Robust Trajectory Optimization Techniques Using a Sweeping Gradient Method and Linear Covariance Analysis

Benjamin W. L. Margolis
NASA Ames Research Center, Moffett Field, CA, 94035

David Woffinden
NASA Johnson Space Center, Houston TX, 77058

We present robust trajectory optimization techniques using a sweeping gradient method for ordinary differential equations with events (SGM) and linear covariance analysis (LinCov). SGM is a method for computing the gradient of trajectory analyses defined by performance indices over initial value problems with events with respect to static parameters. LinCov is an analytic technique for predicting stochastic behavior of dynamical systems. By combining SGM and LinCov, it is possible use efficient, off-the-shelf, gradient-based optimizers to solve robust optimal trajectory design problems. We describe the individual methods and some details on how they can be combined. Then we apply the combined techniques to a variety of orbital trajectory design problems to demonstrate its use, including minimum fuel transfer and mid-course correction burn scheduling.

I. Introduction

Incorporating the impacts of system uncertainty such as navigation errors, disturbance accelerations, burn execution biases and misalignments, orbit insertion dispersions, and a host of other influential factors into the design of a trajectory profile to make it robust to these unknown perturbations is becoming more feasible, particularly with the maturation of linear covariance analysis (LinCov)[1, 2] for closed-loop GN&C systems. LinCov is an analytic technique for predicting stochastic behavior of dynamical systems. Previous studies interfaced a genetic algorithm (GA) with a linear covariance analysis tool to provide the initial demonstration of applying this non-traditional approach to robust trajectory optimization. These robust trajectory design techniques were originally developed for rendezvous applications in low Earth orbit[3]. It was then extended to cis-lunar outbound trajectories to a Near Rectilinear Halo Orbit (NRHO)[4] and introduced for a simple rendezvous approach trajectory in the NRHO for mid-course correction placement[5]. Recently, these robust trajectory optimization principles have been applied to solve cislunar transfers to low-lunar orbit[6], NRHO rendezvous and docking[7], angles-only navigation for NRHO rendezvous [8], lunar powered descent and landing[9], along with Mars aerocapture[10] problems. They were also utilized to determine the optimized trajectory correction burn placement for the upcoming Artemis II free-return cis-lunar trajectory profile[11] and exercised to determine the optimized trajectory correction burn placement for NRHO orbit maintenance anticipated for the Artemis III mission[12].

This paper presents an alternate strategy and a set of robust trajectory optimization techniques that use a sweeping gradient method for ordinary differential equations with events (SGM) and linear covariance analysis. SGM is a method for computing the gradient of trajectory analyses defined by performance indices over initial value problems with events with respect to static parameters[13]. SGM allows analysts to model a trajectory analysis as a black-box mapping $\mathbf{T}\colon \mathbb{R}^{n_\theta} \mapsto \mathbb{R}^{n_J}$, where n_θ is the dimension of a parameter space and n_J is the dimension of the *trajectory analysis output* space, and enables the construction of optimization problems of the form

$$\begin{aligned} \underset{\boldsymbol{\theta}}{\text{minimize}} \quad & \Gamma_0\left(\boldsymbol{J}, \boldsymbol{\theta}\right) \\ \text{subject to} \quad & \Gamma_i\left(\boldsymbol{J}, \boldsymbol{\theta}\right) \leq 0, \qquad i = 1, \ldots, m \\ \text{where} \quad & \boldsymbol{J} = \mathbf{T}\left(\boldsymbol{\theta}\right) \end{aligned} \tag{1}$$

where Γ_0 is an objective function and the Γ_i are constraint functions. Since SGM provides an efficient method for computing the gradients of $\boldsymbol{J}$ with respect to the parameters $\boldsymbol{\theta}$, application of the chain rule enables the computation of the gradient of the objective and constraint functions with respect to the trajectory model parameters. The availability of these gradients permits the use of efficient, off-the-shelf gradient-based solvers for solving optimal control or system design problems of the form (1). By combining SGM and LinCov, these gradient-based optimizers can be used to solve robust optimal trajectory design problems with minimal additional effort. The individual methods and details on how they can be combined is the focus of this research. To illustrate how these combined techniques can be applied, a variety of orbital rendezvous trajectory design problems are solved, including a minimum fuel transfer problem and mid-course correction burn scheduling. Several examples also highlight the limitations of this approach.

Other trajectory optimization techniques have also been used for orbital rendezvous design, including direct collocation methods[14] and convexification[15], which may be considered a special case of direct collocation, and multiple-shooting methods[16]. Collocation methods are often able to converge more rapidly than shooting methods like SGM, particularly for the final convergence near the optimum. However, a number of modeling limitations motivates the use for SGM. In particular, the fixed ordering of trajectory segments is not well suited for certain features of robust orbital rendezvous trajectory modeling such as free-drift safety performance or the search for measurement and maneuver execution times. Methods like hp mesh adaptation may be able to address this with potential computational cost[17]. Convexification methods can be applied to the dynamic nonlinearities[18] in these trajectories but may require significant offline computation before the optimization can be applied. In contrast, modeling the robust orbital rendezvous trajectories as a system of ODEs with events allows the maneuver plan and a variety of relevant performance criteria to be modeled naturally as illustrated in the examples. Gradient-free methods like GA can also be applied with similar modeling approaches and can fully explore a design space without becoming stuck in local optima but lack the efficiency and accuracy of gradient-based methods. These methods exist on a continuum from most efficient to most robust and flexible; we believe the methods presented in this work provide a useful trade-off between performance and flexibility.

The paper is organized as follows. Section II outlines the robust trajectory optimization techniques which includes defining the performance metrics, describing the fundamental linear covariance analysis equations, formulating the sweeping gradient method equations, and combining both SGM and LinCov. Section III provides an overview of the scenario and analysis inputs used for the robust trajectory optimization demonstration which includes the system dynamics models, GN&C system modeling, and the core inputs to utilize SGM and LinCov. Section IV illustrates the proposed robust trajectory optimization techniques using SGM and LinCov with five examples that build on complexity starting with a simple Hohmann transfer scenario that is then expanded to include mid-course correction burns, navigation measurement scheduling, burn magnitude and execution error, initial condition dispersions, and other factors. A conclusion summary is provided in Section V.

II. Robust Trajectory Optimization Techniques

A. Performance Metrics

To develop the techniques of robust trajectory optimization using a sweeping gradient method and linear covariance analysis, there are several performance metrics that must be defined which include the true trajectory dispersions $\delta\mathbf{x}$, the navigation dispersions $\delta\hat{\mathbf{x}}$, the true navigation error $\delta\mathbf{e}$, and the onboard navigation error $\delta\hat{\mathbf{e}}$ as depicted in Figure 1.

The true dispersions $\delta\mathbf{x}$ are defined as the difference between the true state $\mathbf{x}$ and the nominal state $\bar{\mathbf{x}}$. The true state $\mathbf{x}$ is an n-dimensional vector that represents the *real world* environment or the *realization* of the actual state.

$$\delta\mathbf{x} \triangleq \mathbf{x} - \bar{\mathbf{x}} \qquad\qquad \mathbf{D} = E\left[\delta\mathbf{x}\delta\mathbf{x}^{\mathrm{T}}\right] \tag{2}$$

The nominal state $\bar{\mathbf{x}}$ is also an n-dimensional vector that represents the desired or reference state. The covariance of the environment dispersions, $\mathbf{D}$, indicates how precisely the system can follow a desired trajectory.

The navigation dispersions $\delta\hat{\mathbf{x}}$ are defined as the difference between the navigation state $\hat{\mathbf{x}}$ and the nominal state $\bar{\mathbf{x}}$. The navigation state is an $\hat{n}$-dimensional vector ($\hat{n} < n$) that represents the filter's estimated state.

$$\delta\hat{\mathbf{x}} \triangleq \hat{\mathbf{x}} - \mathbf{M}_x\bar{\mathbf{x}} \qquad\qquad \hat{\mathbf{D}} = E\left[\delta\hat{\mathbf{x}}\delta\hat{\mathbf{x}}^{\mathrm{T}}\right] \tag{3}$$

The matrix $\mathbf{M}_x$ is an $(\hat{n} \times n)$ matrix that maps the estimated state in terms of the true and nominal state. The covariance of the navigation dispersions, $\hat{\mathbf{D}}$, reflects how precisely the onboard system thinks it can follow a prescribed reference trajectory.

The true navigation error $\delta\mathbf{e}$ is the difference between the environment and navigation states. It is also the difference between the environment and the navigation dispersions.

$$\delta\mathbf{e} \triangleq \mathbf{M}_x\mathbf{x} - \hat{\mathbf{x}} = \mathbf{M}_x\delta\mathbf{x} - \delta\hat{\mathbf{x}} \qquad\qquad \mathbf{P} = E\left[\delta\mathbf{e}\delta\mathbf{e}^{\mathrm{T}}\right] \tag{4}$$

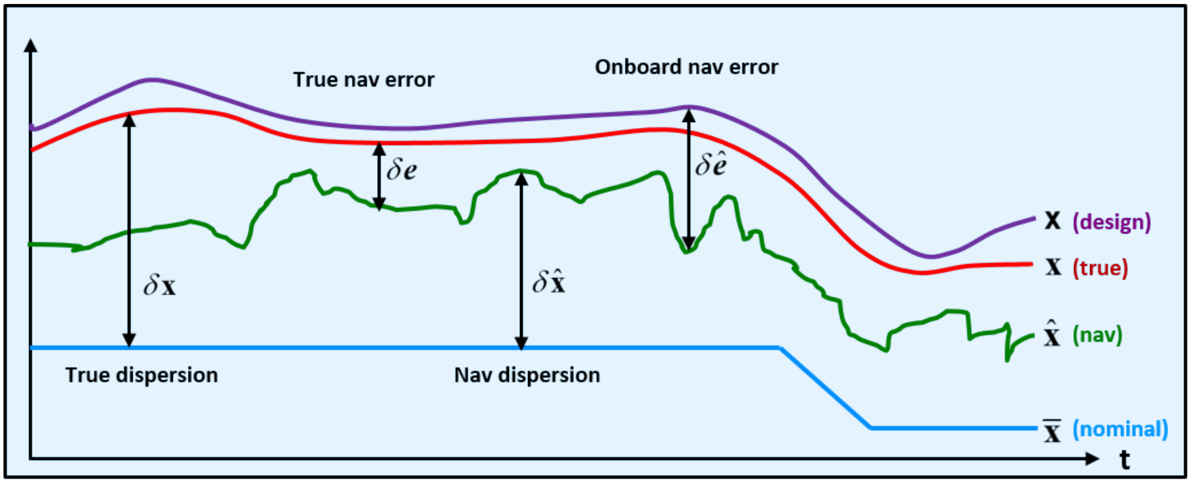


Fig. 1 GN&C Performance Metric Variables

The covariance of the true navigation error, $\mathbf{P}$, quantifies how precisely the onboard navigation system can estimate the actual state.

The onboard navigation error $\delta\hat{\mathbf{e}}$ itself is never computed, but it is used to develop the onboard navigation filter equations. It is defined as the difference between the design state, $\boldsymbol{\mathsf{x}}$, and the navigation state $\hat{\mathbf{x}}$.

$$\delta\hat{\mathbf{e}} \triangleq \boldsymbol{\mathsf{x}} - \hat{\mathbf{x}} \qquad\qquad \hat{\mathbf{P}} = E\left[\delta\hat{\mathbf{e}}\delta\hat{\mathbf{e}}^{\mathrm{T}}\right] \tag{5}$$

The covariance of the onboard navigation error, $\hat{\mathbf{P}}$, quantifies how precisely the onboard navigation system expects it can determine the actual state. The performance of the onboard navigation system is determined by comparing $\hat{\mathbf{P}}$ to the actual navigation performance $\mathbf{P}$. If the *true* states $\mathbf{x}$ and the *design* states $\boldsymbol{\mathsf{x}}$ are assumed to be the same, then the true navigation covariance will equal the onboard navigation covariance.

The covariances of the true dispersions, navigation dispersions, true navigation error, and the onboard navigation error are ultimately used to analyze and assess the performance of a proposed GN&C system. A common approach to obtain these performance metrics is to use a Monte Carlo simulation as outlined in Figure 2, where the sample statistics of hundreds or thousands of runs, N, are used to numerically compute the desired covariance matrices.

$$\mathbf{D} \approx \frac{1}{N-1}\sum \delta\mathbf{x}\delta\mathbf{x}^{\mathrm{T}} \qquad \hat{\mathbf{D}} \approx \frac{1}{N-1}\sum \delta\hat{\mathbf{x}}\delta\hat{\mathbf{x}}^{\mathrm{T}} \qquad \mathbf{P} \approx \frac{1}{N-1}\sum \delta\mathbf{e}\delta\mathbf{e}^{\mathrm{T}} \tag{6}$$

The onboard navigation error covariance $\hat{\mathbf{P}}$ is the navigation filter covariance for each run. This same statistical information can be obtained using linear covariance analysis techniques.

B. Linear Covariance Analysis

In comparison to the hundreds or thousands of runs typically required for Monte Carlo analaysis, LinCov utilizes augmented and onboard state covariance matrices to produce the statistical performance metrics of a closed-loop GN&C system in a single-run. Linear covariance analysis incorporates the non-linear system dynamics models and the non-linear GN&C algorithms to generate a nominal reference trajectory $\bar{\mathbf{x}}$ which is then used to propagate, update, and correct an onboard navigation covariance matrix $\hat{\mathbf{P}}$ and an augmented state covariance matrix $\mathbf{C}$,

$$\mathbf{C} = E\left[\delta\mathbf{X}\delta\mathbf{X}^{\mathrm{T}}\right] \tag{7}$$

where the augmented state $\delta\mathbf{X}^{\mathrm{T}} = [\delta\mathbf{x}^{\mathrm{T}}\ \delta\hat{\mathbf{x}}^{\mathrm{T}}]$ consists of the true dispersions and the navigation dispersions. Pre- and post-multiplying the augmented state covariance matrix by the following mapping matrices results in the covariances for

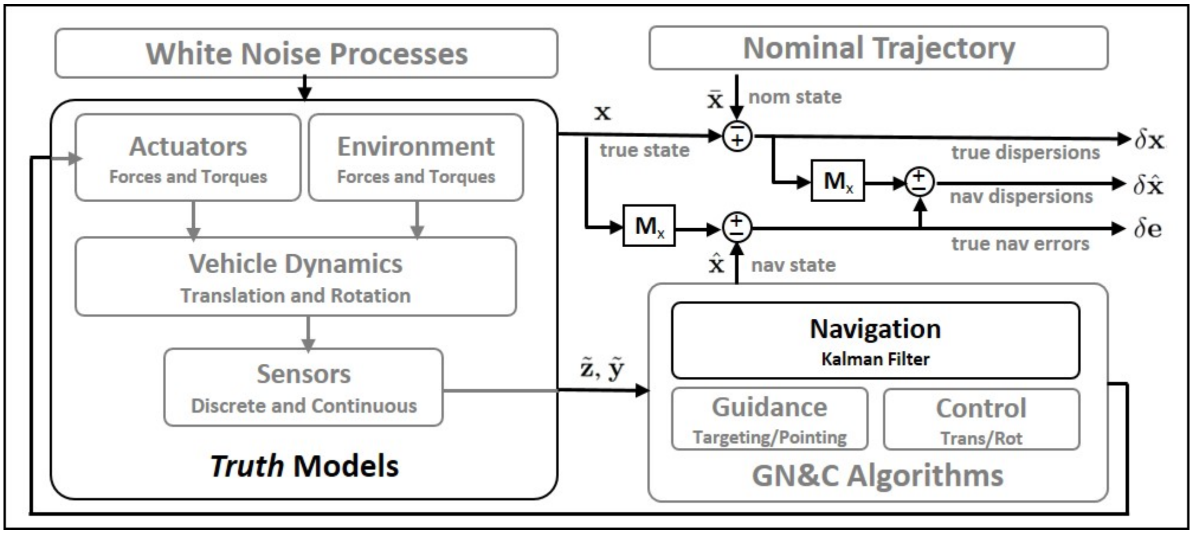


Fig. 2 Extracting GN&C Performance Metrics Using Monte Carlo Techniques

the trajectory dispersions, navigation dispersions, and the navigation error:

$$\begin{aligned}\mathbf{D} &= [\ \mathbf{I}_{n\times n},\ \mathbf{0}_{n\times\hat{n}}\ ]\,\mathbf{C}\,[\ \mathbf{I}_{n\times n},\ \mathbf{0}_{n\times\hat{n}}\ ]^{\mathrm{T}}\\ \hat{\mathbf{D}} &= [\ \mathbf{0}_{\hat{n}\times n},\ \mathbf{I}_{\hat{n}\times\hat{n}}\ ]\,\mathbf{C}\,[\ \mathbf{0}_{\hat{n}\times n},\ \mathbf{I}_{\hat{n}\times\hat{n}}\ ]^{\mathrm{T}}\\ \mathbf{P} &= [\ \mathbf{I}_{\hat{n}\times n},\ -\mathbf{I}_{\hat{n}\times\hat{n}}\ ]\,\mathbf{C}\,[\ \mathbf{I}_{\hat{n}\times n},\ -\mathbf{I}_{\hat{n}\times\hat{n}}\ ]^{\mathrm{T}}\end{aligned} \tag{8}$$

For details regarding the development and implementation of the linear covariance simulation, see the following references[2, 19].

1. Truth Models

The truth models governing the system dynamics, both continuous and discrete sensor measurements, and impulsive state corrections are provided in context of establishing the core LinCov equations. The true state or environment system is governed by the state equation

$$\dot{\mathbf{x}} = \mathbf{f}\,(\mathbf{x}, \hat{\mathbf{u}}, t) + \mathbf{w} \tag{9}$$

where $\mathbf{x}$ are the true states with dimension n, $\hat{\mathbf{u}}$ are the actuator commands with dimension $n_{\hat{u}}$, and $\mathbf{w}$ is the zero-mean white-noise with covariance $\mathbf{S}_w\,(t)\,\delta\,(t-t')$. If continuous measurements such as gyros and accelerometers are utilized, the measured angular rates and accelerations are captured with

$$\tilde{\boldsymbol{y}} = \mathbf{c}\,(\mathbf{x}, t) + \boldsymbol{\eta}$$

where $\boldsymbol{\eta}$ is the zero-mean white noise with covariance $\mathbf{S}_\eta\,(t)\,\delta\,(t-t')$, and the discrete measurement equations are

$$\begin{aligned}\Delta\tilde{\mathbf{y}}_j &= \Delta\mathbf{c}\left(\mathbf{x}_j, t_j\right) + \Delta\boldsymbol{\eta}_j\\ \tilde{z}_k &= \mathbf{h}\,(\mathbf{x}_k, t_k) + \boldsymbol{\nu}_k\end{aligned}$$

where $\Delta\tilde{\mathbf{y}}$ is inertial and $\tilde{z}_k$ is non-inertial, with covariances, $\mathbf{S}_{\Delta\eta}\left(t_j\right)\delta_{jj'}$ and $\mathbf{R}_\nu\,(t_k)\,\delta_{kk'}$, respectively. When discrete measurements are processed by the onboard navigation filter, the truth state does not change at that instant of time.

$$\mathbf{x}_k^+ = \mathbf{x}_k^- \tag{10}$$

However, when the impulsive translational control inputs are modeled, the truth state is instantly corrected according to the mapping $\mathbf{d}$ and accounting for the correction noise $\Delta\mathbf{w}_j$

$$\mathbf{x}_j^{+c} = \mathbf{x}_j^{-c} + \mathbf{d}\left(\mathbf{x}_j^{-c}, \Delta\hat{\mathbf{u}}_j, t_j\right) + \Delta\mathbf{w}_j \tag{11}$$

with covariance $\mathbf{S}_{\Delta w}\left(t_j\right)\delta_{jj'}$.

2. Nominal Models

The nominal state dynamics are a function of the nominal state $\bar{\mathbf{x}}$, the nominal control inputs $\bar{\mathbf{u}}$, and time t, such that there are no disturbance accelerations, maneuver execution errors, navigation errors, or initial condition uncertainty impacting the state propagation,

$$\dot{\bar{\mathbf{x}}} = \bar{\mathbf{f}}\left(\bar{\mathbf{x}}, \bar{\mathbf{u}}, t\right) \tag{12}$$

where the nominal control input is a function of the nominal state and time,

$$\bar{\mathbf{u}} = \bar{\mathbf{g}}\left(\bar{\mathbf{x}}, t\right). \tag{13}$$

When discrete navigation measurements are processed, it does not impact the nominal state so

$$\bar{\mathbf{x}}_k^+ = \bar{\mathbf{x}}_k^-. \tag{14}$$

However, impulsive corrections do impact the nominal state by instantly correcting it according to the targeting algorithm logic

$$\Delta\bar{\mathbf{u}} = \Delta\bar{\mathbf{g}}\left(\bar{\mathbf{x}}_j^{-c}, t_j\right) \tag{15}$$

and mapping $\bar{\mathbf{d}}$. It is assumed that the burn is executed perfectly and is not impacted by navigation errors, so that

$$\bar{\mathbf{x}}_j^{+c} = \bar{\mathbf{x}}_j^{-c} + \bar{\mathbf{d}}\left(\bar{\mathbf{x}}_j^{-c}, \Delta\bar{\mathbf{u}}_j, t_j\right) \tag{16}$$

3. GNC System Model

The dynamics of the navigation state utilized by the guidance, navigation and control (GNC) system is modeled as

$$\dot{\hat{\mathbf{x}}} = \hat{\mathbf{f}}\left(\hat{\mathbf{x}}, \hat{\mathbf{u}}, \tilde{\boldsymbol{y}}, t\right) \tag{17}$$

$$\dot{\hat{\mathbf{P}}} = \left[\hat{\mathbf{F}}_{\hat{x}} + \hat{\mathbf{F}}_{\tilde{y}}\hat{\mathbf{C}}_{\hat{x}}\right]\hat{\mathbf{P}} + \hat{\mathbf{P}}\left[\hat{\mathbf{F}}_{\hat{x}} + \hat{\mathbf{F}}_{\tilde{y}}\hat{C}_{\hat{x}}\right] + \hat{\mathbf{F}}_{\tilde{y}}\hat{\mathbf{S}}_{\eta}\hat{\mathbf{F}}_{\tilde{y}}^T + \hat{\mathbf{S}}_w \tag{18}$$

where $\hat{\mathbf{x}}$ is the navigation system's estimate for the current state and $\hat{\mathbf{P}}$ represents the onboard state covariance of the navigation state relative to the design state defined in equation 5. The continuous onboard controller models are expressed as

$$\hat{\mathbf{u}} = \hat{\mathbf{g}}\left(\hat{\mathbf{x}}, t\right)$$

When discrete measurement updates are processed by the onboard navigation filter, the state and covariance matrix are instantly updated using the Kalman gain, $\hat{\mathbf{K}}$, so that

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \hat{\mathbf{K}}\left(t_k\right)\left[\hat{\mathbf{z}}_k - \hat{\mathbf{h}}\left(\hat{\mathbf{x}}_k, t_k\right)\right] \tag{19}$$

$$\hat{\mathbf{P}}\left(t_k^+\right) = \left[I - \hat{\mathbf{K}}\left(t_k\right)\hat{\mathbf{H}}_{\hat{x}}\left(t_k\right)\right]\hat{\mathbf{P}}\left(t_k^-\right)\left[I - \hat{\mathbf{K}}\left(t_k\right)\hat{\mathbf{H}}_{\hat{x}}\left(t_k\right)\right]^T + \hat{\mathbf{K}}\left(t_k\right)\hat{\mathbf{R}}_{\nu}\hat{\mathbf{K}}^T\left(t_k\right) \tag{20}$$

$$\hat{\mathbf{K}}\left(t_k\right) = \hat{\mathbf{P}}\left(t_k\right)\hat{\mathbf{H}}_{\hat{x}}^T\left(t_k\right)\left[\hat{\mathbf{H}}_{\hat{x}}\left(t_k\right)\hat{\mathbf{P}}\left(t_k\right)\hat{\mathbf{H}}_{\hat{x}}^T\left(t_k\right) + \hat{\mathbf{R}}_{\nu}\left(t_k\right)\right]^{-1} \tag{21}$$

For discrete control inputs such as impulsive translational burns, the navigation state update instantaneously adjusts the vehicles' velocity state and covariance according to,

$$\hat{\mathbf{x}}_j^{+c} = \hat{\mathbf{x}}_j^{-c} + \hat{\mathbf{d}}\left(\hat{\mathbf{x}}_j^{-c}, \Delta\hat{\mathbf{u}}_j, t_j\right) \tag{22}$$

$$\begin{aligned}\hat{\mathbf{P}}\left(t_j^{+c}\right) = {} & \left[\mathbf{I} + \hat{\mathbf{D}}_{\hat{x}}\left(t_j\right) + \hat{\mathbf{D}}_{\Delta\tilde{\boldsymbol{y}}}\left(t_j\right)\Delta\hat{\mathbf{C}}_{\hat{x}}\left(t_j\right)\right]\hat{\mathbf{P}}\left(t_j^{-c}\right)\left[\mathbf{I} + \hat{\mathbf{D}}_{\hat{x}}\left(t_j\right) + \hat{\mathbf{D}}_{\Delta\tilde{y}}\left(t_j\right)\Delta\hat{\mathbf{C}}_{\hat{x}}\left(t_j\right)\right]^T \\ & + \hat{\mathbf{D}}_{\Delta\tilde{y}}\left(t_j\right)\hat{\mathbf{S}}_{\Delta\eta}\hat{\mathbf{D}}_{\Delta\tilde{y}}^T\left(t_j\right) + \hat{\mathbf{S}}_{\Delta w}\left(t_j\right)\end{aligned} \tag{23}$$

where $\Delta\hat{\mathbf{u}}_j$ is given by the impulsive control algorithm model

$$\Delta\hat{\mathbf{u}}_j = \hat{\mathbf{g}}\left(\hat{\mathbf{x}}_j^{-c}, t_j\right)$$

4. Augmented State Covariance Models

In addition to deriving the navigation errors, computing the integrated GN&C performance metrics such as trajectory dispersions and control dispersions, requires the augmented state covariance defined in equation 7 to be initialized, propagated, updated, and corrected using the linear equations

$$\dot{\mathbf{C}} = \boldsymbol{\mathcal{F}}\mathbf{C} + \mathbf{C}\boldsymbol{\mathcal{F}}^T + \boldsymbol{\mathcal{G}}\mathbf{S}_\eta\boldsymbol{\mathcal{G}}^T + \boldsymbol{\mathcal{W}}\mathbf{S}_w\boldsymbol{\mathcal{W}}^T \tag{24}$$

$$\mathbf{C}\left(t_k^+\right) = \boldsymbol{\mathcal{A}}_k\mathbf{C}\left(t_k^-\right)\boldsymbol{\mathcal{A}}_k^T + \boldsymbol{\mathcal{B}}_k\mathbf{R}_\nu\left(t_k\right)\boldsymbol{\mathcal{B}}_k^T \tag{25}$$

$$\mathbf{C}\left(t_j^+\right) = \boldsymbol{\mathcal{D}}_j\mathbf{C}^{-c}\left(t_j^-\right)\boldsymbol{\mathcal{D}}_j^T + \boldsymbol{\mathcal{M}}_j\mathbf{S}_{\Delta\eta}\left(t_j\right)\boldsymbol{\mathcal{M}}_j^T + \boldsymbol{\mathcal{N}}_j\mathbf{S}_{\Delta w}\left(t_j\right)\boldsymbol{\mathcal{N}}_j^T \tag{26}$$

where

$$\boldsymbol{\mathcal{F}} = \begin{bmatrix} \mathbf{F}_x & \mathbf{F}_{\hat{u}}\hat{\mathbf{G}}_{\hat{x}} \\ \hat{\mathbf{F}}_{\tilde{\boldsymbol{y}}}\mathbf{C}_x & \hat{\mathbf{F}}_{\hat{x}} + \hat{\mathbf{F}}_{\hat{u}}\hat{\mathbf{G}}_{\hat{x}} \end{bmatrix} \qquad \boldsymbol{\mathcal{G}} = \begin{bmatrix} \mathbf{0} \\ \hat{\mathbf{F}}_{\tilde{y}} \end{bmatrix} \qquad \boldsymbol{\mathcal{W}} = \begin{bmatrix} \mathbf{I} \\ \mathbf{0} \end{bmatrix}$$

$$\boldsymbol{\mathcal{A}}_k = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \hat{\mathbf{K}}\left(t_k\right)\mathbf{H}_x\left(t_k\right) & \mathbf{I} - \hat{\mathbf{K}}\left(t_k\right)\hat{\mathbf{H}}\left(t_k\right) \end{bmatrix} \qquad \boldsymbol{\mathcal{B}}_k = \begin{bmatrix} \mathbf{0} \\ \hat{\mathbf{K}}\left(t_k\right) \end{bmatrix}$$

$$\boldsymbol{\mathcal{D}}_j = \begin{bmatrix} \mathbf{I} + \mathbf{D}_x\left(t_j\right) & \mathbf{D}_{\Delta\hat{u}}\left(t_j\right)\Delta\hat{\mathbf{G}}_{\hat{x}}\left(t_j\right) \\ \hat{\mathbf{D}}_{\Delta\tilde{y}}\left(t_j\right)\Delta\mathbf{C}_x\left(t_j\right) & \mathbf{I} + \hat{\mathbf{D}}_{\hat{x}}\left(t_j\right) + \hat{\mathbf{D}}_{\Delta\hat{u}}\left(t_j\right)\Delta\hat{\mathbf{G}}_{\hat{x}}\left(t_j\right) \end{bmatrix} \quad \boldsymbol{\mathcal{M}}_j = \begin{bmatrix} \mathbf{0} \\ \hat{\mathbf{D}}_{\Delta\tilde{y}}\left(t_j\right) \end{bmatrix} \quad \boldsymbol{\mathcal{N}}_j = \begin{bmatrix} \mathbf{I} \\ \mathbf{0} \end{bmatrix}$$

C. Sweeping Gradient Method

We assume each component J_i of the trajectory analysis output $\boldsymbol{J}$ is defined by

$$J_i = \phi_i\left(t_f, \boldsymbol{\xi}\left(t_f\right); \boldsymbol{\theta}\right) + \int_{t_0}^{t_f} L_i\left(t, \boldsymbol{\xi}\left(t\right); \boldsymbol{\theta}\right)\,dt$$

for $i = 1, \ldots, n_J$ where ϕ_i is a terminal function and L_i is an on-going function over a trajectory defined by a parameterized initial value problem with events. Every component depends on the same trajectory $\boldsymbol{\xi}\left(t\right)$ with initial condition

$$\boldsymbol{\xi}\left(t_0\right) = \mathbf{h}_0\left(t_0; \boldsymbol{\theta}\right)$$

with the ordinary differential equation governing the continuous evolution of the state is given by

$$\dot{\boldsymbol{\xi}}\left(t\right) = \mathbf{f}\left(t, \boldsymbol{\xi}\left(t\right); \boldsymbol{\theta}\right) \quad .$$

An event i occurs at the zero-crossing of an its event function

$$\gamma_i\left(t_k^-, \boldsymbol{\xi}\left(t_k^-\right); \boldsymbol{\theta}\right) = 0$$

at which point its update equation $\mathbf{h}_i$ is applied to the state so that

$$\boldsymbol{\xi}\left(t_k^+\right) = \mathbf{h}_i\left(t_k^-, \boldsymbol{\xi}\left(t_k^-\right); \boldsymbol{\theta}\right),$$

enabling the modeling of discontinuous changes to the state or dynamics.

The continuous-time SGM method emerged during early analysis of differential equations and was used through the 1960s to solve optimal control problems. The SGM for a trajectory analysis for an initial value problem without events involves a four step process:

1) Evaluate the trajectory by solving IVP with the given set of parameters
2) Evaluate the performance index of interest for the given set of parameters and trajectory.
3) Solve the differential equation adjoint to the trajectory model *and each performance index*, given by

$$\dot{\boldsymbol{\lambda}}\left(t\right) = -\left[\frac{\partial\mathbf{f}}{\partial\boldsymbol{\xi}}\left(t, \boldsymbol{\xi}\left(t\right); \boldsymbol{\theta}\right)\right]^T \boldsymbol{\lambda}\left(t\right) - \left[\frac{\partial L_i}{\partial\boldsymbol{\xi}}\left(t, \boldsymbol{\xi}\left(t\right); \boldsymbol{\theta}\right)\right]^T$$

$$\boldsymbol{\lambda}\left(t_f\right) = \frac{\partial\phi_i}{\partial\boldsymbol{\xi}}\left(t_f, \boldsymbol{\xi}\left(t_f\right); \boldsymbol{\theta}\right)$$

where the forcing function $\frac{\partial L_i}{\partial\xi}$ and terminal condition $\frac{\partial\phi_i}{\partial\xi}$ induce a unique adjoint trajectory for each performance index J_i. Notice also that the adjoint trajectory depends on the particular trajectory determined by the parameters θ as solved in Step 1.

4) Evaluate the gradient of the performance index with respect to the parameters by computing

$$\frac{\partial J_i}{\partial \boldsymbol{\theta}} = \int_{t_0}^{t_f} \boldsymbol{\lambda}^T\left(t\right) \frac{\partial L_i}{\partial \boldsymbol{\theta}}\left(t, \boldsymbol{\xi}\left(t\right); \boldsymbol{\theta}\right)\, dt + \boldsymbol{\lambda}^T\left(t_f\right) \frac{\partial \phi_i}{\partial \boldsymbol{\theta}}\left(t, \boldsymbol{\xi}\left(t\right); \boldsymbol{\theta}\right)$$

This sweeping gradient method was recently extended to trajectories modeled by ODEs with events by inducing an associated adjoint event with update equation

$$\begin{aligned}\boldsymbol{\lambda}\left(t_e^-\right) &= \left[\frac{\partial}{\partial \boldsymbol{\xi}}\mathbf{h}\left(t_e, \boldsymbol{\xi}\left(t_e^+\right); \boldsymbol{\theta}\right)\right]^T \boldsymbol{\lambda}\left(t_e^+\right) \\ &+ \left[\frac{dt_e}{d\boldsymbol{\xi}}\right]^T \left[\frac{\partial}{\partial \boldsymbol{\xi}}\mathbf{h}\left(t_e, \boldsymbol{\xi}\left(t_e^+\right); \boldsymbol{\theta}\right) \mathbf{f}\left(t_e, \boldsymbol{\xi}\left(t_e^-\right); \boldsymbol{\theta}\right) - \mathbf{f}\left(t_e, \boldsymbol{\xi}\left(t_e^+\right); \boldsymbol{\theta}\right)\right]^T \boldsymbol{\lambda}\left(t_e^+\right)\end{aligned} \tag{27}$$

where

$$\frac{dt_e}{d\boldsymbol{\xi}} = \left(\frac{\partial}{\partial \boldsymbol{\xi}}\gamma\left(t_e, \boldsymbol{\xi}\left(t_e^-\right); \boldsymbol{\theta}\right) \mathbf{f}\left(t_e, \boldsymbol{\xi}\left(t_e^-\right); \boldsymbol{\theta}\right)\right)^{-1} \frac{\partial}{\partial \boldsymbol{\xi}}\gamma\left(t_e, \boldsymbol{\xi}\left(t_e^-\right); \boldsymbol{\theta}\right)$$

represents the sensitivity of the event time with respect to the state. The events in the state and adjoint trajectories contribute additional discontinuous terms to the gradient of the form

$$\begin{aligned}\Delta\frac{\partial J_i}{\partial \boldsymbol{\theta}} &= \boldsymbol{\lambda}^T\left(t_e^+\right)\left[\frac{\partial}{\partial \boldsymbol{\theta}}\mathbf{h}\left(t_e, \boldsymbol{\xi}\left(t_e^-\right); \boldsymbol{\theta}\right)\right] \\ &+ \boldsymbol{\lambda}^T\left(t_e^+\right)\left[\frac{\partial}{\partial \boldsymbol{\xi}}\mathbf{h}\left(t_e, \boldsymbol{\xi}\left(t_e^-\right); \boldsymbol{\theta}\right) f\left(t_e, \boldsymbol{\xi}\left(t_e^-\right); \boldsymbol{\theta}\right) - \mathbf{f}\left(t_e, \boldsymbol{\xi}\left(t_e^+\right); \boldsymbol{\theta}\right)\right] \frac{dt_e}{d\boldsymbol{\theta}}\end{aligned} \tag{28}$$

where

$$\frac{dt_e}{d\boldsymbol{\theta}} = -\left(\frac{\partial}{\partial t_e}\gamma\left(t_e, \boldsymbol{\xi}\left(t_e^-\right); \boldsymbol{\theta}\right) + \frac{\partial}{\partial \boldsymbol{\xi}}\gamma\left(t_e, \boldsymbol{\xi}\left(t_e^-\right); \boldsymbol{\theta}\right) \mathbf{f}\left(t_e, \boldsymbol{\xi}\left(t_e^-\right); \boldsymbol{\theta}\right)\right)^{-1} \frac{\partial}{\partial \boldsymbol{\theta}}\gamma\left(t_e, \boldsymbol{\xi}\left(t_e^-\right); \boldsymbol{\theta}\right)$$

represents the sensitivity of the event time with respect to the parameters. For additional details, we refer the interested reader to [13]. To apply this method for $n_J > 1$ trajectory analysis outputs, steps 2-4 are repeated for each output. This approach is extremely efficient when the space of the trajectory analysis output is smaller than the dimension of the parameter space. Alternative sensitivity methods or alternative modeling choices may be warranted for problems where the size of the parameter space is smaller than the size of the trajectory analysis output space.

D. Combining Linear Covariance Analysis and Sweeping Gradient Method

To combine LinCov with SGM, we form the basic *complete* state vector consisting of the nominal state vector $\bar{\mathbf{x}}$, onboard state covariance matrix $\hat{\mathbf{P}}$, and the augmented state covariance matrix $\mathbf{C}$,

$$\boldsymbol{\xi}_0\left(t\right) = \begin{bmatrix} \bar{\mathbf{x}}\left(t\right) \\ \operatorname{vec}\hat{\mathbf{P}}\left(t\right) \\ \operatorname{vec}\mathbf{C}\left(t\right) \end{bmatrix} \tag{29}$$

where the vec operator un-ravels the argument matrix into a column vector, optionally ignoring repeated elements of a symmetric matrix for computational and storage requirement reduction. As is common in optimal control settings, additional state augmentation is used to accumulate performance indices so all performance indices of interest can be evaluated with terminal functions. The basic complete state $\boldsymbol{\xi}_0$ defined by equation (29) is denoted by the sub-script zero so the complete state for any particular scenario may be augmented with the additional elements as needed.

The vector functions of initial conditions, state equation, and update equations are similarly concatenations of the corresponding component functions. In other words, the complete state has an initial condition

$$\boldsymbol{\xi}_0\left(t_0\right) = \begin{bmatrix} \bar{\mathbf{x}}\left(t_0\right) \\ \operatorname{vec}\hat{\mathbf{P}}\left(t_0\right) \\ \operatorname{vec}\mathbf{C}\left(t_0\right) \end{bmatrix} \tag{30}$$

and is governed by the state equation

$$\dot{\boldsymbol{\xi}}_0\left(t\right) = \boldsymbol{\Phi}\left(t, \boldsymbol{\xi}\left(t\right); \boldsymbol{\theta}\right)$$

where

$$\boldsymbol{\Phi}\left(t, \boldsymbol{\xi}\left(t\right); \boldsymbol{\theta}\right) = \begin{bmatrix} \bar{\mathbf{f}}\left(t, \bar{\mathbf{x}}\left(t\right); \boldsymbol{\theta}\right) \\ \operatorname{vec} \mathbf{f}_P\left(t, \bar{\mathbf{x}}\left(t\right), \hat{\mathbf{P}}\left(t\right); \boldsymbol{\theta}\right) \\ \operatorname{vec} \mathbf{f}_C\left(t, \bar{\mathbf{x}}\left(t\right), \hat{\mathbf{P}}\left(t\right), \mathbf{C}\left(t\right); \boldsymbol{\theta}\right) \end{bmatrix} \tag{31}$$

is formed by concatenating the (vec-operated) state equation for each of the constituent components, defined by equations (12), 5, and (24). The cascading dependence of each of the components would produce a lower-block triangular structure to the Jacobian of the complete state equation.

A similar procedure could be used for the update equations. For discrete measurement updates, we assume an event function fully defined by time and a complete state update defined by concatenating the associated measurement updates for each component of the complete state of the form

$$\begin{aligned} \gamma_k\left(t, \boldsymbol{\xi}; \boldsymbol{\theta}\right) &= t - t_k \\ \boldsymbol{\xi}_0\left(t_k^+\right) &= \begin{bmatrix} \bar{\mathbf{x}}\left(t_k^-\right) \\ \operatorname{vec} \hat{\mathbf{P}}\left(t_k^+\right) \\ \operatorname{vec} \mathbf{C}\left(t_k^+\right) \end{bmatrix} \end{aligned} \tag{32}$$

where the nominal state is unchanged by the measurement update as defined in equation (14) and the on-board covariance and augmented covariance updates are defined by equations (20) and (25), respectively. Likewise, impulsive control corrections are modeled with an event function fully defined by time and a complete state update defined by concatenating the associated correction updates for each component of the complete state of the form

$$\begin{aligned} \gamma_j\left(t, \boldsymbol{\xi}\right) &= t - t_j \\ \boldsymbol{\xi}_0\left(t_j^+\right) &= \begin{bmatrix} \bar{\mathbf{x}}\left(t_j^+\right) \\ \operatorname{vec} \hat{\mathbf{P}}\left(t_j^+\right) \\ \operatorname{vec} \mathbf{C}\left(t_j^+\right) \end{bmatrix} \end{aligned} \tag{33}$$

where the nominal state, on-board covariance, and augmented covariance updates are defined by equations (16), 23, and (26), respectively.

The event models for generic discrete measurements (32) and impulsive control actions (33) illustrate how naturally the LinCov and SGM techniques can be combined to formulate robust optimal control and system design problems, particularly for orbital trajectory design where maneuvers are typically modeled impulsively.

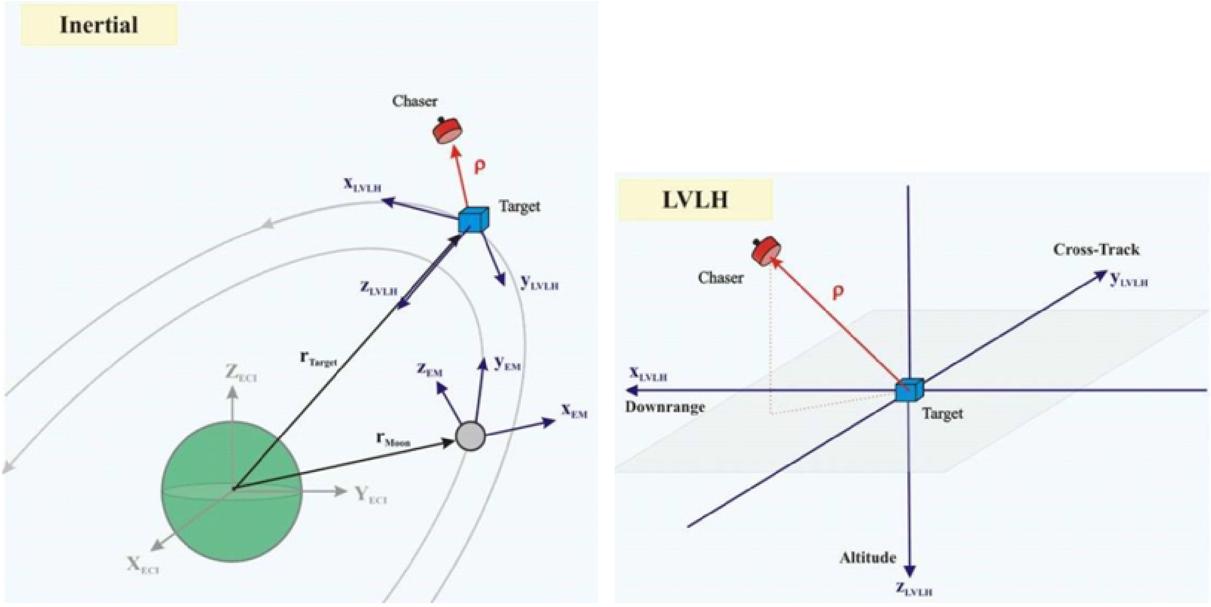
III. Robust Trajectory Optimization Scenario and Analysis Inputs

To demonstrate the robust trajectory optimization, we show its application to a rendezvous and docking problem in low-Earth orbit (LEO). This section provides an overview of the system dynamics model, rendezvous scenario, GN&C system models, and the core SGM and LinCov models used for this application.

A. System Dynamics Model

The relative motion between two orbiting vehicles around a central body can be described with the Clohessy-Wiltshire (CW) equations[20]. The relative reference frame is a rotating reference frame with the target vehicle acting as the origin as shown in Figure 3 with Figure 3a emphasizing the relative motion in the Earth Centered Inertial (ECI) frame and Figure 3b highlighting the relative position in the Local Vertical Local Horizontal (LVLH) frame.

The LVLH reference frame, has three orthogonal basis vectors – $\{\boldsymbol{x}_{LVLH}\ \boldsymbol{y}_{LVLH}\ z_{LVLH}\}$ – defined approximately by the velocity, orbital angular velocity, and position of the target vehicle respectively. The z-axis points along the radial direction from the target vehicle to the central body or Earth and for this reason is often labeled as the *r-bar*. Since this axis describes the altitude or vertical position of the chaser with respect to the target, it is also referred to as the *altitude* component or the *local vertical*. The $\boldsymbol{y}$-axis is perpendicular to the target vehicle's orbital plane in the direction of the



(a) ECI Frame

(b) LVLH Frame

Fig. 3 Relative LVLH reference frame

target's orbital angular velocity. This out-of-plane motion is commonly known as the *cross-track* motion of the chaser. The $\boldsymbol{x}$-axis is the *local horizontal* and is defined as the cross-product of the other two axes. It generally points along the velocity vector of the target and describes how far downrange the chaser is from the target. For these two reasons, this axis is called the *v-bar* or *downrange* direction. Mathematically, each of these axes have the following definition,

$$\begin{aligned} \text{Downrange:} \quad \boldsymbol{x}_{LVLH} &= \boldsymbol{y}_{LVLH} \times \boldsymbol{z}_{LVLH} \\ \text{Cross-Track:} \quad \boldsymbol{y}_{LVLH} &= -\frac{\boldsymbol{\omega}}{|\boldsymbol{\omega}|} \\ \text{Altitude:} \quad \boldsymbol{z}_{LVLH} &= -\frac{\mathbf{r}_t}{|\mathbf{r}_t|} \end{aligned} \tag{34}$$

where the target's angular velocity is proportional to the cross product between its position and velocity vectors

$$\boldsymbol{\omega} = \frac{\mathbf{r}_t \times \mathbf{v}_t}{|\mathbf{r}_t|^2} \tag{35}$$

Using this relative coordinate system, the chaser's relative position, $\boldsymbol{\rho}$, can be completely described as,

$$\boldsymbol{\rho} = [x, y, z]^T \tag{36}$$

where x, y, and z are the downrange, cross-track, and altitude components of the relative position vector respectively. The resulting linear CW-equations modeling the relative motion between two orbiting vehicles where the target is in a circular orbit with the orbital angular rate ω,

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} 2\omega\dot{z} \\ -\omega^2 y \\ 3\omega^2 z - 2\omega\dot{x} \end{bmatrix} \tag{37}$$

The relative position and velocity can be expressed in the state space form such that the CW-system dynamics are

$$\dot{\mathbf{x}} = \mathbf{A}_{cw}\mathbf{x} \tag{38}$$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 2\omega \\ 0 & -\omega^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3\omega^2 & -2\omega & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} \tag{39}$$

B. Relative Trajectory Profile

The demonstration scenario assumes the chaser vehicle starts on a co-elliptic trajectory relative to the target spacecraft at a specified downrange distance. As illustrated in Figure 4, a coelliptic trajectory produces the relative motion profile that causes the chaser to pass either above or below the target vehicle at a constant relative altitude.

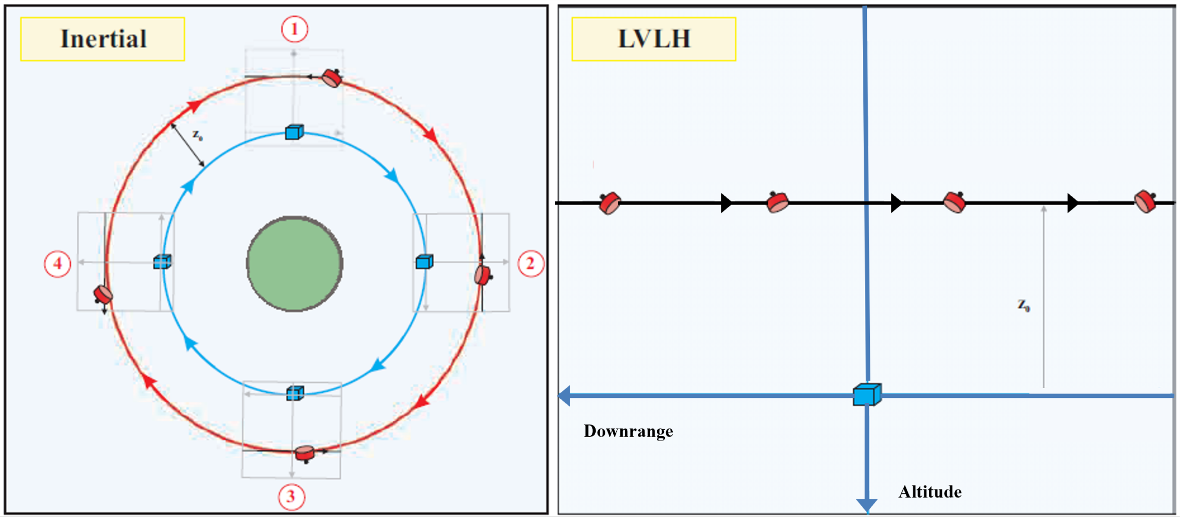


Fig. 4 Notional Coelliptic Trajectory

The scenarios in Section IV mimic a rendezvous with the International Space Station (ISS) in low-Earth orbit, where the chaser spacecraft starts 2 km behind ISS and 1 km below. If no translation burns are executed, the vehicle drifts below the ISS as depicted in Figure 5. Each scenario accounts for the onboard GN&C system, disturbance accelerations acting on the vehicle, navigation errors and periodic discrete measurement updates, initial trajectory dispersions, and impulsive maneuver execution errors.

C. GN&C System Models

The models governing the GN&C system include relative targeting logic and the relative navigation sensors along with the initial relative navigation error, initial relative trajectory dispersions, maneuver execution errors, and the relative disturbance acceleration.

Relative Targeting Model The relative targeting utilizes the CW-equations and targets the relative position $\mathbf{r}_{i+1}$ after some specified transfer time Δt_i. To demonstrate how this is done, the system's state transition matrix Φ is required which is a function of the state dynamics $\mathbf{A}_{cw}$ defined in equation (38) and the transfer time, Δt_i using the matrix exponential

$$\Phi = e^{\mathbf{A}_{cw}\Delta t_i} \tag{40}$$

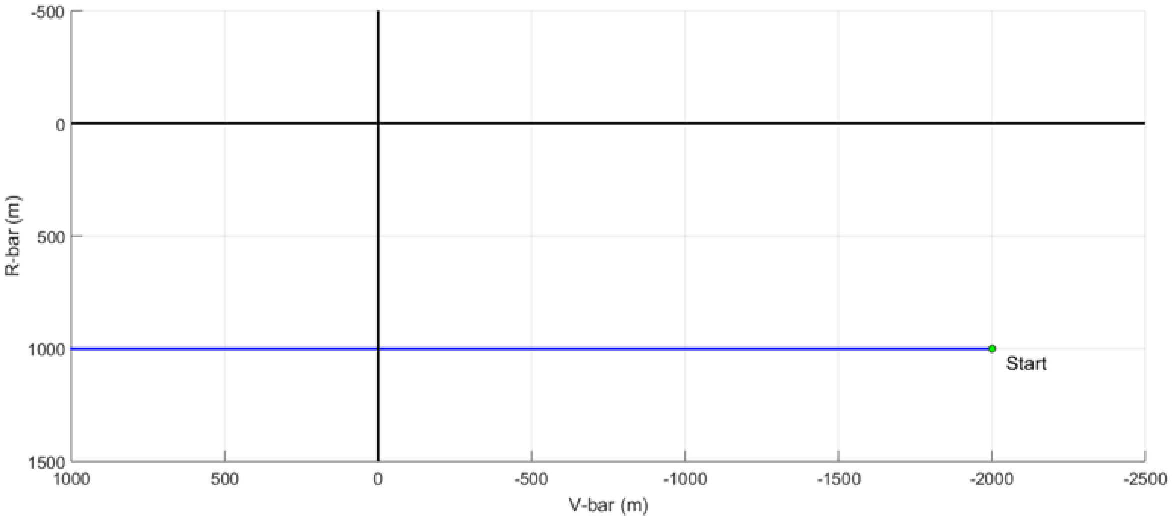


Fig. 5 Simulated Coelliptic Trajectory

The state transition matrix can be divided in terms of the relative position and velocity state,

$$\Phi = \begin{bmatrix} \Phi_{rr} & \Phi_{rv} \\ \Phi_{vr} & \Phi_{vv} \end{bmatrix} \tag{41}$$

Using Φ to determine how an instantaneous change in velocity $\Delta\mathbf{v}$ will influence the relative position at time $t_i + \Delta t_i$,

$$\begin{bmatrix} \mathbf{r}_{i+1} \\ \mathbf{v}_{i+1} \end{bmatrix} = \begin{bmatrix} \Phi_{rr} & \Phi_{rv} \\ \Phi_{vr} & \Phi_{vv} \end{bmatrix} \begin{bmatrix} \mathbf{r}_i \\ \mathbf{v}_i + \Delta\mathbf{v} \end{bmatrix} \tag{42}$$

the three rows associated with the position can be expressed as

$$\mathbf{r}_{i+1} = \Phi_{rr}\mathbf{r}_i + \Phi_{rv}\left(\mathbf{v}_i + \Delta\mathbf{v}\right) \tag{43}$$

Targeting a final relative position state $\mathbf{r}_{i+1}$ from the current state requires an instantaneous velocity change of

$$\Delta\mathbf{v} = \Phi_{rv}^{-1}\left(\mathbf{r}_{i+1} - \Phi_{rr}\mathbf{r}_i\right) - \mathbf{v}_i \tag{44}$$

which is treated as the nominal and on-board control model $\Delta\bar{\mathbf{g}} = \Delta\hat{\mathbf{g}}$ for this scenario. This controller has a corresponding mapping from the nominal control $\Delta\bar{\mathbf{u}}$ to the change in the nominal state by

$$\bar{\mathbf{d}}\left(\bar{\mathbf{x}}_j^{-c}, \Delta\bar{\mathbf{u}}_j, t_j\right) = \left[\mathbf{0}_{3\times3}, \mathbf{I}_{3\times3}\right]^T \Delta\bar{\mathbf{g}}\left(\bar{\mathbf{x}}_j^{-c}, t_j\right)$$

Relative Sensor Models For simplicity, the relative sensor model for this study is a relative GPS measurement that measures the relative position directly, $\boldsymbol{\rho}$. To reduce the computational load, a simplified model is adopted

$$\tilde{\boldsymbol{\rho}} = \boldsymbol{\rho} + \eta_\rho \tag{45}$$

The performance parameters assumed for the maneuver execution error, relative GPS measurement error, initial relative position and velocity navigation error, initial relative position and velocity dispersions, and process noise are defined below in Table 1.

Table 1 GN&C System Performance Parameters

symbol	component description	value
σ_{η_ρ}	Relative GPS Accuracy	15 mm
Δt_k	Time between relative GPS measurements	10 min
$\hat{\mathbf{P}}(0)$	3σ per-axis initial position navigation error	10 m
	3σ per-axis initial velocity navigation error	11.4 mm/s
$3\sigma_{\Delta w}$ and $3\sigma_{\hat{\Delta w}}$	3σ per-axis maneuver execution error (and onboard)	1.5 mm/s
σ_w and $\sigma_{\hat{w}}$	1σ per-axis position process noise (and onboard)	0 mm/s
	1σ per-axis velocity process noise (and onboard)	480 nm/s^2

D. SGM and LINCOV Models

Implementation considerations For these measurement and impulsive control models, the event time is independent of the state so the adjoint update terms containing $dt_e/d\xi = 0$ can be ignored. If the event time is specified as a parameter, the associated differential $dt_e/d\theta$ becomes unity. Most measurement or impulsive maneuver state updates will depend on both the state and parameters, inducing corresponding terms in (27) and (28), respectively. Since these terms are simple partial derivatives of the models described in the preceding sections, their explicit expressions will be left as an exercise for the reader.

Modeling considerations To compute trajectory outputs that are functions of event behavior, the authors have found it useful to create auxiliary components of the complete state that remain constant during the continuous portions of the trajectory with initial condition of zero. The update equations can be used to accumulate each event's contribution to that particular output. For example, we can consider a nominal impulse requirement accumulator state, $\left|\bar{\Delta v}\right|(t)$ which is constant during the continuous portion of the trajectory and during discrete measurements, but takes as an update during impulsive maneuvers

$$\left|\bar{\Delta v}\right|\left(t_j^+\right) = \left|\bar{\Delta v}\right|\left(t_j^-\right) + \Delta v, \tag{46}$$

where the second term is computed according to (44). With this state, we can define the corresponding trajectory output equation as

$$J_{\Delta v}(\boldsymbol{\theta}) = \left|\bar{\Delta v}\right|\left(t_f\right) \tag{47}$$

which may be used in a trajectory design problem to minimize propellant requirements.

IV. Demonstration of Robust Trajectory Optimization Techniques Using SGM and LINCOV

In this section, we present a set of simple cases demonstrating the applicability SGM and LinCov for robust orbital rendezvous trajectory design. As part of the description for each case, any additional elements to the complete state, the trajectory analysis parameters and outputs, and optimization problem formulation will be listed. To solve these problems, we use a mathematical modeling framework in Python called Condor[21] which is in NASA's open-source software release process. IPOPT[22] is used for the optimization and the CVODE solver[23] is used to solve the ODEs for SGM.

A. Minimum fuel circular transfer

In this case, we define the maneuver ignition time, t_{ig}, and the maneuver duration, t_{d}, for a single impulsive maneuver as the parameters to the trajectory. For output, we augment the complete state to include the nominal propellant accumulator defined by the nominal Δv magnitude for a maneuver to transfer between coelliptic circular orbits. We can therefore define this case by the optimization problem,

$$\begin{aligned} &\underset{t_{\text{ig}},t_{\text{d}}}{\text{minimize}} \quad |\Delta\bar{v}|\left(t_f\right) \\ &\text{where} \quad T\colon \begin{bmatrix} t_{\text{ig}} \\ t_{\text{d}} \end{bmatrix} \mapsto \left[|\Delta\bar{v}|\left(t_f\right)\right] \end{aligned}$$

Table 2 Constant values for minimum fuel optimization scenario

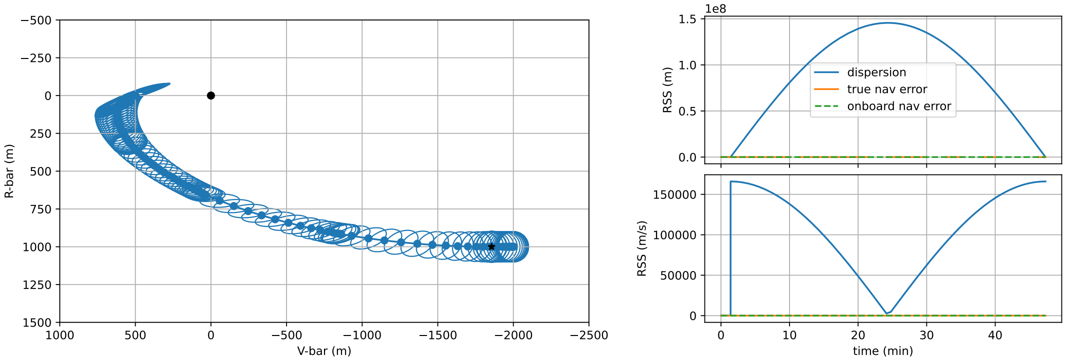
symbol	component description	value
ω	target angular velocity	0.00114
$\bar{x}(0)$	nominal initial position	$[-2, 0, 1]^T$ km
	nominal initial position	$[1.71, 0, 0]^T$ m/s
$\mathbf{C}(0)$	3σ per-axis initial position dispersion	100 m
	3σ per-axis initial velocity dispersion	114 mm/s
$\mathbf{r}_f$	target final position	$[500, 0, 0]^T$ m

Table 3 Comparison of initial and solution values for the minimum fuel transfer

variable	initial value	solution
t_{ig} (s)	156.7	84.1
t_{d} (s)	900	2760
$\lvert\Delta\bar{v}\rvert\left(t_f\right)$ (mm/s)	1480	285

where the accumulator state $|\Delta\bar{v}|(t)$ defined by (46) is appended to the basic complete state of (29). The constants used for the numerical example are listed in Table 2.

The optimizer produces a solution of $t_{\text{ig}} = 84.1$ s and $t_{\text{d}} = 2760$ s, corresponding to a 180 degree transfer angle, recovering the Hohmann solution with an impulse requirement of 285 mm/s. A comparison of the initial value given to the optimizer and the solution is listed in Table 3. The optimized trajectory is shown in Figure 6, with the nominal positions in dots and the 3σ position dispersions marked with corresponding ellipses. The LinCov model of the root sum of squares (RSS) true dispersions, true navigation dispersions, and on-board navigation dispersions for this result is also shown, illustrating the well known singularity of the solution with the true position dispersion on the order of 10^8 m. Since this problem is only two dimensional, we can visualize the design space using the "mission map" shown in Figure 7, which shows the required nominal impulse over the space of ignition time and maneuver durations. The map shows a single local minima that coincides with the solution from the gradient-based optimizer.



(a) Trajectory

(b) Time history

Fig. 6 Nominal fuel optimal solution of Case 1

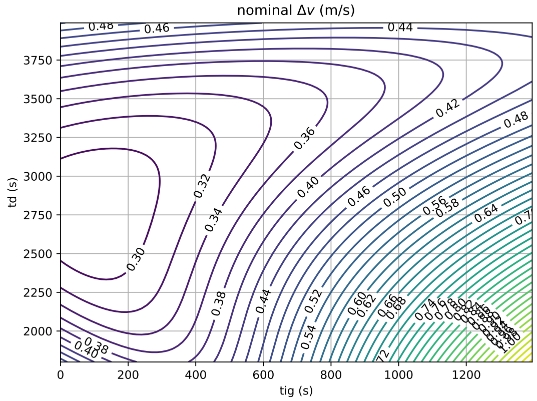


Fig. 7 Mission map for the nominal Δv magnitude of Case 1

B. Robust Minimum total fuel circular transfer

To address the dispersion risk of the Hohmann transfer, we can further augment the complete state with a RSS propellant dispersion accumulator, $\sigma_{|\Delta v|}$ which has an impulsive maneuver update defined by

$$\sigma_{|\Delta v|}\left(t_j^+\right) = \sigma_{|\Delta v|}\left(t_j^-\right) + \sqrt{\mathrm{tr}\left(\mathbf{M}_c \mathbf{C} \mathbf{M}_c^T\right)},$$

with

$$\mathbf{M}_c = \left[\hat{\mathbf{D}}_{\hat{x}}\left(t_j\right) + \hat{\mathbf{D}}_{\Delta\hat{u}}\left(t_j\right)\Delta\hat{\mathbf{G}}_{\hat{x}}\left(t_j\right)\right]\left[\mathbf{0}_{n\times n}, \mathbf{I}_{n\times n}\right]$$

Then a robust minimum fuel transfer can be described by the optimization problem minimizing the so-called total impulse requirement,

$$\underset{t_{\mathrm{ig}}, t_{\mathrm{em}}}{\text{minimize}} \quad |\Delta\bar{v}|\left(t_f\right) + 3\sigma_{|\Delta v|}\left(t_f\right)$$

$$\text{where} \quad T\colon \begin{bmatrix} t_{ig} \\ t_{em} \end{bmatrix} \mapsto \begin{bmatrix} |\Delta\bar{v}|\left(t_f\right) \\ \sigma_{|\Delta v|}\left(t_f\right) \end{bmatrix}$$

with the same constants that were used in the previous case. In this case, the optimizer finds $t_{\mathrm{ig}} = 51.9$ s and $t_{\mathrm{em}} = 2200$ s corresponding to a transfer angle of 144 degrees. This transfer angle is close to a heritage estimated solution of 135 degrees, which was found by averaging the Hohmann solution and the 90 degrees transfer which minimizes the out-of-plane dispersions. The solution has a total impulse requirement of 663 mm/s where 322 mm/s is due to the nominal impulse requirement. We notice that this robust solution has only marginally more nominal impulse required (13%) while risking orders of magnitude less dispersions as shown in Table 4. The optimized trajectory and LinCov model of the true dispersions, true navigation dispersions, and on-board navigation dispersions for this result is shown in Figure 8.

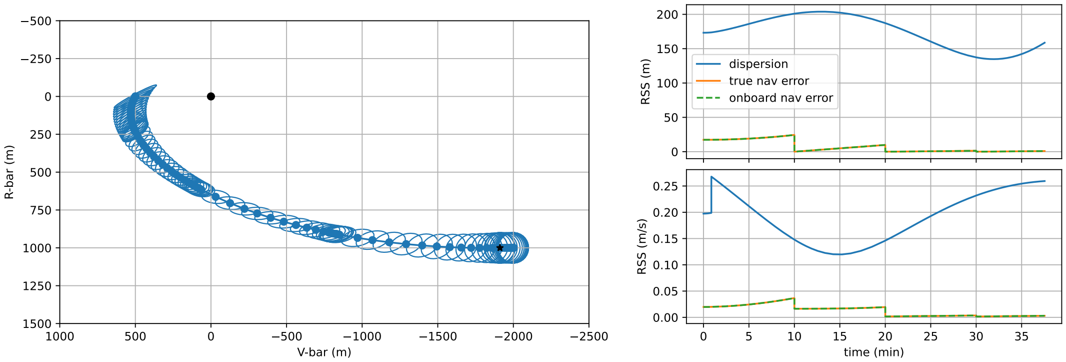
As with the previous case, we can visualize the two-dimensional design space using a mission map for the Total Δv, shown in Figure 9. This map shows the singularity associated with the 180 degree Hohmann transfer (a maneuver duration of 2760 s in this case) is independent of the maneuver execution time. Since the optimizer was provided an initial value below the singularity, the optimizer found the local optimum for a single revolution transfer. If the optimizer is initialized above the singularity, it finds the other local optimum shown in the map which is a multi-revolution transfer with 220 degree transfer angle.

C. Mid-Course correction optimization

In this scenario, we consider a rendezvous of a chaser to a target using one primary burn and one correction burn to recreate the example in [2]. For this scenario, we will consider two different objectives for the optimization problem of

Table 4 Comparison of initial and solution values for the robust minimum fuel transfer

variable	initial value	non-robust solution	robust solution
t_{ig} (s)	156.7	84.1	51.9
t_{d} (s)	900	2755.8	2200
$\lvert\Delta\bar{v}\rvert\,(t_f)$ (mm/s)	1480	285	322
$3\sigma_{\lvert\Delta v\rvert}\,(t_f)$ (m/s)	0.383	166000	0.341



(a) Trajectory

(b) Time history

Fig. 8 Fuel optimal solution for Case 2

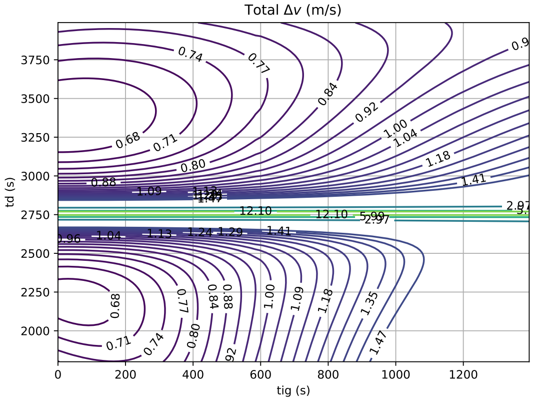


Fig. 9 Mission map for the total Δ*v* magnitude of Case 2

Table 5 Constant Values for the Mid-Course Correction problems

symbol	component description	value
t_p (s)	primary burn maneuver execution time	10
t_d (min)	maneuver duration	20
$\underline{t_c}$ (s)	lower bound, 10 minutes after t_p	610
$\bar{t_c}$ (s)	upper bound, 2 minutes before t_f	1090

Table 6 Comparison of initial and solution values for the Mid-Course Correction problems

variable	initial value	solution minimizing $\Gamma_{\Delta v}$	solution minimizing Γ_{σ_r}
t_c (s)	850	610	1090
$\sigma_{\lvert\Delta v\rvert}$ (mm/s)	148	128	241
Γ_{σ_r} (m)	3.35	3.37	3.33

the form

$$\begin{aligned} \underset{t_c}{\text{minimize}} \quad & \Gamma \\ \text{subject to} \quad & \underline{t_c} \le t_c \le \bar{t_c} \\ \text{where} \quad & T\colon \begin{bmatrix} t_c \\ t_f \end{bmatrix} \mapsto \begin{bmatrix} |\Delta\bar{v}|\left(t_f\right) \\ \sigma_{|\Delta v|}\left(t_f\right) \\ \sigma_r\left(t_f\right) \end{bmatrix} \end{aligned}$$

where t_c defines the maneuver execution time for the correction burn, which along with the primary burn initiating at t_p are targeting the position $\mathbf{r}_f$ at time t_f.

The new trajectory analysis output for the RSS position dispersion

$$\sigma_r\left(\mathbf{C}\left(t_f\right)\right) = \sqrt{\operatorname{tr}\left(\mathbf{M}_r\mathbf{C}\mathbf{M}_r^T\right)}$$

is a function of the final augmented covariance matrix, $\mathbf{C}\left(t_f\right)$, and the constant selection matrix

$$\mathbf{M}_r = \left[I_{3\times3} \quad 0_{3\times9}\right].$$

When using an objective to minimize the total impulse requirement

$$\Gamma_{\Delta v} = |\Delta\bar{v}|\left(t_f\right) + 3\sigma_{|\Delta v|}\left(t_f\right)$$

the optimizer will take the correction burn to the lower bound $\underline{t_c}$ but will incur large final position dispersions. We can alternatively minimize the position dispersions with an objective

$$\Gamma_{\sigma_r} = \sigma_r\left(t_f\right)$$

which the optimizer solves by delaying the maneuver until the upper bound $\bar{t_c}$ since that produces the most accurate maneuver. The trajectory for the solutions to the two optimization problems are shown in Figure 10 with numerical results listed in Table 6. The additional constants required to specify the scenario are listed in Table 5. Since this is a 1-dimensional problem, we can also visualize the design space as shown in Figure 11.

This modeling approach would also permit the design specifications from [2], such as using the total impulse requirement as the objective and enforcing a constraint that the final position dispersions remain below some requirement value. With SGM, the required gradients for both objective and constraints can be computed to solve such a constrained optimization problem.

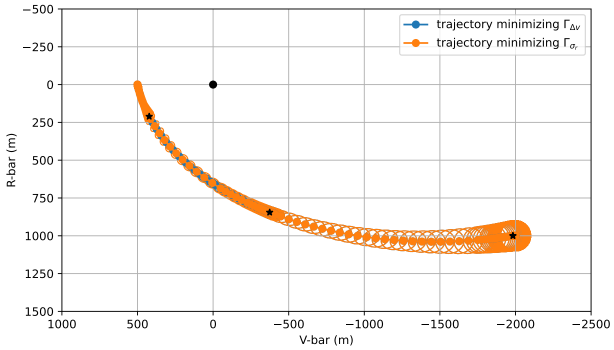


Fig. 10 Trajectory for the two solutions solving the two mid-course correction timing problems, with the location of the two solution corrections

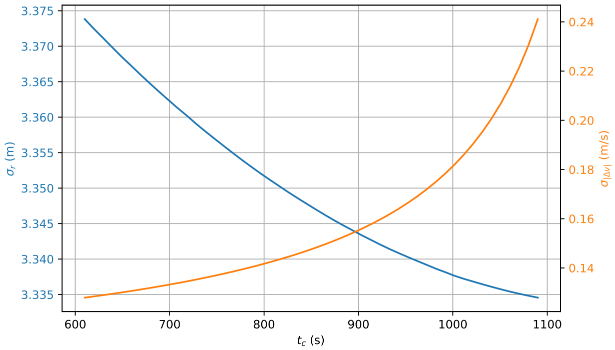


Fig. 11 Trajectory outputs over the design space for Mid-Course Correction timing problems

Table 7 Constant Values for the optimal measurement time problem

symbol	component description	value
t_{ig} (s)	maneuver execution time	850
t_{d} (s)	maneuver duration	1450

Table 8 Comparison of initial and solution values for the optimal measurement time problem

variable	initial value	solution
t_m (s)	0	850
Γ_{σ_r} (m)	26.7	9.02

D. Optimal timing for discrete measurement

Next we consider a single impulsive maneuver with given targeting constants using the initial values from Table 6. In this scenario, we introduce a single discrete measurement from a notional relative-position sensor and wish to minimize the total impulsive requirement by finding the best time to perform the measurement at time t_m. This case is described by the optimization problem

$$\begin{aligned} \underset{t_m}{\text{minimize}} \quad & \Gamma_{\sigma_r} \\ \text{where} \quad & T\colon \left[t_m\right] \mapsto \left[\sigma_r\left(t_f\right)\right] \end{aligned}$$

For this one dimensional problem, we can visualize the design space as shown in Figure 12 using the constant values in Table 7. The measurement has no impact on the nominal impulse requirement, but can greatly increase the accuracy and therefore reduce the dispersions of the maneuver the closer the measurement leads the impulse. Despite the discontinuity in the curve and the strong nonlinearity required to propagate the impact of the measurement on the total impulse requirement, the gradient-based optimizer solves this problem by correctly moving the measurement time to be immediately prior to the maneuver initiation time, within solver tolerance as shown in Table 8. This requires that the optimizer start the measurement before the burn so that it can follow the gradient from SGM to the minimum. If the initial measurement time is after the maneuver execution time, the gradient is null and the optimizer has no information to find the minimum.

Like the robust minimum propellant example in IV.B, this example highlights the primary challenge of this local method: the initial values for the variables must be selected so that the optimizer can use first-order (gradient) information to find a minimum. "Basin hopping" strategies exist to avoid local barriers to the global solution, and essentially amount to sampling the initial value to a gradient-based method over the design space like the 0th iteration of a gradient-free method.

Other techniques would face additional challenges for solving this problem. Collocation would require significant manipulation of the optimization problem to efficiently evaluate the entire design space and would require significant

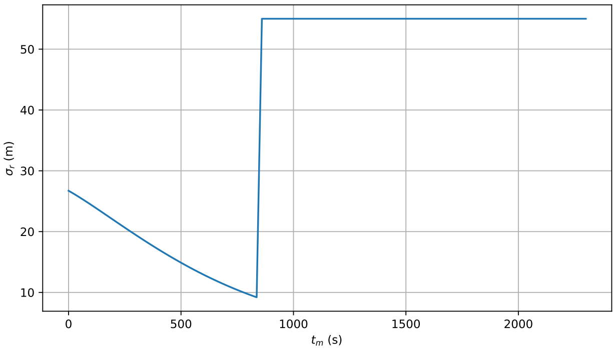


Fig. 12 Final position dispersions over the design space for the optimal measurement time problem

software development to handle the arbitrary ordering and separation of discrete events with sufficient accuracy. Convexification would require pre-computation of all candidate measurement times so that it essentially becomes an exhaustive search. In contrast, by employing SGM, the model can be solved at least for each continuous region of the design space.

E. Multi-Burn Maneuver

A common design scenario involves using multiple burns to acquire some relative position during rendezvous. For these scenarios, free drift safety requires that for each burn, the trajectory is passively safe if no additional burns are completed. To model this, we use a trajectory analysis for each required free drift in addition to the nominal trajectory. For each free drift trajectory, we create an accumulator state that stores the 3σ RSS dispersed point of closest approach (PCA), denoted by $\underline{\sigma}_{|\boldsymbol{\rho}|}$. The DPCA is the minimum value of the 3σ dispersed closest distance (DCD), denoted by $\sigma_{|\boldsymbol{\rho}|}$, over the free-drift trajectory. The DCD can be computed as a function of the complete state

$$\sigma_{|\boldsymbol{\rho}|}\left(\boldsymbol{\xi}\left(t\right)\right) = \left|\boldsymbol{\rho}\left(t\right)\right| - 3\sqrt{\hat{\boldsymbol{\rho}}^T\left(t\right)\mathbf{C}\left(t\right)\hat{\boldsymbol{\rho}}\left(t\right)}$$

where $\hat{\boldsymbol{\rho}}\left(t\right) = \left[\mathbf{0}_{3\times 1}^T, {\boldsymbol{\rho}^T(t)}/{|\boldsymbol{\rho}(t)|}, \mathbf{0}_{6\times 1}^T\right]^T$ captures the position dispersion along the vector from the chaser to target. Then we define a new event that occurs when the relative velocity of the DCD changes sign, which is modeled as an event function

$$\gamma_{\mathrm{DCPA}}\left(t, \boldsymbol{\xi}\left(t\right); \boldsymbol{\theta}\right) = \frac{\partial\sigma_{|\boldsymbol{\rho}|}\left(\boldsymbol{\xi}\left(t\right)\right)}{\partial\boldsymbol{\xi}}\mathbf{f}\left(t, \boldsymbol{\xi}\left(t\right); \boldsymbol{\theta}\right)$$

with a corresponding update

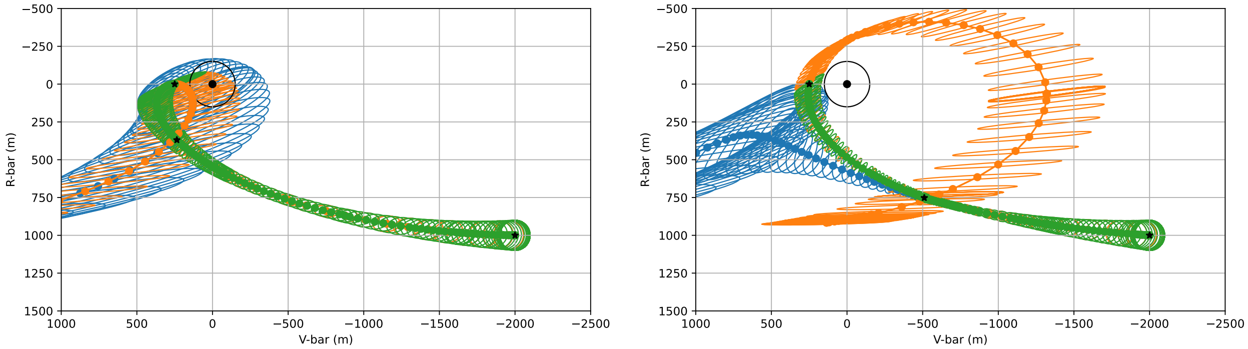
$$\underline{\sigma}_{|\boldsymbol{\rho}|}\left(t^+_{\mathrm{DCPA}}\right) = \begin{cases}\sigma_{|\boldsymbol{\rho}|}\left(\boldsymbol{\xi}\left(t^-_{\mathrm{DCPA}}\right)\right) & \underline{\sigma}_{|\boldsymbol{\rho}|}\left(t^-_{\mathrm{DCPA}}\right) > \sigma_{|\boldsymbol{\rho}|}\left(\boldsymbol{\xi}\left(t^-_{\mathrm{DCPA}}\right)\right) \\ \underline{\sigma}_{|\boldsymbol{\rho}|}\left(t^-_{\mathrm{DCPA}}\right) & \underline{\sigma}_{|\boldsymbol{\rho}|}\left(t^-_{\mathrm{DCPA}}\right) \leq \sigma_{|\boldsymbol{\rho}|}\left(\boldsymbol{\xi}\left(t^-_{\mathrm{DCPA}}\right)\right)\end{cases}$$

which updates the DPCA accumulator $\underline{\sigma}_{|\boldsymbol{\rho}|}$ with the current DCD $\sigma_{|\boldsymbol{\rho}|}$ if it was closer than the current value. The trajectory design can then be formulated as the optimization problem

$$\begin{aligned}
\underset{\boldsymbol{\theta}=\left[t_2, \mathbf{r}_2, t_f\right]^T}{\text{minimize}} \quad & \Gamma_{\Delta v} \\
\text{subject to} \quad & \bar{\sigma}_{|\boldsymbol{\rho}|,1}\left(t_f\right)\left(t_f\right) \geq R_{\mathrm{KOS}} \\
& \bar{\sigma}_{|\boldsymbol{\rho}|,2}\left(t_f\right)\left(t_f\right) \geq R_{\mathrm{KOS}} \\
& t_3 \leq \bar{t}_f \\
& \underline{\Delta t} \leq t_3 - t_2 \\
& \underline{\Delta t} \leq t_2 \\
\text{where} \quad & T_1 \colon \boldsymbol{\theta} \mapsto \left[\bar{\sigma}_{|\boldsymbol{\rho}|,1}\left(t_f\right)\right] \\
& T_2 \colon \boldsymbol{\theta} \mapsto \left[\bar{\sigma}_{|\boldsymbol{\rho}|,2}\left(t_f\right)\right] \\
& T_3 \colon \boldsymbol{\theta} \mapsto \begin{bmatrix}\left|\Delta\bar{v}\right|\left(t_f\right) \\ \sigma_{|\Delta v|}\left(t_f\right)\end{bmatrix}
\end{aligned}$$

where we consider a three burn maneuver using constant values specified in Table 9. It is assumed that the first maneuver occurs at the start of the scenario, targeting $\mathbf{r}_2$ with an end time that coincides with the second maneuver execution time, t_2. The second maneuver targets the final position $\mathbf{r}_3$, ending at time t_3, the execution time of the final maneuver. The final maneuver is a station-keeping burn to bring the relative velocity to rest. The total impulse for all three maneuvers is minimized subject to free drift and timing constraints. The timing constraints represent notional mission requirements for the total flight duration and operational limitations for the time between burns, such as communications losses and target acquisition.

The five-dimensional design space for this problem goes beyond our ability to visualize, but the previous cases build our confidence that combining SGM on the trajectory analysis with a gradient-based optimizer finds the local optimum. To aid in our understanding of this problem, we also solve the problem without the free-drift constraints. Figure 13 shows the optimal trajectories without and with the free-drift constraints, with the target vehicle's position location



(a) Optimal trajectory without free-drift constraints

(b) Optimal trajectory with free-drift constraints

Fig. 13 Optimal trajectories for multi-burn maneuver design

Table 9 Constant Values for multi-burn maneuver design

symbol	component description	value
$\mathbf{r}_3$	target final position	$[250, 0, 0]^T$ m
$\bar{t_f}$	maximum final time	45 minutes
$\underline{\Delta\text{t}}$	minimum time between burns	60 s
R_{KOS}	keep out sphere radius	150 m

Table 10 Comparison of initial and solution values for multi-burn maneuver design

variable	initial value	solution without free-drift constraints	solution with free-drift constraints
t_2 (s)	1800	850	899
$\mathbf{r}_3$ (m)	$[-1250, 0, 667]^T$	$[235, 0, 369]^T$	$[-511, 0, 752]^T$
t_3 (s)	2700	2700	1937
$\lvert\Delta\bar{v}\rvert\left(t_f\right) + 3\sigma_{\lvert\Delta v\rvert}\left(t_f\right)$ (m/s)	4.74	1.02	1.78
$\bar{\sigma}_{\lvert\rho\rvert,0}\left(t_f\right)$ (m)	1278	-269	150
$\bar{\sigma}_{\lvert\rho\rvert,1}\left(t_f\right)$ (m)	3.83	-126	150

marked with a black dot and the keep-out sphere perimeter around it. Table 10 compares the initial values with the two solutions.

The optimal trajectory without free-drift constraints derives a mid-course correction burn like the one used in IV.C; this can be seen visually by noting that nominal trajectories are coincident while the free-drift after the correction burn has reduced dispersions. In contrast, the optimal trajectory with the free-drift constraints starts with the surfaces of the dispersion ellipses closest to the target for each free-drift coincident. As is common with constrained optimization, the solution is driven by the constraints. For example, the solution to the multi-maneuver problem without free-drift constraints has the final time at the upper limit. With the free drift constraints, the solution has the free drift dispersions tangent to the keep-out boundary. The results seen here and in the authors' other experience suggests that optimal trajectories with free-drift constraints will not include true mid-course correction burns but will instead exhibit this modified correction.

V. Conclusion

In this work, we developed techniques to use Linear Covariance Analysis (LinCov) and a sweeping gradient method for ordinary differential equations with events (SGM) to solve robust optimal trajectory design problems with canonical examples for orbital trajectory design. The canonical examples demonstrate the ability of this approach to recover well-known and intuitive solutions to various orbital rendezvous scenarios. The examples demonstrate several modeling approaches to naturally formulate the scenarios as optimization problems. By using SGM to provide sensitivities of the trajectory model outputs with respect to the optimization variables, efficient off-the-shelf solvers can be used. A companion paper [24] applies these techniques to a more sophisticated trajectory design and navigation system requirement specification for a rendezvous scenario.

References

[1] Maybeck, P. S., *Stochastic models, estimation, and control*, Vol. 1, Academic Press, New York, 1979.

[2] Geller, D. K., "Linear Covariance Techniques for Orbital Rendezvous Analysis and Autonomous Onboard Mission Planning," *Journal of Guidance, Control, and Dynamics*, Vol. 29, No. 6, 2006, pp. 1404–1414.

[3] Jin, K., Geller, D. K., and Luo, J., "Robust Trajectory Design for Rendezvous and Proximity Operations with Uncertainties," *Journal of Guidance, Control, and Dynamics*, Vol. 43, No. 4, 2020, pp. 741–753.

[4] Geller, D. K., Shuster, S., Woffinden, D., and Bieniawski, S., "Robust Cislunar Trajectory Optimization via Midcourse Correction and Optical Navigation Scheduling," *44th Annual AAS Guidance, Navigaiton and Control Conference*, AAS 22-065, Breckenridge, CO, 2022.

[5] Woffinden, D., Shuster, S., and Geller, S., David Kand Bieniawski, "Robust Trajectory Optimization and GN&C Performance Analysis For NRHO Rendezvous," *2022 AAS/AIAA Astrodynamics Specialist Conference*, 22-564, Charlotte, North Carolina, 2022.

[6] Geller, D., Woffinden, D., and Bieniawski, S., "Sensitivity of Optimal Midcourse Correction Scheduling for Robust Cislunar Trajectory Design," AAS 23-061, Breckenridge, CO, 2023.

[7] Goulet, T., Woffinden, D., Collins, N., and Andrews, B., "Robust Trajectory Design for Rendezvous in a Near Rectilinear Halo Orbit," AAS 23-066, Breckenridge, CO, 2023.

[8] Cavesmith, T., Woffinden, D., and Collins, N., "Angles-Only Robust Trajectory Optimization for NRHO Rendezvous," AAS 24-168, Breckenridge, CO, 2024.

[9] Calkins, G., Woffinden, D., and Putnam, Z., "Robust Trajectory Optimization for Guided Powered Descent and Landing," *2022 AAS/AIAA Astrodynamics Specialist Conference*, AAS 22-660, Charlotte, NC, 2022.

[10] Joshi, J., Woffinden, D., and Putnam, Z., "End-to-End Mars Aerocapture Analysis Using Linear Covariance Techniques and Robust Trajectory Optimization," *2022 AAS/AIAA Astrodynamics Specialist Conference*, AAS 22-678, Charlotte, NC, 2022.

[11] Woffinden, D., Eckman, R., and Robinson, S., "Optimized Trajectory Correction Burn Placement for the NASA Artemis II Mission," AAS 23-062, Breckenridge, CO, 2023.

[12] Woffinden, D., and Barton, B., "Optimized Trajectory Correction Burn Placement for NRHO Orbit Maintenance," *33rd AAS/AIAA Space Flight Mechanics Meeting*, AAS 23-364, Austin, TX, 2023.

[13] Margolis, B. W. L., "A Sweeping Gradient Method for Ordinary Differential Equations with Events," *Journal of Optimization Theory and Applications*, Vol. 199, No. 2, 2023, pp. 600–638. doi:10.1007/s10957-023-02303-3.

[14] Sosa, A., and Tuminello, R. M., "Optimized Launch to Rendezvous Trajectories for a Mars Ascent Vehicle Using Direct Collocation," *ASCEND 2021*, American Institute of Aeronautics and Astronautics, 2021. doi:10.2514/6.2021-4074.

[15] Lu, P., and Liu, X., "Autonomous Trajectory Planning for Rendezvous and Proximity Operations by Conic Optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 36, No. 2, 2013, pp. 375–389. doi:10.2514/1.58436.

[16] OCAMPO, C., "AN ARCHITECTURE FOR A GENERALIZED SPACECRAFT TRAJECTORY DESIGN AND OPTIMIZATION SYSTEM," *Libration Point Orbits and Applications*, WORLD SCIENTIFIC, 2003. doi:10.1142/9789812704849_0023.

[17] Sagliano, M., "Generalized hp Pseudospectral-Convex Programming for Powered Descent and Landing," *Journal of Guidance, Control, and Dynamics*, Vol. 42, No. 7, 2019, pp. 1562–1570. doi:10.2514/1.g003731.

[18] Malyuta, D., Reynolds, T. P., Szmuk, M., Lew, T., Bonalli, R., Pavone, M., and Açıkmeşe, B., "Convex Optimization for Trajectory Generation: A Tutorial on Generating Dynamically Feasible Trajectories Reliably and Efficiently," *IEEE Control Systems*, Vol. 42, No. 5, 2022, pp. 40–113. doi:10.1109/mcs.2022.3187542.

[19] Woffinden, D., Robinson, S., Williams, J., and Putnam, Z., "Linear Covariance Analysis Techniques to Generate Navigation and Sensor Requirements for the Safe and Precise Landing - Integrated Capabilities Evolution (SPLICE) Project," *AIAA Scitech 2019 Forum*, AIAA 2019-0662, San Diego, CA, 2019.

[20] Clohessy, W. H., and Wiltshire, R., "Terminal Guidance System for Satellite Rendezvous," *Journal of the Aero/Space Sciencess*, Vol. 27, 1960, pp. 653—658.

[21] Margolis, B. W. L., and Lyons, K. R., "Condor," New Technology Report ARC-18996-1, Nov. 2023.

[22] Wächter, A., and Biegler, L. T., "On the Implementation of an Interior-Point Filter Line-Search Algorithm for Large-Scale Nonlinear Programming," *Mathematical Programming*, Vol. 106, No. 1, 2005. doi:10.1007/s10107-004-0559-y.

[23] Hindmarsh, A. C., Brown, P. N., Grant, K. E., Lee, S. L., Serban, R., Shumaker, D. E., and Woodward, C. S., "SUNDIALS: Suite of nonlinear and differential/algebraic equation solvers," *ACM Transactions on Mathematical Software (TOMS)*, Vol. 31, No. 3, 2005, pp. 363–396. doi:10.1145/1089014.1089020.

[24] Margolis, B. W. L., and Woffinden, D., "Co-Optimization of Navigation System Requirements and Trajectory Design Using a Sweeping Gradient Method and Linear Covariance Analysis," *AIAA and AAS Astrodynamics Specialist Conference*, Broomfield, CO, 2024.