AAS 26-973

CLOSED-LOOP NONLINEAR COVARIANCE ANALYSIS FOR GN&C APPLICATIONS

Quinn P. Moon[*], Brandon A. Jones[†], and Ryan P. Russell [‡]

The standard format of Closed-Loop Linear Covariance Analysis (LinCov) is modified to include higher-order effects. The propagation step includes higher-order State Transition Tensors (STTs); an alternative augmented state is used to avoid splitting the STTs along continuous trajectory segments. The higher-order sensitivities of spacecraft targeting maneuvers are incorporated, capturing the dominant nonlinear effects arising from engine execution errors and nonlinear targeting solutions. The consistency between the onboard filter and the true estimation errors are found by approximating the measurement update step to higher-order, yielding an analytic method to determine appropriate measurement underweighting methods. The algorithm is called a *Generalized Higher-Order Covariance Analysis*, or *GenCov* and provides an alternative to Monte Carlo simulations by validating LinCov when nonlinearities are insignificant or by replacing LinCov when significant nonlinearities are considered. While slower than LinCov, GenCov remains orders of magnitude faster than Monte Carlo simulations.

INTRODUCTION

Linear Covariance (LinCov) analysis is an increasingly common alternative to Monte Carlo simulations for closed-loop spacecraft mission analysis [1]. Provided the underlying system has insignificant nonlinearities, LinCov provides an accurate and rapid approximation of the true state dispersions and estimation errors based on closed-loop targeting [1]. However, LinCov's accuracy deteriorates when either larger state dispersions or stronger nonlinear equations are present; for significant nonlinear environments, LinCov either cannot be used at all or requires careful validation from Monte Carlo simulations. Here, the LinCov paradigm is extended to include nonlinearities, leading to a closed-loop nonlinear covariance analysis. As a significant body of literature and operational software tools are built upon the LinCov concept [1–3]—and the various first-order approximations in LinCov are generalized to a higher-, yet not strictly equal, order—the resulting algorithm is referred to as a *Generalized Higher-Order Covariance Analysis*, or *GenCov*.

Closed-loop analyses implementing and verifying LinCov are prevalent throughout the literature. Most studies, however, are applied to mission environments where the underlying functions are either linear or dominated by linear effects. Indeed certain missions are classically ideal for LinCov. A few of these significant examples from the literature are highlighted here. Orbital rendezvous missions frequently linearize about the target spacecraft trajectory, leading to both linear dynamics and linear rendezvous maneuvers [1,2]. Powered descent missions to the Moon [4,5] and Mars [6]

[*]PhD Student, Aerospace Engineering and Engineering Mechanics, quinnpmoon@utexas.edu, http://orcid.org/0000-0002-6766-7577
[†]Associate Professor, brandon.jones@utexas.edu, http://orcid.org/0000-0003-3480-6320
[‡]Professor, ryan.russell@austin.utexas.edu, http://orcid.org/0000-0001-7672-0408

utilize the linear Apollo descent guidance law. Trajectory Correction Maneuvers (TCMs) for final state targeting and station keeping linearize about the nominal trajectory with the State Transition Matrix (STM) [7, 8], providing a linear targeting law. Interplanetary missions similarly use linearized B-plane targeting [9]. For measurement updates with the Extended Kalman Filter (EKF), LinCov provides an accurate approximation of the true estimation errors as the EKF itself relies on a local linearization about the filter estimate [10]. In summary, LinCov is well suited for missions where linearization is used for in-flight systems or models, including the dynamics, targeting and measurement updates.

LinCov's ability to handle *nonlinear* systems is dependent on either weak or insignificant nonlinearities. Nonlinearities are weak when either higher-order terms are small (a weak *real* nonlinearity) or when state dispersions are small (a weak *induced* nonlinearity) [11]. Strong nonlinearities may be *insignificant* in the presence of multiple linear, Gaussian error sources [11]. Closed-loop mission analysis assumes Gaussian errors arise from multiple sources—including measurement errors, IMU errors, engine execution errors, and random process noise—causing the strong system nonlinearities to become insignificant. As such, LinCov is able to approximate the behavior of nonlinear systems. Notable missions with strong nonlinear dynamics and nonlinear guidance laws assessed with LinCov include lunar ascent [12] atmospheric entry [13, 14], and navigation analysis in near rectilinear halo orbits [15].

As the statistics provided by LinCov are approximations, LinCov is frequently validated against an empirical Monte Carlo analysis before further implementation into autonomous systems or optimization routines; many of the cited examples include Monte Carlo validations. Analytic assessments of a system's nonlinearity provide a faster alternative to Monte Carlo studies. Higher-order expansions of the dynamics with State Transition Tensors (STTs) are common, with nonlinearity metrics found via statistical moments [16] and nonlinearity indices [17]. Additional nonlinearity metrics include tensor eigenpairs [18], induced tensor norms [19, 20], and unscented transforms [21]. Analytic nonlinear checks of linearized systems are powerful, but including higher-order effects into a complete closed-loop framework remains a ripe area of research [22–24].

Efforts to incorporate higher-order terms into *individual* aspects of closed-loop mission design and analysis are more common. Uncertainty propagation beyond linearization methods is a rich field of study, including STT expansions [16, 25], directional STT expansions [26], unscented transforms [21], polynomial chaos expansions [27], conic based analytic models [28, 29], and Gaussian mixture models [30]. Nonlinear guidance methods are intimately tied to STTs, both in providing the targeting scheme [16, 31] and analyzing the subsequent impact on the nonlinear dynamics [32]. Higher-order filtering combines aspects of nonlinear propagation and nonlinear measurement approximations, including the Higher-Order Extended Kalman Filters (HOEKF) [33] and the Unscented Kalman Filter (UKF) [21]. However, specific efforts to combine the higher-order expansions of the dynamics, targeting, and measurement update into a single algorithm, like LinCov, for closed-loop mission analysis are less common. Work by the authors to incorporate higher-order terms into LinCov include a second-order STT expansion [22] and a second-order expansion of impulsive correction maneuvers fired with multiplicative engine errors [23].

In this work, a single algorithm generalized to higher-order is introduced for nonlinear closed-loop mission analysis, combining individual aspects from the existing literature. The resulting tool, GenCov, incorporates higher-order terms but maintains the basic structure of LinCov. The remaining agenda of the current work follows. First, relevant background from the literature on higher-order partials and Gaussian moments is provided. Second, an overview of closed-loop mission design and LinCov analysis is presented, with discussion provided on the relevant shortcomings of LinCov. Third, the primary components of LinCov (state propagation, instantaneous state correction, and state measurement updates) are individually scrutinized and higher-order terms incorporated. Lastly, a case study is performed for a satellite orbiting Bennu, highlighting the efficacy of GenCov compared against LinCov.

RELEVANT BACKGROUND

The current work relies heavily on the theory and application of tensors, STTs about a reference trajectory, and mapping Gaussian moments with higher-order expansions. This section provides an overview of the necessary equations and notations, summarizing contributions from the literature for completeness in the current work [16, 34].

Tensors and Einstein Notation

The notation necessary to include higher-order tensors in GenCov is presented in this section, specifically with respect to dynamics and STTs; for further reading see [16]. The nonlinear dynamics $\boldsymbol{\phi}$ (representing the numerically integrated ordinary differential equations) map an initial state $\boldsymbol{x}_0$ to the final state $\boldsymbol{x}_f$ according to $\boldsymbol{x}_f = \boldsymbol{\phi}(\boldsymbol{x}_0)$. The STTs represent the partials of the flow about a reference trajectory, i.e., the pth-order sensitivity of the final state $\boldsymbol{x}_f$ with respect to the initial state $\boldsymbol{x}_0$ is [16]

$$\phi^{i,k_1\cdots k_p} := \frac{\partial x_f^i}{\partial x_0^{k_1}\cdots\partial x_0^{k_p}}. \tag{1}$$

The superscript notation $(i, k_1\cdots k_p)$ denotes the partial of the ith state with respect to the states k_1 through k_p. The final state dispersion can be approximated with the sth-order Taylor series [16]

$$\delta x_f^i \approx \sum_{p=1}^{s} \frac{1}{p!}\phi^{i,k_1\cdots k_p}\delta x_0^{k_1}\cdots\delta x_0^{k_p}, \tag{2}$$

where repeated indices in the superscript are summed using Einstein's convention. The notation in Eqs. 1 and 2 is valid for any nonlinear, differentiable function.

For a differentiable nonlinear multivariate function $\boldsymbol{f}$, the first-order partial of $\boldsymbol{f}$ with respect to a vector $\boldsymbol{x}$ is commonly denoted as $\boldsymbol{F_x}$, yielding a matrix [1–4]. The notation $\boldsymbol{F_{xx}}$ thus denotes the second-order tensor of $\boldsymbol{f}$ with respect to $\boldsymbol{x}$. As tensor algebra is facilitated with Einstein's notation, it is more convenient to express matrices and tensors according to their individual elements as

$$f_{xx}^{i,jk} := \frac{\partial f^i}{\partial x^j \partial x^k}, \tag{3}$$

where the partial of the ith component of $\boldsymbol{f}$ is taken with respect to the jth and kth component of $\boldsymbol{x}$. In similar fashion, the notation commonly utilized for sub-matrices of the STM, $\boldsymbol{\Phi_{r,v}}$ denotes the first-order sensitivity of the final position $\boldsymbol{r}$ with respect to the initial velocity $\boldsymbol{v}$, is extended to tensor notation. Eq. 4 denotes the second-order partial of the ith final position with respect to the jth initial position and the kth initial velocity:

$$\phi_{r,rv}^{i,jk} := \frac{\partial r_f^i}{\partial r_0^j \partial v_0^k}. \tag{4}$$

Lastly, the elements of any matrix can be expressed using Einstein notation according to $\boldsymbol{A} = A^{ij}$. The partial of the ijth element of a matrix with respect to the kth state of $\boldsymbol{x}$ is denoted as

$$A_x^{ij,k} = \frac{\partial A^{ij}}{\partial x^k}. \tag{5}$$

As vectors and matrices are frequently marked with superscripts for closed-loop analyses, the superscript notations "-" and "+" commonly denote before and after, respectively, an instantaneous change to the state occurs, the *i*th element of a vector $\boldsymbol{x}^+$ is noted as $[x^+]^i$.

Mapping Gaussian Moments

Random variables in aerospace applications are frequently assumed to be Gaussian, as the Gaussian distribution is completely described by its mean $\boldsymbol{m}$ and covariance $\boldsymbol{P}$. LinCov follows the Gaussian assumption, as linearization preserves Gaussianity. Expanding the Taylor series and including higher Gaussian moments improves the approximation of the mean and covariance; for further reading, see [16, 33]. For a normally distributed random variable $\delta\boldsymbol{x} \sim \mathcal{N}(\delta\boldsymbol{m}, \boldsymbol{P})$, the higher-order moments are completely described with the mean and covariance using Isserlis' theorem [34]. The first four non-central Gaussian moments are reported as [16]

$$E[\delta x^i] = \delta m^i, \tag{6}$$

$$E[\delta x^i \delta x^j] = \delta m^i \delta m^j + P^{ij}, \tag{7}$$

$$E[\delta x^i \delta x^j \delta x^k] = \delta m^i \delta m^j \delta m^k + \left(\delta m^i P^{jk} + \delta m^j P^{ik} + \delta m^k P^{ij}\right), \tag{8}$$

$$\begin{aligned} E[\delta x^i \delta x^j \delta x^k \delta x^l] &= \delta m^i \delta m^j \delta m^k \delta m^l + P^{ij} P^{kl} + P^{ik} P^{jl} + P^{il} P^{jk} + \\ &\left(\delta m^i \delta m^j P^{kl} + \delta m^i \delta m^k P^{jl} + \delta m^j \delta m^k P^{il} + \delta m^i \delta m^l P^{jk} + \delta m^j \delta m^l P^{ik} + \delta m^k \delta m^l P^{ij}\right). \end{aligned} \tag{9}$$

The mean and covariance of a nonlinear function are expressed, using the STTs as an example, to the sth-order via [16]

$$\delta m_f^i \approx \sum_{p=1}^{s} \frac{1}{p!} \phi^{i,k_1 \cdots k_p} E\left[\delta x_0^{k_1} \cdots \delta x_0^{k_p}\right], \tag{10}$$

$$P_f^{ij} \approx \left(\sum_{p=1}^{s} \sum_{q=1}^{s} \frac{1}{p!q!} \phi^{i,k_1 \cdots k_p} \phi^{j,l_1 \cdots l_q} E\left[\delta x_0^{k_1} \cdots \delta x_0^{k_p} \delta x_0^{l_1} \cdots \delta x_0^{l_q}\right] \right) - \delta m_f^i \delta m_f^j. \tag{11}$$

GENCOV FRAMEWORK IN RELATION TO LINCOV

In this section, the necessary framework and notations for closed-loop covariance analysis is provided. The framework for GenCov is identical to that of LinCov [1–4] and is included in the current work for completeness and to give context to the advancements provided by GenCov.

Closed-loop mission analysis is the study of a spacecraft's behavior with respect to a nominal trajectory $\bar{\boldsymbol{x}}$. The true state $\boldsymbol{x}$ is unknown, estimated using filtering techniques to provide a filter state, or *nav* state $\hat{\boldsymbol{x}}$. Three quantities of interest for closed-loop mission analysis are the *true dispersions* $\delta\boldsymbol{x}$, the *nav dispersions* $\delta\hat{\boldsymbol{x}}$, and the *true estimation errors* $\delta\boldsymbol{e}$, defined as [1,2]

$$\delta\boldsymbol{x} = \boldsymbol{x} - \bar{\boldsymbol{x}}, \quad \delta\hat{\boldsymbol{x}} = \hat{\boldsymbol{x}} - \bar{\boldsymbol{x}}, \quad \delta\boldsymbol{e} = \boldsymbol{x} - \hat{\boldsymbol{x}}. \tag{12}$$

The mean of the dispersions in Eq. 12 are, in order, $\delta \boldsymbol{m_x}$, $\delta \boldsymbol{m_{\hat{x}}}$, and $\delta \boldsymbol{m_e}$. Via a Monte Carlo study, the individual dispersions and estimation errors are collected from the N runs and the covariances empirically computed as

$$\boldsymbol{D}_{\text{true}} \approx \frac{1}{N-1}\sum_{i=1}^{N}(\delta \boldsymbol{x}_i - \delta \boldsymbol{m_x})(\delta \boldsymbol{x}_i - \delta \boldsymbol{m_x})^{\mathsf{T}}, \quad \boldsymbol{D}_{\text{nav}} \approx \frac{1}{N-1}\sum_{i=1}^{N}(\delta \hat{\boldsymbol{x}}_i - \delta \boldsymbol{m_{\hat{x}}})(\delta \hat{\boldsymbol{x}}_i - \delta \boldsymbol{m_{\hat{x}}})^{\mathsf{T}},$$

$$\boldsymbol{P} \approx \frac{1}{N-1}\sum_{i=1}^{N}(\delta \boldsymbol{e}_i - \delta \boldsymbol{m_e})(\delta \boldsymbol{e}_i - \delta \boldsymbol{m_e})^{\mathsf{T}}, \tag{13}$$

The empirical Monte Carlo statistics are treated as the actual representation of the dispersions, provided sufficiently large N.

LinCov linearizes about the nominal trajectory to provide analytic approximations of the desired covariances in Eq. 13; LinCov assumes the means are zero. The linearized augmented state $\delta \boldsymbol{x_a}$ consists of the true dispersion $\delta \boldsymbol{x}$ and the nav dispersion $\delta \hat{\boldsymbol{x}}$, leading to an augmented covariance $\boldsymbol{\mathcal{C}_a}$. The covariances of interest in Eq. 13 are obtained by extracting the corresponding components of $\boldsymbol{\mathcal{C}_a}$ according to [1,2]

$$\boldsymbol{D}_{true} = \left[\boldsymbol{I}_{n\times n},\ \boldsymbol{0}_{n\times\hat{n}}\right] \boldsymbol{\mathcal{C}_a} \left[\boldsymbol{I}_{n\times n},\ \boldsymbol{0}_{n\times\hat{n}}\right]^{\mathsf{T}}, \quad \boldsymbol{D}_{nav} = \left[\boldsymbol{0}_{\hat{n}\times n},\ \boldsymbol{I}_{\hat{n}\times\hat{n}}\right] \boldsymbol{\mathcal{C}_a} \left[\ \boldsymbol{0}_{\hat{n}\times n},\ \boldsymbol{I}_{\hat{n}\times\hat{n}}\ \right]^{\mathsf{T}},$$

$$\boldsymbol{P} = \left[\boldsymbol{M_x},\ -\boldsymbol{I}_{\hat{n}\times\hat{n}}\right] \boldsymbol{\mathcal{C}_a} \left[\boldsymbol{M_x},\ -\boldsymbol{I}_{\hat{n}\times\hat{n}}\right]^{\mathsf{T}}, \tag{14}$$

where n and $\hat{n}$ are the number of true states and nav states, respectively. The matrices $\boldsymbol{I}$ and $\boldsymbol{0}$ are the identity matrix and the zero matrix, respectively. Generally, there are more true states than nav states, $n \geq \hat{n}$, thus, the matrix $\boldsymbol{M_x}$ consists of identity matrices that map the desired elements from the true state $\boldsymbol{x}$ to the nav state $\hat{\boldsymbol{x}}$, yielding the estimation error $\delta \boldsymbol{e}$.

An additional pseudo-metric is the *onboard estimation error* $\delta \hat{\boldsymbol{e}}$, which is never calculated directly [2]. The covariance of $\delta \hat{\boldsymbol{e}}$ is represented by the onboard filter covariance $\hat{\boldsymbol{P}}$. The formulation of $\delta \hat{\boldsymbol{e}}$ is necessary to develop the onboard filter update equations, whereas the true estimation covariance $\boldsymbol{P}$ is the performance of the estimation errors. The onboard filter covariance $\hat{\boldsymbol{P}}$ is designed to represent the true estimation error covariance $\boldsymbol{P}$, but may fail to do so if significant system nonlinearities are present; this inconsistency is one of the motivations for the current work.

Addressing the Shortcomings of LinCov with GenCov

Some aspects of closed-loop mission analysis taken for granted in LinCov only become apparent when a Monte Carlo analysis is performed. Beyond simply misrepresenting the covariance, LinCov omits the mean of the dispersions in Eq. 12. LinCov assumes the initial means are zero, starting at the nominal state, and never become non-zero. However, changes to the mean are captured with the second-order approximation in Eq. 10. The mean true dispersion $\delta \boldsymbol{m_x}$ primarily arises from nonlinear dynamics and spacecraft targeting schemes. For example, first-order targeting methods once applied to the true state and propagated through the nonlinear dynamics yield a non-zero mean dispersion $\delta \boldsymbol{m_x}$ ultimately omitted by LinCov. The filtering bias, represented by $\delta \boldsymbol{m_e}$, is similarly ignored in a LinCov study.

The onboard filter consistency can be quantified with LinCov using the Normalized Estimation Error Squared (NEES), or ε, but may be inaccurate since the filter bias $\delta \boldsymbol{m_e}$ is omitted. For Monte

Carlo studies, the NEES is calculated using the filter covariance $\hat{\boldsymbol{P}}$ and the true estimation error $\delta\boldsymbol{e}$ according to [10]

$$\varepsilon = \delta\boldsymbol{e}^{\mathsf{T}}\hat{\boldsymbol{P}}^{-1}\delta\boldsymbol{e}. \tag{15}$$

The expected value of ε, or $\bar{\varepsilon}$, is equal to the number of filter states $\hat{n}$. Within the LinCov framework, the expected NEES $\bar{\varepsilon}$ is expressed in tensor notation using the true estimation errors and the filter design covariance $\hat{\boldsymbol{P}}$ as

$$\bar{\varepsilon} = \left[\hat{P}^{-1}\right]^{ij}\left(P^{ij} - \delta m_e^i \delta m_e^j\right). \tag{16}$$

In LinCov, if the expected NEES value $\bar{\varepsilon}$ is greater than the number of filter states $\hat{n}$, then the onboard filter is inconsistent. LinCov captures filter inconsistency due to dynamic or measurement mismodeling [1]. However, even a filter with no modeling error can become biased or inconsistent due to the filter's design. LinCov omits the mean estimation error $\delta\boldsymbol{m_e}$, misrepresenting the bias and inconsistency of the onboard filter.

Expanding LinCov to higher-order captures the means $\{\delta\boldsymbol{m_x}, \delta\boldsymbol{m_e}\}$, and properly represents the augmented covariance $\boldsymbol{\mathcal{C}_a}$ and the expected NEES metric $\bar{\varepsilon}$. With the expanded GenCov tool, the viability of both targeting and filter strategies are properly addressed. To incorporate higher-order terms into LinCov, the three primary components of closed-loop mission analysis are considered. First, *propagation* of the state through nonlinear dynamics. Second, the instantaneous *correction* of the state through translational maneuvers. Third, the *measurement update* of the nav state with an onboard filter. Each component exhibits its own challenges for implementation and performance, and each is discussed in detail in the following sections.

COVARIANCE PROPAGATION

State and covariance propagation with higher-order STTs is a well studied field [16,33]. In Fig. 1, the nonlinear spacecraft dispersion is depicted along with the first- and second-order approximation. LinCov approximates the dynamics to first-order using the nominal STMs, but expanding to higher-order better represents the nonlinear dynamics. Though STTs are well studied for spacecraft propagation and targeting [16, 26, 31], incorporating STTs to the propagation step in LinCov is a more recent endeavor [22]. This section discusses how higher-order STTs are incorporated into the LinCov framework by introducing a modified augmented state and covariance. The STT expansions are presented to second-order for brevity, but can be extended to arbitrary order.

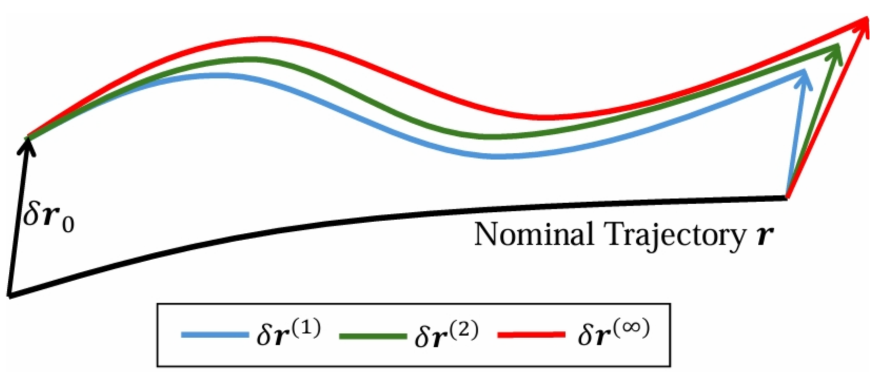


Figure 1: First- and second-order approximations $\{\delta\boldsymbol{r}^{(1)}, \delta\boldsymbol{r}^{(2)}\}$ of the nonlinear dispersion $\delta\boldsymbol{r}^{(\infty)}$.

One method of incorporating STTs into LinCov expands the dynamics to higher-order between measurement times t_{m_1} and t_{m_2} [22]. The augmented state dispersion is approximated to second-order with the augmented STT $\phi_a^{i,k_1\cdots k_p}$ according to

$$\delta x_a^i(t_{m_2}) = \phi_a^{i,\alpha}(t_{m_2}, t_{m_1})\delta x_a^\alpha(t_{m_1}) + \frac{1}{2}\phi_a^{i,\alpha\beta}(t_{m_2}, t_{m_1})\delta x_a^\alpha(t_{m_1})\delta x_a^\beta(t_{m_1}). \tag{17}$$

The approach in Eq. 17 under-utilizes the STTs by splitting the flow of the true dynamics whenever a measurement is taken, losing information from the STT expansion [33]. For first-order dynamics, no fidelity is lost when chaining the STMs. However, chaining second-order dynamics introduces higher-order terms that are neglected when the chained expression is truncated to second-order. To maintain the flow of the true dynamics, an alternative augmented state is introduced to avoid interrupting the true state propagation.

The true state $\boldsymbol{x}$ is propagated continuously until an instantaneous correction is performed (e.g., an impulsive Δv maneuver). The nav state $\hat{\boldsymbol{x}}$, equal to $\boldsymbol{x} + \delta\boldsymbol{e}$, is more frequently updated through measurement processing, though only the measurement error is altered. Therefore, a more effective implementation of the STTs is introduced by decoupling the estimation error $\delta\boldsymbol{e}$ from the nav state. Instead of using Eq. 17, the true dispersion is propagated from an initial time t_0 to the next discontinuity at time $t_{\Delta v}$, or until final time t_f if no maneuvers are present, according to

$$\delta x^i(t_{\Delta v}) = \phi^{i,\alpha}(t_{\Delta v}, t_0)\delta x^\alpha(t_0) + \frac{1}{2}\phi^{i,\alpha\beta}(t_{\Delta v}, t_0)\delta x^\alpha(t_0)\delta x^\beta(t_0). \tag{18}$$

The estimation errors are propagated from one measurement time t_{m_1} to the next measurement time t_{m_2} according to

$$\delta e^i(t_{m_2}) = \phi^{i,\alpha}(t_{m_2}, t_{m_1})\delta e^\alpha(t_{m_1}) + \frac{1}{2}\phi^{i,\alpha\beta}(t_{m_2}, t_{m_1})\delta e^\alpha(t_{m_1})\delta e^\beta(t_{m_1}). \tag{19}$$

A simple linear mapping decouples the estimation errors $\delta\boldsymbol{e}$ from the nav state, creating an alternative augmented state dispersion $\delta\boldsymbol{x}_{\boldsymbol{a}}'$ via

$$\delta\boldsymbol{x}_{\boldsymbol{a}}' := \begin{bmatrix} \delta\boldsymbol{x} \\ \delta\boldsymbol{e} \end{bmatrix} = \begin{bmatrix} \boldsymbol{I}_{n\times n} & \boldsymbol{0}_{n\times\hat{n}} \\ \boldsymbol{M}_{\boldsymbol{x}} & -\boldsymbol{I}_{\hat{n}\times\hat{n}} \end{bmatrix} \begin{bmatrix} \delta\boldsymbol{x} \\ \delta\hat{\boldsymbol{x}} \end{bmatrix},$$

$$\delta\boldsymbol{x}_{\boldsymbol{a}}' = \boldsymbol{M}_{\boldsymbol{a}}\delta\boldsymbol{x}_{\boldsymbol{a}}, \tag{20}$$

where $\boldsymbol{M}_{\boldsymbol{x}}$ maps the true states to the nav states, previously discussed in Eq. 14. Conveniently, the matrix $\boldsymbol{M}_{\boldsymbol{a}}$ maps the alternative augmented state $\delta\boldsymbol{x}_{\boldsymbol{a}}'$ back to $\delta\boldsymbol{x}_{\boldsymbol{a}}$ ($\boldsymbol{M}_{\boldsymbol{a}}^2 = \boldsymbol{I}$). The mean and covariance of $\delta\boldsymbol{x}_{\boldsymbol{a}}'$ are expressed as $\delta\boldsymbol{m}_{\boldsymbol{a}}'$ and $\boldsymbol{\mathcal{C}}_{\boldsymbol{a}}'$, respectively. Thus, the higher-order mapping using the STTs remains uninterrupted until a maneuver is fired.

The steps of the propagation phase in GenCov, implementing Eqs. 18 and 19, are shown in Alg. 1. Essentially, the propagation step is performed linearly between arbitrary times t_k to t_{k+1}, adding in the higher-order effects calculated from time t_0 to t_{k+1}. Though more intricate, the GenCov propagation in Alg. 1 more accurately captures the state dispersions than the sequential method in Eq 17. The higher-order propagation is intimately tied to the targeting phase of GenCov, discussed in greater detail in the next section

Algorithm 1 GenCov propagation algorithm: no targeting performed.

Input: a) initial mean and covariance $\{\delta \boldsymbol{m}_{\boldsymbol{a}}(t_0), \boldsymbol{\mathcal{C}}_{\boldsymbol{a}}(t_0)\}$,
b) measurement indices k in time array t,
c) maneuver time $t_{\Delta v}$.

while $t_k \leq t_{\Delta v}$ **do**
  $\{\delta \boldsymbol{m}_{\boldsymbol{a}}(t_{k+1}), \boldsymbol{\mathcal{C}}_{\boldsymbol{a}}(t_{k+1})\} \leftarrow$first-order propagation of $\{\delta \boldsymbol{m}_{\boldsymbol{a}}(t_k), \boldsymbol{\mathcal{C}}_{\boldsymbol{a}}(t_k)\}$. ▷ See LinCov [1–4]
  $\{\delta \boldsymbol{m}'_{\boldsymbol{a}}(t_{k+1}), \boldsymbol{\mathcal{C}}'_{\boldsymbol{a}}(t_{k+1})\} \leftarrow$ mapping $\boldsymbol{M}_{\boldsymbol{a}}$ of $\{\delta \boldsymbol{m}_{\boldsymbol{a}}(t_{k+1}), \boldsymbol{\mathcal{C}}_{\boldsymbol{a}}(t_{k+1})\}$. ▷ Eq. 20
  $\{\delta \boldsymbol{m}_{\boldsymbol{x}}(t_{k+1}), \boldsymbol{D}_{\boldsymbol{xx}}(t_{k+1})\}^{(p)} \leftarrow p$th-order mapping of $\{\delta \boldsymbol{m}_{\boldsymbol{a}}(t_0), \boldsymbol{\mathcal{C}}_{\boldsymbol{a}}(t_0)\}$. ▷ Eqs. 10, 11, and 18
  $\{\delta \boldsymbol{m}'_{\boldsymbol{x}}(t_{k+1}), \boldsymbol{D}'_{\boldsymbol{xx}}(t_{k+1})\} \leftarrow \{\delta \boldsymbol{m}_{\boldsymbol{x}}(t_{k+1}), \boldsymbol{D}_{\boldsymbol{xx}}(t_{k+1})\}^{(p)}$
  Modify $\{\delta \boldsymbol{m}'_{\boldsymbol{e}}(t_{k+1}), \boldsymbol{D}'_{\boldsymbol{ee}}(t_{k+1})\}$ by adding higher-order terms. ▷ Eqs. 10, 11, and 19
  $\{\delta \boldsymbol{m}_{\boldsymbol{a}}(t_{k+1}), \boldsymbol{\mathcal{C}}_{\boldsymbol{a}}(t_{k+1})\} \leftarrow$ mapping $\boldsymbol{M}_{\boldsymbol{a}}$ of $\{\delta \boldsymbol{m}'_{\boldsymbol{a}}(t_{k+1}), \boldsymbol{\mathcal{C}}'_{\boldsymbol{a}}(t_{k+1})\}$. ▷ Eq. 20
end while

CORRECTION STEP AND NONLINEAR TARGETING

The correction step refers to any instantaneous change to the true state $\boldsymbol{x}$, subsequently tracked by an instantaneous change to the nav state $\hat{\boldsymbol{x}}$. The corrections of interest in this work are translational maneuvers altering the spacecraft's velocity. The current section provides an overview of closed-loop spacecraft targeting errors [1, 2], highlighting the necessity for instantaneous higher-order covariance corrections [23]. Next, the higher-order sensitivities of nonlinear targeting solutions are presented based on the reference STTs. Lastly, the higher-order sensitivities of the dynamics implicit on the fired maneuvers are derived via the multivariate implicit value theorem [35], introduced for closed-loop targeting analysis. As before, the higher-order expansions are presented to second-order for brevity but can be expanded to arbitrary order; see Appendix A.

Closed-Loop Correction Modeling

For closed-loop maneuvers, the commanded maneuver $\boldsymbol{u}_c$ is calculated from the nav state $\hat{\boldsymbol{x}}$ and then corrupted according to the maneuver execution model proposed by Geller [1]. The resulting true maneuver $\boldsymbol{u}$ is modeled according to [2]:

$$\boldsymbol{u} = (\boldsymbol{I}_{3\times 3} + \mathrm{diag}(\boldsymbol{s}_{\boldsymbol{u}}) + [\boldsymbol{e}_{\boldsymbol{u}}\times])\, \boldsymbol{u}_c + \boldsymbol{b}_{\boldsymbol{u}}, \tag{21}$$

where the parameter $\boldsymbol{s}_{\boldsymbol{u}}$ is the three-dimensional scaling error, $\boldsymbol{e}_{\boldsymbol{u}}$ is the misalignment error, and $\boldsymbol{b}_{\boldsymbol{u}}$ is the engine bias. The true maneuver $\boldsymbol{u}$ in Eq. 21 is added directly to the true velocity. The true maneuver is sensed by the onboard IMUs and an instantaneous, integrated measurement is expressed as

$$\boldsymbol{y} = (\boldsymbol{I}_{3\times 3} + \mathrm{diag}(\boldsymbol{s}_{\boldsymbol{y}}) + [\boldsymbol{e}_{\boldsymbol{y}}\times])\, \boldsymbol{u} + \boldsymbol{b}_{\boldsymbol{y}}, \tag{22}$$

where the parameter $\boldsymbol{s}_{\boldsymbol{y}}$ is the IMU scaling error, $\boldsymbol{e}_{\boldsymbol{y}}$ is the IMU misalignment error, and $\boldsymbol{b}_{\boldsymbol{y}}$ is the IMU bias. The IMU nav states provide an estimated maneuver $\hat{\boldsymbol{u}}$, and the nav velocity updated accordingly [1]. The estimated maneuver is expressed as

$$\hat{\boldsymbol{u}} = (\boldsymbol{I}_{3\times 3} - \mathrm{diag}(\hat{\boldsymbol{s}}_{\boldsymbol{y}}) - [\hat{\boldsymbol{e}}_{\boldsymbol{y}}\times]) \left(\boldsymbol{y} - \hat{\boldsymbol{b}}_{\boldsymbol{y}}\right). \tag{23}$$

The maneuver execution and IMU models in Eqs. 21- 23 are implemented for closed-loop targeting in most LinCov analyses, with minor variations presented throughout the literature to include attitude misalignment or random noise.

The nonlinear correction altering the true and nav velocity is expressed as a nonlinear function of the augmented state

$$\boldsymbol{x}_{\boldsymbol{a}}^{+} = \boldsymbol{d}(\boldsymbol{x}_{\boldsymbol{a}}^{-}), \tag{24}$$

where the superscripts "-" and "+" denote before and after the correction. The nonlinear function $\boldsymbol{d}$ representing a TCM fired based on the nav dispersion $\delta\hat{\boldsymbol{x}}$ is called a stochastic maneuver or zero-nominal maneuver; under nominal conditions no correction is necessary. In LinCov, the first-order partial $d_{x_a}^{i,j}$ omits the multiplicative engine errors $\{\boldsymbol{s_u}, \boldsymbol{e_u}\}$ and IMU errors $\{\boldsymbol{s_y}, \boldsymbol{e_y}, \hat{\boldsymbol{s}}_{\boldsymbol{y}}, \hat{\boldsymbol{e}}_{\boldsymbol{y}}\}$. Once analyzed at the nominal state for a stochastic TCM, the first-order sensitivities with respect to the multiplicative engine and IMU errors become zero. The engine and IMU multiplicative errors are nonlinear, necessitating a higher-order covariance correction. The second-order expansion of Eq. 24 is

$$\left[\delta x_a^{+}\right]^{i} = d_{x_a}^{i,\alpha}\left[\delta x_a^{-}\right]^{\alpha} + \frac{1}{2} d_{x_a x_a}^{i,\alpha\beta}\left[\delta x_a^{-}\right]^{\alpha}\left[\delta x_a^{-}\right]^{\beta}. \tag{25}$$

Implementing a quadratic expansion captures the dominant terms of the multiplicative engine and IMU errors.

In summary, the impact of the engine errors $\{\boldsymbol{s_u}, \boldsymbol{e_u}\}$ on the true dispersions $\delta\boldsymbol{x}$ and the IMU errors $\{\boldsymbol{s_y}, \boldsymbol{e_y}, \hat{\boldsymbol{s}}_{\boldsymbol{y}}, \hat{\boldsymbol{e}}_{\boldsymbol{y}}\}$ on the estimation errors $\delta\boldsymbol{e}$ are omitted in LinCov, but represented in GenCov. A study on the engine execution errors with *linearized* dynamics is presented by the authors in [23]. Additional higher-order terms arise from instantaneous corrections due to nonlinear dynamics and targeting methods.

Closed-Loop Higher-Order Targeting Sensitivities

For closed-loop targeting about a reference trajectory, the most common objective is to steer the spacecraft back to the reference at some final time t_f. Given an initial nav dispersion $\delta\hat{\boldsymbol{r}}_0$, the root-solved nonlinear targeting solution $\boldsymbol{u}^*$ drives the final dispersion $\delta\hat{\boldsymbol{r}}_f$ to zero under nonlinear dynamics. The nonlinear targeting solution $\boldsymbol{u}^*$ is numerically obtained through shooting methods [31]. Alternatively, first-order maneuvers $\boldsymbol{u}_{(1)}$ [7] approximate the nonlinear targeting solution $\boldsymbol{u}^*$ using the reference STMs

$$\boldsymbol{u}_{(1)} = -\boldsymbol{\Phi}_{\boldsymbol{rv}}^{-1}\boldsymbol{\Phi}_{\boldsymbol{rr}}\delta\hat{\boldsymbol{r}} - \delta\hat{\boldsymbol{v}}. \tag{26}$$

First-order maneuvers are ideal for LinCov as the maneuver itself is linear. Assuming no measurement errors or engine execution errors, the first-order maneuver reaches the target state according to *linear* dynamics, depicted below in Fig. 2. The first-order maneuver applied to the true state and propagated through the nonlinear dynamics misses the final target. This discrepancy is ignored in LinCov as a linear approximation is implemented. Expanding the dynamics to higher-order in GenCov better represents the nonlinear dynamics, depicted in Fig. 2. Though first-order maneuvers are appealing for in-flight use due to their simplicity, the nonlinear targeting solution $\boldsymbol{u}^*$ may be required under stronger nonlinear conditions.

Figure 3 depicts the nonlinear dynamics, and the first- and second-order approximations of the dynamics, after the maneuver $\boldsymbol{u}^*$ is fired under perfect conditions, i.e., no engine execution errors or state estimation errors. Note that the linear and quadratic approximations reach the final target, as well as the nonlinear dynamics [24]. In order to incorporate the nonlinear targeting solution $\boldsymbol{u}^*$ into GenCov, the higher-order sensitivities of the maneuver *and* the maneuver's impact on the dynamics are needed. A thorough discussion of nonlinear targeting solutions is provided in [24] based on the multivariate implicit value theorem [35], with specific applications to closed-loop implementation presented here.

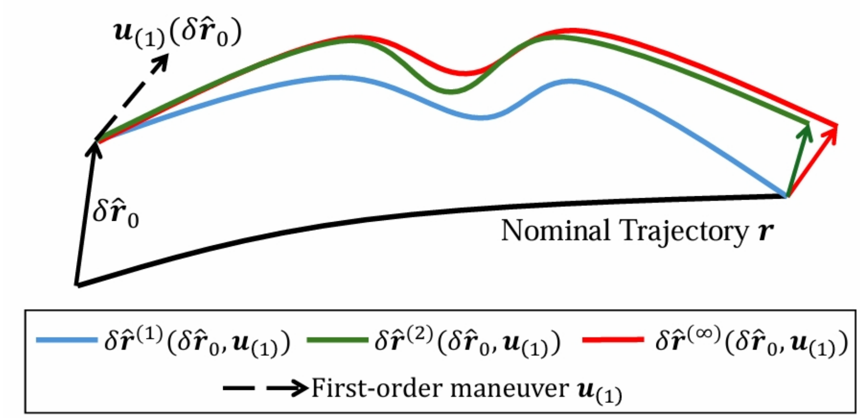


Figure 2: The first-order maneuver $\boldsymbol{u}_{(1)}$ applied to the nav state reaches the final target under *linear* dynamics ($\delta\boldsymbol{r}_f^{(1)} = \boldsymbol{0}$), but misses under second-order $\delta\boldsymbol{r}_f^{(2)}$ and nonlinear dynamics $\delta\boldsymbol{r}_f^{(\infty)}$.

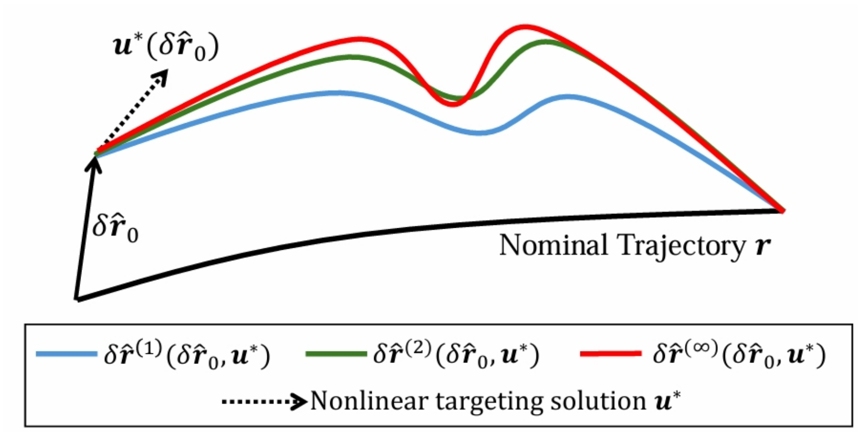


Figure 3: The nonlinear targeting solution $\boldsymbol{u}^*$ applied to the nav state reaches the *final* target under *all* dynamics approximations according to the implicit value theorem [24, 35]. The second-order approximation $\delta\boldsymbol{r}^{(2)}$ outperforms the first-order approximation $\delta\boldsymbol{r}^{(1)}$ prior to the final time.

Targeting solutions are found using the *model* dynamics $\hat{\boldsymbol{\phi}}$; the model dynamics differ from the true dynamics $\boldsymbol{\phi}$ due to simulation fidelity. Consider the final position nav dispersion $\delta\hat{\boldsymbol{r}}_f$ defined by the nonlinear model dynamics $\hat{\boldsymbol{\phi}}$, defined as a function of the current nav dispersion $\delta\hat{\boldsymbol{r}}_0$ and the maneuver $\boldsymbol{u}^*$:

$$\delta\hat{\boldsymbol{x}}_f = \hat{\boldsymbol{\phi}}\left(\delta\hat{\boldsymbol{r}}_0, \boldsymbol{u}^*(\delta\hat{\boldsymbol{r}}_0)\right). \tag{27}$$

The nonlinear targeting solution $\boldsymbol{u}^*$ drives the final position dispersion $\delta\hat{\boldsymbol{r}}_f$ to zero. As such the sensitivities of $\delta\hat{\boldsymbol{r}}_f$ with respect to $\delta\hat{\boldsymbol{r}}_0$ are also zero. The first- and second-order closed-loop targeting sensitivities, $u_{\hat{r}}^{*i,j}$ and $u_{\hat{r}\hat{r}}^{*i,jk}$ are found by taking the partials of the position elements of Eq. 27 with respect to the initial position and isolating the sensitivities [24], yielding

$$u_{\hat{r}}^{*i,j} = -\left[\hat{\phi}_{\hat{r},\hat{v}}^{-1}\right]^{i,\alpha} \hat{\phi}_{\hat{r},\hat{r}}^{\alpha,j}, \tag{28}$$

$$u_{\hat{r}\hat{r}}^{*i,jk} = -\left[\hat{\phi}_{\hat{r},v}^{-1}\right]^{i,\alpha} \left(\hat{\phi}_{\hat{r},\hat{r}\hat{r}}^{\alpha,jk} + \hat{\phi}_{\hat{r},\hat{r}\hat{v}}^{\alpha,j\beta} u_{\hat{r}}^{*\beta,k} + \hat{\phi}_{\hat{r},\hat{v}\hat{r}}^{\alpha,\beta k} u_{\hat{r}}^{*\beta,j} + \hat{\phi}_{\hat{r},\hat{v}\hat{v}}^{\alpha,\beta\gamma} u_{\hat{r}}^{*\beta,j} u_{\hat{r}}^{*\gamma,k}\right). \tag{29}$$

Note that several first-order sensitivities $u_{\hat{r}}^{*i,j}$ are embedded in the second-order sensitivity and are computed using the nav, or model, STTs $\hat{\phi}^{i,k_1\cdots k_p}$. The third-order targeting sensitivity is presented in Appendix A; for further reading see [24, 35].

The nonlinear targeting solution $\boldsymbol{u}^*$ is then plugged into Eq. 21 as the commanded maneuver and is subsequently corrupted by the engine execution errors. Figure 4 depicts the closed-loop $\boldsymbol{u}^*$ maneuver fired and the subsequent first- and second-order approximations of the *true* nonlinear dynamics. Note that none of the approximations reach the final target, as the maneuver is computed based on the nav state $\hat{\boldsymbol{x}}$ and corrupted by engine error. To approximate the nonlinear dynamics implicit on a maneuver $\boldsymbol{u}^*$, the STTs are combined with the targeting sensitivities in Eqs. 28 and 29.

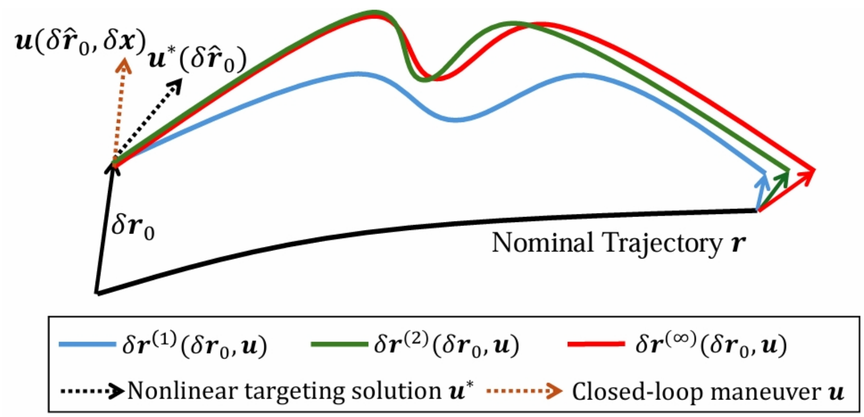


Figure 4: The nonlinear targeting solution $\boldsymbol{u}^*$ is corrupted to produce $\boldsymbol{u}$ and is applied to the *true* state. The first-order approximation $\delta\boldsymbol{r}^{(1)}$ no longer reaches the target due to the error, while the second-order approximation $\delta\boldsymbol{r}^{(2)}$ better captures the true miss $\delta\boldsymbol{r}^{(\infty)}$.

The true maneuver is thus a nonlinear function of the true state and the nav state. The true state $\boldsymbol{x}$ at arbitrary time t is a function of the true dynamics $\boldsymbol{\phi}$ implicit on the maneuver fired at time t_0

$$\boldsymbol{x}_t = \boldsymbol{\phi}\left(\boldsymbol{x}_0, \boldsymbol{u}(\boldsymbol{x}_0, \hat{\boldsymbol{x}}_0)\right). \tag{30}$$

Taking the first and second-order partials of the nonlinear dynamics in Eq. 30 yields the sensitivities of the dynamics *implicit* on the closed-loop targeting. To first-order, the sensitivities of $\boldsymbol{x}_t$ with respect to $\boldsymbol{x}_0$ and $\hat{\boldsymbol{x}}_0$ are

$$\frac{\partial x_t^i}{\partial x_0^j} = \phi_{x,x}^{i,j} + \phi_{x,v}^{i,\alpha} u_x^{\alpha,j}, \qquad \frac{\partial x_t^i}{\partial \hat{x}_0^j} = \phi_{x,v}^{i,\alpha} u_{\hat{x}}^{\alpha,j}. \tag{31}$$

The tensor $\phi_{x,x}^{i,j}$ is simply the first-order TCM. The second-order sensitivities are

$$\frac{\partial^2 x_t^i}{\partial x_0^j \partial x_0^k} = \phi_{x,xx}^{i,jk} + \phi_{x,xv}^{i,j\alpha} u_x^{\alpha,k} + \phi_{x,vx}^{i,\alpha k} u_x^{\alpha,j} + \phi_{x,vv}^{i,\alpha\beta} u_x^{\alpha,j} u_x^{\beta,k} + \phi_{x,v}^{i,\alpha} u_{xx}^{\alpha,jk},$$

$$\frac{\partial^2 x_t^i}{\partial x_0^j \partial \hat{x}_0^k} = \phi_{x,xv}^{i,j\alpha} u_{\hat{x}}^{\alpha,k} + \phi_{x,vv}^{i,\alpha\beta} u_x^{\alpha,j} u_{\hat{x}}^{\beta,k} + \phi_{x,v}^{i,\alpha} u_{x\hat{x}}^{\alpha,jk},$$

$$\frac{\partial^2 x_t^i}{\partial \hat{x}_0^j \partial x_0^k} = \phi_{x,vx}^{i,\alpha k} u_{\hat{x}}^{\alpha,j} + \phi_{x,vv}^{i,\alpha\beta} u_{\hat{x}}^{\alpha,j} u_{x}^{\beta,k} + \phi_{x,v}^{i,\alpha} u_{\hat{x}x}^{\alpha,jk},$$

$$\frac{\partial^2 x_t^i}{\partial \hat{x}_0^j \partial \hat{x}_0^k} = \phi_{x,vv}^{i,\alpha\beta} u_{\hat{x}}^{\alpha,j} u_{\hat{x}}^{\beta,k} + \phi_{x,v}^{i,\alpha} u_{\hat{x}\hat{x}}^{\alpha,jk}. \tag{32}$$

Equations 31 and 32 are the first- and second-order sensitivities of the targeting implicit dynamics; the individual components compose a new set of tensors denoted as $\tilde{\phi}^{i,k_1\cdots k_p}$, referred to as the *targeting STTs*. The true state dispersion at time t is expressed to second-order using the targeting STTs and the augmented state $\delta\boldsymbol{x_a}$ *before* the maneuver time $t_{\Delta v}$

$$\delta x^i(t) = \tilde{\phi}^{i,\alpha}(t, t_{\Delta v}) \left[\delta x_a^-(t_{\Delta v})\right]^\alpha + \frac{1}{2}\tilde{\phi}^{i,\alpha\beta}(t, t_{\Delta v}) \left[\delta x_a^-(t_{\Delta v})\right]^\alpha \left[\delta x_a^-(t_{\Delta v})\right]^\beta. \tag{33}$$

With the targeting STTs, the impact of nonlinear targeting methods on the true dynamics are properly captured. Algorithm 2 describes how GenCov utilizes the higher-order expansions to instantaneously correct the augmented covariance and subsequently propagate the true state dispersions. Similar to the propagation algorithm in Alg. 1, the targeting algorithm in Alg. 2 is more intricate than LinCov, but better represents the mean and covariance from nonlinear engine execution errors and nonlinear targeting methods. Prior to a targeting maneuver, the mean and covariance are propagated using Alg. 1; after a targeting maneuver, Alg. 2 is used instead.

Algorithm 2 GenCov targeting and propagation algorithm.

Input: a) mean and covariance prior to targeting $\{\delta\boldsymbol{m}_{\boldsymbol{a}}^-(t_{\Delta v}), \boldsymbol{\mathcal{C}}_{\boldsymbol{a}}^-(t_{\Delta v})\}$,
b) measurement indices k in time array t,
c) maneuver time $t_{\Delta v}$.

1: $\{\delta\boldsymbol{m}_{\boldsymbol{a}}^+(t_{\Delta v}), \boldsymbol{\mathcal{C}}_{\boldsymbol{a}}^+(t_{\Delta v})\} \leftarrow$ higher-order correction of $\{\delta\boldsymbol{m}_{\boldsymbol{a}}^+(t_{\Delta v}), \boldsymbol{\mathcal{C}}_{\boldsymbol{a}}^+(t_{\Delta v})\}$. ▷ Eqs. 10, 11, 25- 29
2: $\tilde{\phi}^{i,k_1\cdots k_p} \leftarrow$ Compute targeting STTs. ▷ Eqs. 31- 32
3: **while** $t_k \leq t_{\Delta v}$ **do**
4: $\{\delta\boldsymbol{m}_{\boldsymbol{a}}(t_{k+1}), \boldsymbol{\mathcal{C}}_{\boldsymbol{a}}(t_{k+1})\} \leftarrow$first-order propagation of $\{\delta\boldsymbol{m}_{\boldsymbol{a}}(t_k), \boldsymbol{\mathcal{C}}_{\boldsymbol{a}}(t_k)\}$. ▷ See LinCov [1–4]
5: $\{\delta\boldsymbol{m}'_{\boldsymbol{a}}(t_{k+1}), \boldsymbol{\mathcal{C}}'_{\boldsymbol{a}}(t_{k+1})\} \leftarrow$ mapping $\boldsymbol{M_a}$ of $\{\delta\boldsymbol{m}_{\boldsymbol{a}}(t_{k+1}), \boldsymbol{\mathcal{C}}_{\boldsymbol{a}}(t_{k+1})\}$. ▷ Eq. 20
6: $\{\delta\boldsymbol{m}_{\boldsymbol{x}}(t_{k+1}), \boldsymbol{D}_{\boldsymbol{xx}}(t_{k+1})\}^{(p)} \leftarrow p$th-order mapping of $\{\delta\boldsymbol{m}_{\boldsymbol{a}}^-(t_{\Delta v}), \boldsymbol{\mathcal{C}}_{\boldsymbol{a}}^-(t_{\Delta v})\}$ using the targeting STTs $\tilde{\phi}^{i,k_1\cdots k_p}$. ▷ Eqs. 10, 11, 33
7: $\{\delta\boldsymbol{m}'_{\boldsymbol{x}}(t_{k+1}), \boldsymbol{D}'_{\boldsymbol{xx}}(t_{k+1})\} \leftarrow \{\delta\boldsymbol{m}_{\boldsymbol{x}}(t_{k+1}), \boldsymbol{D}_{\boldsymbol{xx}}(t_{k+1})\}^{(p)}$
8: Modify $\{\delta\boldsymbol{m}'_{\boldsymbol{e}}(t_{k+1}), \boldsymbol{D}'_{\boldsymbol{ee}}(t_{k+1})\}$ by adding higher-order terms. ▷ Eqs. 10, 11, and 19
9: $\{\delta\boldsymbol{m}_{\boldsymbol{a}}(t_{k+1}), \boldsymbol{\mathcal{C}}_{\boldsymbol{a}}(t_{k+1})\} \leftarrow$ mapping $\boldsymbol{M_a}$ of $\{\delta\boldsymbol{m}'_{\boldsymbol{a}}(t_{k+1}), \boldsymbol{\mathcal{C}}'_{\boldsymbol{a}}(t_{k+1})\}$. ▷ Eq. 20
10: **end while**

MEASUREMENT UPDATE

In this section, the measurement update produced by an EKF and its impact on the nav state $\hat{\boldsymbol{x}}$ and true estimation errors $\delta\boldsymbol{e}$ are discussed. The objective is to introduce a higher-order expansion of the measurement update to assess whether the filter's estimation error $\delta\hat{\boldsymbol{e}}$ is consistent with the true estimation error $\delta\boldsymbol{e}$. It is crucial to note that a higher-order expansion of the measurement update is *not* equivalent to a nonlinear Kalman filter, such as the HOEKF or the UKF; the former *assesses* the filter's performance, while the latter *changes* the filter's performance.

Kalman Filter Design and LinCov Implementation

The onboard Kalman filter updates the nav state $\hat{\boldsymbol{x}}$ with true measurements $\boldsymbol{h}$, predicted measurements $\hat{\boldsymbol{h}}$, and the Kalman gain $\hat{\boldsymbol{K}}$ according to [10]

$$\hat{\boldsymbol{x}}^{+} = \hat{\boldsymbol{x}}^{-} + \hat{\boldsymbol{K}}(\hat{\boldsymbol{x}}^{-}) \left(\boldsymbol{h}(\boldsymbol{x}^{-}, \boldsymbol{\nu}) - \hat{\boldsymbol{h}}(\hat{\boldsymbol{x}}^{-}) \right), \tag{34}$$

where the true measurement $\boldsymbol{h}$ is corrupted by random measurement noise $\boldsymbol{\nu}$. As before, the notation "-" and "+" denotes before and after a measurement update, respectively. The outputs of the measurement functions $\boldsymbol{h}$ and $\hat{\boldsymbol{h}}$ are frequently expressed as $\boldsymbol{z}$ and $\hat{\boldsymbol{z}}$, respectively, with $\delta\hat{\boldsymbol{z}} = \boldsymbol{z} - \hat{\boldsymbol{z}}$. The Kalman gain is computed according to the filter's estimation errors $\delta\hat{\boldsymbol{e}}$ and the measurement difference $\delta\hat{\boldsymbol{z}}$ according to [10]

$$\boldsymbol{K} = \boldsymbol{P}_{\boldsymbol{x}\boldsymbol{z}} \boldsymbol{P}_{\boldsymbol{z}\boldsymbol{z}}^{-1} = E[\delta\hat{\boldsymbol{e}}\delta\hat{\boldsymbol{z}}^{\mathsf{T}}] \left(E[\delta\hat{\boldsymbol{z}}\delta\hat{\boldsymbol{z}}^{\mathsf{T}}] \right)^{-1}. \tag{35}$$

The expectations in Eq. 35 are zero-mean, linearized about the current state estimate. For in-flight implementation, the EKF utilizes a first-order expansion about the current nav state $\hat{\boldsymbol{x}}$ to approximate the expectation operations in Eq. 35, yielding

$$\hat{\boldsymbol{K}} = \hat{\boldsymbol{P}} \hat{\boldsymbol{H}}_{\hat{\boldsymbol{x}}}^{\mathsf{T}} \left(\hat{\boldsymbol{H}}_{\hat{\boldsymbol{x}}} \hat{\boldsymbol{P}} \hat{\boldsymbol{H}}_{\hat{\boldsymbol{x}}}^{\mathsf{T}} + \boldsymbol{R} \right)^{-1}, \tag{36}$$

where $\hat{\boldsymbol{P}}$ is the *filter* covariance, $\hat{\boldsymbol{H}}_{\hat{x}}$ is the first-order measurement sensitivity with respect to the nav state, and $\boldsymbol{R}$ is the measurement noise covariance.

If the measurement function is strongly nonlinear, the estimation errors are large, or the measurement noise is small, the EKF's filter estimation error $\delta\hat{\boldsymbol{e}}$ may become inconsistent with the true estimation error $\delta\boldsymbol{e}$. As such, the Kalman gain in Eq. 35 can be approximated to higher-order with the HOEKF [33] or the UKF [21]. Alternatively, underweighting the measurements via Lear's method [36] approximates the effects of a second-order Kalman filter according to

$$\hat{\boldsymbol{K}} = \hat{\boldsymbol{P}} \hat{\boldsymbol{H}}_{\hat{\boldsymbol{x}}}^{\mathsf{T}} \left((1+\beta) \hat{\boldsymbol{H}}_{\hat{\boldsymbol{x}}} \hat{\boldsymbol{P}} \hat{\boldsymbol{H}}_{\hat{\boldsymbol{x}}}^{\mathsf{T}} + \boldsymbol{R} \right)^{-1}, \tag{37}$$

where β is frequently set to 0.2 [36]; for further reading on measurement underweighting methods, see [37].

Regardless of what kind of Kalman filter is used, in LinCov the filter design errors $\delta\hat{\boldsymbol{e}}$ are modeled by propagating and updating the covariance along the nominal trajectory [1, 2, 10]. The first-order expansion of the measurement update is found by linearizing Eq. 34 according to

$$\delta\hat{\boldsymbol{x}}^{+} = \delta\hat{\boldsymbol{x}}^{-} + \hat{\boldsymbol{K}} \left(\boldsymbol{H}_{\boldsymbol{x}} \delta\boldsymbol{x}^{-} - \hat{\boldsymbol{H}}_{\hat{\boldsymbol{x}}} \delta\hat{\boldsymbol{x}}^{-} \right) + \hat{\boldsymbol{K}} \delta\boldsymbol{\nu}. \tag{38}$$

The linearization in Eq. 38 is then implemented to update the augmented covariance $\boldsymbol{\mathcal{C}}_{\boldsymbol{a}}$, extracting the true estimation error covariance $\boldsymbol{P}$ as needed using Eq. 14.

Note that the Kalman gain within Eq. 38 represents *any* Kalman gain, including from the EKF, HOEKF, UKF, or other filtering strategy. The key takeaway is that LinCov's first-order expansion in Eq. 38 approximates updates to the estimation error $\delta\boldsymbol{e}$ performed by onboard filters. As discussed previously, LinCov quantifies the filter inconsistency arising from measurement mismodeling, but does not properly capture the impact of the filter design itself. In practice, filter inconsistencies may arise from a poor approximation of the expectations in Eq. 35. To properly quantify the performance of the onboard filter, the first-order approximation in Eq. 38 is expanded to higher-order.

Higher-Order Expansion of the Measurement Update

The measurement update equation in Eq. 34, presented to first-order in Eq. 38, is expanded to higher-order to assess whether the filter's design negatively impacts the true estimation errors $\delta \boldsymbol{e}$. Expanding Eq. 34 with tensor notation yields the higher-order sensitivities of the measurement update with respect to the true state and nav state. For convenience and notation consistency, the first-order sensitivities of the measurement update $\hat{\boldsymbol{x}}^+$, corresponding to Eq. 38, are

$$\frac{\partial[\hat{x}^+]^i}{\partial[x^-]^j} = \hat{K}^{i\alpha} h_x^{\alpha,j}, \qquad \frac{\partial[\hat{x}^+]^i}{\partial[\hat{x}^-]^j} = I^{ij} - \hat{K}^{i\alpha} \hat{h}_{\hat{x}}^{\alpha,j}, \tag{39}$$

where I^{ij} represents the identity tensor. Next, the second-order sensitivities are taken, taking specific note that the Kalman gain is a function of the nav state $\hat{\boldsymbol{x}}^-$:

$$\frac{\partial[\hat{x}^+]^i}{\partial[x^-]^j \partial[x^-]^k} = \hat{K}^{i\alpha} h_{xx}^{\alpha,jk}, \qquad \frac{\partial[\hat{x}^+]^i}{\partial[x^-]^j \partial[\hat{x}^-]^k} = \hat{K}_{\hat{x}}^{i\alpha,k} h_x^{\alpha,j},$$
$$\frac{\partial[\hat{x}^+]^i}{\partial[\hat{x}^-]^j \partial[x^-]^k} = \hat{K}_{\hat{x}}^{i\alpha,j} h_x^{\alpha,k}, \qquad \frac{\partial[\hat{x}^+]^i}{\partial[\hat{x}^-]^j \partial[\hat{x}^-]^k} = -\hat{K}^{i\alpha} \hat{h}_{\hat{x}\hat{x}}^{\alpha,jk} - \hat{K}_{\hat{x}}^{i\alpha,k} \hat{h}_{\hat{x}}^{\alpha,j} - \hat{K}_{\hat{x}}^{i\alpha,j} \hat{h}_{\hat{x}}^{\alpha,k}. \tag{40}$$

The first- and second-order sensitivities in Eqs. 39- 40 apply to the measurement update of *any* Kalman filter. However, the partials of the Kalman gain $K^{ij,k_1 \cdots k_p}$ change based on what kind of filter is used. As the EKF remains the most commonly implemented onboard filter, the partials of the Kalman gain for an EKF are presented in Appendix B. The second-order expansion captures the dominant nonlinear effects to the mean and covariance, but for convenience the third-order sensitivities are presented in Eq. 47 in Appendix A.

For implementation into GenCov, the higher-order approximation of the measurement update is included at every measurement time t_m. GenCov thus directly approximates the filter's bias (represented as the mean of the true estimation errors $\delta \boldsymbol{m_e}$) and the filter's inconsistency (represented as the expected NEES $\bar{\varepsilon}$ in Eq. 16). GenCov provides an analytic method to determine whether the onboard filter is tuned properly and, if not, adjust the filter design and underweighting. If the estimation errors produced by GenCov match those produced by LinCov, the filter is judged to be performing as desired.

CASE STUDY FOR GENCOV

To demonstrate the improvements provided by GenCov, a case study is performed on an elliptical orbit about Bennu. The nominal orbit is depicted in Fig. 5, representing a polar orbit with semi-major axis $a = 1.85$ km and an eccentricity of $e = 0.1351$, based roughly on the preliminary survey orbit of the OSIRIS-REx mission [38]. The correction scheme targets the final position after approximately one orbital period, or 2.6 days. Bennu is a fast spinner, seen in the body-fixed frame in Fig. 5.

Landmark based measurements are taken every hour using the pinhole camera equation [39]. Fifty known landmarks are placed across the surface, where each landmark in the body-fixed frame $\boldsymbol{r}_l^B$ is rotated to the inertial frame using a 3–1–3 Euler rotation matrix $\boldsymbol{T_I^B}$, found using the body's pole prime meridian θ_p, declination δ_p, and right ascension α_p

$$\boldsymbol{T_I^B} = \boldsymbol{T}_3(\theta_p) \boldsymbol{T}_1 \left(\frac{\pi}{2} - \delta_p \right) \boldsymbol{T}_3 \left(\frac{\pi}{2} + \alpha_p \right). \tag{41}$$

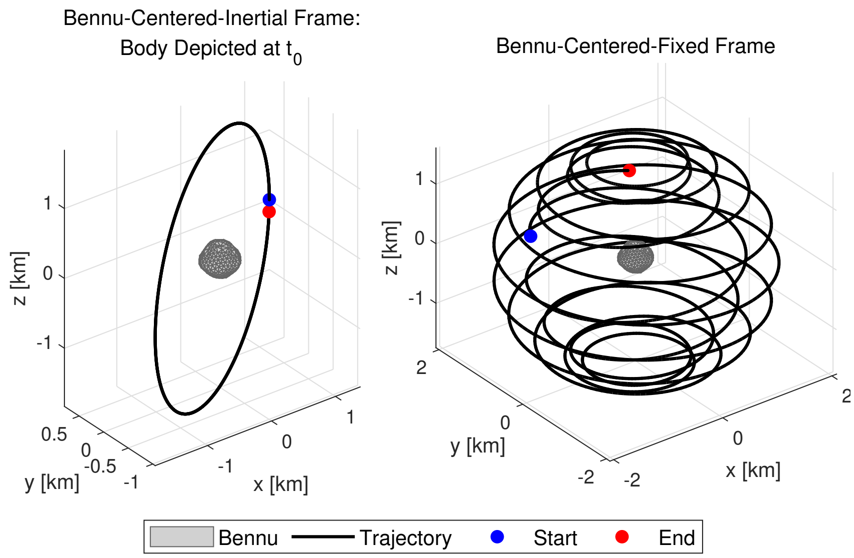


Figure 5: Nominal Bennu orbit.

The relative distance from the landmark to the spacecraft in the inertial frame $\boldsymbol{r}^I$ is rotated to the camera frame via $\boldsymbol{T}_{\boldsymbol{C}}^{\boldsymbol{I}}$, then corrupted by a small angle rotation $\delta\boldsymbol{T}$ [2] due to attitude knowledge errors:

$$\boldsymbol{s} = \delta\boldsymbol{T}(\boldsymbol{q})\boldsymbol{T}_{\boldsymbol{C}}^{\boldsymbol{I}}\left[\boldsymbol{T}_{\boldsymbol{I}}^{\boldsymbol{B}}\boldsymbol{r}_l^B - \boldsymbol{r}^I\right]. \tag{42}$$

The relative vector $\boldsymbol{s}$ is then projected into the image space using the pinhole camera equation [39]:

$$\boldsymbol{h}(\boldsymbol{x}) = \boldsymbol{M}\boldsymbol{\rho}(\boldsymbol{s}) = \boldsymbol{M}\left[\begin{array}{ccc} -x/z, & y/z, & 1 \end{array}\right]^{\mathsf{T}}. \tag{43}$$

The matrix $\boldsymbol{M}$ denotes a linear map corresponding to the camera's Field of View (FOV) [39], and for the current work is set to $20.8°$ based on the camera specifications of OSIRIS-REx [38]. Random, Gaussian noise is added to the true measurement. Landmark-based navigation using the pinhole camera equation is strongly nonlinear, making it an interesting case study for nonlinear covariance analysis.

The closed-loop analysis uses the true state in Eq. 44, consisting of the spacecraft position $\boldsymbol{r}$ and velocity $\boldsymbol{v}$, attitude knowledge errors $\boldsymbol{q}$, pole elements $\boldsymbol{\lambda_p}$ consisting of the euler angles $\{\alpha_p, \delta_p, \theta_p\}$ and the body spin-rate $\dot{\theta}_p$ according to a principal axis rotation, IMU errors $\boldsymbol{p_y}$, and engine execution errors $\boldsymbol{p_u}$:

$$\boldsymbol{x} = \left[\begin{array}{cccccc} \boldsymbol{r}^{\mathsf{T}}, & \boldsymbol{v}^{\mathsf{T}}, & \boldsymbol{q}^{\mathsf{T}}, & \boldsymbol{\lambda_p}^{\mathsf{T}}, & \boldsymbol{p_y}^{\mathsf{T}}, & \boldsymbol{p_u}^{\mathsf{T}} \end{array}\right]^{\mathsf{T}}. \tag{44}$$

The nav state $\hat{\boldsymbol{x}}$ contains the same state elements with the exception of the engine execution errors $\boldsymbol{p_u}$. Table 1 records the initial true dispersions and estimation errors of the augmented state $\boldsymbol{x_a}$. The attitude knowledge errors, engine errors, and IMU errors are modeled as Exponentially Correlated Random Variables (ECRVs) with a time constant of $\tau = 1000$ s. The true and nav dynamics are the equivalently modeling using two-body dynamics.

Table 1: Bennu orbit initial 3σ dispersions and estimation errors.

(a) State dispersions and estimation errors.

Component	Value
True position dispersion $\delta\boldsymbol{r}$.	10 m
True velocity dispersion $\delta\boldsymbol{v}$.	1 mm/s
Position estimation $\delta\boldsymbol{e_r}$.	3 m
Velocity estimation $\delta\boldsymbol{e_v}$.	0.5 mm/s
Pole estimation $\delta\boldsymbol{e}_{\lambda_p}$.	3 deg
Spin estimation $\delta\boldsymbol{e}_{\dot{\theta}_p}$.	0.025 deg/hr
Attitude knowledge error $\delta\boldsymbol{e_q}$.	10 arcsec

(b) Engine, IMU, and camera errors.

Component	Value
Engine bias $\delta\boldsymbol{b_u}$.	15 μm/s
Engine scaling $\delta\boldsymbol{s_u}$.	1%
Engine misalignment $\delta\boldsymbol{e_u}$.	10 mrad
IMU bias $\delta\boldsymbol{b_y}$.	1 μm/s
IMU scaling $\delta\boldsymbol{s_y}$.	0.3%
IMU misalignment $\delta\boldsymbol{e_y}$.	1.5 mrad
Camera Noise $\boldsymbol{\nu}$.	0.15 pixels

Simulation Results

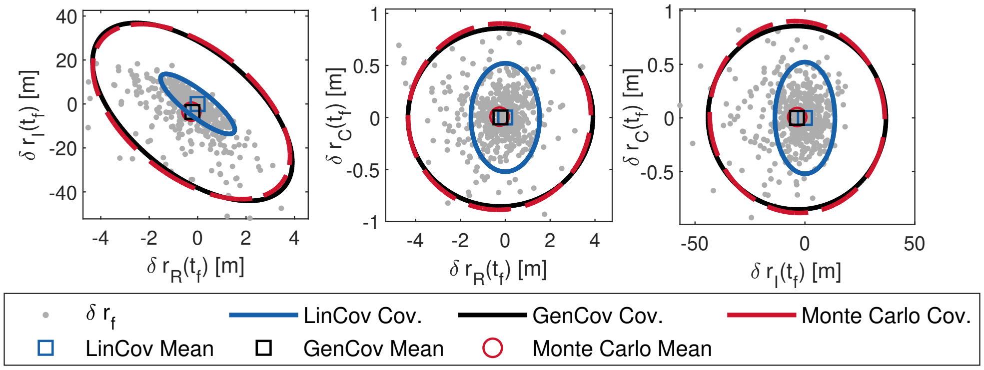
From the initial dispersions in Table 1, 5,000 Monte Carlo runs are generated as a baseline comparison to the LinCov and GenCov results. A maneuver is performed 16 hours into the orbit (or about a fourth of the orbital period) targeting the final nominal position. The final state dispersions and errors are recorded and presented in the Radial, In-Track, Cross-Track (RIC) frame.

First, consider the final position dispersions depicted in Fig. 6a due to firing a first-order, non-iterative targeting maneuver using the nominal STMs [7]. The maneuver is fired based on the nav state $\hat{\boldsymbol{x}}$ provided by the EKF with no measurement underweighting, i.e., Lear's coefficient $\beta = 0$. Note that the mean and covariance produced by LinCov under-approximate the true dispersions, whereas GenCov captures the statistics provided by the Monte Carlo analysis. With the statistics provided by GenCov, mission designers may determine that the final dispersions provided by the first-order targeting strategy are too large and too far offset from the nominal.

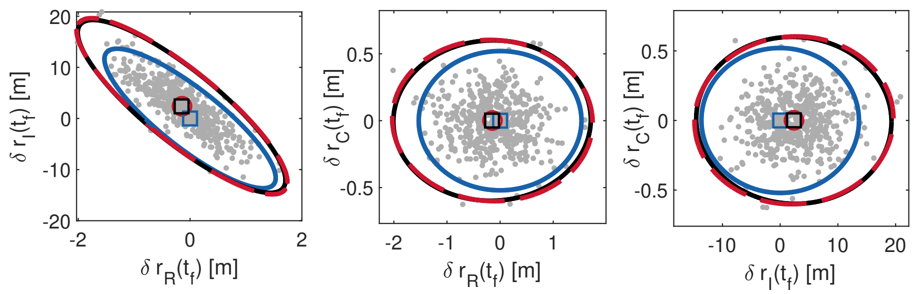
Next, the first-order maneuver is swapped with the nonlinear targeting solution $\boldsymbol{u}^*$, where the maneuver is refined through a shooting method. Figure 6b depicts the final position dispersions from LinCov, GenCov, and the Monte Carlo data. According to GenCov, the nonlinear targeting solution $\boldsymbol{u}^*$ yields smaller final position dispersions, whereas the mean and covariance from LinCov do not change at all from Fig. 6a to Fig. 6b. The nonlinear targeting solution reduces the final position dispersion covariance, but the dispersions are still offset by a non-zero mean $\delta\boldsymbol{m_x}$ arising from the filter's design.

The filter consistency affects the estimation errors, subsequently impacting the closed-loop targeting performance. Consider the true estimation errors shown in Fig. 7a, depicting the position errors in the radial direction from time t_0 to the maneuver time $t_{\Delta v}$. The onboard EKF with no measurement underweighting, i.e., $\beta = 0$, is clearly biased as $\delta\boldsymbol{m_e}$ is non-zero. The standardized expected NEES, or $\bar{\varepsilon}/\hat{n}$, is ideally equal to one, but for the current filter is 4.82. Adopting an aggressive underweighting strategy by setting $\beta = 1.0$, the filter bias decreases and the standardized expected NEES drops to 1.04, shown in Fig. 7b.

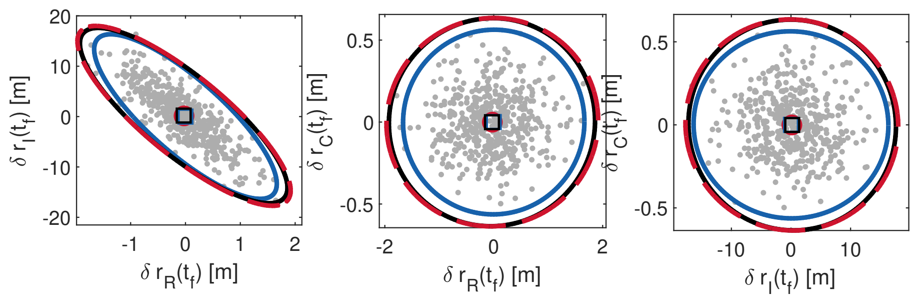
With the underweighted filter, the maneuver fired at $t_{\Delta v}$ has nearly non-biased estimation errors. As such, the final position dispersion errors improve once again from Fig. 6b to Fig. 6c. The final position dispersions are nearly zero-mean and the the covariance once again shrinks. The remaining discrepancies between LinCov and GenCov in Fig. 6c are primarily due to the engine execution errors $(\boldsymbol{s_u}, \boldsymbol{e_u})$, omitted from LinCov but captured in GenCov.



(a) Final position dispersions $\delta \boldsymbol{r}$ in the RIC frame from first-order targeting, $\beta = 0.0$: $||\delta \boldsymbol{m_r}(t_f)|| = 3.61$ m.



(b) Final position dispersions $\delta \boldsymbol{r}$ in the RIC frame from nonlinear targeting, $\beta = 0.0$: $||\delta \boldsymbol{m_r}(t_f)|| = 2.37$ m.

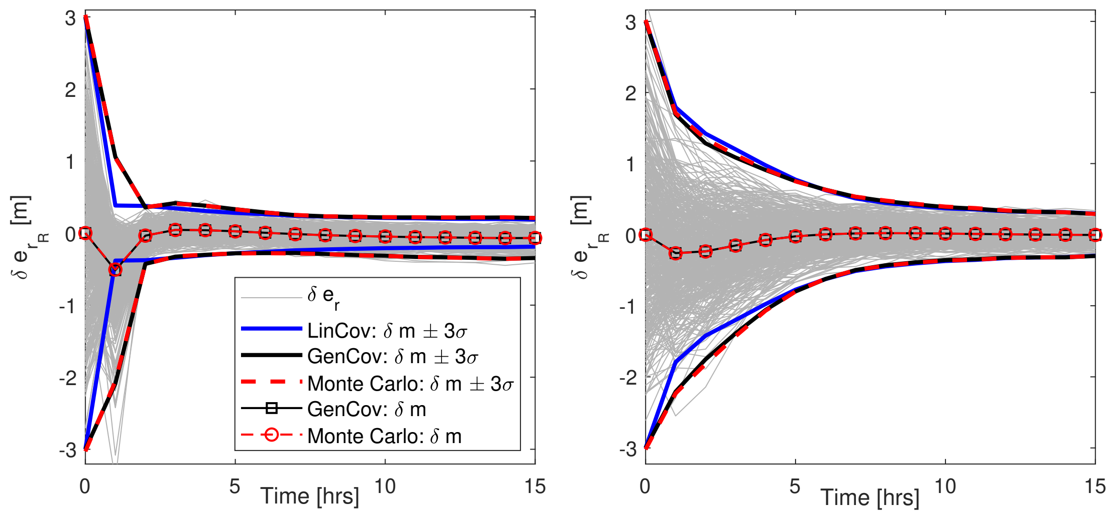


(c) Final position dispersions $\delta \boldsymbol{r}$ in the RIC frame from nonlinear targeting, $\beta = 1.0$: $||\delta \boldsymbol{m_r}(t_f)|| = 0.20$ m.

Figure 6: Nonlinear covariance analysis for mission design using GenCov.

Computational Complexity and Applications of GenCov

GenCov more accurately represents the Monte Carlo statistics than LinCov, but is more computationally expensive to implement. Tensor multiplications are naturally more complicated than matrix multiplications. Furthermore, the Gaussian moments up to fourth-order, shown earlier in Eq. 11, are required at every higher-order operation. However, significant time is saved by only expanding the higher-order operations with the function-dependent states. For example, the measurement equation is not dependent on the IMU states; as such, the dimension of the measurement tensor is



(a) No measurement underweighting
$\beta = 0.0 \rightarrow \bar{\varepsilon}_{max}/\hat{n} = 4.82$

(b) Measurement underweighting
$\beta = 1.0 \rightarrow \bar{\varepsilon}_{max}/\hat{n} = 1.04$

Figure 7: The impact of measurement underweighting on position estimation errors in the radial direction.

reduced by excluding the IMU states. Additional time can also be saved by omitting the higher-order STTs when propagating the true estimation errors $\delta \boldsymbol{e}$ in Eq. 19, as most closed-loop scenarios assume a measurement rich environment. For the presented mission simulation about Bennu, GenCov runs about *five times* slower than LinCov. While the cost is not trivial, GenCov still runs orders of magnitude faster than a Monte Carlo study.

The GenCov algorithm presented in the current work, and demonstrated in the closed-simulation about Bennu, implements a quadratic expansion of the propagation, correction, and measurement update steps. The expansions presented can be extended to higher-order—and do not have to be extended to the same order for the propagation, targeting, and measurement update—depending on the desired application of GenCov. One application of GenCov is to replace the Monte Carlo validation of LinCov. Instead of running a Monte Carlo to match LinCov, run GenCov; a match between LinCov and GenCov is a strong indication that the higher-order effects are negligible. If the quadratic expansion in GenCov matches LinCov, it is unlikely that a higher-order expansion in GenCov will return differing results. The other application of GenCov replaces LinCov when nonlinearities exist and need to be properly represented to higher-order, as demonstrated for the Bennu simulation. A full expansion to third-order requires up to the sixth Gaussian moment in Eq. 11, but can be truncated to only include up to the fourth Gaussian moment, capturing the dominant effects arising between the first- and third-order partial. The closed-loop targeting-implicit STTs and the partials of the Kalman gain are derived and presented to third-order in Appendix A.

CONCLUSIONS

Maintaining the basic framework of LinCov, the first-order approximations are generalized to higher-order, introducing a new Generalized Covariance Analysis, GenCov. The new algorithm better represents the statistics provided by a Monte Carlo analysis, specifically addressing the nonlinearities arising from spacecraft propagation, targeting, and measurement updates. Though slower than LinCov, GenCov runs hundreds of times faster than a Monte Carlo analysis. GenCov presents an alternative tool to validate the statistics provided by LinCov and, when necessary, replace LinCov as a closed-loop mission analysis tool. The utility of an analytic nonlinear covariance analysis expands the application of LinCov to mission environments not traditionally considered, including missions to small bodies and missions with extended periods of operation. While LinCov is ideal for understanding how sensor and engine specifications impact mission performance, GenCov provides an understanding into the targeting and filter design.

ACKNOWLEDGMENTS

The research is funded by Intuitive Machines, 3700 Bay Area Blvd #600, Houston, TX 77058.

APPENDIX A: THIRD-ORDER TARGETING STTS AND MEASUREMENT UPDATE

The third-order targeting sensitivity of the closed-loop, nonlinear targeting solution $\boldsymbol{u}^*$ is

$$\begin{aligned} u^{*i,jkl}_{\hat{r}\hat{r}\hat{r}} = & -\left[\hat{\phi}^{-1}_{\hat{r},\hat{v}}\right]^{i,\alpha} \Big(\hat{\phi}^{\alpha,jkl}_{\hat{r},\hat{r}\hat{r}\hat{r}} + \hat{\phi}^{\alpha,jk\beta}_{\hat{r},\hat{r}\hat{r}\hat{v}} u^{*\beta,l}_{\hat{r}} + \hat{\phi}^{\alpha,j\beta l}_{\hat{r},\hat{r}\hat{v}\hat{r}} u^{*\beta,k}_{\hat{r}} + \hat{\phi}^{\alpha,\beta kl}_{\hat{r},\hat{v}\hat{r}\hat{r}} u^{*\beta,j}_{\hat{r}} + \hat{\phi}^{\alpha,j\beta\gamma}_{\hat{r},\hat{r}\hat{v}\hat{v}} u^{*\beta,k}_{\hat{r}} u^{*\gamma,l}_{\hat{r}} \\ & + \hat{\phi}^{\alpha,\beta k\gamma}_{\hat{r},\hat{v}\hat{r}\hat{v}} u^{*\beta,j}_{\hat{r}} u^{*\gamma,l}_{\hat{r}} + \hat{\phi}^{\alpha,\beta\gamma l}_{\hat{r},\hat{v}\hat{v}\hat{r}} u^{*\beta,j}_{\hat{r}} u^{*\gamma,k}_{\hat{r}} + \hat{\phi}^{\alpha,\beta l}_{\hat{r},\hat{v}\hat{r}} u^{*\beta,jk}_{\hat{r}\hat{r}} + \hat{\phi}^{\alpha,j\beta}_{\hat{r},\hat{r}\hat{v}} u^{*\beta,kl}_{\hat{r}\hat{r}} + \hat{\phi}^{\alpha,\beta k}_{\hat{r},\hat{v}\hat{r}} u^{*\beta,jl}_{\hat{r}\hat{r}} \\ & + \hat{\phi}^{\alpha,\beta\gamma\delta}_{r,\hat{v}\hat{v}\hat{v}} u^{*\beta,j}_{r} u^{*\gamma,k}_{r} u^{*\delta,l}_{r} + \hat{\phi}^{\alpha,\beta\gamma}_{r,\hat{v}\hat{v}} \left[u^{*\beta,jl}_{rr} u^{*\gamma,k}_{r} + u^{*\beta,j}_{r} u^{*\gamma,kl}_{rr} + u^{*\beta,jk}_{rr} u^{*\gamma,l}_{r} \right] \Big) . \end{aligned} \tag{45}$$

The closed-loop, targeting-implicit third-order STTs are expressed with the sensitivities:

$$\begin{aligned} \frac{\partial \delta x^i_f}{\partial \delta x^j_0 \partial \delta x^k_0 \partial \delta x^l_0} = & \phi^{i,jkl}_{x,xxx} + \phi^{i,jk\alpha}_{x,xxv} u^{\alpha,l}_{x} + \phi^{i,j\alpha l}_{x,xvx} u^{\alpha,k}_{x} + \phi^{i,\alpha kl}_{x,vxx} u^{\alpha,j}_{x} + \phi^{i,j\alpha\beta}_{x,xvv} u^{\alpha,k}_{x} u^{\beta,l}_{x} \\ & + \phi^{i,\alpha k\beta}_{x,vxv} u^{\alpha,j}_{x} u^{\beta,l}_{x} + \phi^{i,\alpha\beta l}_{x,vvx} u^{\alpha,j}_{x} u^{\beta,k}_{x} + \phi^{i,\alpha l}_{x,vx} u^{\alpha,jk}_{xx} + \phi^{i,j\alpha}_{x,xv} u^{\alpha,kl}_{xx} + \phi^{i,\alpha k}_{x,vx} u^{\alpha,jl}_{xx} \\ & + \phi^{i,\alpha\beta\gamma}_{x,vvv} u^{\alpha,j}_{x} u^{\beta,k}_{x} u^{\gamma,l}_{x} + \phi^{i,\alpha\beta}_{x,vv} \left[u^{\alpha,jl}_{xx} u^{\beta,k}_{x} + u^{\alpha,j}_{x} u^{\beta,kl}_{xx} + u^{\alpha,jk}_{xx} u^{\beta,l}_{x} \right] + \phi^{i,\alpha}_{x,v} u^{\alpha,jkl}_{xxx} \end{aligned}$$

$$\begin{aligned} \frac{\partial \delta x^i_f}{\partial \delta x^j_0 \partial \delta x^k_0 \partial \delta \hat{x}^l_0} = & \phi^{i,jk\alpha}_{x,xxv} u^{\alpha,l}_{\hat{x}} + \phi^{i,j\alpha\beta}_{x,xvv} u^{\alpha,k}_{x} u^{\beta,l}_{\hat{x}} + \phi^{i,\alpha k\beta}_{x,vxv} u^{\alpha,j}_{x} u^{\beta,l}_{\hat{x}} + \phi^{i,j\alpha}_{x,xv} u^{\alpha,kl}_{x\hat{x}} + \phi^{i,\alpha k}_{x,vx} u^{\alpha,jl}_{x\hat{x}} \\ & + \phi^{i,\alpha\beta\gamma}_{x,vvv} u^{\alpha,j}_{x} u^{\beta,k}_{x} u^{\gamma,l}_{\hat{x}} + \phi^{i,\alpha\beta}_{x,vv} \left[u^{\alpha,jl}_{x\hat{x}} u^{\beta,k}_{x} + u^{\alpha,j}_{x} u^{\beta,kl}_{x\hat{x}} + u^{\alpha,jk}_{xx} u^{\beta,l}_{\hat{x}} \right] + \phi^{i,\alpha}_{x,v} u^{\alpha,jkl}_{xx\hat{x}} \end{aligned}$$

$$\begin{aligned} \frac{\partial \delta x^i_f}{\partial \delta x^j_0 \partial \delta \hat{x}^k_0 \partial \delta x^l_0} = & \phi^{i,j\alpha l}_{x,xvx} u^{\alpha,k}_{\hat{x}} + \phi^{i,j\alpha\beta}_{x,xvv} u^{\alpha,k}_{\hat{x}} u^{\beta,l}_{x} + \phi^{i,\alpha\beta l}_{x,vvx} u^{\alpha,j}_{x} u^{\beta,k}_{\hat{x}} + \phi^{i,\alpha l}_{x,vx} u^{\alpha,jk}_{x\hat{x}} + \phi^{i,j\alpha}_{x,xv} u^{\alpha,kl}_{\hat{x}x} \\ & + \phi^{i,\alpha\beta\gamma}_{x,vvv} u^{\alpha,j}_{x} u^{\beta,k}_{\hat{x}} u^{\gamma,l}_{x} + \phi^{i,\alpha\beta}_{x,vv} \left[u^{\alpha,jl}_{xx} u^{\beta,k}_{\hat{x}} + u^{\alpha,j}_{x} u^{\beta,kl}_{\hat{x}x} + u^{\alpha,jk}_{x\hat{x}} u^{\beta,l}_{x} \right] + \phi^{i,\alpha}_{x,v} u^{\alpha,jkl}_{x\hat{x}x} \end{aligned}$$

$$\begin{aligned} \frac{\partial \delta x^i_f}{\partial \delta \hat{x}^j_0 \partial \delta x^k_0 \partial \delta x^l_0} = & \phi^{i,\alpha kl}_{x,vxx} u^{\alpha,j}_{\hat{x}} + \phi^{i,\alpha k\beta}_{x,vxv} u^{\alpha,j}_{\hat{x}} u^{\beta,l}_{x} + \phi^{i,\alpha\beta l}_{x,vvx} u^{\alpha,j}_{\hat{x}} u^{\beta,k}_{x} + \phi^{i,\alpha l}_{x,vx} u^{\alpha,jk}_{\hat{x}x} + \phi^{i,\alpha k}_{x,vx} u^{\alpha,jl}_{\hat{x}x} \\ & + \phi^{i,\alpha\beta\gamma}_{x,vvv} u^{\alpha,j}_{\hat{x}} u^{\beta,k}_{x} u^{\gamma,l}_{x} + \phi^{i,\alpha\beta}_{x,vv} \left[u^{\alpha,jl}_{\hat{x}x} u^{\beta,k}_{x} + u^{\alpha,j}_{\hat{x}} u^{\beta,kl}_{xx} + u^{\alpha,jk}_{\hat{x}x} u^{\beta,l}_{x} \right] + \phi^{i,\alpha}_{x,v} u^{\alpha,jkl}_{\hat{x}xx} \end{aligned}$$

$$\begin{aligned} \frac{\partial \delta x^i_f}{\partial \delta x^j_0 \partial \delta \hat{x}^k_0 \partial \delta \hat{x}^l_0} = & \phi^{i,j\alpha\beta}_{x,xvv} u^{\alpha,k}_{\hat{x}} u^{\beta,l}_{\hat{x}} + \phi^{i,j\alpha}_{x,xv} u^{\alpha,kl}_{\hat{x}\hat{x}} + \phi^{i,\alpha\beta\gamma}_{x,vvv} u^{\alpha,j}_{x} u^{\beta,k}_{\hat{x}} u^{\gamma,l}_{\hat{x}} \\ & + \phi^{i,\alpha\beta}_{x,vv} \left[u^{\alpha,jl}_{x\hat{x}} u^{\beta,k}_{\hat{x}} + u^{\alpha,j}_{x} u^{\beta,kl}_{\hat{x}\hat{x}} + u^{\alpha,jk}_{x\hat{x}} u^{\beta,l}_{\hat{x}} \right] + \phi^{i,\alpha}_{x,v} u^{\alpha,jkl}_{x\hat{x}\hat{x}} \end{aligned}$$

$$
\begin{aligned}
\frac{\partial \delta x_f^i}{\partial \delta \hat{x}_0^j \partial \delta x_0^k \partial \delta \hat{x}_0^l} &= \phi_{x,vxv}^{i,\alpha k\beta} u_{\hat{x}}^{\alpha,j} u_{\hat{x}}^{\beta,l} + \phi_{x,vx}^{i,\alpha k} u_{\hat{x}\hat{x}}^{\alpha,jl} + \phi_{x,vvv}^{i,\alpha\beta\gamma} u_{\hat{x}}^{\alpha,j} u_{x}^{\beta,k} u_{\hat{x}}^{\gamma,l} \\
&\quad + \phi_{x,vv}^{i,\alpha\beta} \left[u_{\hat{x}\hat{x}}^{\alpha,jl} u_{x}^{\beta,k} + u_{\hat{x}}^{\alpha,j} u_{x\hat{x}}^{\beta,kl} + u_{\hat{x}x}^{\alpha,jk} u_{\hat{x}}^{\beta,l} \right] + \phi_{x,v}^{i,\alpha} u_{\hat{x}x\hat{x}}^{\alpha,jkl} \\
\frac{\partial \delta x_f^i}{\partial \delta \hat{x}_0^j \partial \delta \hat{x}_0^k \partial \delta x_0^l} &= \phi_{x,vvx}^{i,\alpha\beta l} u_{\hat{x}}^{\alpha,j} u_{\hat{x}}^{\beta,k} + \phi_{x,vx}^{i,\alpha l} u_{\hat{x}\hat{x}}^{\alpha,jk} + \phi_{x,vvv}^{i,\alpha\beta\gamma} u_{\hat{x}}^{\alpha,j} u_{\hat{x}}^{\beta,k} u_{x}^{\gamma,l} \\
&\quad + \phi_{x,vv}^{i,\alpha\beta} \left[u_{\hat{x}x}^{\alpha,jl} u_{\hat{x}}^{\beta,k} + u_{\hat{x}}^{\alpha,j} u_{\hat{x}x}^{\beta,kl} + u_{\hat{x}\hat{x}}^{\alpha,jk} u_{x}^{\beta,l} \right] + \phi_{x,v}^{i,\alpha} u_{\hat{x}\hat{x}x}^{\alpha,jkl}
\end{aligned}
$$

$$
\frac{\partial \delta x_f^i}{\partial \delta \hat{x}_0^j \partial \delta \hat{x}_0^k \partial \delta \hat{x}_0^l} = \phi_{x,vvv}^{i,\alpha\beta\gamma} u_{\hat{x}}^{\alpha,j} u_{\hat{x}}^{\beta,k} u_{\hat{x}}^{\gamma,l} + \phi_{x,vv}^{i,\alpha\beta} \left[u_{\hat{x}\hat{x}}^{\alpha,jl} u_{\hat{x}}^{\beta,k} + u_{\hat{x}}^{\alpha,j} u_{\hat{x}\hat{x}}^{\beta,kl} + u_{\hat{x}\hat{x}}^{\alpha,jk} u_{\hat{x}}^{\beta,l} \right] + \phi_{x,v}^{i,\alpha} u_{\hat{x}\hat{x}\hat{x}}^{\alpha,jkl} \tag{46}
$$

Measurement Update Third-Order Sensitivities

The third-order partials of the measurement update are

$$
\frac{\partial [\hat{x}^+]^i}{\partial [x^-]^j \partial [x^-]^k \partial [x^-]^l} = \hat{K}^{i\alpha} h_{xxx}^{\alpha,jkl}, \qquad \frac{\partial [\hat{x}^+]^i}{\partial [x^-]^j \partial [x^-]^k \partial [\hat{x}^-]^l} = \hat{K}_{\hat{x}}^{i\alpha,l} h_{xx}^{\alpha,jk},
$$

$$
\frac{\partial [\hat{x}^+]^i}{\partial [x^-]^j \partial [\hat{x}^-]^k \partial [x^-]^l} = \hat{K}_{\hat{x}}^{i\alpha,k} h_{xx}^{\alpha,jl}, \qquad \frac{\partial [\hat{x}^+]^i}{\partial [x^-]^j \partial [\hat{x}^-]^k \partial [\hat{x}^-]^l} = \hat{K}_{\hat{x}\hat{x}}^{i\alpha,kl} h_{x}^{\alpha,j},
$$

$$
\frac{\partial [\hat{x}^+]^i}{\partial [\hat{x}^-]^j \partial [x^-]^k \partial [x^-]^l} = \hat{K}_{\hat{x}}^{i\alpha,j} h_{xx}^{\alpha,kl}, \qquad \frac{\partial [\hat{x}^+]^i}{\partial [\hat{x}^-]^j \partial [x^-]^k \partial [\hat{x}^-]^l} = \hat{K}_{\hat{x}\hat{x}}^{i\alpha,jl} h_{x}^{\alpha,k},
$$

$$
\frac{\partial [\hat{x}^+]^i}{\partial [\hat{x}^-]^j \partial [\hat{x}^-]^k \partial [x^-]^l} = \hat{K}_{\hat{x}\hat{x}}^{i\alpha,jk} h_{x}^{\alpha,l},
$$

$$
\begin{aligned}
\frac{\partial [\hat{x}^+]^i}{\partial [\hat{x}^-]^j \partial [\hat{x}^-]^k \partial [\hat{x}^-]^l} &= -\hat{K}_{\hat{x}\hat{x}}^{i\alpha,jk} \hat{h}_{\hat{x}}^{\alpha,l} - \hat{K}_{\hat{x}\hat{x}}^{i\alpha,jl} \hat{h}_{\hat{x}}^{\alpha,k} - \hat{K}_{\hat{x}\hat{x}}^{i\alpha,kl} \hat{h}_{\hat{x}}^{\alpha,j} - \ldots \\
&\quad \hat{K}_{\hat{x}}^{i\alpha,j} h_{\hat{x}\hat{x}}^{\alpha,kl} - \hat{K}_{\hat{x}}^{i\alpha,k} h_{\hat{x}\hat{x}}^{\alpha,jl} - \hat{K}_{\hat{x}}^{i\alpha,l} h_{\hat{x}\hat{x}}^{\alpha,jk} - \hat{K}^{i\alpha} h_{\hat{x}\hat{x}\hat{x}}^{\alpha,jkl}.
\end{aligned} \tag{47}
$$

Note that the third-order partials presented in Eq. 47 apply for an Kalman filter; the partials of the Kalman gain $K_{\hat{x}}^{ij,k_1\cdots k_p}$ are dependent on the filter type, and are presented in Appendix B for an EKF.

APPENDIX B: PARTIALS OF THE KALMAN GAIN

The Kalman gain, is expressed with the covariances $\boldsymbol{P_{xz}}$ and $\boldsymbol{P_{zz}}$, seen in Eq. 35 [10]. The tensor form of the Kalman gain is expressed as

$$
\hat{K}^{ij} = [P_{xz}]^{i\alpha} \left[P_{zz}^{-1}\right]^{\alpha j}, \tag{48}
$$

The first- and second-order partials of the Kalman gain are

$$
\hat{K}_{\hat{x}}^{ij,k} = [P_{xz}]_{\hat{x}}^{i\alpha,k} \left[P_{zz}^{-1}\right]^{\alpha j} + [P_{xz}]^{i\alpha} \left[P_{zz}^{-1}\right]_{\hat{x}}^{\alpha j,k}, \tag{49}
$$

$$
\hat{K}_{\hat{x}\hat{x}}^{ij,kl} = [P_{xz}]_{\hat{x}}^{i\alpha,k} \left[P_{zz}^{-1}\right]_{\hat{x}}^{\alpha j,l} + [P_{xz}]_{\hat{x}\hat{x}}^{i\alpha,kl} \left[P_{zz}^{-1}\right]^{\alpha j} + [P_{xz}]^{i\alpha} \left[P_{zz}^{-1}\right]_{\hat{x}\hat{x}}^{\alpha j,kl} + [P_{xz}]_{\hat{x}}^{i\alpha,l} \left[P_{zz}^{-1}\right]_{\hat{x}}^{\alpha j,k}. \tag{50}
$$

The partials of the Kalman gain presented in Eqs. 49 and 50 are valid for the EKF, UKF, and HOEKF. However, the individual partials of $\boldsymbol{P_{xz}}$ and $\boldsymbol{P_{zz}}$ change based on the filter type. The partials of $\boldsymbol{P_{xz}}$ for the EKF are

$$[P_{xz}]^{ij,k}_{\hat{x}} = \hat{P}^{i\alpha}\hat{h}^{j,\alpha k}_{\hat{x}\hat{x}}, \qquad [P_{xz}]^{ij,kl}_{\hat{x}\hat{x}} = \hat{P}^{i\alpha}\hat{h}^{j,\alpha kl}_{\hat{x}\hat{x}\hat{x}}. \tag{51}$$

The partials of $\boldsymbol{P_{zz}}$ for an EKF are

$$[P_{zz}]^{ij,k}_{\hat{x}} = \left(\hat{h}^{i,\alpha k}_{\hat{x}\hat{x}}\hat{h}^{j,\beta}_{\hat{x}} + \hat{h}^{i,\alpha}_{\hat{x}}\hat{h}^{j,\beta k}_{\hat{x}\hat{x}}\right)\hat{P}^{\alpha\beta}, \tag{52}$$

$$[P_{zz}]^{ij,kl}_{\hat{x}\hat{x}} = \left(\hat{h}^{i,\alpha kl}_{\hat{x}\hat{x}\hat{x}}\hat{h}^{j,\beta}_{\hat{x}} + \hat{h}^{i,\alpha k}_{\hat{x}\hat{x}}\hat{h}^{j,\beta l}_{\hat{x}} + \hat{h}^{i,\alpha l}_{\hat{x}\hat{x}}\hat{h}^{j,\beta k}_{\hat{x}} + \hat{h}^{i,\alpha}_{\hat{x}}\hat{h}^{j,\beta kl}_{\hat{x}\hat{x}\hat{x}}\right)\hat{P}^{\alpha\beta}. \tag{53}$$

Lastly the partials of $\boldsymbol{P_{zz}^{-1}}$ are

$$\left[P_{zz}^{-1}\right]^{ij,k}_{\hat{x}} = -\left[P_{zz}^{-1}\right]^{i\alpha}\left[P_{zz}^{-1}\right]^{\beta j}[P_{zz}]^{\alpha\beta,k}_{\hat{x}}, \tag{54}$$

$$\begin{aligned}\left[P_{zz}^{-1}\right]^{ij,kl}_{\hat{x}\hat{x}} = &-\left[P_{zz}^{-1}\right]^{i\alpha,l}_{\hat{x}}\left[P_{zz}^{-1}\right]^{\beta j}[P_{zz}]^{\alpha\beta,k}_{\hat{x}} - \left[P_{zz}^{-1}\right]^{i\alpha}\left[P_{zz}^{-1}\right]^{\beta j,l}_{\hat{x}}[P_{zz}]^{\alpha\beta,k}_{\hat{x}} \dots \\ &- \left[P_{zz}^{-1}\right]^{i\alpha}\left[P_{zz}^{-1}\right]^{\beta j}[P_{zz}]^{\alpha\beta,kl}_{\hat{x}\hat{x}}.\end{aligned} \tag{55}$$

REFERENCES

[1] D. K. Geller, "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, https://doi.org/10.2514/1.19447.

[2] D. C. Woffinden and D. K. Geller, "Relative angles-only navigation and pose estimation for autonomous orbital rendezvous," *Journal of Guidance, Control, and Dynamics*, Vol. 30, No. 5, 2007, pp. 1455–1469, https://doi.org/10.2514/1.28216.

[3] R. S. Christensen and D. Geller, "Linear covariance techniques for closed-loop guidance navigation and control system design and analysis," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, Vol. 228, No. 1, 2014, pp. 44–65, https://doi.org/10.1177/0954410012467717.

[4] D. K. Geller and D. P. Christensen, "Linear covariance analysis for powered lunar descent and landing," *Journal of Spacecraft and Rockets*, Vol. 46, No. 6, 2009, pp. 1231–1248, https://doi.org/10.2514/1.38641.

[5] J. Jang, S. Bhatt, M. Fritz, and D. Woffinden, "Linear covariance analysis for a lunar lander," *AIAA Guidance, Navigation, and Control Conference*, 2017, p. 1499, https://doi.org/10.2514/6.2017-1499.

[6] J. Williams, W. E. Brandenburg, D. Woffinden, and Z. R. Putnam, "Validation of linear covariance techniques for Mars entry, descent, and landing guidance and navigation performance analysis," *AIAA SciTech 2022 Forum*, 2022, p. 0745, https://doi.org/10.2514/6.2022-0745.

[7] R. H. Battin, *An introduction to the mathematics and methods of astrodynamics*. AIAA, 1999.

[8] D. Geller, D. Woffinden, and C. York, "Generalized Reference Targeting for Spaceflight," *47th Annual American Astronautical Society (AAS) Guidance, Navigation and Control (GN&C) Conference*, No. AAS-25-099, American Astronomical Society, 2025, https://ntrs.nasa.gov/citations/20250000047.

[9] N. B. Stastny and D. K. Geller, "Autonomous optical navigation at Jupiter: a linear covariance analysis," *Journal of Spacecraft and Rockets*, Vol. 45, No. 2, 2008, pp. 290–298, https://doi.org/10.2514/1.28451.

[10] P. S. Maybeck, *Stochastic models, estimation, and control*, Vol. 3. Academic press, 1982.

[11] A. H. Jazwinski, *Stochastic processes and filtering theory*. Courier Corporation, 2007.

[12] M. B. Rose and D. Geller, "Linear covariance techniques for powered ascent," *AIAA guidance, navigation, and control conference*, 2010, p. 8175, https://doi.org/10.2514/6.2010-8175.

[13] W. E. Brandenburg, J. Williams, D. Woffinden, and Z. R. Putnam, "Demonstration of Linear Covariance Analysis Techniques to Evaluate Entry Descent and Landing Guidance Algorithms, Vehicle Configurations, and Trajectory Profiles," *AIAA SciTech 2022 Forum*, 2022, p. 1215, https://doi.org/10.2514/6.2022-1215.

[14] K. Jin, D. Geller, and J. Luo, "Development and validation of linear covariance analysis tool for atmospheric entry," *Journal of Spacecraft and Rockets*, Vol. 56, No. 3, 2019, pp. 854–864, https://doi.org/10.2514/1.A34297.

[15] S. Yun, K. Tuggle, R. Zanetti, and C. D'Souza, "Sensor configuration trade study for navigation in near rectilinear halo orbits," *The Journal of the astronautical sciences*, Vol. 67, No. 4, 2020, pp. 1755–1774, https://doi.org/10.1007/s40295-020-00224-1.

[16] R. S. Park and D. J. Scheeres, "Nonlinear mapping of Gaussian statistics: theory and applications to spacecraft trajectory design," *Journal of guidance, Control, and Dynamics*, Vol. 29, No. 6, 2006, pp. 1367–1375, https://doi.org/10.2514/1.20177.

[17] J. L. Junkins and P. Singla, "How nonlinear is it? A tutorial on nonlinearity of orbit and attitude dynamics," *The Journal of the Astronautical Sciences*, Vol. 52, No. 1, 2004, pp. 7–60, https://doi.org/10.1007/BF03546420.

[18] E. L. Jenson and D. J. Scheeres, "Semianalytical measures of nonlinearity based on tensor eigenpairs," *Journal of Guidance, Control, and Dynamics*, Vol. 46, No. 4, 2023, pp. 638–653, https://doi.org/10.2514/1.G006760.

[19] J. Kulik, M. Ruth, C. Orton-Urbina, and D. Savransky, "Applications of induced tensor norms to guidance navigation and control," *Journal of Guidance, Control, and Dynamics*, Vol. 48, No. 10, 2025, pp. 2180–2198, https://doi.org/10.2514/1.G009054.

[20] J. Kulik, B. Hastings, and K. A. LeGrand, "Linear covariance fidelity checks and measures of non-Gaussianity," *Proceedings of the 2025 AAS/AIAA Space Flight Mechanics Meeting*, No. AAS 25-412, 2025.

[21] E. A. Wan and R. Van Der Merwe, "The unscented Kalman filter for nonlinear estimation," *Proceedings of the IEEE 2000 adaptive systems for signal processing, communications, and control symposium (Cat. No. 00EX373)*, Ieee, 2000, pp. 153–158, https://doi.org/10.1109/ASSPCC.2000.882463.

[22] D. A. Cunningham, R. P. Russell, and D. C. Woffinden, "Robust Trajectory Optimization for NRHO Rendezvous Using SPICE Kernel Relative Motion," *47th Annual American Astronautical Society (AAS) Guidance, Navigation and Control (GN&C) Conference*, No. AAS 25-115, American Astronautical Society, 2025, https://ntrs.nasa.gov/citations/20250000758.

[23] Q. P. Moon, B. A. Jones, R. P. Russell, and S. Stewart, "Improving LinCov Dispersion Analysis with Quadratic Covariance Correction: IM-2 Mission Analysis," *AAS GNC Conference*, No. AAS 25-067, 2025.

[24] Q. P. Moon, B. A. Jones, and R. P. Russell, "Generalized Nonlinear Spacecraft Targeting with Implicit Sensitivities," *AAS/AIAA Astrodynamics Specialist Conference*, No. AAS 26-970, 2026.

[25] D. Cunningham and R. P. Russell, "An interpolated second-order relative motion model for gateway," *The Journal of the Astronautical Sciences*, Vol. 70, No. 4, 2023, p. 26, https://doi.org/10.1007/s40295-023-00393-9.

[26] S. Boone and J. McMahon, "Directional state transition tensors for capturing dominant nonlinear effects in orbital dynamics," *Journal of Guidance, Control, and Dynamics*, Vol. 46, No. 3, 2023, pp. 431–442, https://doi.org/10.2514/1.G006910.

[27] B. A. Jones, A. Doostan, and G. H. Born, "Nonlinear propagation of orbit uncertainty using non-intrusive polynomial chaos," *Journal of Guidance, Control, and Dynamics*, Vol. 36, No. 2, 2013, pp. 430–444, https://doi.org/10.2514/1.57599.

[28] R. P. Russell, "Complete Kepler Propagator Including Second-Order Sensitivities," *Journal of Guidance, Control, and Dynamics*, 2026. to appear, `https://doi.org/10.2514/1.G009657`.

[29] J. Leith and R. P. Russell, "Mean to Osculating Element Transformations for Averaged Primer Vector Theory," *AAS/AIAA Astrodynamics Specialist Conference*, No. AAS 26-1008, 2026.

[30] K. J. DeMars, R. H. Bishop, and M. K. Jah, "Entropy-based approach for uncertainty propagation of nonlinear dynamical systems," *Journal of Guidance, Control, and Dynamics*, Vol. 36, No. 4, 2013, pp. 1047–1057, https://doi.org/10.2514/1.5898.

[31] S. Boone and J. McMahon, "Orbital guidance using higher-order state transition tensors," *Journal of Guidance, Control, and Dynamics*, Vol. 44, No. 3, 2021, pp. 493–504, https://doi.org/10.2514/1.G005493.

[32] Z. Yang, Y.-Z. Luo, and J. Zhang, "Nonlinear semi-analytical uncertainty propagation of trajectory under impulsive maneuvers," *Astrodynamics*, Vol. 3, No. 1, 2019, pp. 61–77, https://doi.org/10.1007/s42064-018-0036-7.

[33] R. S. Park and D. J. Scheeres, "Nonlinear semi-analytic methods for trajectory estimation," *Journal of guidance, control, and dynamics*, Vol. 30, No. 6, 2007, pp. 1668–1676, https://doi.org/10.2514/1.29106.

[34] L. Isserlis, "On a formula for the product-moment coefficient of any order of a normal frequency distribution in any number of variables," *Biometrika*, Vol. 12, No. 1/2, 1918, pp. 134–139, https://doi.org/10.2307/2331932.

[35] J. L. Junkins, J. D. Turner, and M. Majji, "Generalizations and applications of the lagrange implicit function theorem," *The Journal of the Astronautical Sciences*, Vol. 57, No. 1, 2009, pp. 313–345, https://doi.org/10.1007/BF03321507.

[36] A. D. Wylie and H. G. deVENZIN, "Onboard rendezvous navigation for the Space Shuttle," *Navigation*, Vol. 32, No. 3, 1985, pp. 197–220, 10.1002/j.2161-4296.1985.tb00905.x.

[37] R. Zanetti, K. J. DeMars, and R. H. Bishop, "Underweighting nonlinear measurements," *Journal of guidance, control, and dynamics*, Vol. 33, No. 5, 2010, pp. 1670–1675, https://10.2514/1.50596.

[38] P. G. Antreasian, C. D. Adam, K. Berry, J. Geeraert, K. M. Getzandanner, D. Highsmith, J. M. Leonard, E. J. Lessac-Chenen, A. H. Levine, J. V. McAdams, *et al.*, "OSIRIS-REx proximity operations and navigation performance at Bennu," *AIAA SCITECH 2022 Forum*, No. 2470, 2022, https://doi.org/10.2514/6.2022-2470.

[39] J. A. Christian, "A tutorial on horizon-based optical navigation and attitude determination with space imaging systems," *IEEE Access*, Vol. 9, 2021, pp. 19819–19853, https://doi.org/10.1109/ACCESS.2021.3051914.