - t_{p,\text{ref}}| \leq \varepsilon_{t_f,\text{targ}}. \tag{16c}$$

Compared to the Newton-Raphson update (12), the formulation in (16) includes separate targeting constraints on the state (16b) and phase (16c), removing the need for a weight like $W_{t_f}$ in the residual function (14) from the DC-based approach. Note that by making the propagation time of the steered state $t_f$ a variable, the formulation in (16) no longer ensures the targeted state $\boldsymbol{\vartheta}_f$ is at perilune; rather, the choice of $\boldsymbol{\vartheta}_f$ should be made in a way that promotes the steered state to occur approximately at perilune. One may explicitly add a constraint that ensures the steered state occurs at a perilune, for example ensuring the final steered state satisfies $\boldsymbol{r}(t_f)^T \boldsymbol{v}(t_f) \approx 0$; however, our experiments found that choosing the targeted state components set as $\mathcal{M} = \{v_x, v_y\}$ is sufficient for station-keeping over multiple years, and is thus adopted in this work.

*3.4.2 Sequential Linearization* The general NLP (16) is solved iteratively, each time linearizing the dynamics about the state estimate. Let $\bar{t}_f^{(k)}$ be the time until the $N^{\text{th}}$ perilune before applying the control maneuver during the $k^{\text{th}}$ iteration. Then, without any control maneuver, the targeted state components during the $k^{\text{th}}$ iteration $\bar{\boldsymbol{\vartheta}}_f^{(k)} \in \mathbb{R}^m$ is given by

$$\bar{\boldsymbol{\vartheta}}_f^{(k)} = \left( \boldsymbol{T}_{\text{EM}}^{\text{Inr}} \int_{t_0}^{\bar{t}_f^{(k)}} f(\boldsymbol{\theta}_0^{(k)}, \tau) \mathrm{d}\tau \right)_{\mathcal{M}}. \tag{17}$$

The expression for the deviation in final state due to $\Delta\boldsymbol{v}$ and $\delta t_f$ is given by

$$\delta\boldsymbol{\theta}_f^{(k)} = \boldsymbol{T}_{\text{EM}}^{\text{Inr}} \left( \begin{bmatrix} \Phi^{\boldsymbol{rv}}(\bar{t}_f^{(k)}, t_0) \\ \Phi^{\boldsymbol{vv}}(\bar{t}_f^{(k)}, t_0) \end{bmatrix} \Delta\boldsymbol{v}^{(k)} + \frac{\partial\boldsymbol{\theta}(\bar{t}_f^{(k)})}{\partial t} \delta t_f^{(k)} \right), \tag{18}$$

where $\Phi^{\boldsymbol{rv}} \in \mathbb{R}^{3\times3}$ and $\Phi^{\boldsymbol{vv}} \in \mathbb{R}^{3\times3}$ denote the top-right and bottom-right submatrices of $\Phi$, respectively. Let $\delta\boldsymbol{\vartheta}_f^{(k)}$ denote the components in $\mathcal{M}$ from the state deviation $\delta\boldsymbol{\theta}_f^{(k)}$. In matrix form, the variation in the final targeted state $\delta\boldsymbol{\vartheta}_f^{(k)}$ is given by

$$\delta\boldsymbol{\vartheta}_f^{(k)} = \left(\delta\boldsymbol{\theta}_f^{(k)}\right)_{\mathcal{M}} = \begin{bmatrix} \boldsymbol{B}_{\mathcal{M}}^{(k)} & \boldsymbol{C}_{\mathcal{M}}^{(k)} \end{bmatrix} \begin{bmatrix} \Delta\boldsymbol{v}^{(k)} \\ \delta t_f^{(k)} \end{bmatrix}, \tag{19}$$

where $\boldsymbol{B}_{\mathcal{M}}^{(k)} \in \mathbb{R}^{m\times3}$ and $\boldsymbol{C}_{\mathcal{M}}^{(k)} \in \mathbb{R}^{m\times1}$ are given by

$$\boldsymbol{B}_{\mathcal{M}}^{(k)} = \left( \boldsymbol{T}_{\text{EM}}^{\text{Inr}} \begin{bmatrix} \Phi^{\boldsymbol{rv}}(\bar{t}_f^{(k)}, t_0) \\ \Phi^{\boldsymbol{vv}}(\bar{t}_f^{(k)}, t_0) \end{bmatrix} \right)_{\mathcal{M}}, \tag{20}$$

$$\boldsymbol{C}_{\mathcal{M}}^{(k)} = \left( \boldsymbol{T}_{\text{EM}}^{\text{Inr}} \frac{\partial\boldsymbol{\theta}(\bar{t}_f^{(k)})}{\partial t} \right)_{\mathcal{M}}. \tag{21}$$

Through the linearization of constraints (16b) and (16c) by making use of the linearized final state $\bar{\boldsymbol{\vartheta}}_f^{(k)}$ and final state deviation $\delta\boldsymbol{\vartheta}_f^{(k)}$, the NLP (16) becomes a SOCP, where the two-norm objective may be replaced with a second-order cone constraint. While the two-norm function $\|\Delta\boldsymbol{v}\|_2$ and the quadratic function $\Delta\boldsymbol{v}^T\Delta\boldsymbol{v}$ have the same minimum, the former has higher sensitivity at small $\Delta\boldsymbol{v}$ magnitudes. In this work, we use the SOCP formulation to avoid issues

with having to fine-tune the scales of the input vectors and matrices to the SOCP; the problem is given by

$$\min_{\Delta \boldsymbol{v}^{(k)}, \delta t_f^{(k)}, \eta} \quad \eta, \tag{22a}$$

$$\text{such that} \quad \|\Delta \boldsymbol{v}^{(k)}\|_2 \leq \eta, \tag{22b}$$

$$\left| \left( \bar{\boldsymbol{\vartheta}}_f^{(k)} + \delta \boldsymbol{\vartheta}_f^{(k)} \right) - \boldsymbol{\vartheta}_{f,\text{ref}} \right|_i \leq \boldsymbol{\varepsilon}_{\boldsymbol{\vartheta}_i}, \quad i = 1, \ldots, m, \tag{22c}$$

$$\left| \left( \bar{t}_f^{(k)} + \delta t_f^{(k)} \right) - t_{p,\text{ref}} \right| \leq \varepsilon_{t_f,\text{targ}}, \tag{22d}$$

where the notation $|\cdot|_i$ in constraint (22c) denotes the absolute value of the $i^{\text{th}}$ component of the vector inside the absolute value operator. Constraints (22c) and (22d) can be written in standard form as

$$-\begin{bmatrix} \boldsymbol{B}_{\mathcal{M}}^{(k)} & \boldsymbol{C}_{\mathcal{M}}^{(k)} \end{bmatrix} \begin{bmatrix} \Delta \boldsymbol{v}^{(k)} \\ \delta t_f^{(k)} \end{bmatrix} \leq \boldsymbol{\varepsilon}_{\boldsymbol{\vartheta},\text{targ}} + \left( \bar{\boldsymbol{\vartheta}}_f^{(k)} - \boldsymbol{\vartheta}_{f,\text{ref}} \right), \tag{23}$$

$$\begin{bmatrix} \boldsymbol{B}_{\mathcal{M}}^{(k)} & \boldsymbol{C}_{\mathcal{M}}^{(k)} \end{bmatrix} \begin{bmatrix} \Delta \boldsymbol{v}^{(k)} \\ \delta t_f^{(k)} \end{bmatrix} \leq \boldsymbol{\varepsilon}_{\boldsymbol{\vartheta},\text{targ}} - \left( \bar{\boldsymbol{\vartheta}}_f^{(k)} - \boldsymbol{\vartheta}_{f,\text{ref}} \right), \tag{24}$$

and

$$-\delta t_f^{(k)} \leq \varepsilon_{t_f,\text{targ}} + \left( t_{f,0} - t_{p,\text{ref}} \right), \tag{25}$$

$$\delta t_f^{(k)} \leq \varepsilon_{t_f,\text{targ}} - \left( t_{f,0} - t_{p,\text{ref}} \right). \tag{26}$$

After solving problem (22), both $\Delta \boldsymbol{v}^{(k)}$ and $\delta t_f^{(k)}$ are used to update $\boldsymbol{\theta}_0$ and $t_{f,0}$ respectively, such that

$$t_{f,0}^{(k+1)} = \bar{t}_f^{(k)} + \Delta t_f^{(k)}, \tag{27a}$$

$$\boldsymbol{\theta}_0^{(k+1)} = \boldsymbol{\theta}_0^{(k)} + \begin{bmatrix} \boldsymbol{0}_{3\times 1} \\ \Delta \boldsymbol{v}^{(k)} \end{bmatrix}, \tag{27b}$$

where the superscript $(\cdot)^{(k)}$ and $(\cdot)^{(k+1)}$ denote the $k^{\text{th}}$ and $(k+1)^{\text{th}}$ iteration solving the linearized problem (22) respectively. Then, matrices $\boldsymbol{B}_{\mathcal{M}}^{(k+1)}$ and $\boldsymbol{C}_{\mathcal{M}}^{(k+1)}$ are recomputed using equations (20) and (21). Due to the sequential nature of linearizing and then forming the SOCP, this method is hereafter referred to as the phase-constrained sequential cone program (PC-SCoP). An example implementation of the PC-SCoP is shown in Algorithm 1. Within the algorithm, a few aliases for functions are being used:

- `PropagateUntilNthPerilune`$(t_0, \boldsymbol{\theta}_0, N)$ propagates the initial state $\boldsymbol{\theta}_0$ at time $t_0$ until the $N^{\text{th}}$ perilune

**Algorithm 1** Phase-constrained sequential second-order cone program

**Inputs**: $t_0$, $\boldsymbol{\theta}_0$, $N$, $\boldsymbol{\vartheta}_{f,\text{ref}}$, $k_{\max}$, $\boldsymbol{\varepsilon}_{\boldsymbol{\vartheta},\text{targ}}$, $\varepsilon_{t_f,\text{targ}}$
**Outputs**: convergence_flag, $\Delta\boldsymbol{v}$

convergence_flag $\leftarrow 0$ ▷ Initialize flag
$\Delta\boldsymbol{v} \leftarrow \mathbf{0}_{3\times 1}$ ▷ Initialize cumulative $\Delta\boldsymbol{v}$
$\Delta\boldsymbol{v}^{(0)}, \delta t_f^{(0)} \leftarrow \mathbf{0}_{3\times 1}, 0$ ▷ Initialize incremental correction
$\bar{t}_f^{(0)} \leftarrow$ `PropagateUntilNthPerilune`$(t_0, \boldsymbol{\theta}_0, N)$
**for** $k$ in $1, \ldots, k_{\max}$ **do**
  $\boldsymbol{\theta}_0^{(k)} \leftarrow \boldsymbol{\theta}_0 + [\mathbf{0}_{3\times 1}; \Delta\boldsymbol{v}^{(k-1)}]$
  $\bar{t}_f^{(k)} \leftarrow \bar{t}_f^{(k-1)} + \delta t_f^{(k-1)}$
  $\bar{\boldsymbol{\theta}}_f^{(k)}, \Phi_{\boldsymbol{rv}}, \Phi_{\boldsymbol{vv}} \leftarrow$ `Propagate`$(t_0, \bar{t}_f^{(k)}, \boldsymbol{\theta}_0^{(k)})$
  $\boldsymbol{T}_{\text{EM}}^{\text{Inr}} \leftarrow$ `sxform`$(\mathcal{F}_{\text{Inr}}, \mathcal{F}_{\text{EM}}, \bar{t}_f^{(0)})$ ▷ SPICE function
  $\bar{\boldsymbol{\vartheta}}_f^{(k)} \leftarrow \boldsymbol{T}_{\text{EM}}^{\text{Inr}} \bar{\boldsymbol{\theta}}_f^{(k)}$
  **if** $\bar{\boldsymbol{\vartheta}}_f^{(k)} \leq \boldsymbol{\varepsilon}_{\boldsymbol{\vartheta},\text{targ}}$ and $\bar{t}_f^{(k)} \leq \varepsilon_{t_f,\text{targ}}$ **then**
    convergence_flag $\leftarrow 1$
    break
  **end if**
  $\boldsymbol{B}_{\mathcal{M}}^{(k)}, \boldsymbol{C}_{\mathcal{M}}^{(k)} \leftarrow$ eqn. (20), (21)
  $\Delta\boldsymbol{v}^{(k)}, \delta t_f^{(k)} \leftarrow$ `SOCP`$(\bar{\boldsymbol{\vartheta}}_f^{(k)}, \boldsymbol{B}_{\mathcal{M}}^{(k)}, \boldsymbol{C}_{\mathcal{M}}^{(k)}, 0.9\boldsymbol{\varepsilon}_{\boldsymbol{\vartheta},\text{targ}}, 0.9\varepsilon_{t_f,\text{targ}},)$
  $\Delta\boldsymbol{v} \leftarrow \Delta\boldsymbol{v} + \Delta\boldsymbol{v}^{(k)}$ ▷ Update cumulative $\Delta\boldsymbol{v}$
**end for**

through the use of event detection;

- `Propagate`$(t_0, t_f, \boldsymbol{\theta}_0)$ propagates the initial state $\boldsymbol{\theta}_0$ along with the STM from time $t_0$ until time $t_f$;
- `sxform`$(\mathcal{F}_{\text{Inr}}, \mathcal{F}_{\text{EM}}, \bar{t}_f^{(0)})$ generates the transformation matrix from $\mathcal{F}_{\text{Inr}}$ to $\mathcal{F}_{\text{EM}}$ at epoch $\bar{t}_f^{(0)}$; and
- `SOCP`$(\bar{\boldsymbol{\vartheta}}_f^{(k)}, \boldsymbol{B}_{\mathcal{M}}^{(k)}, \boldsymbol{C}_{\mathcal{M}}^{(k)}, \boldsymbol{\varepsilon}_{\boldsymbol{\vartheta},\text{targ}}, \varepsilon_{t_f,\text{targ}})$ builds and solves problem (22) through the use of a SOCP solver, such as ECOS[37] or SCS.[38]

Note that the sequential nature of the algorithm is necessitated purely due to the linearization of the dynamics; this is similar in nature to differential correction, where the correction only considers the Jacobian $DF$ from equation (11); thus, in essence, the PC-SCoP approach replaces the Newton-Raphson update from equation (12) by the SOCP. The number of iterations required to either conduct the Newton-Raphson update or solve the SOCP is similar, usually taking under 5 iterations. It is also noted that while solving the SOCP is computationally more expensive than simply computing the Newton-Raphson update, the primary computational cost is by far the propagation of the STM from $t_0$ until the targeted time $t_f$, where each iteration in both approaches integrates the high-fidelity ephemeris model over several days with the perilune event detection. Thus, the PC-SCoP does not result in a significant increase in terms of computational cost compared to differential correction-based approaches.

We also note that the linearization process of the dynamics in both DC and the PC-SCoP makes them susceptible to not converging if the initial guess, i.e., the state $\boldsymbol{\theta}_0$ at $t_0$, lies far from the baseline such that the linearized dynamics

up to $t_f$ is inaccurate. While such a case was not encountered for the levels of error assumed in this work, one may replace the single-shooting paradigm used to represent the dynamics by a multiple-shooting one, with the introduction of state $\boldsymbol{\theta}$ at some time(s) $t$ between $t_0$ and $t_f$ as additional variables, to attenuate the sensitivity of the linearized dynamics.

Previous works on DC-based $x$-axis crossing control report the method to provide sufficient robustness against realistic sources and levels of uncertainty, provided that the targeting tolerances and $W_{t_f}$ in (14) are tuned appropriately, for durations of several months to a few years. At a cadence of once every few years, additional corrective maneuvers, such as targeting the $y$-component at perilune passage, are additionally necessitated to return the spacecraft closer to its intended dynamical regime.[1] Through numerical experiments, we empirically find that the PC-SCoP possesses a similar robustness property, provided that $\boldsymbol{\varepsilon}_{\boldsymbol{\vartheta},\mathrm{targ}}$ and $\varepsilon_{t_f,\mathrm{targ}}$ are not set to excessively strict or loose values.

## 4 RECURSIVE SIMULATION SETUP

The actual performance of station-keeping algorithms can be assessed only through recursive simulation; per-maneuver performance may reveal initial insights, but the extent to which the controller stabilizes the steered path is revealed by the cumulative cost rather than the per-maneuver cost. The performance is further affected by the presence of uncertainty, primarily arising from navigation uncertainty, maneuver execution error, and variability on perturbation terms such as the SRP. As such, a recursive, Monte-Carlo experiment must be conducted.

Figure 3 shows the recursive simulation setup. We define two distinct threshold parameters, namely a *trigger* threshold $\boldsymbol{\varepsilon}_{\mathrm{trig}}$, used to determine whether a maneuver is warranted, and a *targeting* threshold $\boldsymbol{\varepsilon}_{\mathrm{targ}}$, used within the control scheme. These two thresholds are separately defined to study the effect of choosing $\boldsymbol{\varepsilon}_{\mathrm{trig}} > \boldsymbol{\varepsilon}_{\mathrm{targ}}$ instead of simply setting $\boldsymbol{\varepsilon}_{\mathrm{trig}} = \boldsymbol{\varepsilon}_{\mathrm{targ}}$. By selecting a tighter $\boldsymbol{\varepsilon}_{\mathrm{targ}}$ than $\boldsymbol{\varepsilon}_{\mathrm{trig}}$, the propagation of the true state executing a maneuver computed based on an erroneous state estimate and under other sources of errors is more likely to lie within $\boldsymbol{\varepsilon}_{\mathrm{trig}}$ at subsequent revolutions. Due to such a hysteresis-type effect, we expect selecting $\boldsymbol{\varepsilon}_{\mathrm{trig}} > \boldsymbol{\varepsilon}_{\mathrm{targ}}$ to reduce the cumulative control cost. Also, note that the control maneuver must be computed based on a corrupted state due to navigation error.

The recursive simulation is terminated if the station-keeping algorithm does not converge; in addition to clearing the targeting tolerances within the threshold $\boldsymbol{\varepsilon}_{\mathrm{targ}}$, we require the maneuver magnitude to be within a predefined maximum executable maneuver magnitude, denoted as $\Delta v_{\max}$. This threshold serves as an indicator for cases where the controller has failed to maintain the tracking error under control. We do not implement corrective procedures such as reducing the targeting perilune horizon $N$ as is implemented in Davis et al.,[1] electing to focus on the nominal performance of the controller rather than on corrective cases when the nominal procedure fails.

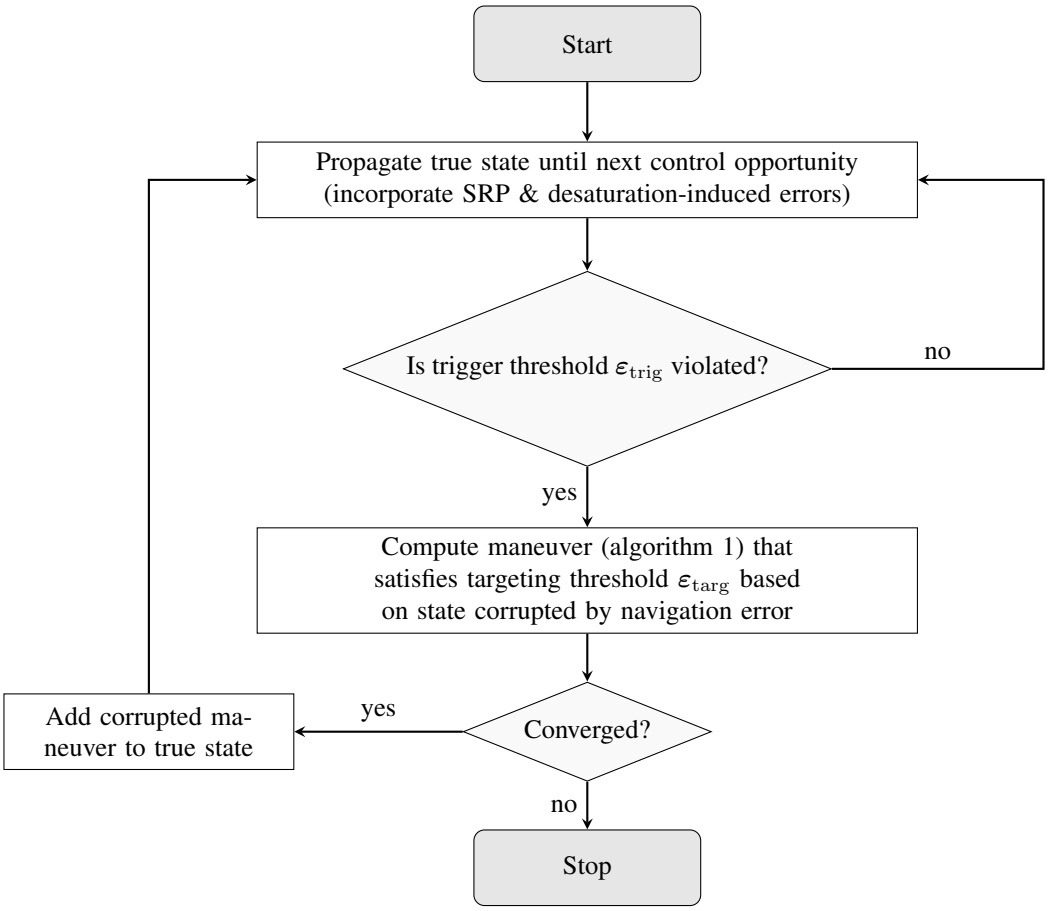



**Figure 3**: Flow-chart for recursive simulation of station-keeping algorithms

### 4.1 Error Simulation

The type and values of errors associated with the simulation are taken from Davis et al.;[1] namely, dynamics error in terms of SRP coefficients, velocity perturbations induced by desaturation, navigation error, and maneuver execution error are incorporated. We provide details on how each of these errors is introduced.

*4.1.1 Solar Radiation Pressure Error* Immediately after a maneuver is executed, the SRP area-to-mass ratio $A/m$ and reflective coefficient $C_R$ are modified with respect to a preset nominal value, denoted by $(A/m)_0$ and $C_{R,0}$, with a perturbation defined based on relative standard deviations $\sigma_{\bar{C}_r}$ and $\sigma_{\overline{(A/m)}}$.

$$C_R = C_{R,0}\left(1 + \delta\bar{C}_R\right), \quad \delta\bar{C}_R \sim \mathcal{N}(0, \sigma^2_{\bar{C}_r}), \tag{28}$$

$$(A/m) = (A/m)_0\left(1 + \delta\overline{(A/m)}\right), \quad \delta\overline{(A/m)} \sim \mathcal{N}(0, \sigma^2_{\overline{(A/m)}}). \tag{29}$$

*4.1.2 Desaturation Error* Desaturation of the momentum wheel results in small velocity perturbations on the spacecraft. This is modeled as a random $\delta\boldsymbol{v}_{\text{desat}}$ occurring instantaneously when the desaturation occurs. Let $\sigma_{\delta v_{\text{desat}}}$ denote the standard deviation of the magnitude of this perturbation, then

$$\boldsymbol{v}^+ = \boldsymbol{v}^- + \delta v_{\text{desat}}\hat{\boldsymbol{i}}, \quad \delta v_{\text{desat}} \sim \mathcal{N}\left(0, \sigma^2_{\delta v_{\text{desat}}}\right), \tag{30}$$

where $\boldsymbol{v}^-$ and $\boldsymbol{v}^+$ denote, respectively, the velocity vector immediately before and after the desaturation event, and $\hat{\boldsymbol{i}} \in \mathbb{R}^3$ is a random unit vector given by

$$\hat{\boldsymbol{i}} = \frac{1}{\sqrt{i_x^2 + i_y^2 + i_z^2}}\begin{bmatrix} i_x \\ i_y \\ i_z \end{bmatrix}, \quad i_x, i_y, i_z \sim \mathcal{U}(-1, 1). \tag{31}$$

*4.1.3 Navigation Error* Navigation error is given solely to the controller, mimicking the fact that the control action must be computed from the state estimates rather than the true state. The position and velocity vector estimates, $\hat{\boldsymbol{r}}$ and $\hat{\boldsymbol{v}}$, are given by

$$\hat{\boldsymbol{r}} = \boldsymbol{r} + \delta\boldsymbol{r}_{\text{nav}}, \quad \delta\boldsymbol{r}_{\text{nav}} \sim \mathcal{N}(0, \sigma^2_{\boldsymbol{r}}), \tag{32a}$$

$$\hat{\boldsymbol{v}} = \boldsymbol{r} + \delta\boldsymbol{v}_{\text{nav}}, \quad \delta\boldsymbol{v}_{\text{nav}} \sim \mathcal{N}(0, \sigma^2_{\boldsymbol{v}}). \tag{32b}$$

*4.1.4 Maneuver Execution Error* The actual maneuver that is executed cannot exactly match the output of the control algorithm due to actuation errors. The maneuver execution model is based on the analysis provided by Gates.[39] Let $\Delta\bar{\boldsymbol{v}}$ denote the *ideal* control action computed by the algorithm, and let $\Delta\boldsymbol{v}$ denote the control action corrupted by

noise and executed on the spacecraft, such that

$$\boldsymbol{v}^{+} = \boldsymbol{v}^{-} + \Delta\boldsymbol{v}(\Delta\bar{\boldsymbol{v}}, \delta\tilde{\boldsymbol{v}}_{\mathrm{abs}}, \delta\tilde{\boldsymbol{v}}_{\mathrm{rel}}, \delta\phi), \tag{33}$$

where $\boldsymbol{v}^{-}$ and $\boldsymbol{v}^{+}$ denote, respectively, the velocity vector immediately before and after the maneuver. The execution error is realized in three parts: in terms of a relative magnitude error, $\delta\tilde{\boldsymbol{v}}_{\mathrm{rel}}$, an absolute magnitude error, $\delta\tilde{\boldsymbol{v}}_{\mathrm{abs}}$, and a direction error, $\delta\phi$. The relative magnitude error is modeled by a relative standard deviation $\sigma_{\delta v_{\mathrm{rel}}}$, such that

$$\delta\tilde{\boldsymbol{v}}_{\mathrm{rel}} = \|\Delta\bar{\boldsymbol{v}}\|\delta v_{\mathrm{rel}}\hat{\boldsymbol{i}}, \quad \delta v_{\mathrm{rel}} \sim \mathcal{N}\left(0, \sigma^2_{\delta v_{\mathrm{rel}}}\right), \tag{34}$$

where $\hat{\boldsymbol{i}}$ follows the definition from (31), but realized separately. The absolute magnitude error is modeled by a standard deviation $\sigma_{\delta v_{\mathrm{abs}}}$, such that

$$\delta\tilde{\boldsymbol{v}}_{\mathrm{abs}} = \delta v_{\mathrm{abs}}\hat{\boldsymbol{i}}, \quad \delta v_{\mathrm{abs}} \sim \mathcal{N}\left(0, \sigma^2_{\delta v_{\mathrm{abs}}}\right). \tag{35}$$

Finally, the direction error is modeled by a random rotation of the quantity $\Delta\bar{\boldsymbol{v}} + \delta\tilde{\boldsymbol{v}}_{\mathrm{rel}} + \delta\tilde{\boldsymbol{v}}_{\mathrm{abs}}$ by an angle $\delta\phi$, with standard deviation $\sigma_\phi$. The corresponding transformation matrix $\boldsymbol{T}(\delta\phi)$ is given by the Rodrigues' rotation formula

$$\boldsymbol{T}(\delta\phi) = \cos(\delta\phi)\boldsymbol{I}_3 + \sin(\delta\phi)\boldsymbol{i}^{\times} + [1 - \cos(\delta\phi)]\boldsymbol{i}\boldsymbol{i}^{T}, \quad \delta\phi \sim \mathcal{N}(0, \sigma^2_{\phi}), \tag{36}$$

where $\boldsymbol{i}^{\times}$ is the skew-symmetric form of $\boldsymbol{i}$. Combining all errors together, the corrupted maneuver is given by

$$\Delta\boldsymbol{v} = \boldsymbol{T}(\delta\phi)\left[\Delta\bar{\boldsymbol{v}} + \delta\tilde{\boldsymbol{v}}_{\mathrm{rel}} + \delta\tilde{\boldsymbol{v}}_{\mathrm{abs}}\right], \tag{37}$$

and is substituted back into equation (33).

## 5 NUMERICAL RESULTS

We report the results of the proposed PC-SCoP approaches, both in terms of station-keeping cost and phase-tracking capability. Simulation parameters for the recursive simulation are given in Table 1; parameters on the errors are chosen to match the simulation in Davis et al.[1] as closely as possible. Table 2 summarizes the parameters used to tune the controllers. In each case, 100 Monte-Carlo samples are realized for a duration of 300 revolutions, which corresponds to approximately 5.40 years on the 9:2 resonant NRHO from NASA.[35] Table 3 summarizes the success rate, maneuver utilization rate, and station-keeping cost statistics obtained from the various controllers and their chosen parameters. For comparison, Davis et al.[1] reports a mean annual cost of $134\,\mathrm{cm/s}$ with a range varying from $120\,\mathrm{cm/s}$ and $160\,\mathrm{cm/s}$. We then conduct a separate Monte-Carlo experiment for 600 revolutions, taking the best-performing

**Table 1**: Parameters for numerical simulation

| Simulation Parameter | | Value |
|---|---|---|
| Simulation duration, revolutions | | 300 / 500 |
| Maneuver location true anomaly, deg | | 200 |
| Maximum maneuver magnitude $\Delta v_{\max}$, m/s | | 1.0 |
| SRP error | Relative $A/m$ $3\sigma_{\overline{(A/m)}}$, % | 30 |
| | Relative $C_r$ $3\sigma_{\bar{C}_r}$, % | 15 |
| Desaturation | Induced velocity error $3\sigma_{\delta v_{\text{desat}}}$, cm/s | 1.0 |
| | Location true anomaly, deg | 340, 350, 10, 190 |
| Navigation error | Position $3\sigma_{\boldsymbol{r}}$, km | 1.5 |
| | Velocity $3\sigma_{\boldsymbol{v}}$, cm/s | 0.8 |
| Maneuver execution error | Relative magnitude $3\sigma_{\delta v_{\text{rel}}}$, % | 1.5 |
| | Absolute magnitude $3\sigma_{\delta v_{\text{abs}}}$, mm/s | 1.42 |
| | Direction $3\sigma_{\phi}$, deg | 1 |

**Table 2**: Parameters for controllers

| Parameter | | Differential Correction | PC-SCoP |
|---|---|---|---|
| Targeted future perilune, $N$ | | $7^{\text{th}}$ | $7^{\text{th}}$ |
| Targeted state components $\mathcal{M}$ | | $v_x$ | $v_x$, $v_z$ |
| State tolerance, m/s | $\boldsymbol{\varepsilon}_{\boldsymbol{\vartheta},\text{trig}}$ | 20 | 20 |
| | $\boldsymbol{\varepsilon}_{\boldsymbol{\vartheta},\text{targ}}$ | 20 | 20/5 |
| Phase tolerance, min | $\varepsilon_{tf,\text{trig}}$ | - | 20 |
| | $\varepsilon_{tf,\text{targ}}$ | - | 20 / 10 |

parameters from the former experiment.

### 5.1 Performance Sensitivity of Phase-Constrained Sequential Cone Program

As a baseline, Figure 4 shows Monte-Carlo results of the cumulative cost and the perilune deviation, using a DC process with no phase constraint. The perilune deviation is computed by comparing the perilune epochs along the realized, steered path to the baseline's perilune passes. As expected, the phase, as measured by the crossing time of perilune, experiences a secular drift, growing up to around 2.1 hours in 5 years. Neglecting the phase deviation, the

**Table 3**: DC and PC-SCoP performances of yearly cost $\Delta v_{\text{yr}}$ from 300-revolutions Monte-Carlo experiment

| Controller | Case | $\boldsymbol{\varepsilon}_{\boldsymbol{\vartheta},\text{targ}}$, m/s | $\varepsilon_{tf,\text{targ}}$, min | Success | Utilization, % | $\Delta v_{\text{yr}}$, cm/s | | |
|---|---|---|---|---|---|---|---|---|
| | | | | | | Mean | $95^{\text{th}}$-% | 1-$\sigma$ |
| DC | - | 20 | n.a. | 100 / 100 | 83.22 | 103.59 | 164.58 | 27.56 |
| PC-SCoP | (a) | 20 | 20 | 100 / 100 | 79.94 | 118.53 | 134.87 | 9.83 |
| | (b) | 5 | 20 | 100 / 100 | 70.93 | 103.27 | 117.45 | 9.10 |
| | (c) | 5 | 10 | 97 / 100 | 65.97 | 136.34 | 209.53 | 38.66 |

In all cases, $\boldsymbol{\varepsilon}_{\boldsymbol{\vartheta},\text{trig}} = 20$ m/s and $\varepsilon_{tf,\text{trig}} = 20$ min (for PC-SCoP only) are used.
Utilization rate and maneuver statistics are computed from successful cases only.

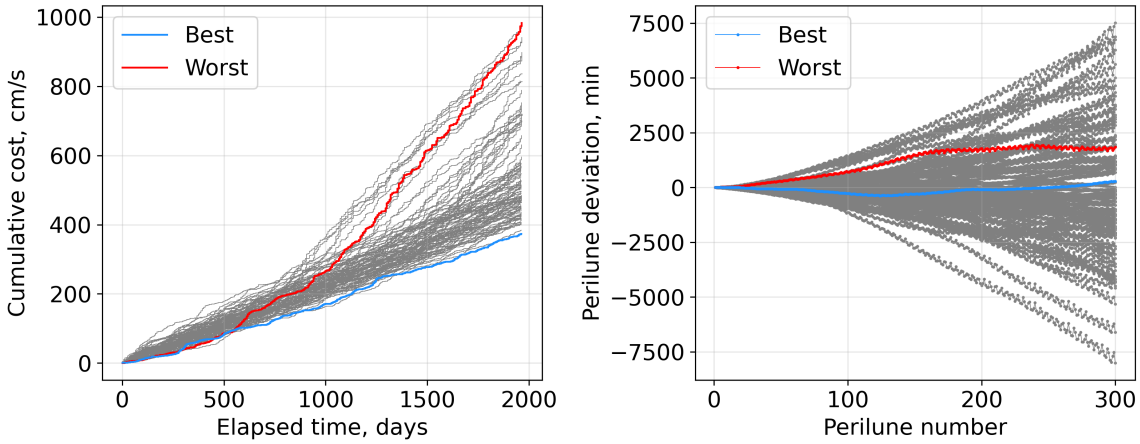



**Figure 4**: Monte-Carlo results over 300 revolutions using differential correction with no phase constraint

DC approach is able to track the baseline throughout 300 revolutions in all 100 Monte-Carlo trials, as summarized in Table 3. Notably, while the mean yearly cost is low, its standard deviation is considerable; the high standard deviation corresponds to cases with significant perilune time deviation, where the targeted perilune state degrades in effectiveness for maintaining the spacecraft on the NRHO.

Figure 5 shows the Monte-Carlo results from the PC-SCoP, using different levels of $\varepsilon_{\text{trig}}$ and $\varepsilon_{\text{targ}}$. In cases (a) and (b), where $t_{p,\text{targ}} = 20$ min is used, the perilune epoch deviation initially undergoes secular growth but becomes bounded within approximately the targeting tolerance $t_{p,\text{targ}}$ chosen for the entire duration of the simulation. In contrast, with case (c) where a stricter $t_{p,\text{targ}} = 10$ min is used, we observe instability occurring by around the $200^{\text{th}}$ revolution, resulting in a quick ramp-up of the station-keeping cost as well. These trends are quantified in Table 3, where both the mean and standard deviation of the yearly cost are high.

Note that choosing a tighter state targeting threshold $\varepsilon_{\boldsymbol{\vartheta},\text{targ}}$, as is done in case (b), results in reduced cumulative cost compared to (a), where $\varepsilon_{\boldsymbol{\vartheta},\text{trig}} = \varepsilon_{\boldsymbol{\vartheta},\text{targ}}$ is used. The difference in performance between (a) and (b) highlights the importance of choosing a tighter $\varepsilon_{\boldsymbol{\vartheta},\text{targ}}$. As hypothesized, the resulting maneuvers are less likely to immediately violate the triggering tolerance upon execution, even under the modeled errors. As a consequence, fewer maneuvers are required, quantified by a lower maneuver utilization ratio. With either DC or PC-SCoP case (a), the maneuver utilization is around 80%, whereas with case (b), where a hysteresis in terms of $\varepsilon_{\boldsymbol{\vartheta},\text{trig}} > \varepsilon_{\boldsymbol{\vartheta},\text{targ}}$ is introduced, the utilization drops by about 10%. Comparing cases (b) and (c), we see that a hysteresis in terms of $\varepsilon_{t_f,\text{trig}} > \varepsilon_{t_f,\text{targ}}$ also drops the utilization by another 5%; however, as noted before, this tighter $\varepsilon_{t_f,\text{targ}}$ results in instability, and hence does not translate to a reduction in cumulative station-keeping cost.

Greater insight into the behavior of the PC-SCoP may be interpreted by looking at Figure 6, which shows the state deviation between the steered path's perilune passes and the baseline's perilune passes. Due to the choice of $\mathcal{M}$ consisting only of velocity state components, we see that the secular growth of the velocity state deviations is slower

than that of the position state deviations. Note that the controller achieves the reported performance without any additional procedures, such as reducing the targeting horizon $N$ in case the controller fails, or introducing additional corrective maneuvers, as is done in Davis et al.[1] If the PC-SCoP is to be implemented in an actual mission, the controller's station-keeping performance over the course of 5 years gives ample time to design an additional corrective maneuver targeting the full state components of the baseline at some point, thus attenuating the error in perilune deviation.

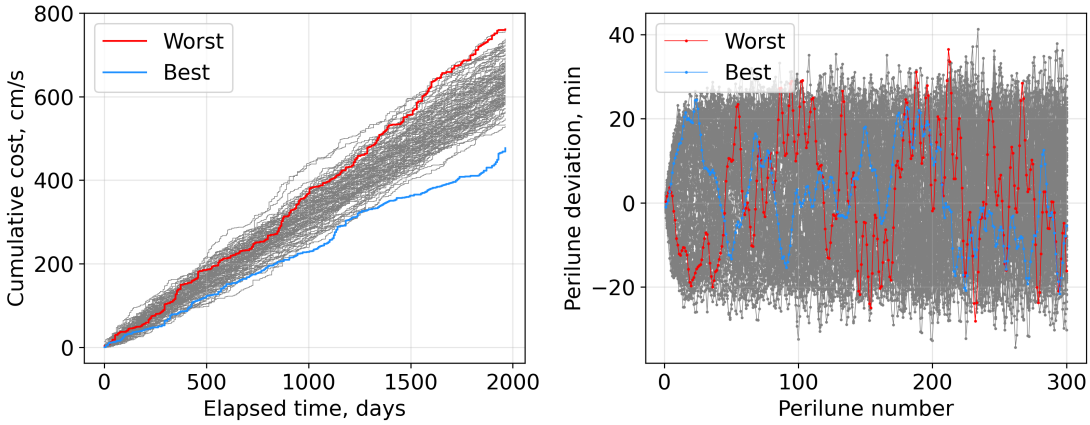
### 5.2 Long-Term Performance of Phase-Constrained Sequential Cone Program

Figure 7 shows the Monte-Carlo results from 600 revolutions, or approximately 10.75 years, using the PC-SCoP with $v_{x,\mathrm{trig}} = v_{z,\mathrm{trig}} = 20\ \mathrm{m/s}$, $v_{x,\mathrm{targ}} = v_{z,\mathrm{targ}} = 5\ \mathrm{m/s}$, and $t_{p,\mathrm{trig}} = 20\ \mathrm{min}$, $t_{p,\mathrm{targ}} = 20\ \mathrm{min}$. Here, we observe a success rate of 95%, with a mean yearly cost of $123.74\ \mathrm{cm/s}$ and a standard deviation of $22.77\ \mathrm{cm/s}$. Note that the yearly mean and standard deviation are significantly higher than the Monte-Carlo results with 300 revolutions due to poorer performance of the controller on a significant number of realizations beyond around the $2500^{\mathrm{th}}$ day.

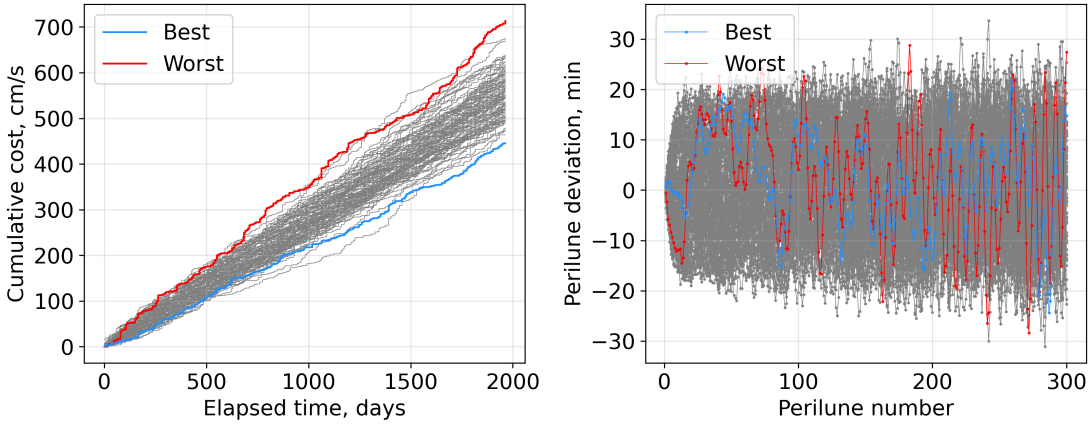
In practice, it is unrealistic for a spacecraft on the NRHO to necessitate autonomous station-keeping ability for such a long duration. When excessive divergence from the baseline is observed on the ground or onboard, a corrective maneuver that aligns the spacecraft back to the baseline through full-state targeting could be designed and implemented. The design of such a maneuver may, for example, follow a simple targeting scheme and is beyond the scope of this work. Based on the performance of the controller in Figure 7a, a check every 100 revolutions would be sufficient to realign the spacecraft back to its intended path. Comparing the trends of the perilune deviation from the best- and worst-performing Monte-Carlo samples, it is possible to observe that the worst-performing case exhibits a short-period oscillation in perilune deviation with a period of one to a few revolutions; in contrast, the best-performing case only exhibits a slower oscillation with a period of 10's of revolutions. This characteristic may serve the purpose of an early indicator for divergence to trigger cautionary measures, such as the aforementioned full-state targeting maneuver.

## 6 CONCLUSIONS

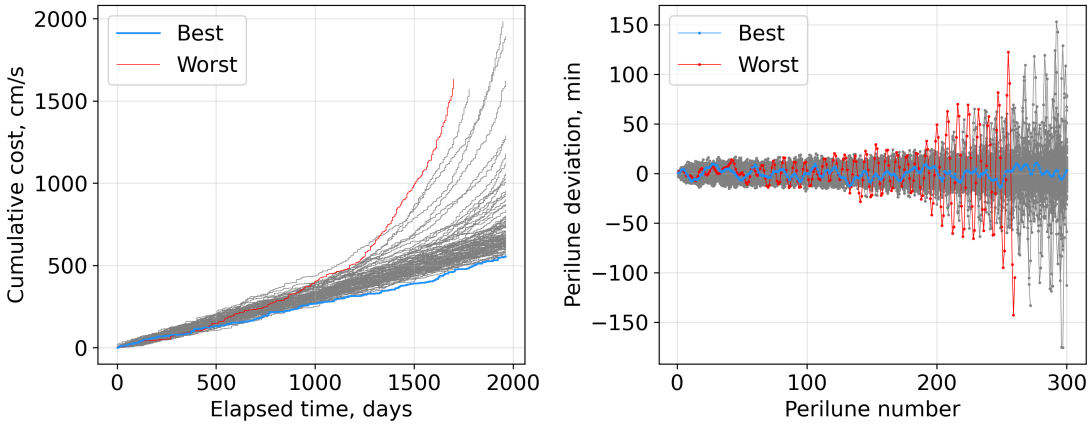
In this work, we proposed an optimization-based station-keeping algorithm for maintaining the spacecraft in the vicinity of a baseline cislunar LPO, while ensuring the phase along the LPO is tracked. The proposed algorithm, coined as the PC-SCoP, is inspired by the $x$-axis crossing control technique, which has widely been studied and used in combination with a differential correction algorithm; instead, this work poses the $x$-axis crossing control framework as a minimization problem. The resulting NLP is sequentially solved via linearization of the dynamics with respect to increment in velocity and time of flight, with an objective based on the two-norm $\Delta V$ cost, thus resulting in a SOCP. The SOCP is repeatedly solved, each time re-linearizing the dynamics about the current state estimate, appending the control corresponding to the minimizing solution from the previous iteration. The advantage of the PC-SCoP is that the explicit optimization problem allows for handling targeting requirements of different quantities, namely, position,




a $v_{x,\text{trig}} = v_{z,\text{trig}} = 20$ m/s, $v_{x,\text{targ}} = v_{z,\text{targ}} = 20$ m/s, $t_{p,\text{trig}} = 20$ min, $t_{p,\text{targ}} = 20$ min

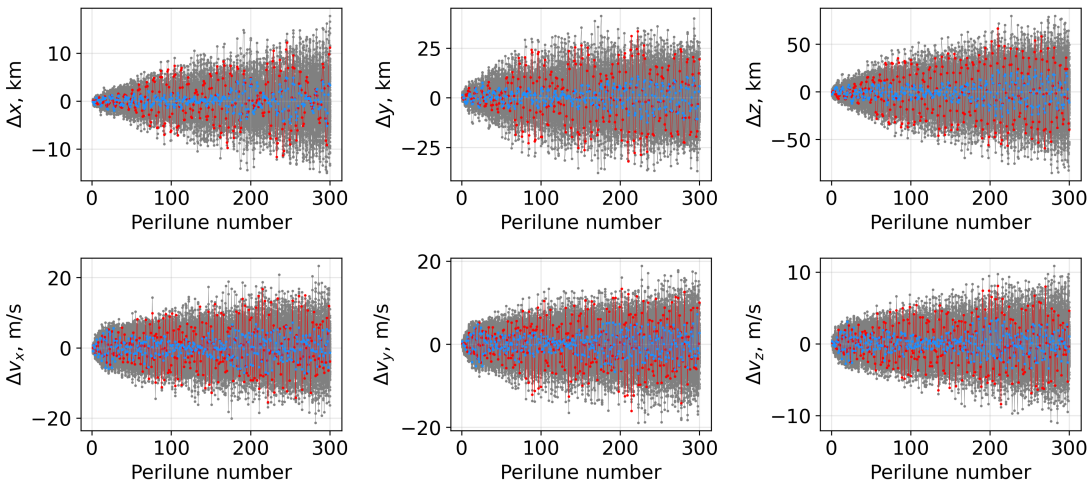



b $v_{x,\text{trig}} = v_{z,\text{trig}} = 20$ m/s, $v_{x,\text{targ}} = v_{z,\text{targ}} = 5$ m/s, $t_{p,\text{trig}} = 20$ min, $t_{p,\text{targ}} = 20$ min

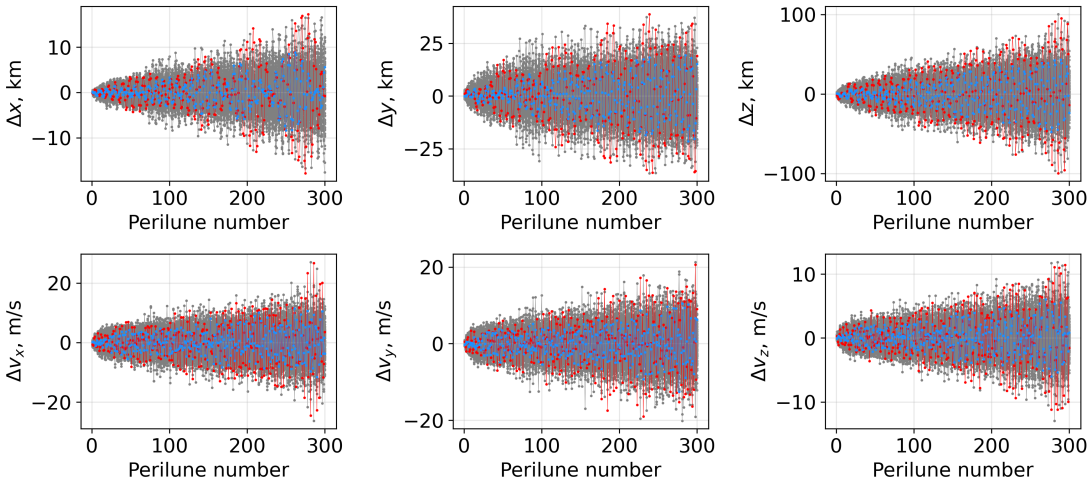



c $v_{x,\text{trig}} = v_{z,\text{trig}} = 20$ m/s, $v_{x,\text{targ}} = v_{z,\text{targ}} = 5$ m/s, $t_{p,\text{trig}} = 20$ min, $t_{p,\text{targ}} = 10$ min

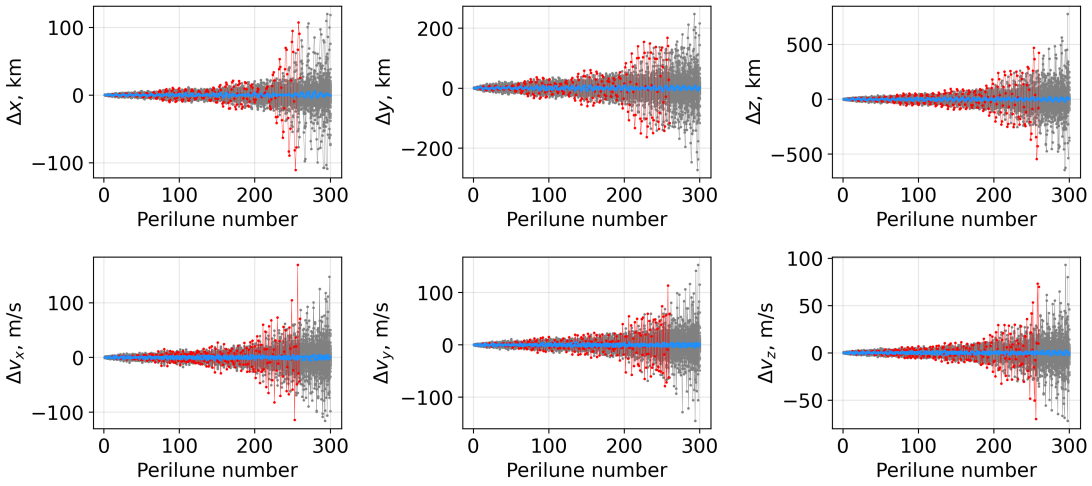
**Figure 5**: Monte-Carlo results over 300 revolutions of cumulative cost and perilune epoch deviation with the PC-SCoP




a $v_{x,\mathrm{trig}} = v_{z,\mathrm{trig}} = 20$ m/s, $v_{x,\mathrm{targ}} = v_{z,\mathrm{targ}} = 20$ m/s, $t_{p,\mathrm{trig}} = 20$ min, $t_{p,\mathrm{targ}} = 20$ min

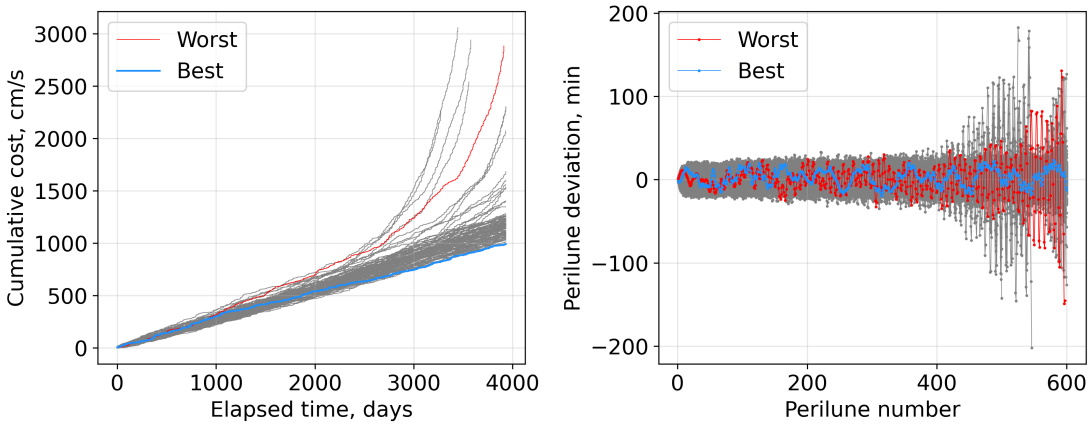



b $v_{x,\mathrm{trig}} = v_{z,\mathrm{trig}} = 20$ m/s, $v_{x,\mathrm{targ}} = v_{z,\mathrm{targ}} = 5$ m/s, $t_{p,\mathrm{trig}} = 20$ min, $t_{p,\mathrm{targ}} = 20$ min

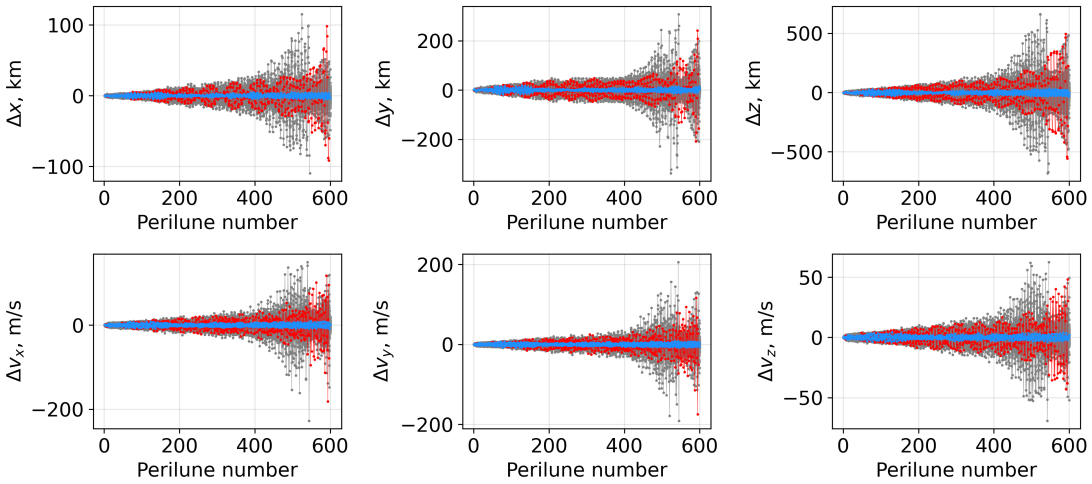



c $v_{x,\mathrm{trig}} = v_{z,\mathrm{trig}} = 20$ m/s, $v_{x,\mathrm{targ}} = v_{z,\mathrm{targ}} = 5$ m/s, $t_{p,\mathrm{trig}} = 20$ min, $t_{p,\mathrm{targ}} = 10$ min

**Figure 6**: Monte-Carlo results over 300 revolutions of perilune state deviation with the PC-SCoP




a Cumulative cost and perilune epoch deviation




b Perilune state deviation

**Figure 7**: Monte-Carlo results over 600 revolutions with PC-SCoP, using $v_{x,\text{trig}} = v_{z,\text{trig}} = 20$ m/s, $v_{x,\text{targ}} = v_{z,\text{targ}} = 5$ m/s, $t_{p,\text{trig}} = 20$ min, $t_{p,\text{targ}} = 20$ min

velocity, and/or timing, as separate constraints; the constraints replace the use of weights to scale the various quantities into a single residual vector with targeting tolerances for each component, which are intuitive, physical quantities that are easier to tune.

The PC-SCoP has been demonstrated on the 9:2 resonant NRHO in the high-fidelity ephemeris model, incorporating realistic sources of uncertainties, namely navigation error, control execution error, dynamics error in the solar radiation pressure term, and impulse imparted from momentum dumping. With appropriately chosen targeting tolerances, the PC-SCoP is found to be able to track the reference LPO for an extended duration of 300 revolutions, or about 5.4 years. During this time, the state deviation, recorded at each perilune pass experiences a secular growth, but this does not prohibit its use for autonomous station-keeping over the course of extended durations. Furthermore, through a Monte-Carlo simulation lasting 600 revolutions, a rapid oscillatory trend in perilune deviation that corresponds to cases with poor tracking performance and eventual divergence has been identified; the oscillatory trend can be used as a trigger to occasionally design and execute a simple targeting maneuver that realigns the spacecraft closer to the intended baseline; still, such a maneuver should be seldom required, for example at intervals of every 100 revolutions, or about 1.79 years.

Overall, the proposed optimization-based framework provides a more explainable framework for station-keeping, which also delivers performance that matches state-of-the-art $x$-axis crossing control approaches. Future work includes applying the PC-SCoP to other LPOs, as well as the design of corrective maneuvers necessitated once every few years to realign the spacecraft back to the baseline in order to achieve nearly autonomous station-keeping for a duration exceeding 300 revolutions.

## REFERENCES


[1] D. C. Davis, S. T. Scheuerle, D. A. Williams, F. S. Miguel, E. M. Zimovan-Spreen, and K. C. Howell, "Orbit Maintenance Burn Details for Spacecraft in a Near Rectilinear Halo Orbit," *AAS/AIAA Astrodynamics Specialists Conference*, 2022.

[2] D. A. P. Williams, K. C. Howell, and D. C. Davis, "A Comparison of Station-Keeping Strategies for Halo Orbits," *AAS/AIAA Astrodynamics Specialist Conference*, Vol. 231, 2023, pp. 1–20.

[3] J. K. Vendl and M. J. Holzinger, "Cislunar Periodic Orbit Analysis for Persistent Space Object Detection Capability," *Journal of Spacecraft and Rockets*, Vol. 58, No. 4, 2021, pp. 1174–1185, 10.2514/1.A34909.

[4] L. Visonneau, Y. Shimane, and K. Ho, "Optimizing Multi-spacecraft Cislunar Space Domain Awareness Systems via Hidden-Genes Genetic Algorithm," *The Journal of the Astronautical Sciences*, Vol. 70, jul 2023, p. 22, 10.1007/s40295-023-00386-8.

[5] Y. Shimane, K. Tomita, and K. Ho, "Strategic Regions for Monitoring Incoming Low-Energy Transfers to Low-Lunar Orbit," *Advanced Maui Optical and Space Surveillance Technologies (AMOS) Conference*, 2023.

[6] D. Folta and F. Vaughn, "A survey of earth-moon libration orbits: Stationkeeping strategies and intra-orbit transfers," *Collection of Technical Papers - AIAA/AAS Astrodynamics Specialist Conference*, Vol. 1, 2004, pp. 87–106, 10.2514/6.2004-4741.

[7] M. Shirobokov, S. Trofimov, and M. Ovchinnikov, "Survey of station-keeping techniques for libration point orbits," *Journal of Guidance, Control, and Dynamics*, Vol. 40, No. 5, 2017, pp. 1085–1105, 10.2514/1.G001850.

[8] W. Wiesel and W. Shelton, "Modal Control of an Unstable Periodic Orbit.," *Journal of the Astronautical Sciences*, Vol. 31, No. 1, 1983, pp. 63–76.

[9] C. Simó, G. Gómez, J. Llibre, R. Martínez, and J. Rodríguez, "On the optimal station keeping control of halo orbits," *Acta Astronautica*, Vol. 15, No. 6-7, 1987, pp. 391–397, 10.1016/0094-5765(87)90175-5.

[10] M. Xin, S. N. Balakrishnan, and H. J. Pernicka, "Libration point stationkeeping using the $\theta$ - D technique," *Journal of the Astronautical Sciences*, Vol. 56, No. 2, 2008, pp. 231–250, 10.1007/BF03256550.

[11] L. Bucci, M. Lavagna, and R. Jehn, "Station Keeping Techniques for Near Rectilinear Halo Orbits in the Earth-Moon System," *10th International ESA Conference on Guidance, Navigation & Control Systems*, 2017.

[12] F. Colombi, A. Colagrossi, and M. Lavagna, "Floquet modes and stability analysis of periodic orbit-attitude solutions along Earth–Moon halo orbits," *Celestial Mechanics and Dynamical Astronomy*, Vol. 133, No. 7, 2021, pp. 1–33, 10.1007/s10569-021-10030-y.

[13] P. Elango, S. Di Cairano, U. Kalabic, and A. Weiss, "Local Eigenmotion Control for Near Rectilinear Halo Orbits," *Proceedings of the American Control Conference*, Vol. 2022-June, 2022, pp. 1822–1827, 10.23919/ACC53348.2022.9867672.

[14] D. Guzzetti, E. M. Zimovan, K. C. Howell, and D. C. Davis, "Stationkeeping Analysis for Spacecraft in Lunar Near Rectilinear Halo Orbits," *AAS/AIAA Space Flight Mechanics Meeting*, 2017.

[15] X. Fu, N. Baresi, and R. Armellin, "Stochastic optimization for stationkeeping of periodic orbits using a high-order Target Point Approach," *Advances in Space Research*, Vol. 70, jul 2022, pp. 96–111, 10.1016/j.asr.2022.04.039.

[16] R. Zhang, Y. Wang, Y. Shi, C. Zhang, and H. Zhang, "Performance analysis of impulsive station-keeping strategies for cis-lunar orbits with the ephemeris model," *Acta Astronautica*, Vol. 198, No. January, 2022, pp. 152–160, 10.1016/j.actaastro.2022.05.054.

[17] Y. Lian, G. Gómez, J. J. Masdemont, and G. Tang, "Station-keeping of real Earth-Moon libration point orbits using discrete-time sliding mode control," *Communications in Nonlinear Science and Numerical Simulation*, 2014, 10.1016/j.cnsns.2014.03.026.

[18] M. Elobaid, M. Mattioni, S. Monaco, and D. Normand-Cyrot, "Station-Keeping of L2 Halo Orbits Under Sampled-Data Model Predictive Control," *Journal of Guidance, Control, and Dynamics*, Vol. 45, No. 7, 2022, pp. 1337–1346, 10.2514/1.G006349.

[19] Y. Shimane, S. Di Cairano, K. Ho, and A. Weiss, "Revolution-Spaced Output-Feedback Model Predictive Control for Station Keeping on Near-Rectilinear Halo Orbits," *IEEE Transactions on Control Systems Technology*, 2025, 10.1109/TCST.2025.3614324.

[20] N. Kumagai and K. Oguri, "Robust NRHO Station-Keeping Planning with Maneuver Location Optimization under Operational Uncertainties," *AAS/AIAA Astrodynamics Specialist Conference*, 2023, pp. 1–20.

[21] F. Gettatelli, A. Zavoli, B. Benedikter, and R. Furfaro, "Stationkeeping of Earth-Moon L2 Libration Point Orbits via Optimal Covariance Control," *AAS/AIAA Astrodynamics Specialist Conference*, 2023, pp. 1–20.

[22] D. Davis, S. Bhatt, K. Howell, J.-W. Jang, R. Whitley, F. Clark, D. Guzzetti, E. Zimovan, and G. Barton, "Orbit Maintenance and Navigation of Human Spacecraft at Cislunar Near Rectilinear Halo Orbits," *AAS/AIAA Space Flight Mechanics Meeting*, 2017.

[23] C. P. Newman, D. C. Davis, R. J. Whitley, J. R. Guinn, and M. S. Ryne, "Stationkeeping, Orbit Determination, and Attitude Control for Spacecraft in Near Rectilinear Halo Orbits," *AAS Astrodynamics Specialists Conference*, 2018.

[24] B. Cheetham, T. Gardner, and A. Forsman, "Cislunar autonomous positioning system technology operations and navigation experiment (Capstone)," *Accelerating Space Commerce, Exploration, and New Discovery conference, ASCEND 2021*, American Institute of Aeronautics and Astronautics Inc, AIAA, 2021, 10.2514/6.2021-4128.

[25] D. Malyuta, T. P. Reynolds, M. Szmuk, T. Lew, R. Bonalli, M. Pavone, and B. Açıkmeşe, "Convex Optimization for Trajectory Generation: A Tutorial on Generating Dynamically Feasible Trajectories Reliably and Efficiently," *IEEE Control Systems*, Vol. 42, oct 2022, pp. 40–113, 10.1109/MCS.2022.3187542.

[26] D. Morgan, S. J. Chung, and F. Y. Hadaegh, "Model predictive control of swarms of spacecraft using sequential convex programming," *Journal of Guidance, Control, and Dynamics*, Vol. 37, No. 6, 2014, pp. 1725–1740, 10.2514/1.G000218.

[27] G. Tang, F. Jiang, and J. Li, "Fuel-Optimal Low-Thrust Trajectory Optimization Using Indirect Method and Successive Convex Programming," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 54, aug 2018, pp. 2053–2066, 10.1109/TAES.2018.2803558.

[28] Z. Wang and M. J. Grant, "Minimum-Fuel Low-Thrust Transfers for Spacecraft: A Convex Approach," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 54, No. 5, 2018, pp. 2274–2290, 10.1109/TAES.2018.2812558.

[29] Y. Kayama, K. C. Howell, M. Bando, and S. Hokamoto, "Low-Thrust Trajectory Design with Successive Convex Optimization for Libration Point Orbits," *Journal of Guidance, Control, and Dynamics*, Vol. 45, apr 2022, pp. 623–637, 10.2514/1.G005916.

[30] N. Bernardini, N. Baresi, and R. Armellin, "State-dependent trust region for successive convex programming for autonomous spacecraft," *Astrodynamics*, apr 2024, 10.1007/s42064-024-0200-1.

[31] N. Kumagai and K. Oguri, "Adaptive-Mesh Sequential Convex Programming for Space Trajectory Optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 47, jun 2024, pp. 1–8, 10.2514/1.G008107.

[32] A. C. Morelli, A. Morselli, C. Giordano, and F. Topputo, "Convex Trajectory Optimization Using Thrust Regularization," *Journal of Guidance, Control, and Dynamics*, Vol. 47, feb 2024, pp. 339–346, 10.2514/1.G007646.

[33] P. Elango, S. Di Cairano, K. Berntorp, and A. Weiss, "Sequential linearization-based station keeping with optical navigation for NRHO," *AAS/AIAA Astrodynamics Specialist Conference*, 2022.

[34] R. S. Park, W. M. Folkner, J. G. Williams, and D. H. Boggs, "The JPL Planetary and Lunar Ephemerides DE440 and DE441," *The Astronomical Journal*, Vol. 161, feb 2021, p. 105, 10.3847/1538-3881/abd414.

[35] D. E. Lee, "Gateway Destination Orbit Model: A Continuous 15 Year NRHO Reference Trajectory," tech. rep., NASA Johnson Space Center, 2019.

[36] K. Rivera Lopez and M. Holzinger, "Statistical Analysis of Optimal Stationkeeping Location and Coast Duration Using Stretching Directions," *Journal of the Astronautical Sciences*, Vol. 71, feb 2024, 10.1007/s40295-023-00427-2.

[37] A. Domahidi, E. Chu, and S. Boyd, "ECOS: An SOCP solver for embedded systems," *European Control Conference (ECC)*, 2013, pp. 3071–3076.

[38] B. O'Donoghue, E. Chu, N. Parikh, and S. Boyd, "Conic Optimization via Operator Splitting and Homogeneous Self-Dual Embedding," *Journal of Optimization Theory and Applications*, Vol. 169, June 2016, pp. 1042–1068.

[39] C. R. Gates, "A Simplified Model of Midcourse Maneuver Execution Errors," tech. rep., Jet Propulsion Laboratory, 1963.