OPTIMIZATION-BASED PHASE-CONSTRAINED STATION-KEEPING CONTROL ON LIBRATION POINT ORBIT

Shimane, Yuri; Ho, Koki; Weiss, Avishai

TR2024-109 August 15, 2024

Abstract

In this work, an optimization-based station-keeping algorithm for colinear libration point orbits (LPO) based on the xz-plane crossing control technique is conceived. The optimization problem is cast as a sequential second-order cone pro- gram and incorporates an explicit constraint on the perilune pass epoch to en- sure 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 the Gateway's Near Rectilinear Halo Orbit (NRHO) in the full-ephemeris dynamics with realistic error models.

AAS/AIAA Astrodynamics Specialist Conference 2024

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

(Preprint) AAS 24-499

OPTIMIZATION-BASED PHASE-CONSTRAINED STATION-KEEPING CONTROL ON LIBRATION POINT ORBIT

Yuri Shimane[*], Koki Ho[†], and Avishai Weiss[‡]

In this work, an optimization-based station-keeping algorithm for colinear libration point orbits (LPO) based on the xz-plane crossing control technique 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 the Gateway's Near Rectilinear Halo Orbit (NRHO) in the full-ephemeris dynamics with realistic error models.

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, the 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 necessary to maintain; 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 regards 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 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 the 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 ΔV 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]

[*] 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

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,[8–11] eigenmotion,[12] and Cauchy Green Tensor-based approaches,[2,13] while the latter includes approaches such as targeting control,[14,15] sliding mode,[16] model predictive control (MPC),[17] and, more recently, robust control strategies.[18,19]

In the context of impulsive station-keeping on the NRHO in high-fidelity dynamics, the xz-plane crossing control, belonging to the class of targeting-based approach, has been shown to provide exceptional performance by multiple authors,[1,13,20,21] and has also demonstrated in flight on the recent CAPSTONE mission.[22] The classical xz-plane 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 (e.g. x-component velocity) and time, which is non-intuitive.

In this work, we propose an alternative optimization-based approach to achieve phase-constrained xz-plane crossing control. We leverage the sequential linearization-based formulation previously proposed by Elango et al[23] and extend it to include a phase maintenance constraint. The extended formulation results in a sequentially solved second-order cone program (SOCP), where the dynamics are linearized over each iteration. Due to the use of an explicit optimization problem formulation, our approach replaces the non-intuitive weight of[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.

BACKGROUND

We consider a spacecraft governed by full-ephemeris dynamics in the vicinity of the Moon under 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.

Equations of Motion

We consider the motion of the spacecraft in the inertial frame, $\mathcal{F}_{\text{Inr}}$, centered at the Moon. 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 $\mathcal{F}_{\text{Inr}}$ and the derivative of $\boldsymbol{r}$ in $\mathcal{F}_{\text{Inr}}$, 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

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

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

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

where $\boldsymbol{T}_{\mathrm{Inr}}^{\mathrm{PA}} \in \mathbb{R}^{3\times 3}$ is the transformation matrix from the body's principle axes frame $\mathcal{F}_{\mathrm{PA}}$, to $\mathcal{F}_{\mathrm{Inr}}$, 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^{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^{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^{th} body, and $d_i = \|\boldsymbol{d}_i\|_2$. The state-transition matrix (STM) is given by solving the nonlinear initial value problem (IVP)

$$\dot{\boldsymbol{\theta}}(t) = \boldsymbol{f}(t, \boldsymbol{\theta}), \tag{5a}$$

$$\dot{\Phi}(t, t_0) = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{\theta}}\Phi(t, t_0), \tag{5b}$$

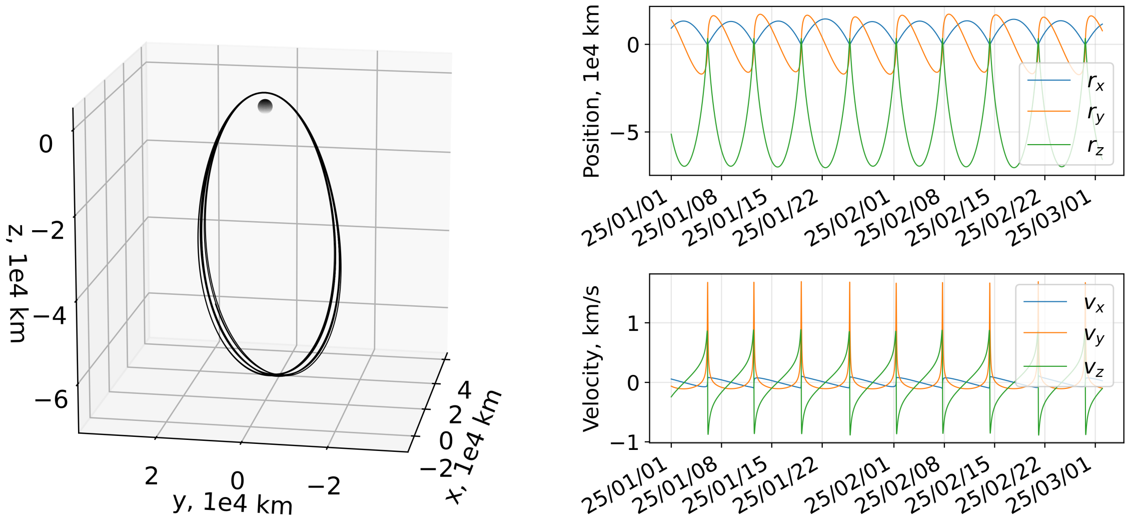
with initial conditions $\boldsymbol{\theta}(t_0) = \boldsymbol{\theta}_0$ and $\Phi_0 = \boldsymbol{I}_6$.

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.[24] Figure 1 shows a portion of the baseline NRHO trajectory for a duration of 60 days in the Earth-Moon rotating frame, centered at the Moon.

STATION-KEEPING CONTROL

The basis of this work's control framework is xz-plane crossing control. In this section, we begin by providing an overview of xz-plane crossing control using differential correction (DC);[1,2,13,20–22] the original algorithm[13,20–22] does not consider tracking with respect to the phase but is presented



(a) Position space

(b) State history

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

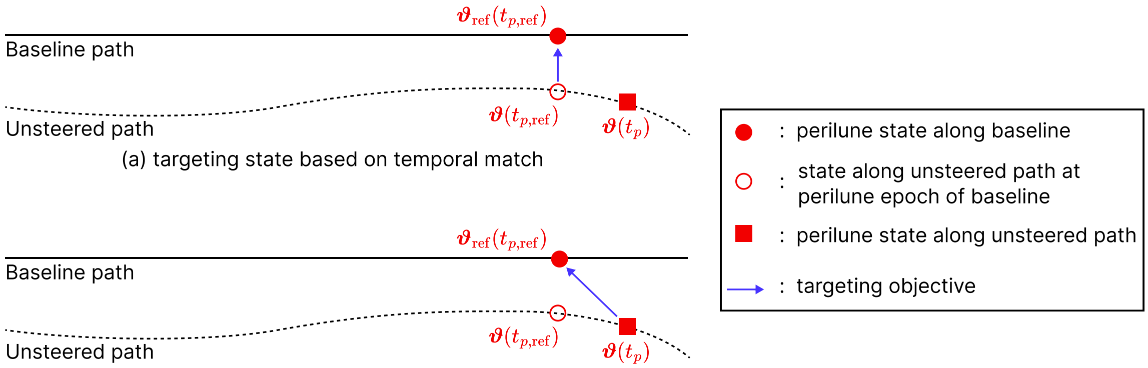
for context. We then provide a brief discussion on the phase-constrained xz-plane crossing with DC, as reported in recent works.[1,2] Finally, we introduce our optimization-based formulation for the same problem, replacing the non-intuitive weights in the DC algorithm with explicit constraints on the phase, which can be derived from mission requirements.

Preliminaries on xz-plane Crossing Control

Background on xz-plane crossing control is presented in terms of the definition of the event where the targeting is to occur, the state components to be chosen for targeting, and the location(s) along the LPO where the maneuver is to be executed.

Targeted Event The controller aims to maintain the steered motion of the spacecraft in the vicinity of a precomputed, ballistic reference path, which will be referred to as the *baseline* hereafter. Consider the Earth-Moon rotating frame $\mathcal{F}_{\text{EM}}$, 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 xz-plane 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 $\mathcal{F}_{\text{EM}}$. The intersection with the xz-plane crossing happens at the perilune in simplified dynamics models such as the circular restricted three-body problem, while in the full-ephemeris system, 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 xz-plane 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 xz-plane 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


$\vartheta_{\mathrm{ref}}(t_{p,\mathrm{ref}})$
Baseline path
Unsteered path
$\vartheta(t_{p,\mathrm{ref}})$
$\vartheta(t_p)$
● : perilune state along baseline
○ : state along unsteered path at perilune epoch of baseline
■ : perilune state along unsteered path
→ : targeting objective

(a) targeting state based on temporal match

(b) targeting state based on event match

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

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 xz-plane 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.

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,

$$\boldsymbol{\vartheta}(t) = \left(\boldsymbol{T}_{\mathrm{EM}}^{\mathrm{Inr}}(t)\boldsymbol{\theta}(t)\right)_{\mathcal{M}}, \tag{6}$$

where $\boldsymbol{T}_{\mathrm{EM}}^{\mathrm{Inr}}(t) \in \mathbb{R}^{6\times 6}$ is the transformation matrix from $\mathcal{F}_{\mathrm{Inr}}$ to $\mathcal{F}_{\mathrm{EM}}$, 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^{th} and 6^{th} components of the vector $\boldsymbol{T}_{\mathrm{EM}}^{\mathrm{Inr}}(t)\boldsymbol{\theta}(t)$. Note that the transformation matrix $\boldsymbol{T}_{\mathrm{EM}}^{\mathrm{Inr}}$ is time-dependent, as the Earth-Moon rotating frame is non-inertial.

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.[13,25] 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 with controls to occur exactly on the osculating apolune. 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.[1,13,20]

Phase-Free Differential Correction

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

$$F = \boldsymbol{\vartheta}_f - \boldsymbol{\vartheta}_{\text{ref}}, \tag{7}$$

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

$$\boldsymbol{\vartheta}_f = \left(\boldsymbol{T}_{\text{EM}}^{\text{Inr}} \int_{t_0}^{t_f} \boldsymbol{f}(t, \boldsymbol{\theta}_0) \mathrm{d}t \right)_{\mathcal{M}}, \tag{8}$$

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 (8) is computed based on state estimates. The free variables vector $X \in \mathbb{R}^3$ is the initial variation in velocity,

$$X = \begin{bmatrix} \Delta v_{x,0} \\ \Delta v_{y,0} \\ \Delta v_{z,0} \end{bmatrix}. \tag{9}$$

The corresponding Jacobian DF is given by

$$DF = \frac{\partial F}{\partial X} = \frac{\partial \boldsymbol{\vartheta}_f}{\partial \boldsymbol{v}_0}. \tag{10}$$

The DC problem involves iteratively solving for $F = \mathbf{0}$, 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

$$X^{(k+1)} = X^{(k)} - \left(DF^{(k)} \right)^T \left[(DF^{(k)})(DF^{(k)})^T \right]^{-1} F(X^{(k)}). \tag{11}$$

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. The iteration is stopped when components of F is smaller than corresponding components in a tolerance vector $\boldsymbol{\varepsilon}_{\boldsymbol{\vartheta},\text{targ}} \in \mathbb{R}^m$,

$$| \left(\boldsymbol{\vartheta}_f - \boldsymbol{\vartheta}_{\text{ref}} \right)_i | \leq \varepsilon_{\boldsymbol{\vartheta}_i,\text{targ}} \quad i = 1, \ldots, m. \tag{12}$$

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 (7)-(10), 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

$$F = \begin{bmatrix} \boldsymbol{\vartheta}_f - \boldsymbol{\vartheta}_{\text{ref}} \\ W_{t_f} \left(t_f - t_{p,\text{ref}} \right) \end{bmatrix}, \tag{13}$$

where W_{t_f} is a 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 (12) and the phase within tolerance $\varepsilon_{t_f,\text{targ}}$ given by

$$|t_f - t_{p,\text{ref}}| \leq \varepsilon_{t_f,\text{targ}}. \tag{14}$$

This algorithm is the proposed station-keeping scheme for the Gateway.

Phase-Constrained Sequential Cone Program

In this work, we propose an alternative approach to phase-constrained xz-plane crossing control by formulating an explicit nonlinear program (NLP) that achieves the phase-constrained xz-plane crossing control. 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).

Formulation The general NLP is given by

$$\min_{\Delta \boldsymbol{v}, t_f} \quad \|\Delta \boldsymbol{v}\|_2, \tag{15a}$$

$$\text{such that} \quad |\left(\boldsymbol{\vartheta}_f - \boldsymbol{\vartheta}_{\text{ref}}\right)_i| \leq \boldsymbol{\varepsilon}_{\boldsymbol{\vartheta}_i,\text{targ}} \quad i = 1, \dots, m, \tag{15b}$$

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

Compared to the Newton-Raphson update (11), the formulation in (15) includes separate targeting constraints on the state (15b) and phase (15c), removing the need for a weight like W_{t_f} in the residual function (13) from the DC-based approach. Note that by making the propagation time of the steered state t_f a variable, the formulation in (15) 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.

Sequential Linearization Let $\bar{t}_f^{(k)}$ be the time until the N^{th} perilune before applying the control maneuver during the k^{th} iteration. Then, without any control maneuver, the targeted state components during the k^{th} iteration $\bar{\boldsymbol{\vartheta}}_f^{(k)} \in \mathbb{R}^m$ is given by

$$\bar{\boldsymbol{\vartheta}}_f^{(k)} = \left(\boldsymbol{T}_{\text{EM}}^{\text{Inr}} \int_{t_0}^{\bar{t}_f^{(k)}} f(\boldsymbol{\theta}_0^{(k)}, \tau) \mathrm{d}\tau \right)_{\mathcal{M}}. \tag{16}$$

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

$$\delta \boldsymbol{\theta}_f^{(k)} = \boldsymbol{T}_{\text{EM}}^{\text{Inr}} \left(\begin{bmatrix} \Phi_{rv}(\bar{t}_f^{(k)}, t_0) \\ \Phi_{vv}(\bar{t}_f^{(k)}, t_0) \end{bmatrix} \Delta \boldsymbol{v}^{(k)} + \frac{\partial \boldsymbol{\theta}(\bar{t}_f^{(k)})}{\partial t} \delta t_f^{(k)} \right). \tag{17}$$

In matrix form, the variation in the final targeted state $\delta \boldsymbol{\vartheta}_f^{(k)}$ is given by

$$\delta \boldsymbol{\vartheta}_f^{(k)} = \begin{bmatrix} \boldsymbol{B}_{\mathcal{M}}^{(k)} & \boldsymbol{C}_{\mathcal{M}}^{(k)} \end{bmatrix} \begin{bmatrix} \Delta \boldsymbol{v}^{(k)} \\ \delta t_f^{(k)} \end{bmatrix}, \tag{18}$$

where $\boldsymbol{B}_{\mathcal{M}}^{(k)} \in \mathbb{R}^{m \times 3}$ and $\boldsymbol{C}_{\mathcal{M}}^{(k)} \in \mathbb{R}^{m \times 1}$ are given by

$$\boldsymbol{B}_{\mathcal{M}}^{(k)} = \left(\boldsymbol{T}_{\text{EM}}^{\text{Inr}} \begin{bmatrix} \Phi_{rv}(\bar{t}_f^{(k)}, t_0) \\ \Phi_{vv}(\bar{t}_f^{(k)}, t_0) \end{bmatrix} \right)_{\mathcal{M}}, \tag{19}$$

$$\boldsymbol{C}_{\mathcal{M}}^{(k)} = \left(\boldsymbol{T}_{\text{EM}}^{\text{Inr}} \frac{\partial \boldsymbol{\theta}(\bar{t}_f^{(k)})}{\partial t} \right)_{\mathcal{M}}. \tag{20}$$

Through the linearization of constraints (15b) and (15c) by making use of the linearized final state $\bar{\boldsymbol{\vartheta}}_f^{(k)}$ and final state deviation $\delta\boldsymbol{\vartheta}_f^{(k)}$, the NLP (15) 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

$$\min_{\Delta\boldsymbol{v}^{(k)},\delta t_f^{(k)},\eta} \quad \eta, \tag{21a}$$

$$\text{such that} \quad \|\Delta\boldsymbol{v}^{(k)}\|_2 \leq \eta, \tag{21b}$$

$$\left|\left(\bar{\boldsymbol{\vartheta}}_f^{(k)} + \delta\boldsymbol{\vartheta}_f^{(k)}\right) - \boldsymbol{\vartheta}_{f,\mathrm{ref}}\right|_i \leq \boldsymbol{\varepsilon}_{\boldsymbol{\vartheta}_i}, \quad i = 1,\ldots,m, \tag{21c}$$

$$\left|\left(\bar{t}_f^{(k)} + \delta t_f^{(k)}\right) - t_{p,\mathrm{ref}}\right| \leq \varepsilon_{t_f,\mathrm{targ}}, \tag{21d}$$

where the notation $|\cdot|_i$ in constraint (21c) denotes the absolute value of the i^{th} component of the vector inside the absolute value operator. Constraints (21c) and (21d) can be written in standard form as

$$-\begin{bmatrix}\boldsymbol{B}_{\mathcal{M}}^{(k)} & \boldsymbol{C}_{\mathcal{M}}^{(k)}\end{bmatrix}\begin{bmatrix}\Delta\boldsymbol{v}^{(k)} \\ \delta t_f^{(k)}\end{bmatrix} \leq \boldsymbol{\varepsilon}_{\boldsymbol{\vartheta},\mathrm{targ}} + \left(\bar{\boldsymbol{\vartheta}}_f^{(k)} - \boldsymbol{\vartheta}_{f,\mathrm{ref}}\right), \tag{22}$$

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

$$-\delta t_f^{(k)} \leq \varepsilon_{t_f,\mathrm{targ}} + \left(t_{f,0} - t_{p,\mathrm{ref}}\right), \tag{24}$$

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

After solving problem (21), both $\Delta\boldsymbol{v}^{(k)}$ and $\delta t_f^{(k)}$ are used to update $\boldsymbol{\theta}_0$ and $t_{f,0}$ respectively, such that

$$t_{f,0}^{(k+1)} = \bar{t}_f^{(k)} + \Delta t_f^{(k)}, \tag{26a}$$

$$\boldsymbol{\theta}_0^{(k+1)} = \boldsymbol{\theta}_0^{(k)} + \begin{bmatrix}\mathbf{0}_{3\times 1} \\ \Delta\boldsymbol{v}^{(k)}\end{bmatrix}, \tag{26b}$$

where the superscript $(\cdot)^{(k)}$ and $(\cdot)^{(k+1)}$ denote the k^{th} and $(k+1)^{\mathrm{th}}$ iteration solving the linearized problem (21) respectively. Then, matrices $\boldsymbol{B}_{\mathcal{M}}^{(k+1)}$ and $\boldsymbol{C}_{\mathcal{M}}^{(k+1)}$ are recomputed using equations (19) and (20). 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:

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

Algorithm 1 Phase-constrained sequential second-order cone program

```
Inputs: $t_0$, $\boldsymbol{\theta}_0$, $N$, $\boldsymbol{\vartheta}_{f,\mathrm{ref}}$, $k_{\max}$, $\boldsymbol{\varepsilon}_{\boldsymbol{\vartheta},\mathrm{targ}}$
Outputs: $convflag$, $\Delta\boldsymbol{v}$
$convflag \leftarrow 0$                                        ▷ 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 $1, \dots, k_{\max}$ 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($t_0, \bar{t}_f^{(k)}, \boldsymbol{\theta}_0^{(k)}$)
    $\boldsymbol{T}_{\mathrm{EM}}^{\mathrm{Inr}} \leftarrow$sxform($\mathcal{F}_{\mathrm{Inr}}, \mathcal{F}_{\mathrm{EM}}, \bar{t}_f^{(0)}$)          ▷ SPICE function
    $\bar{\boldsymbol{\vartheta}}_f^{(k)} \leftarrow \boldsymbol{T}_{\mathrm{EM}}^{\mathrm{Inr}} \bar{\boldsymbol{\theta}}_f^{(k)}$
    if $\bar{\boldsymbol{\vartheta}}_f^{(k)} \leq \boldsymbol{\varepsilon}_{\boldsymbol{\vartheta},\mathrm{targ}}$ then
        $convflag \leftarrow 1$
        break
    end if
    $\boldsymbol{B}_{\mathcal{M}}^{(k)}, \boldsymbol{C}_{\mathcal{M}}^{(k)} \leftarrow$ eqn. (19), (20)
    $\Delta\boldsymbol{v}^{(k)}, \delta t_f^{(k)} \leftarrow$ SOCP($\bar{\boldsymbol{\vartheta}}_f^{(k)}, \boldsymbol{B}_{\mathcal{M}}^{(k)}, \boldsymbol{C}_{\mathcal{M}}^{(k)}, 0.9\boldsymbol{\varepsilon}_{\boldsymbol{\vartheta},\mathrm{targ}}$)
    $\Delta\boldsymbol{v} \leftarrow \Delta\boldsymbol{v} + \Delta\boldsymbol{v}^{(k)}$                  ▷ Update cumulative $\Delta\boldsymbol{v}$
end for
```

- `sxform`($\mathcal{F}_{\mathrm{Inr}}, \mathcal{F}_{\mathrm{EM}}, \bar{t}_f^{(0)}$) generates the transformation matrix from $\mathcal{F}_{\mathrm{Inr}}$ to $\mathcal{F}_{\mathrm{EM}}$ at epoch $\bar{t}_f^{(0)}$); and

- `SOCP`($\bar{\boldsymbol{\vartheta}}_f^{(k)}, \boldsymbol{B}_{\mathcal{M}}^{(k)}, \boldsymbol{C}_{\mathcal{M}}^{(k)}, \boldsymbol{\varepsilon}_{\boldsymbol{\vartheta},\mathrm{targ}}$) builds and solves problem (21) through the use of a SOCP solver, such as ECOS[26] or SCS.[27–29]

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 (10); thus, in essence, the PC-SCoP approach replaces the Newton-Raphson update from equation (11) 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, which is necessitated by both approaches. Thus, the PC-SCoP does not result in significant increase in terms of computational cost compared to differential correction-based approaches.

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

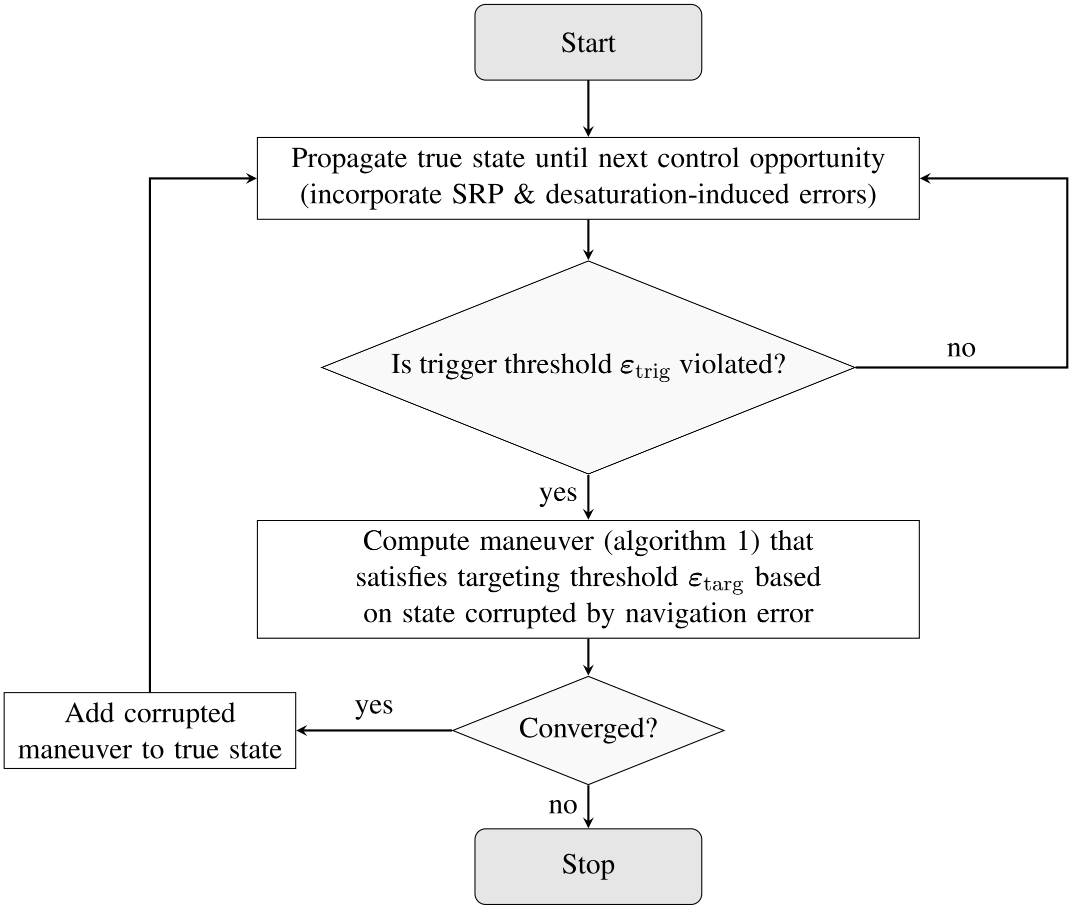

Start
Propagate true state until next control opportunity
(incorporate SRP & desaturation-induced errors)
Is trigger threshold $\varepsilon_{\mathrm{trig}}$ violated?
no
yes
Compute maneuver (algorithm 1) that
satisfies targeting threshold $\varepsilon_{\mathrm{targ}}$ based
on state corrupted by navigation error
Converged?
yes
Add corrupted
maneuver to true state
no
Stop

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

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 bring to the readers' attention the use of two thresholds, one for checking whether a maneuver is warranted, coined the trigger threshold $\varepsilon_{\mathrm{trig}}$, and another which serves as the upper bound threshold for the targeting control scheme, $\varepsilon_{\mathrm{targ}}$. We study the effect of choosing $\varepsilon_{\mathrm{trig}} > \varepsilon_{\mathrm{targ}}$ instead of setting $\varepsilon_{\mathrm{trig}} = \varepsilon_{\mathrm{targ}}$; the motivation for the former is to ensure the computed maneuvers, even under the presence of uncertainties, delivers a maneuver that doesn't immediately violate the trigger condition. 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 $\varepsilon_{\mathrm{targ}}$, we require the maneuver magnitude to be within a predefined maximum executable maneuver magnitude, denoted as Δv_{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.

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 are introduced.

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 $\sigma_{\overline{(A/m)}}$.

$$C_R = C_{R,0}\left(1 + \delta\bar{C}_R\right), \quad \delta\bar{C}_R \sim \mathcal{N}(0, \sigma_{\bar{C}_R}^2), \tag{27}$$

$$(A/m) = (A/m)_0\left(1 + \delta\overline{(A/m)}\right), \quad \delta\overline{(A/m)} \sim \mathcal{N}(0, \sigma_{\overline{(A/m)}}^2). \tag{28}$$

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

$$\boldsymbol{v}^+ = \boldsymbol{v}^- + \delta v_{\text{desat}}\hat{\boldsymbol{i}}, \quad \delta v_{\text{desat}} \sim \mathcal{N}\left(0, \sigma_{\delta v_{\text{desat}}}^2\right), \tag{29}$$

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

$$\hat{\boldsymbol{i}} = \frac{1}{\sqrt{i_x^2 + i_y^2 + i_z^2}}\begin{bmatrix} i_x \\ i_y \\ i_z \end{bmatrix}, \quad i_x, i_y, i_z \sim \mathcal{U}(-1, 1). \tag{30}$$

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

$$\hat{\boldsymbol{r}} = \boldsymbol{r} + \delta\boldsymbol{r}_{\text{nav}}, \quad \delta\boldsymbol{r}_{\text{nav}} \sim \mathcal{N}(0, \sigma_{\boldsymbol{r}}^2), \tag{31a}$$

$$\hat{\boldsymbol{v}} = \boldsymbol{r} + \delta\boldsymbol{v}_{\text{nav}}, \quad \delta\boldsymbol{v}_{\text{nav}} \sim \mathcal{N}(0, \sigma_{\boldsymbol{v}}^2). \tag{31b}$$

Maneuver Execution Error The actual maneuver that is executed cannot exactly match the output of the control algorithm due to actuation errors. 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

$$\boldsymbol{v}^+ = \boldsymbol{v}^- + \Delta\boldsymbol{v}(\Delta\bar{\boldsymbol{v}}, \delta\tilde{\boldsymbol{v}}_{\text{abs}}, \delta\tilde{\boldsymbol{v}}_{\text{rel}}, \delta\phi), \tag{32}$$

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, $\delta\tilde{\boldsymbol{v}}_{\text{rel}}$, an absolute magnitude error, $\delta\tilde{\boldsymbol{v}}_{\text{abs}}$, and a direction error, $\delta\phi$. The relative magnitude error is modeled by a relative standard deviation $\sigma_{\delta v_{\text{rel}}}$, such that

$$\delta\tilde{\boldsymbol{v}}_{\text{rel}} = \|\Delta\bar{\boldsymbol{v}}\|\delta v_{\text{rel}}\hat{\boldsymbol{i}}, \quad \delta v_{\text{rel}} \sim \mathcal{N}\left(0, \sigma_{\delta v_{\text{rel}}}^2\right), \tag{33}$$

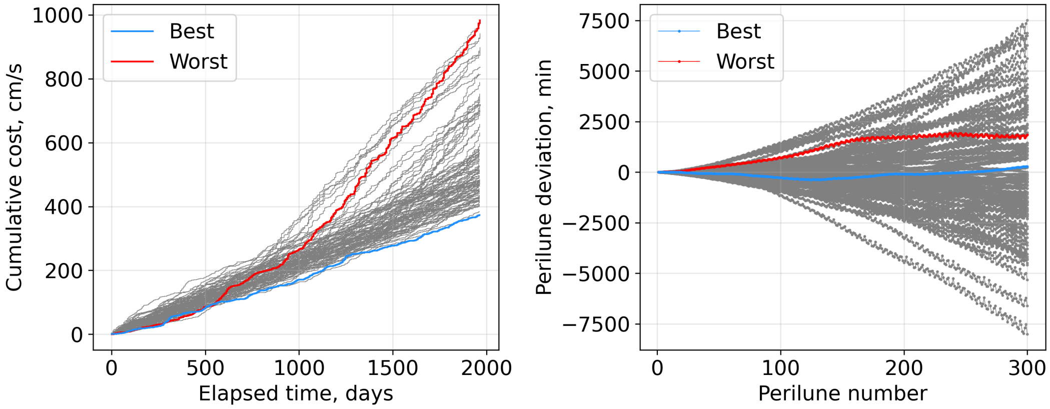


Figure 4: Monte-Carlo results over 300 revolutions using differential correction with no phase constraint

where $\hat{\boldsymbol{i}}$ follows the definition from (30), but realized separately. The absolute magnitude error is modeled by a standard deviation $\sigma_{\delta v_{\text{abs}}}$, such that

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

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

$$\boldsymbol{T}(\delta\phi) = \cos(\delta\phi)\boldsymbol{I}_3 + \sin(\delta\phi)\boldsymbol{i}^\times + [1 - \cos(\delta\phi)]\boldsymbol{i}\boldsymbol{i}^T, \quad \delta\phi \sim \mathcal{N}(0, \sigma_\phi^2), \tag{35}$$

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

$$\Delta\boldsymbol{v} = \boldsymbol{T}(\delta\phi)\left[\Delta\bar{\boldsymbol{v}} + \delta\tilde{\boldsymbol{v}}_{\text{rel}} + \delta\tilde{\boldsymbol{v}}_{\text{abs}}\right], \tag{36}$$

and is substituted back into equation (32).

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.[24] Table 3 summarizes the station-keeping cost obtained from the various controllers and their chosen parameters. For comparison, Davis et al[1] reports a mean annual cost of $134\,\text{cm/s}$ with a range varying from $120\,\text{cm/s}$ and $160\,\text{cm/s}$. We then conduct a separate Monte-Carlo experiment for 600 revolutions, taking the best-performing parameters from the former experiment.

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

Table 1: Parameters for numerical simulation

Simulation Parameter	Value
Simulation duration, number of revolutions	300 / 500
Maneuver location true anomaly, deg	200
SRP area relative error 3-σ ($3\sigma_{\bar{C}_R}$), $\%$	30
SRP reflection pressure coefficient relative error 3-σ ($3\sigma_{\overline{(A/m)}}$), $\%$	15
Desaturation-induced velocity error 3-σ ($3\sigma_{\delta v_{\mathrm{desat}}}$), $\mathrm{cm/s}$	1.0
Desaturation location true anomaly, deg	340, 350, 10, 190
Navigation error position 3-σ ($3\sigma_{\boldsymbol{r}}$), km	1.5
Navigation error velocity 3-σ ($3\sigma_{\boldsymbol{v}}$), $\mathrm{cm/s}$	0.8
Maneuver execution relative magnitude error 3-σ ($3\sigma_{\delta v_{\mathrm{rel}}}$), $\%$	1.5
Maneuver execution absolute magnitude error 3-σ ($3\sigma_{\delta v_{\mathrm{abs}}}$), $\mathrm{mm/s}$	1.42
Maneuver execution direction error 3-σ ($3\sigma_{\phi}$), deg	1
Maximum maneuver magnitude $\Delta v_{\max}$, $\mathrm{m/s}$	1.0

Table 2: Parameters for controllers

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

Table 3: PC-SCoP performances of yearly cost Δv_{yr} from 300-revolutions Monte-Carlo experiment

Controller	$v_{x/z,\mathrm{targ}}$, $\mathrm{m/s}$	$t_{p,\mathrm{targ}}$, min	Success, $\%$	Δv_{yr}, $\mathrm{cm/s}$		
				Mean	95^{th}-$\%$	1-σ
DC	20	n.a.	100	103.59	164.58	27.56
PC-SCoP	20	20	100	118.53	134.87	9.83
	5	20	100	103.27	117.45	9.10
	5	10	97	136.34	209.53	38.66

In all cases, $v_{x/z,\mathrm{trig}} = 20$ $\mathrm{m/s}$ and $t_{p,\mathrm{trig}} = 20$ min (for PC-SCoP only) are used.

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 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 $\varepsilon_{\mathrm{trig}}$ and $\varepsilon_{\mathrm{targ}}$. In the first two cases, where $t_{p,\mathrm{targ}} = 20$ min is used, the perilune epoch deviation initially undergoes secular growth but becomes bounded within approximately the targeting tolerance $t_{p,\mathrm{targ}}$ chosen for the entire duration of the simulation. In contrast, using a higher $t_{p,\mathrm{targ}} = 10$ min results in instability occurring by around the 200^{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.

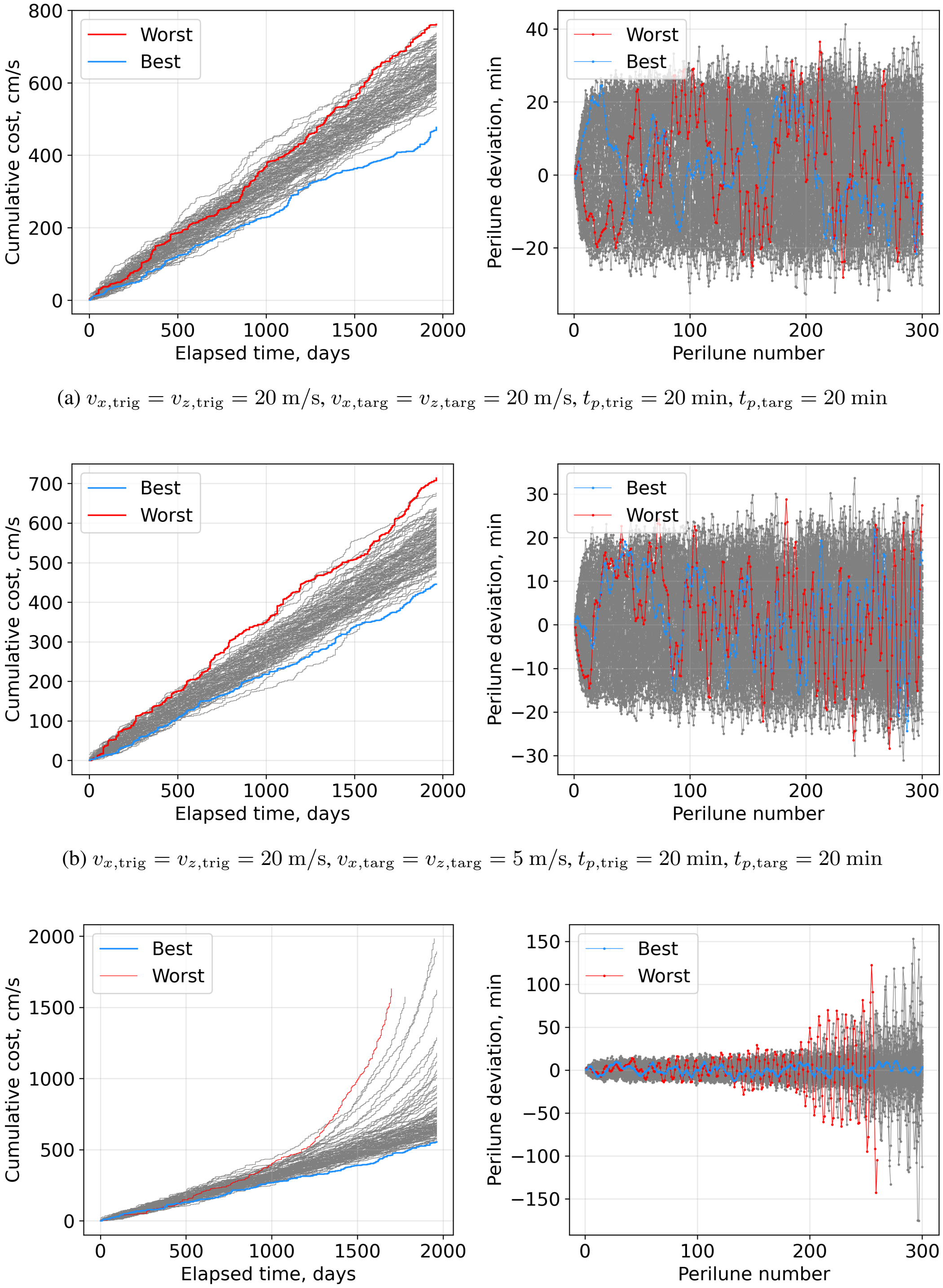
Focusing now on the first two cases, we can see that choosing a tighter state targeting threshold $\varepsilon_{\boldsymbol{\vartheta},\mathrm{targ}}$, as is done in case (b), results in reduced cumulative cost compared to (a), where $\varepsilon_{\mathrm{trig}} = \varepsilon_{\mathrm{targ}}$ is used. The difference in performance between (a) and (b) highlights the importance of choosing a tighter $\varepsilon_{\mathrm{targ}}$: as hypothesized, the resulting maneuvers are less likely to immediately violate the triggering tolerance upon execution even under the modeled errors, thus resulting in lower cumulative station-keeping costs.

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 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 six state of the baseline at some point, thus attenuating the error in perilune deviation.

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 $v_{x,\mathrm{trig}} = v_{z,\mathrm{trig}} = 20$ m/s, $v_{x,\mathrm{targ}} = v_{z,\mathrm{targ}} = 5$ m/s, and $t_{p,\mathrm{trig}} = 20$ min, $t_{p,\mathrm{targ}} = 20$ min. Here, we observe a success rate of 95%, with a mean yearly cost of 123.74 cm/s and a standard deviation of 22.77 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^{th} day.

In practice, it is unlikely for a spacecraft on 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 7(a), a check every 100 revolutions would be sufficient to

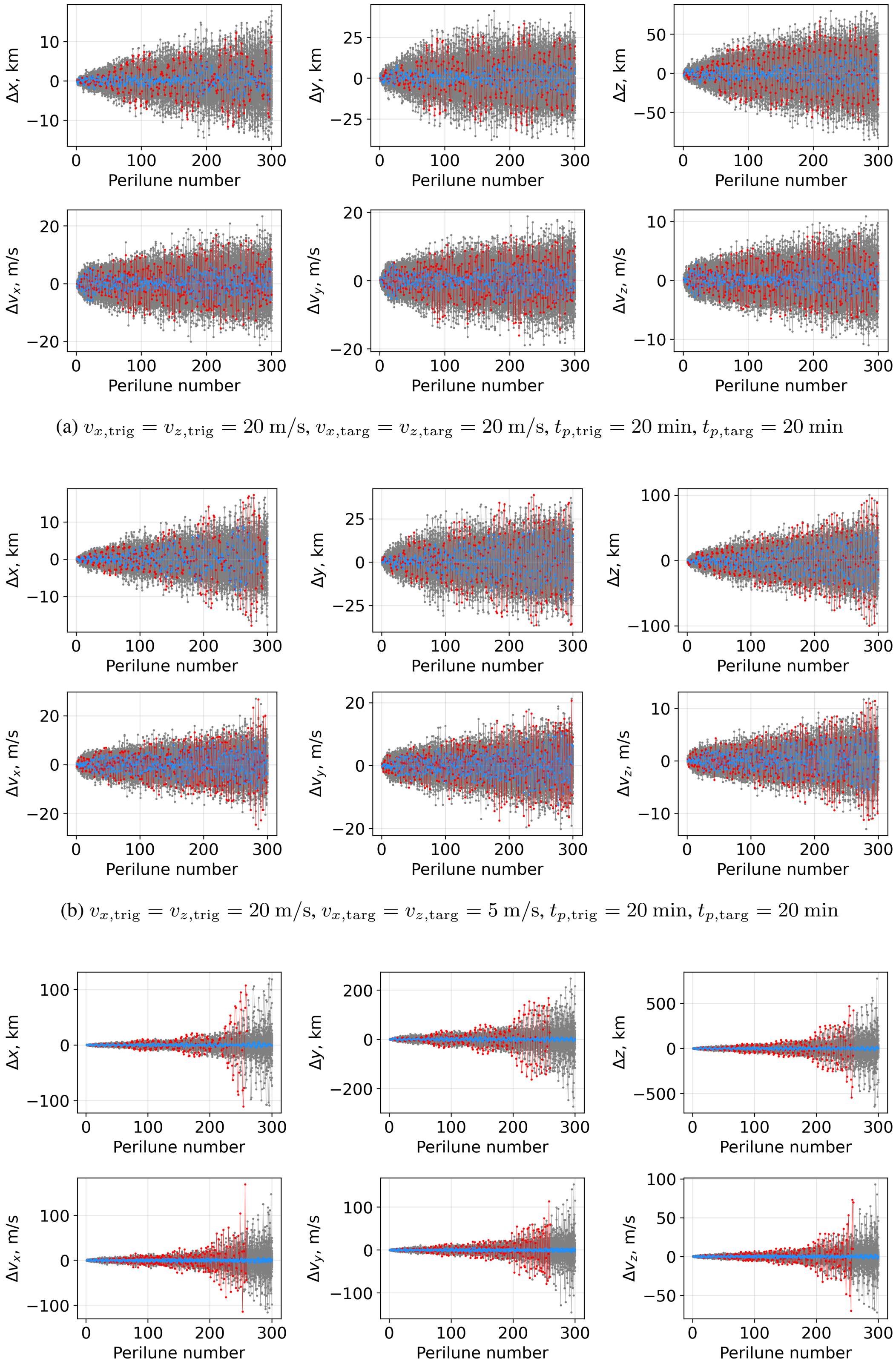


(a) $v_{x,\mathrm{trig}} = v_{z,\mathrm{trig}} = 20\ \mathrm{m/s}$, $v_{x,\mathrm{targ}} = v_{z,\mathrm{targ}} = 20\ \mathrm{m/s}$, $t_{p,\mathrm{trig}} = 20\ \mathrm{min}$, $t_{p,\mathrm{targ}} = 20\ \mathrm{min}$

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

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

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

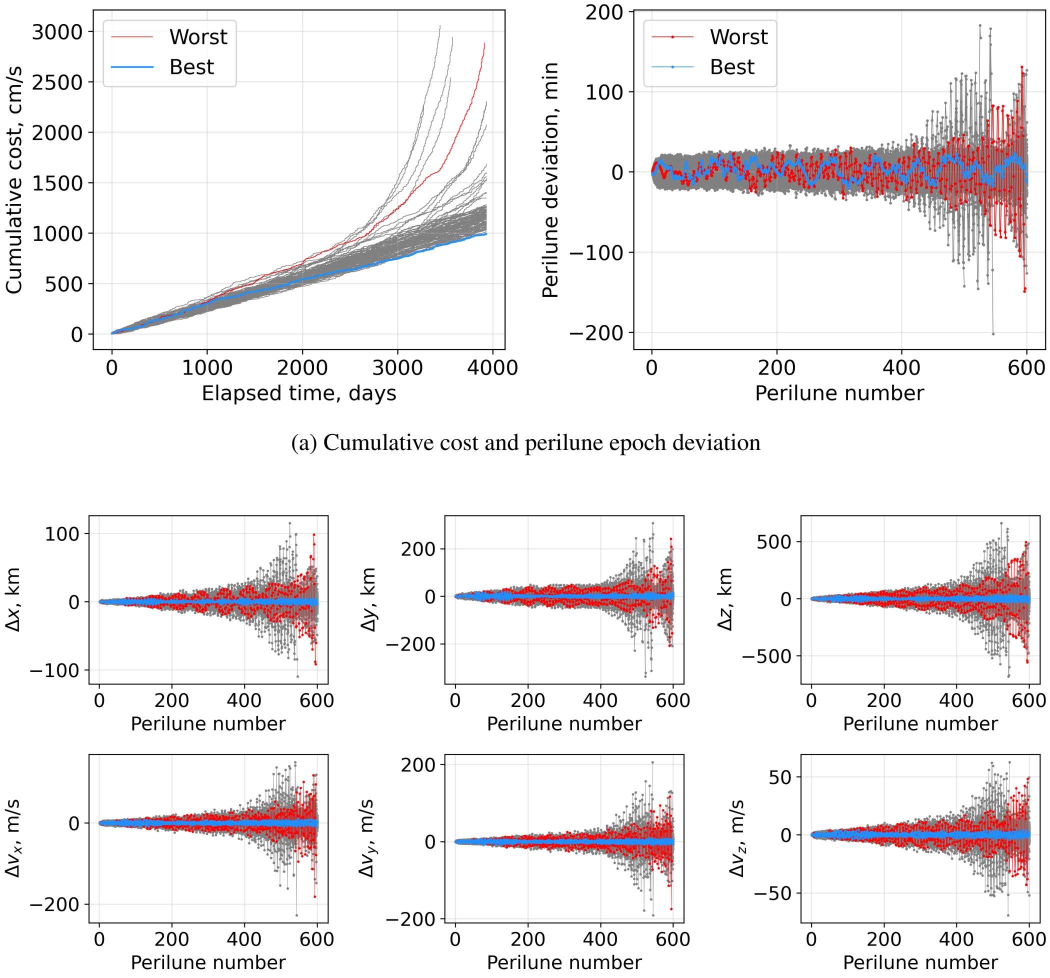


(a) $v_{x,\mathrm{trig}} = v_{z,\mathrm{trig}} = 20$ m/s, $v_{x,\mathrm{targ}} = v_{z,\mathrm{targ}} = 20$ m/s, $t_{p,\mathrm{trig}} = 20$ min, $t_{p,\mathrm{targ}} = 20$ min

(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

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.

CONCLUSION

In this work, we proposed an optimization-based station-keeping algorithm for maintaining the spacecraft in the vicinity of a baseline cislunar LPO, while also ensuring the phase along the LPO is tracked. The proposed algorithm, coined as the PC-SCoP is inspired by the xz-plane crossing control technique, which has widely been studied and used in combination with a differential correction algorithm; instead, this work poses the xz-plane crossing problem as a SOCP with linearized dynamics and inequality constraints. The SOCP is sequentially solved, each time re-linearizing the dynamics about the current state based on the incremental correction obtained by 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, 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 full-ephemeris dynamics, incorporating realistic sources of error such as navigation error, control execution error, 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 experienced a secular growth, but this does not prohibit its use for autonomous station-keeping over the course of 300 revolutions. 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 xz-plane crossing control approaches.

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 θ - D technique,” *Journal of the Astronautical Sciences*, Vol. 56, No. 2, 2008, pp. 231–250, 10.1007/BF03256550.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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,” 2014, 10.1016/j.cnsns.2014.03.026.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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,” 2018.

[22] 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.

[23] 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.

[24] D. E. Lee, “Gateway Destination Orbit Model: A Continuous 15 Year NRHO Reference Trajectory,” tech. rep., 2019.

[25] 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.

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

[27] 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.

[28] B. O’Donoghue, “Operator Splitting for a Homogeneous Embedding of the Linear Complementarity Problem,” *SIAM Journal on Optimization*, Vol. 31, August 2021, pp. 1999–2023.

[29] B. O’Donoghue, E. Chu, N. Parikh, and S. Boyd, “SCS: Splitting Conic Solver, version 3.2.4,” `https://github.com/cvxgrp/scs`, Nov. 2023.