GAUSSIAN PROCESS SEQUENTIAL FILTERING FOR SMALL BODY SLAM WITH SILHOUETTE-BASED MEASUREMENTS

Quinn P. Moon[1*], Courtney Hollenberg[2], Enrico M. Zucchelli[1] Noah Lifset[1], Brandon A. Jones[3], Ryan P. Russell[4], Shyam Bhaskaran[2], Daniel P. Lubey[2]; [1]PhD Student, Aerospace Engineering, University of Texas at Austin, Austin, TX, 78712. [2] Navigation Engineer, Outer Planet Navigation Group, Mission Design and Navigation Section, Jet Propulsion Laboratory, California Institute of Technology, [3] Associate Professor, Aerospace Engineering, University of Texas at Austin, Austin, TX, 78712. [4] Professor, Aerospace Engineering, University of Texas at Austin, Austin, TX, 78712. [*][quinnpmoon@utexas.edu]

Abstract. *A Gaussian Process Sequential Filter (GPSF) is developed for Simultaneous Localization and Mapping (SLAM). The GPSF models the small body shape through spatially correlated basis nodes and radii, and simultaneously estimates the body's shape, orientation, and spin, along with the spacecraft's relative position and velocity. The formulation of the GPSF is provided, including the root-solved analytic partials, measurement underweighting techniques, and covariance inflation methods. The performance of the GPSF is highlighted with a Monte Carlo study about multiple small bodies, including Lutetia, Eros, Toutatis, and Bennu.*

Introduction. Stereo-Photoclinometry (SPC) has proven successful for mapping and relative terrain navigation[1,2] in a variety of missions, including the Dawn mission to Ceres and Vesta[3,4] and the OSIRIS-REx mission to Bennu.[5] However, SPC is computationally intensive and requires considerable ground team support. Autonomous spacecraft reduce ground costs and time for maneuver planning. Here Simultaneous Localization and Mapping (SLAM) algorithms are implemented as a step towards the ultimate goal of unsupervised approach and navigation.

Significant work has been dedicated to the theory of small body shape modeling, including LIDAR imagery to track landmarks on Eros,[6] flash LIDAR imagery about Itokawa,[7] infrared imagery of Bennu,[8] and light-curve inversion.[9,10] Preliminary work by the authors implemented silhouette-based measurements for SLAM applications.[11–13] Silhouette-based measurements are readily produced by onboard visible spectrum cameras and can be used as direct measurements of the central body's convex hull. Silhouette-based measurements are thus well-suited for autonomous SLAM applications. This work presents the cumulative work of the authors to produce a sequential filter that incorporates silhouette-based measurements for SLAM applications about small bodies.

To reduce the computational intensity of the shape model, a Gaussian Process (GP) is implemented to model the small body's shape with basis nodes and radii. Wahlstrom and Ozkan introduced a sequential GP estimation filter for target tracking,[14] and we have implemented the GP technique for shape and pole estimation of Eros.[11,12] A GP Iterated Extended Kalman Filter was developed in a SLAM algorithm to also simultaneously estimate the observing spacecraft's relative position and velocity.[13]

The complete development and derivation of the GP Sequential Filter (GPSF) is thus presented. The GPSF incorporates silhouette-based measurements to simultaneously estimate the spacecraft's relative position and velocity, while estimating the small body's pole, spin, and shape. The methodology of modeling small bodies with GP nodes and radii are described and implemented into the filter. The GPSF incorporates the complete analytic measurement partials to account for data association sensitivities. The GPSF underweights measurements to account for measurement nonlinearity and data association error. As SLAM filters often become overconfident,[15,16] the GPSF incorporates covariance inflation to ensure filter pessimism. As a proper Monte Carlo analysis, where the truth is perturbed based on a statistically consistent covariance matrix,[17] is yet to be performed, the main objective of the current paper is to fill this important gap in the sequential filter design and validation process. The Monte Carlo studies are performed about multiple small bodies, including Lutetia, Eros, Toutatis, and Bennu. The initial covariances for the Monte Carlo studies are obtained through a maximum likelihood GP Batch Filter,[18] which ensures that the highly correlated elements of the GP states are consistent.

Gaussian Process Small Body Shape Modeling. The small body shapes are modeled via GP regression. The GP regression models the entire body with radii at discrete nodes and a covariance function (or kernel function). More basis nodes and radii increase the fidelity of the shape model, as illustrated in Fig. 1

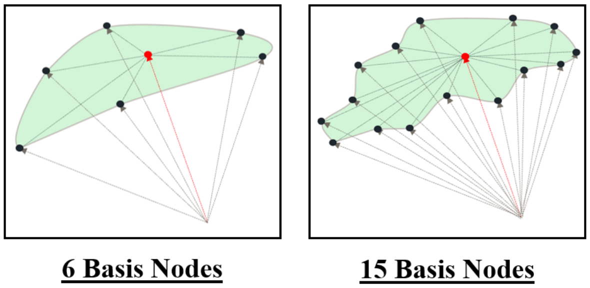


Figure 1. GP shape modeling on a representative 2D body.

The GP basis set, or simply the basis set, consists of nodes as unit vectors $\boldsymbol{e}'$ and their associated radii f'. A single basis node is denoted as $\boldsymbol{e}'$, whereas the set of n basis nodes are denoted as a 3 by n matrix $\boldsymbol{E}'$:

$$\boldsymbol{E}' = \left[\begin{array}{cccc} \boldsymbol{e}'_1 & \boldsymbol{e}'_2 & \cdots & \boldsymbol{e}'_n \end{array} \right]. \tag{1}$$

The basis set, denoted with a prime, is used to predict the radii $\boldsymbol{f}$ at a set of measured nodes $\boldsymbol{E}$ according to the GP prediction equation:[19]

$$\boldsymbol{f}(\boldsymbol{E}) = \boldsymbol{C}(\boldsymbol{E}, \boldsymbol{E}')\boldsymbol{C}^{-1}(\boldsymbol{E}', \boldsymbol{E}')\boldsymbol{f}', \tag{2}$$

where $\boldsymbol{C}(\cdot, \cdot)$ is the GP covariance function (kernel function). The covariance function aids in representing complicated, multi-dimensional data simply; for greater detail on GPs and covariance functions, see Rasmussen and Williams.[19] Thus, GP regression models the entire shape of the small body with just a few nodes, radii, and their kernel function, making them well-suited for autonomous navigation.

The GPSF and GP batch filter[12, 13, 18] previously implemented a covariance function that correlates nodes through their angular distance according to

$$\boldsymbol{C}_{n,m}(\boldsymbol{E}, \boldsymbol{E}') = \sigma_c^2 \exp\left(-\frac{\text{acos}(\boldsymbol{E}_n^T \boldsymbol{E}'_m)}{l^2}\right), \tag{3}$$

where $\boldsymbol{C}(\cdot, \cdot)$ is a matrix with each element analyzed at the nth and mth node of the inputs. Thus, the dimension of the covariance function depends on the number of input nodes. The strength of the correlation decays with increasing angular distance between two nodes. The scaling factor l adjusts the spatial correlation's strength, and σ_c^2 scales the entire covariance function. The GPSF operates as desired using the covariance function in Eq. 3, and is able to fully estimate the central shape. However, when a measurement is processed at a basis node ($\boldsymbol{e} = \boldsymbol{e}'$), a singularity is exhibited when taking the measurement partials. This singularity is clearly seen below when analyzing the measurement partials analytically in Appendix D. While no true singularity typically occurs during simulation, numerical issues arise causing poor estimation performance. Thus, the GPSF no longer incorporates the covariance function in Eq. 3.

Different covariance functions yield different predictions, and some are better suited to specific scenarios than others.[20] A common choice is the Radial Basis Function (RBF), or Gaussian Kernel:

$$\boldsymbol{C}_{n,m}(\boldsymbol{E}, \boldsymbol{E}') = \sigma_c^2 \exp\left(-\frac{||\boldsymbol{E}_n - \boldsymbol{E}'_m||^2}{l^2}\right). \tag{4}$$

The RBF kernel has no singularities associated with its measurement partials. However, the RBF kernel suffers from instability due to numerical precision with its rank. When implemented within the GPSF, the RBF yields poor shape representation and is unable to accurately capture the truth state.

The current GPSF thus utilizes a rectified version of the covariance function in Eq. 3, maintaining previous filter performance while avoiding errors associated to singularity proximity. The rectified covariance function is

$$\boldsymbol{C}_{n,m}(\boldsymbol{E}, \boldsymbol{E}') = \sigma_c^2 \exp\left(\frac{\text{acos}(\kappa) - \text{acos}(\kappa \boldsymbol{E}_n^T \boldsymbol{E}'_m)}{l^2}\right), \tag{5}$$

where $0 < \kappa < 1$. The rectification avoids any singularities within the measurement partials, and maintains autocorrelation when $\boldsymbol{E} = \boldsymbol{E}'$. For all simulations presented, the rectification is set at $\kappa = 0.999999$. Setting $\kappa = 1$ makes Eq. 3 and Eq. 5 equivalent. Further detail on the rectified kernel's singularity avoidance is found in Appendix A.

Thus, a small body is modeled by a tractable GP basis. Consider Fig. 2, depicting Eros modeled with a high fidelity polyhedral model. The GP regression aims to model this high fidelity shape with fewer basis nodes.

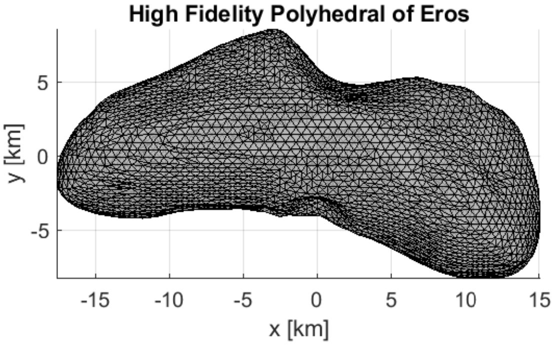
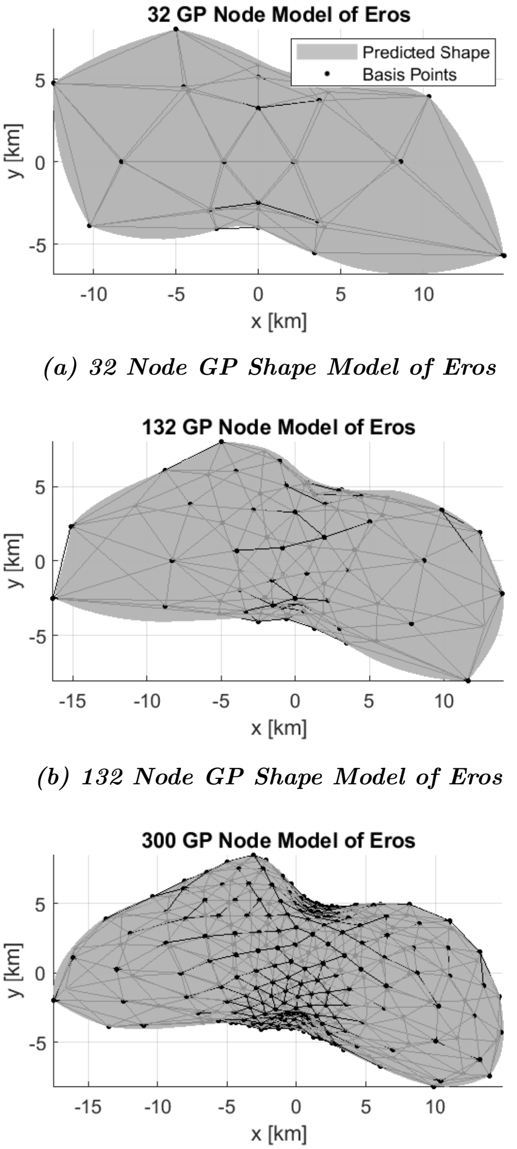


Figure 2. High Fidelity Polyhedral Model of Eros

The shape of Eros is similarly captured using GP regression with far fewer points. Figure 3 depicts GP models of Eros with spherically distributed basis nodes. Note that adding nodes to the GP basis set increases the shape model fidelity. While greater shape fidelity is desired, increasing the number of states significantly slows down the filter. Currently, the basis nodes are distributed by evenly spacing the nodes on a unit sphere. When basis nodes are spherically distributed, increasing the number of basis nodes demonstrates significant diminishing returns. Illustrated in Fig. 3, few nodes are distributed along the extreme ends of Eros. Thus, a spherical distribution of basis nodes is not the best distribution for elongated or highly non-spherical bodies such as Eros. To best utilize the basis nodes, a mission may initially model the small body with spherically distributed nodes, but rearrange the basis nodes once the GPSF obtains a shape estimate. An improved distribution of basis nodes is an area of ongoing research.

The truth shapes used in simulation are modeled as a GP with 300 basis nodes, see Fig. 3c. Monte Carlo analysis requires randomly perturbed shapes from a prior distribution, i.e., randomly sampled radii at the basis nodes. We use a maximum likelihood GP batch filter[21] to produce GP prior distributions that can be randomly sampled. Further details on the batch estimator are provided below. The truth shape models used in the batch process are obtained from GP regression with high fidelity polyhedral models. The high fidelity shape models are obtained from NASA's Planetary Data System, pulled from Clark[22] and Hergenrother[23] for Bennu, Hudson[24] for Toutatis, Gaskell[25] for Eros, and Gaskell and Carry[26] for Lutetia.



(a) 32 Node GP Shape Model of Eros

(b) 132 Node GP Shape Model of Eros

(c) 300 Node GP Shape Model of Eros

Figure 3. GP Shape Models of Eros

The silhouette-based measurements are generated directly from the GP modeled shapes. Previous work discussed the methodology to process images of the asteroid's real shape and convert the data to lower fidelity GP shapes,[13] akin to processing real-time gravitational acceleration with a finite set of Stokes' coefficients. Here the truth data and images are simulated using the GP truth models directly, akin to modeling the true dynamics with a finite degree of spherical harmonics model. Greater detail on image generation is provided when discussing the pinhole camera measurement function.

Silhouette-Based Measurements. Silhouette-based measurements are generated by extracting the visible edge from an image of the partially illuminated body. Figure 4 depicts a Blender simulated image of a partially illuminated Eros.[27] An image processing tool then extracts the longest continuous edge, obtaining the visible edge of the body.[13] The GPSF has successfully implemented this measurement generation technique,[13] demonstrating feasibility of a suitable image processing pipeline, though future work is needed in this area to address more of the challenges associated with working with real spacecraft imagery. Blender simulations are, however, time consuming, especially when the true asteroid's shape and orientation are perturbed for hundreds of Monte Carlo simulations.

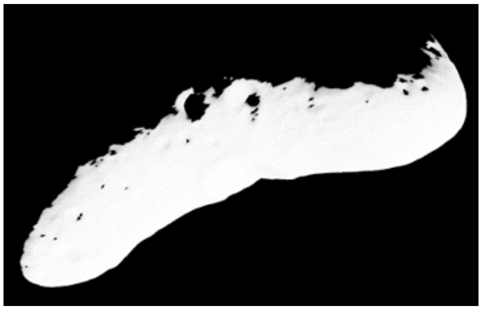


Figure 4. Blender Simulated Image of Eros

Alternatively, Fig. 5 depicts a simulated image from the new, customized measurement generation algorithm in MATLAB. Thousands of points on the truth shape are projected into the image plane, and MATLAB's *boundary* command provides a discrete set of points representing the potential visible edge. For each boundary point, a ray is traced from the point toward the Sun. Rays traced from self-shadowed boundary points will cross into the small body, and are thus not along the visible edge. The gold line in Fig. 5 depicts the calculated visible edge after trimming and checking for self-shadowing. Note the customized routine provides a visible edge similar to the Blender image in Fig. 4.

The new measurement generation routine readily produces silhouette-based measurements for any small body and mission scenario, ideal for Monte Carlo simulations. Each Monte Carlo generates measurements from a randomly perturbed shape, and incorporates the data into the GPSF. Note that the described customized routine assumes perfect knowledge of the truth state to create truth measurements, and is thus only used for simulation purposes.

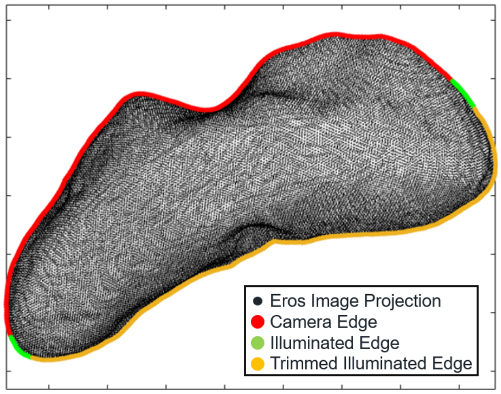


Figure 5. MATLAB Simulated Image of Eros

Pinhole Camera. The measurement equation is based on a pinhole camera where discrete points along the three-dimensional visible edge are projected into the two-dimensional camera plane. The measurement equations are expressed as:

$$\begin{bmatrix} x_j \\ y_j \\ z_j \end{bmatrix} = \boldsymbol{s} = \boldsymbol{T}_C^I \left[\boldsymbol{T}_I^A \boldsymbol{r}^A_{e_j/a} - \boldsymbol{r}^I_{s/a} \right], \tag{6}$$

$$\boldsymbol{h}(\boldsymbol{x}) = \begin{bmatrix} u_j \\ v_j \end{bmatrix} = \boldsymbol{M}\boldsymbol{\rho}(\boldsymbol{s}) = \boldsymbol{M} \begin{bmatrix} -x_j/z_j \\ y_j/z_j \\ 1 \end{bmatrix}, \tag{7}$$

where $\boldsymbol{M}$ denotes a linear mapping corresponding to the pinhole camera's field of view and geometric parameters, $\boldsymbol{r}^I_{s/a}$ the relative position of the spacecraft with respect to the asteroid in the inertial frame, $\boldsymbol{r}^A_{e/a}$ the position of a visible edge point relative to the asteroid in the asteroid fixed frame, calculated with

$$\boldsymbol{r}^A_{e_j/a} = f_j \boldsymbol{e}_j. \tag{8}$$

The rotation matrix $\boldsymbol{T}_C^I$ maps the inertial frame to the camera frame. The rotation matrix $\boldsymbol{T}_I^A$ maps the asteroid frame to the inertial frame. The subscript j denotes the jth data point along the visible edge.

The MATLAB boundary routine generates measurements using the truth state and Eqs. 6-7. The pinhole camera equations also generate predicted measurements, but requires a procedure to associate each measurement to a predicted measurement.

Data Association. Each discrete measurement, generated along the visible edge of the shape, needs to be associated to a body fixed location, generating a predicted measurement. A perfect data association procedure would correctly match a predicted measurement with its corresponding node $\boldsymbol{e}$. However, the correct match is not known to the filter, necessitating a data association procedure.

The GPSF uses geometric constraints to match a point on the silhouette to a point along the edge of the predicted projection of the body's shape. The unit vector pointing from the spacecraft to the asteroid's visible edge can be extracted from the vector $\boldsymbol{\rho}(\boldsymbol{s})$. This unit vector is here denoted as $\boldsymbol{i_a}$. A search plane for data association is defined by $\boldsymbol{i_a}$ and the estimated position vector $\hat{\boldsymbol{r}}$ according to

$$\boldsymbol{\zeta} = \hat{\boldsymbol{r}} \times \boldsymbol{i_a}, \tag{9}$$

where $\boldsymbol{\zeta}$ represents the vector normal to the search plane. The unit vector of the state position estimate $\hat{\boldsymbol{r}}$ is denoted as $\boldsymbol{i_{\hat{r}}}$. The unit vector $\boldsymbol{i_{\hat{r}}}$ is rotated within the plane by a rotation angle γ to generate estimated measured nodes according to:

$$\boldsymbol{\Gamma_\zeta} = \text{expm}\left(\left[\boldsymbol{i_\zeta} \times \right] \gamma \right), \tag{10}$$

$$\boldsymbol{e} = \boldsymbol{T}_A^I \boldsymbol{\Gamma_\zeta} \boldsymbol{i_r}, \tag{11}$$

where $[\cdot\times]$ denotes the cross-product matrix, and $\boldsymbol{\Gamma_\zeta}$ the matrix applied to rotate $\boldsymbol{i_r}$ in the plane. An exact expression for the matrix exponential exists for skew-symmetric matrices, such that

$$\boldsymbol{\Gamma_\zeta} = \boldsymbol{I} + \sin(\gamma) \left[\boldsymbol{i_\zeta} \times \right] + (1 - \cos(\gamma)) \left[\boldsymbol{i_\zeta} \times \right]^2. \tag{12}$$

Using Eq. 12 is faster computationally than numerical techniques to calculate the matrix exponential, and is useful for the measurement partials.

The associated measurement node $\boldsymbol{e}$ then predicts the estimated radii using the GP regression equation in Eq. 2. The rotation angle γ is obtained by minimizing the objective function with respect to γ[13]

$$J(\gamma) = -\frac{f(\boldsymbol{e}(\gamma)) \sin(\gamma)}{\sqrt{f^2(\boldsymbol{e}(\gamma)) + r^2 - 2f(\boldsymbol{e}(\gamma)) r \cos(\gamma)}}. \tag{13}$$

Minimizing the $J(\gamma)$ provides the visible edge of the central body. Thus each measured data point is associated to an estimated measurement through the data association algorithm.

The data association process is known to be naive, as the truth node is not guaranteed to lie in the search plane. Thus, the node produced by the data association process for the predicted measurements is expected to deviate from the true node. This difference is referred to as data association error, expressed in Eq. 14

$$\Delta \boldsymbol{e} = \boldsymbol{e} - \hat{\boldsymbol{e}}. \tag{14}$$

Data association errors within the image plane are expected to shrink as the position and pole state estimates converge to the truth. Out of plane data association errors are typically due to shape estimate errors, causing the shape estimate to be updated incorrectly. Due to the nature of SLAM algorithms,[13,15,16,21] large, incorrect updates to the shape will also cause the remaining states to diverge.

More sophisticated means of data association have been formulated, but significantly increase the filter's run-time to undesirable levels for autonomous missions. To account for the data association error, the input nodes and output radius can be treated as stochastic processes,[28, 29] and an appropriate compensation added to underweight the measurements. Provided a covariance exists for the distribution of $\Delta \boldsymbol{e}$, it can then be linearly mapped to the measurement space according to

$$\boldsymbol{U} = \boldsymbol{H_e} \boldsymbol{P_e} \boldsymbol{H_e}^T, \tag{15}$$

where $\boldsymbol{P_e}$ denotes the data association error covariance, and $\boldsymbol{H_e}$ denotes the partial of the measurement function with respect to a measured node. However, no covariance of the data association error is readily available. Furthermore, the errors are non-Gaussian, and demonstrate no identifiable pattern.

Previous work with the GP batch filter demonstrated that compensation for the data association error can be achieved by underweighting the measurements.[18] While the covariance $\boldsymbol{P_e}$ is not known, steps are taken to conservatively apply measurement underweighting based on the technique in Eq. 15. This technique is described in greater detail in a later section with the GPSF's measurement underweighting.

Mapping data association errors from the input nodes to the measurement space requires the measurement partials. Previous work on the GPSF utilized numerical, complex step partial derivatives,[13, 30] which are slow and do not provide intermediate partials, such as $\boldsymbol{H_e}$ needed for the data association compensation. As such, the steps taken to obtain the complete, root-solved analytic partials are provided below.

Measurement Partials. This section summarizes the process of deriving the measurement partial derivative. The full results are provided in Appendix A. The measurement equation with data association is a function of the state $\boldsymbol{x}$, the jth measured radius f_j, node $\boldsymbol{e}_j$, and optimization angle γ_j:

$$\boldsymbol{h}\left[\boldsymbol{x}, \boldsymbol{e}_j(\boldsymbol{x}, \gamma_j(\boldsymbol{x})), f_j(\boldsymbol{x}, \boldsymbol{e}_j(\boldsymbol{x}, \gamma_j(\boldsymbol{x})))\right]. \tag{16}$$

Thus, the measurement derivative of the jth data point with respect to a state element $\boldsymbol{x}_i$ can be found through the chain rule according to:

$$\begin{aligned}\frac{d\boldsymbol{h}}{d\boldsymbol{x}_i} = \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}_i} + \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{e}}\left(\frac{\partial \boldsymbol{e}}{\partial \boldsymbol{x}_i} + \frac{\partial \boldsymbol{e}}{\partial \gamma}\frac{\partial \gamma}{\partial \boldsymbol{x}_i}\right) \\ + \frac{\partial \boldsymbol{h}}{\partial f}\left(\frac{\partial f}{\partial \boldsymbol{x}_i} + \frac{\partial f}{\partial \boldsymbol{e}}\left(\frac{\partial \boldsymbol{e}}{\partial \boldsymbol{x}_i} + \frac{\partial \boldsymbol{e}}{\partial \gamma}\frac{d\gamma}{d\boldsymbol{x}_i}\right)\right),\end{aligned} \tag{17}$$

where the j subscript is dropped for compactness. Previous implementations of the GPSF omitted the root-solved partials with respect to γ.[11–13] Finding the measurement partials numerically with the root-solved components proves to be time consuming, as additional runs of the data association routine are required. Fixing the optimization angle γ, and thus omitting the root-solved partials, permits the use of fast, complex step numerical derivatives.[30] For numerous simulations, omitting the root-solved partials can be justified by the inequality

$$\frac{\partial \boldsymbol{e}}{\partial \boldsymbol{x}_i} \gg \frac{\partial \boldsymbol{e}}{\partial \gamma}\frac{d\gamma}{d\boldsymbol{x}_i}. \tag{18}$$

However, assuming the root-solved partials are always inconsequential yields issues with robustness. Furthermore, including the root-solved partials has been demonstrated to generally improve filter performance. As such, the complete set of partials is implemented.

Furthermore, calculating the analytic partials is approximately 500-1000 times faster than calculating them numerically. This significant runtime improvement marks a significant step towards implementing the GPSF for autonomous SLAM applications.

Gaussian Process Sequential Filter. The GPSF estimates the state vector

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{r}^T & \boldsymbol{v}^T & \alpha & \delta & \dot{\theta} & \theta_0 & \boldsymbol{f}'^T \end{bmatrix}^T, \tag{19}$$

where $\boldsymbol{r}$ and $\boldsymbol{v}$ denote the spacecraft's position and velocity with respect to the asteroid, α and δ the right ascension and declination of the asteroid axis of rotation, $\dot{\theta}$ the asteroid's spin about the prime meridian θ_0, and $\boldsymbol{f}'$ the basis radii of the asteroid at the predetermined basis set $\boldsymbol{E}'$. Recall the basis nodes and radii are marked with a prime to distinguish between measured and basis values.

The pole elements map the inertial frame to the asteroid frame using Euler angles according to the direction cosine matrix

$$\boldsymbol{T}_A^I = \boldsymbol{T}_3(\dot{\theta}t + \theta_0)\boldsymbol{T}_1(\tfrac{\pi}{2} - \delta)\boldsymbol{T}_3(\tfrac{\pi}{2} + \alpha), \tag{20}$$

where $\boldsymbol{T}_1$ and $\boldsymbol{T}_3$ with arbitrary input ϕ are:

$$\boldsymbol{T_1}(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & \sin(\phi) \\ 0 & -\sin(\phi) & \cos(\phi) \end{bmatrix}, \tag{21}$$

$$\boldsymbol{T_3}(\phi) = \begin{bmatrix} \cos(\phi) & \sin(\phi) & 0 \\ -\sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{22}$$

The GPSF is derived from nonlinear conditional probability and follows the propagation and update steps of an iterated extended Kalman filter.[17, 31, 32] The state covariance is propagated forward according to

$$\boldsymbol{P}_{k+1} = \boldsymbol{\Phi}_{k+1,k}\boldsymbol{P}_k\boldsymbol{\Phi}_{k+1,k}^T, \tag{23}$$

where $\boldsymbol{\Phi}$ is the State Transition Matrix, calculated by propagating forward along the estimated trajectory using 2-body gravitational relative motion. The state is updated according to:

$$\boldsymbol{K} = \boldsymbol{P}^-\boldsymbol{H}\boldsymbol{S}^{-1}, \tag{24}$$

$$\hat{\boldsymbol{x}}^+ = \hat{\boldsymbol{x}}^- + \boldsymbol{K}(\boldsymbol{z} - \boldsymbol{h}\left(\hat{\boldsymbol{x}}^-\right)), \tag{25}$$

where the superscripts "$-, +$" denote before and after a measurement update, respectively. The state covariance is updated according to

$$\boldsymbol{P}^{+} = (\boldsymbol{I} - \boldsymbol{K}\boldsymbol{H})\boldsymbol{P}^{-}(\boldsymbol{I} - \boldsymbol{K}\boldsymbol{H})^{T} + \boldsymbol{K}\boldsymbol{R}^{*}\boldsymbol{K}^{T} + \boldsymbol{P}_{lb}, \quad (26)$$

where $\boldsymbol{R}^{*}$ is later defined when discussing measurement underweighting. The matrix $\boldsymbol{P}_{lb}$ serves as a lower bound on the filter covariance, commonly applied to artificially inflate the filter covariance in SLAM filters.[15,16] It is common in SLAM applications for the filter to become over-confident, especially with respect to the states associated with the measurement source.[15,16] This pattern results in the GPSF becoming over-confident in estimating the central body's shape. Thus, the matrix $\boldsymbol{P}_{lb}$ is set as a diagonal matrix with non-zero elements for the shape states. Ultimately, the covariance inflation implemented through $\boldsymbol{P}_{lb}$ ensures the GPSF does not become over-confident, while also decorrelating the shape states after update.

The innovation covariance $\boldsymbol{S}$, classically formulated as $\boldsymbol{S} = \boldsymbol{H}\boldsymbol{P}^{-}\boldsymbol{H}^{T} + \boldsymbol{R}$, includes additional terms to underweight the measurements. The process of simultaneously estimating the state in Eq. 19 is a sensitive process, where large state updates can cause the filter to diverge from the truth. Thus, the GPSF utilizes the combined innovation covariance:

$$\boldsymbol{R}^{*} = \beta\boldsymbol{H}\boldsymbol{P}^{-}\boldsymbol{H}^{T} + \boldsymbol{H}_{f}\boldsymbol{\mathcal{C}}\boldsymbol{H}_{\boldsymbol{f}}{}^{T} + \boldsymbol{R} + \boldsymbol{U} + \boldsymbol{L}, \quad (27)$$

$$\boldsymbol{S} = \boldsymbol{H}\boldsymbol{P}^{-}\boldsymbol{H}^{T} + \boldsymbol{R}^{*}, \quad (28)$$

where $\boldsymbol{R}$ is the measurement noise covariance

$$\boldsymbol{R} = \sigma_{\rho}^{2}\boldsymbol{I}. \quad (29)$$

The scalar β serves to approximate the innovation weight that would be applied in a second order EKF using Lear's method.[33,34] The matrix $\boldsymbol{\mathcal{C}}$ arises from GP regression errors, and are mapped to the measurement space with $\boldsymbol{H}_{\boldsymbol{f}}$. The matrices $\boldsymbol{L}$ and $\boldsymbol{U}$ serve as a measurement rejection scheme and data error compensation, respectively. These measurement underweighting schemes are further described in detail here.

GP Regression Error. Shape modeling with GP regression requires calculating the conditional probability of the measured radii to the basis radii. Originally proposed by Wahlstrom and Ozkan for extended target tracking,[14] the covariance of the measurements $\boldsymbol{z}$ conditioned on the measured radii is expressed as:

$$\boldsymbol{\mathcal{C}} = \boldsymbol{C}(\boldsymbol{E}, \boldsymbol{E}) - \boldsymbol{C}(\boldsymbol{E}, \boldsymbol{E}')\boldsymbol{C}^{-1}(\boldsymbol{E}', \boldsymbol{E}')\boldsymbol{C}(\boldsymbol{E}', \boldsymbol{E}), \quad (30)$$

$$\boldsymbol{P}_{\boldsymbol{z}|\boldsymbol{f}} = \boldsymbol{H}_{\boldsymbol{f}}\boldsymbol{\mathcal{C}}\boldsymbol{H}_{\boldsymbol{f}}{}^{T}. \quad (31)$$

The conditional probability is then added to the innovation. For further detail on the derivation of the conditional probability, see the original work by Wahlstrom and Ozkan[14] and previous work on the GP filter.[11–13,21]

The scale of the covariance in Eq. 31 is determined by σ_{c}^{2} i.e., the vertical scale in Eq. 3. When the true shape is not modeled as a GP, or modeled with more nodes than the estimated shape, the scaling factor should be properly tuned. However, as the truth shapes are here modeled with the same number of nodes as the estimated shapes, σ_{c}^{2} can be quite small. Furthermore, small values of σ_{c}^{2} are recommend by Whalstrom and Ozkan when measurements are produced by the target's surface, as is the case for silhouette-based measurements.

The strength of the correlations is determined by l, or the horizontal scale. The l factor is crucial, as it allows for the entire shape to be updated with silhouette-based measurements. Regions of the surface that experience constant shadowing, such as craters, can still be estimated when incorporating a larger l scale, though caution is required as no direct observations of these regions are available. The GPSF thus incorporates errors associated with GP regression.

Measurement Rejection. A measurement rejection scheme is implemented to mitigate filter divergence. The scheme follows the idea set forth by Navon,[35] where measurements are rejected through additional underweighting. The measurements are rejected by checking the normalized innovations according to

$$\overline{\boldsymbol{\delta z}}_{i} = \tfrac{\boldsymbol{z}_{i} - \hat{\boldsymbol{z}}_{i}}{\sqrt{\boldsymbol{S}_{ii}}}, \quad (32)$$

$$\boldsymbol{l}_{i} = \begin{cases} 1, \text{ for } \overline{\boldsymbol{\delta z}}_{i} > 3, \\ 0, \text{ otherwise} \end{cases} \quad (33)$$

$$\boldsymbol{L} = c_{l} \cdot \operatorname{diag}(\boldsymbol{l}). \quad (34)$$

The parameter c_{l} varies the strength of the measurement rejection. The measurement rejection matrix $\boldsymbol{L}$ is then added to the innovation covariance. Note the measurement rejection scheme is applied using the innovation $\boldsymbol{S}$ prior to other underweighting sources.

Lear's Method. To account for nonlinearities associated with the measurement equation, the GPSF utilizes Lear's method to provide a conservative estimate of the covariance from a second order Kalman filter.[34] The underweighting is done to provide a lower-bound estimate to the second order weight matrix $\boldsymbol{B}$, according to

$$\beta \cdot \operatorname{trace}\left(\boldsymbol{H}\boldsymbol{P}\boldsymbol{H}^{T}\right) \geq \operatorname{trace}\left(\boldsymbol{B}\right). \quad (35)$$

When the second order measurement partials are known, β can be calculated to ensure the inequality in Eq. 35 holds. It is difficult to select a β that satisfies the inequality for all measurements, since the second order terms are not known.

The GPSF implements an adaptive approximation of β_{k} at time t_{k} according to[13,34]

$$\beta_{k} = \beta_{0}\frac{\operatorname{trace}(\boldsymbol{P}_{k})^{2}}{\operatorname{trace}(\boldsymbol{H}_{k}\boldsymbol{P}_{k}\boldsymbol{H}_{k}^{T})}\frac{\operatorname{trace}(\boldsymbol{H}_{0}\boldsymbol{P}_{0}\boldsymbol{H}_{0}^{T})}{\operatorname{trace}(\boldsymbol{P}_{0})^{2}}. \quad (36)$$

Previous implementations of the GPSF used $\beta_{0} = 0.2$, justified by the successful implementation on the space shuttle.[33] However, numerically calculating the second order measurement terms according to Roth and Gustafsson[36] has shown that $\beta_0 \approx 0.01$ is sufficient to satisfy Eq. 35 at t_0 for Eros. Thus the term $\beta_k \boldsymbol{H}_k \boldsymbol{P}_k \boldsymbol{H}_k^T$ is added to the innovation covariance at t_k to account for measurement nonlinearity.

Data Association Error Underweighting. The data association can incorrectly match measurements with points on the edge of the predicted image. This produces errors in the GPSF-estimated shape. In order to account for data association errors, a measurement underweighting scheme is introduced that treats the node error as white noise.

The node error covariance $\boldsymbol{P_e}$ is a 2×2 matrix expressed as

$$\boldsymbol{P_e} = \sigma_e^2 \cdot \boldsymbol{I}, \tag{37}$$

where σ_e represents the data association error in latitude and longitude in the body fixed frame. To conservatively approximate the data association errors, each data point is underweighted according to

$$\boldsymbol{U} = \frac{||\boldsymbol{H_e} \boldsymbol{P_e} \boldsymbol{H_e}^T||}{n_m} \cdot \boldsymbol{I}_{n_m}, \tag{38}$$

where n_m is the number of data points of an image. This technique applies a weighted average to underweight each measurement point. The matrix $\boldsymbol{U}$ is then added to the innovation covariance.

The strength of this underweighting scheme is dependent on the input σ_e^2. The data association errors are known to decrease as the estimated state converges to the truth. Thus, the value σ_e^2 used in Eq. 37 is a function of time according to

$$\sigma_e^2(t_k) = \sigma_e^2(t_0) \exp\left[5\left(1 - \frac{\text{trace}(\boldsymbol{P_{rf}}(t_0))}{\text{trace}(\boldsymbol{P_{rf}}(t_k))}\right)\right] + \sigma_{e_{lb}}^2, \tag{39}$$

where $\boldsymbol{P_{rf}}$ denotes the condensed state covariance, consisting only of the position and shape elements. The term $\sigma_{e_{lb}}^2$ enforces a lower bound. Equation 39 causes the data association noise to decrease as the filter covariance shrinks.

Various schemes can be used to change the data association noise over time with various degrees of efficacy. The exponential decay method in Eq. 39 is chosen based on how the position and shape errors converge with a perfect data association Monte Carlo study. Future work includes alternative methods to compensate for data association errors.

Filter Tuning Parameters. The GPSF has multiple tuning parameters, summarized in Table 1. Without properly tuning these additional parameters, the GPSF produces optimistic estimates that may not converge to the true states. The parameters are tuned in order to provide estimates that converge to the truth, but do so slowly and conservatively. Greater detail is provided below when describing the simulation setup what values are appropriate for small body SLAM applications.

Table 1. GPSF Tuning Parameters

Symbol	Adjusts	Units
σ_c	Kernel Vertical Scale	km
l	Kernel Horizontal Scale	–
σ_ρ	Measurement Noise	Pixels
β_0	Lear's Method	–
σ_e,$\sigma_{e_{lb}}$	Data Association Error	Radians
c_l	Measurement Rejection	Pixels
$\boldsymbol{P}_{lb}$	Filter Covariance Lower Bound	km^2

Monte Carlo Simulation Setup. Each Monte Carlo test randomly samples from the prior distribution to perturb the truth state from an initial, nominal state. The gravitational parameter, spin rate about the prime meridian, and average radius for Lutetia, Eros, Toutatis, and Bennu are provided in Table 2. The gravitational parameters and spin rate are obtained from NASA's Planetary Data System.[37] Note that the spin is assumed to be about a single axis starting at the prime meridian, despite some demonstrating non-principal axis rotation.[24] The mean radii are calculated from the 300 GP radii states. Each asteroid is modeled as a point mass with the gravitational parameter. Future work will incorporate non-spherical gravity for mission phases with lower altitudes.

Table 2. Asteroid Parameters

Asteroid	$\mu \left[\frac{\text{km}^3}{\text{s}^2}\right]$	f_{mean} [km]	$\dot{\theta} \left[\frac{\text{deg}}{\text{hr}}\right]$
Lutetia	1.14×10^{-1}	48.15	44.07
Eros	3.92×10^{-4}	7.33	68.31
Toutatis	1.27×10^{-6}	1.15	2.77
Bennu	4.87×10^{-9}	0.25	84.19

Each trajectory is a circular, polar orbit with a period of 52 hours. For each simulation, an image is taken and processed at 30 minute intervals. The total simulation time is set to one week, about 3.23 orbital periods. The lighting conditions are set such that the images taken by the spacecraft pass from fully illuminated to fully eclipsed. The nominal orbit, presented as the magnitude of the position and velocity vectors, and asteroid orientation are presented in Table 3. The pole values are the same as those used by Hollenberg,[13] while the orbit for each asteroid is adjusted to have the same 52 hour period, approximately modeled after the NEAR mission to Eros.[38]

Table 3. Asteroid Parameters

Asteroid	$\|\|\boldsymbol{r}\|\|$ [km]	$\|\|\boldsymbol{v}\|\|$ [km]	α [deg]	δ [deg]	θ_0 [deg]
Lutetia	457.56	15.82	0	60	-27
Eros	69.19	2.39	0	60	-27
Toutatis	10.23	0.36	0	60	-27
Bennu	1.60	0.05	0	60	-27

Figure 6 depicts one orbital period about each asteroid in the body-fixed frame. The trajectory about Toutatis provides the least coverage while the trajectory about Bennu provides the most. This is due to the relatively slow and fast spin rates of Toutatis and Bennu, respectively.

For each Monte Carlo run, the prior distribution is sampled to define a true shape, rotation rate and axis, and relative spacecraft position and velocity states. All states are modeled as Gaussian except for the shape, modeled as a GP distribution. The values of the initial standard deviations are reported in Table 4.

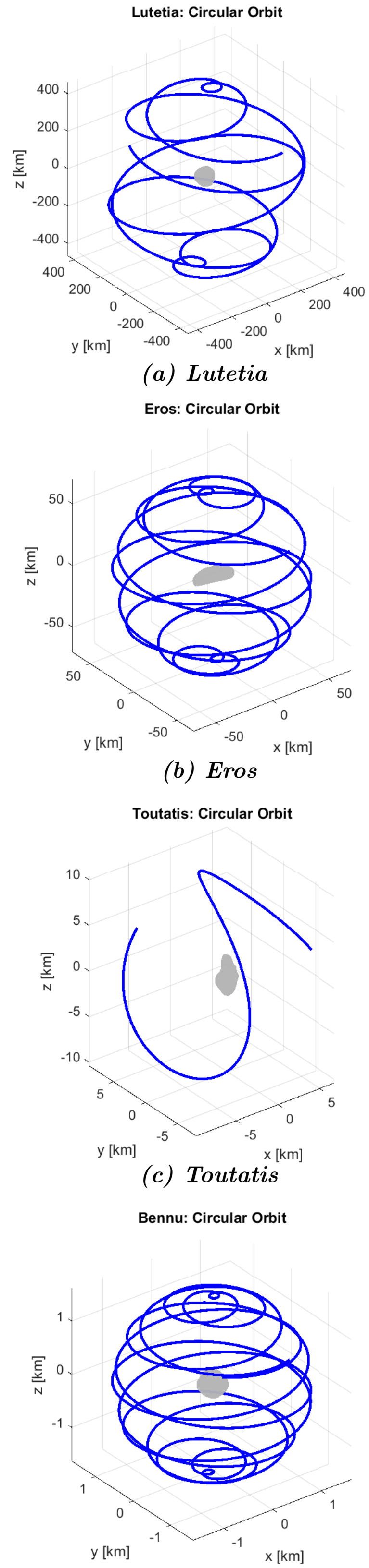
Table 4. Monte Carlo Initial Standard Deviation Values

Std. Dev.	Units	Lutetia	Eros	Toutatis	Bennu
σ_r	[km]	6.50	1.00	0.15	0.02
σ_v	[cm/s]	8.82	1.33	0.20	0.03
σ_α	[Deg.]	7.50	7.50	7.50	7.50
σ_δ	[Deg.]	7.50	7.50	7.50	7.50
$\sigma_{\dot{\theta}}$	[Deg./hr]	1.32	2.05	0.08	2.53
σ_{θ_0}	[Deg.]	0.01	0.01	0.01	0.01
RSS($\sigma_{f'}$)	[m]	2,433.1	393.3	59.1	12.3

The initial position standard deviations are approximately 1.5% of the nominal orbit's semi-major axis. The initial velocity standard deviations cover a 5% maneuver execution error when injecting into circular orbit from a hyperbolic approach. The deviation on each axis is such that 99.7% of initial velocity perturbations fall under the 5% maneuver execution error.[39] The standard deviations for the pole elements are the same for each asteroid. The spin perturbations are 3% of the nominal spin.

Previous work has demonstrated that the prime meridian has difficulty converging to the truth, pointing to observability issues. As such, the perturbations on the prime meridian are small to ensure the lack of observability does not significantly impact the convergence of the other states. If the prime meridian is unobservable, the results will indicate no additional convergence to the truth. This analysis is crucial, as it determines whether the state should be discarded from future analyses.

Perturbing the shape is a highly sensitive process. Without the proper correlations, perturbed truth shapes are likely to be unrealistic to how small bodies naturally form, including peaks, craters, and negative radii. To produce proper perturbed shapes, a maximum likelihood GP batch filter is processed on a hyperbolic approach and distribution used for sampling.[18] However, GP density produced from the batch filter is quite confident, yielding uninteresting shape perturbations for the Monte Carlo studies. Thus, the GP batch filter is used here to obtain the spatial correlations of the shape covariance, and the standard deviations inflated to 5% of the nominal radii. This approach has provided the most consistent initial covariances desired for Monte Carlo simulations.



(a) Lutetia

(b) Eros

(c) Toutatis

(d) Bennu

Figure 6. Satellite Trajectories in the Body-Fixed Frame

The tuning parameters for the GPSF are the same for each asteroid. The vertical scaling σ_c is set to $1m$, and the horizontal scale factor l set to one. The camera noise σ_ρ is set to 0.1 pixels according to standard camera noise levels.[40] The initial data association error is set to match the initial standard deviation of the pole states, 7.5 degrees, with a lower bound of 1 degree. The value for β_0 is set at 0.01, and is found to be appropriate for the simulations presented.[36] The level of rejection c_l is set to 1×10^6, which is sufficient for measurement rejection. The covariance inflation factor $\boldsymbol{P}_{lb}$ is set such that shape standard deviations do not go below 7.5% of their initial values, according to

$$p_{lb} = 0.075 \tag{40}$$

$$\boldsymbol{P}_{lb} = \begin{bmatrix} \mathbf{0}_{10\times10} & \mathbf{0}_{10\times300} \\ \mathbf{0}_{300\times10} & p_{lb}^2 \text{diag}\left(\boldsymbol{P}_f(t_0)\right) \end{bmatrix}. \tag{41}$$

The navigation errors are presented for 500 Monte Carlo runs. The Monte Carlo navigation errors are used to calculate an empirical covariance, measuring the performance of the filter. An analytical covariance study, or Linear Covariance (LinCov) study, is performed by propagating the initial covariance and incorporating measurements along the nominal trajectory.[17] It is often desired to demonstrate that the covariance produced by a LinCov analysis is approximately equivalent to the covariance produced by a Monte Carlo study. This relationship has been validated for a variety of scenarios, including Mars atmospheric entry,[41,42] powered lunar descent,[43] and cislunar ground-based navigation.[44] Furthermore, it is highly useful to provide the covariance validation as an analytic solution can be readily utilized within optimization algorithms for maneuver planning[45] and navigation planning.[21] However, by artificially underweighting the measurements and applying covariance inflation, the GPSF true navigation errors are consistently smaller than the individual filter covariances. Thus, the LinCov results are not expected to match the empirical data calculated from the Monte Carlo simulations.

The inconsistency between the LinCov and Monte Carlo results is further analyzed by calculating the normalized estimation error squared (NEES) for each simulation

$$\epsilon = (\boldsymbol{x} - \hat{\boldsymbol{x}})^T \hat{\boldsymbol{P}}^{-1} (\boldsymbol{x} - \hat{\boldsymbol{x}}), \tag{42}$$

where $\boldsymbol{x}$ is the true state, $\hat{\boldsymbol{x}}$ the filter state estimate, and $\hat{\boldsymbol{P}}$ the filter covariance. For efficient filters, the NEES follows a chi-square distribution with an expected value equal to the number of states. The covariances produced by LinCov and Monte Carlo studies are expected to match in scenarios where the filter is efficient and remains within the region of linearity. Due to the measurement underweighting and covariance inflation terms, the GPSF is not statistically consistent. Thus, the LinCov and Monte Carlo covariances are not expected to match.

While efficient filters are desired, the GPSF requires measurement underweighting and covariance inflation to prevent filter saturation. The tuning parameters are set to provide a pessimistic filter, as denoted by low NEES values inconsistent with a chi-square distribution. Furthermore, small body missions may last far longer than a week,[46] as simulated here in the Monte Carlo runs. More aggressive underweighting techniques may be desired and acceptable given the available time to estimate the state. Approaches for autonomously reducing the amount of underweighting for scenarios with longer observation periods is an area of ongoing research.

The tuning parameters here are selected based on numerous case studies studying state estimate convergence and filter efficiency. While the GPSF tuning parameters here are the same about each asteroid, this does not indicate that these values are optimal for all mission scenarios. Furthermore, these values cannot be treated as optimal for the GPSF about any of the small bodies. Table 5 presents the approximate range of appropriate values for the tuning parameters. Measurement noise is excluded, as it is selected based on sensor specifications. The covariance vertical scale σ_c is also excluded, as it strongly depends on the asteroid's shape and how GP regression errors exhibit.

Table 5. Acceptable GPSF Tuning Parameters

Parameter	Range	Units
l	[.75, 2.5]	–
β_0	[0.01, 0.20]	–
σ_e	[5, 10]	Degrees
$\sigma_{e_{lb}}$	[.5, 2]	Degrees
c_l	[1×10^3, 1×10^6]	Pixels
p_{lb}	[0.05, 0.15]	-

The lower bounds provided generally ensure GPSF state convergence with filter pessimism. The upper bounds are not absolute, as higher values will simply make the GPSF converge more conservatively. Further note that exaggerating any one of the tuning parameters may eliminate the impact and necessity of the other parameters

Thus the GPSF is tuned and tested in various mission scenarios with Monte Carlo studies. The results from the Monte Carlo simulations for Eros are presented, consisting of the navigation errors for the position, velocity, pole, spin, and shape states. The NEES values for each filter run are also provided. The Monte Carlo results for Lutetia, Toutatis, and Bennu are presented in Appendices B-D.

Monte Carlo Results: Eros. Figure 7 depicts the position navigation errors for 500 Monte Carlo simulations. Along with the individual navigation errors from the GPSF, the empirical standard deviations are plotted in red, and the analytical (produced by LinCov) in blue. The plots are zoomed in to better visualize the navigation errors near the end of the simulations. Figure 8 depicts the velocity navigation errors. Note that the analytical and empirical standard deviations do not match, as discussed. The analytical covariance tends to be larger than the empirical covariance, particularly as time progresses.

The position and velocity navigation errors demonstrate the GPSF's ability to estimate the true position and velocity of the spacecraft. The final position navigation errors are less than 5 meters per axis after a week of measurement data, while the initial errors start at 1 kilometer per axis. The final velocity navigation errors are less than 0.37 mm/s per axis, while the initial errors start at 1.33 cm/s. Future work will focus on determining what level of final errors are necessary for various feasibility metrics, however the ability of the GPSF to estimate the position and velocity are clearly demonstrated.

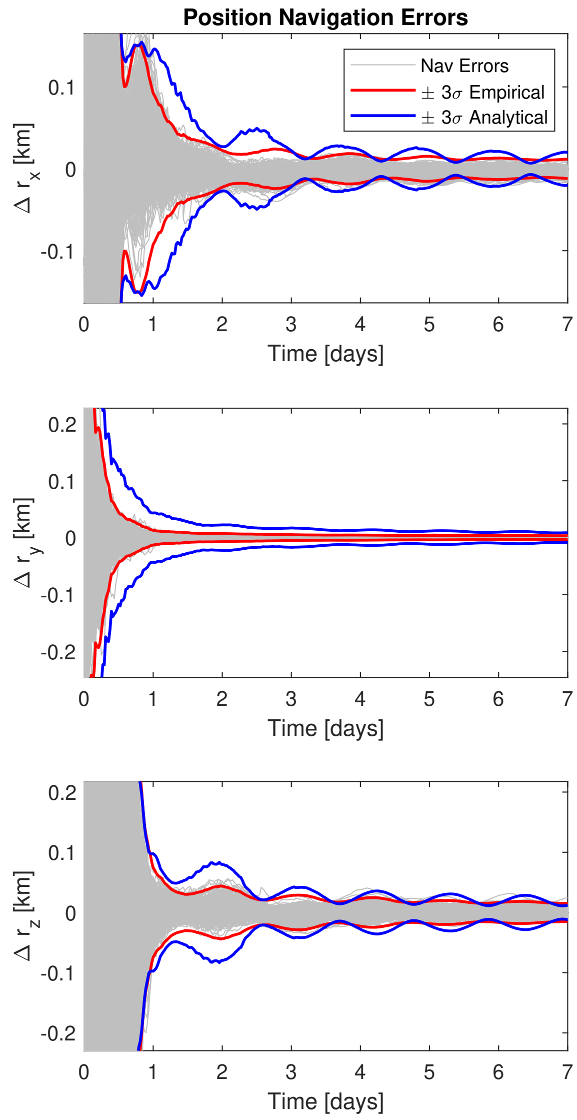


Figure 7. Eros Position Navigation Errors

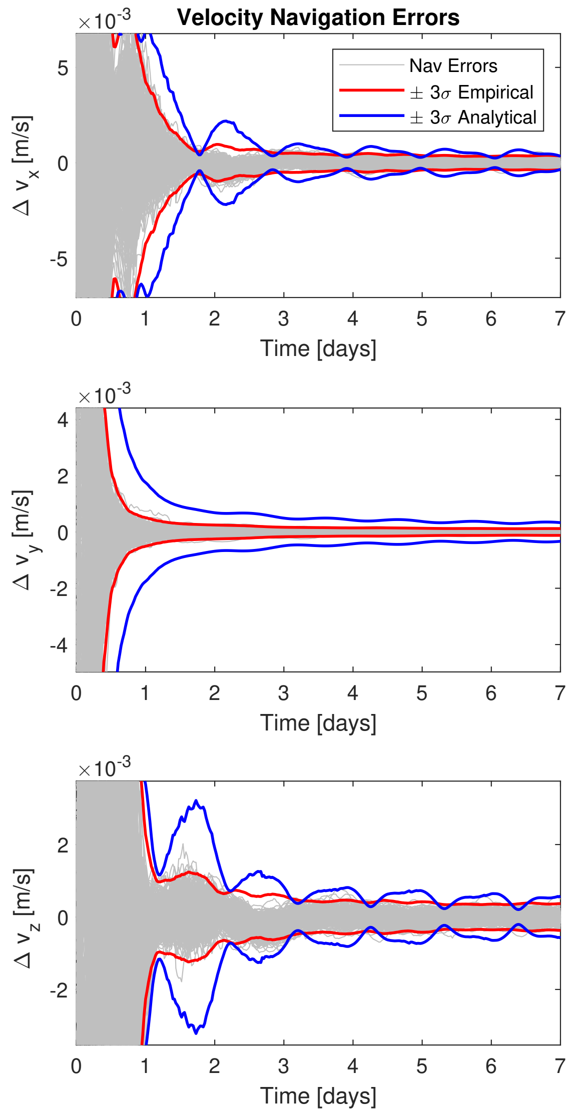


Figure 8. Eros Velocity Navigation Errors

Figure 9 depicts the navigation errors for the right ascension, declination, and spin rate. Consider the significant difference between the analytical and empirical standard deviations. Over a week of measurements, the right ascension errors decrease from a standard deviation of 7.5 degrees to 0.02 degrees, the declination from 7.5 degrees to 0.01 degrees, and the spin rate from 2.05 degrees per hour to 0.00083 degrees per hour (or from 3% to 0.0012%). The navigation errors for the prime meridian are not shown as the errors do not improve with measurement incorporation. Thus, it is concluded that the prime meridian is unobservable under the current silhouette-based measurement system.

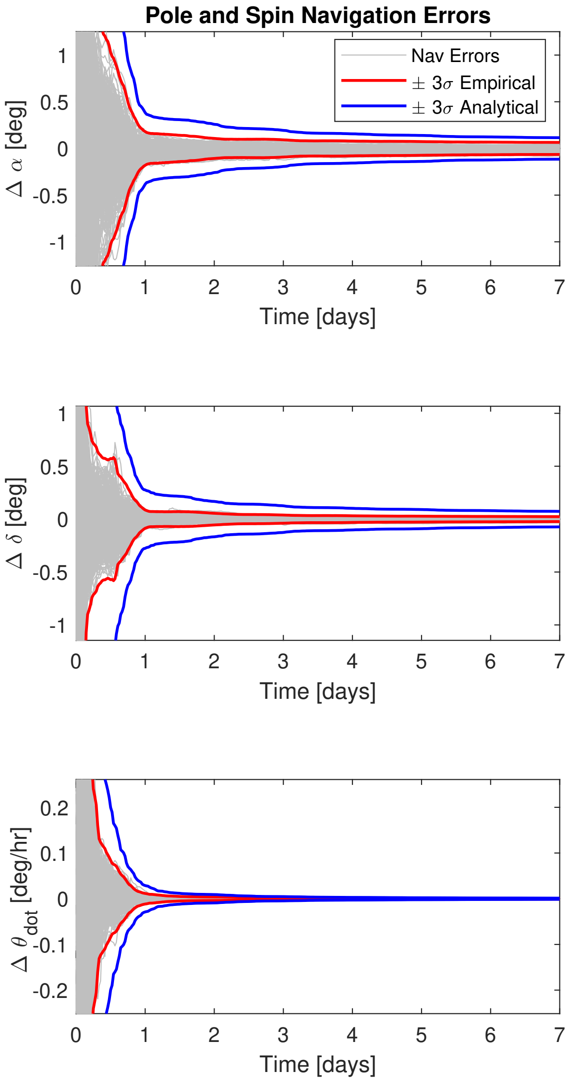


Figure 9. Eros Spin/Pole Navigation Errors

Figure 10 depicts the shape navigation errors for the 500 Monte Carlo simulations. The estimated radii at the 300 basis nodes are condensed by calculating the Root Mean Squared (RMS) of the navigation shape errors. The mean of the RMS shape errors is also calculated and plotted. Note the shape errors quickly converge, then begin to level off. Recall that the shape elements of the filter's covariance are inflated at each measurement step to ensure filter pessimism. For longer mission durations, where it is no longer wise to base the inflation off the initial covariance, the covariance inflation can be changed to ensure further convergence.

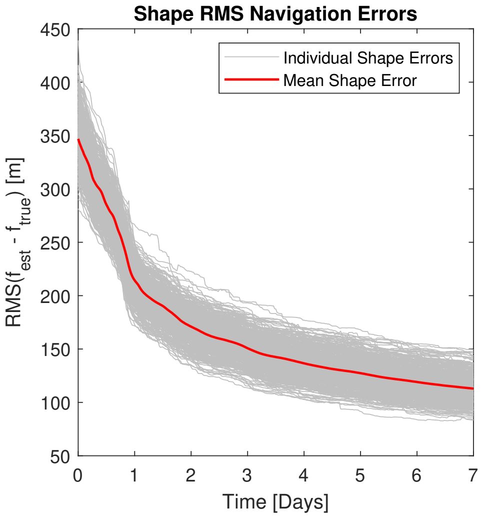


Figure 10. Eros Navigation Shape Errors

Figure 11 depicts the NEES for each Monte Carlo run. As designed, there is a clear bias towards filter pessimism due to the measurement underweighting and covariance inflation. This filter inefficiency explains why the LinCov covariance tends to be larger than the covariance calculated empirically from the Monte Carlo results.

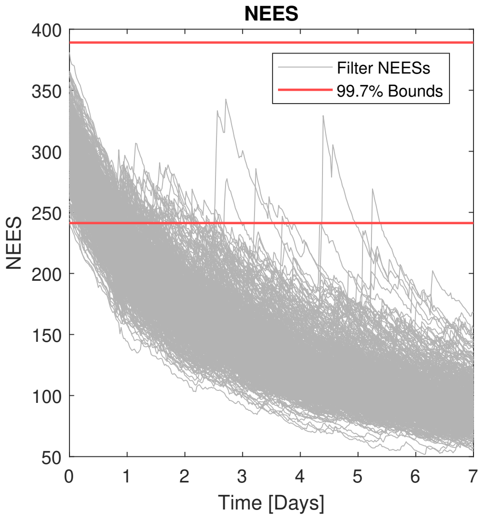


Figure 11. Eros NEES Values

Figure 12 depicts a heat map of Eros, capturing the shape estimation errors at the final time. The heat map is represented as a Delaunay triangulation over the nominal shape of Eros. The nominal shape is consistent with the mean, as the shape is perturbed normally from the initial shape covariance. The initial shape errors are 5%. Therefore, while some regions of Eros have higher shape errors at the final time, the filter provides an accurate shape of Eros.

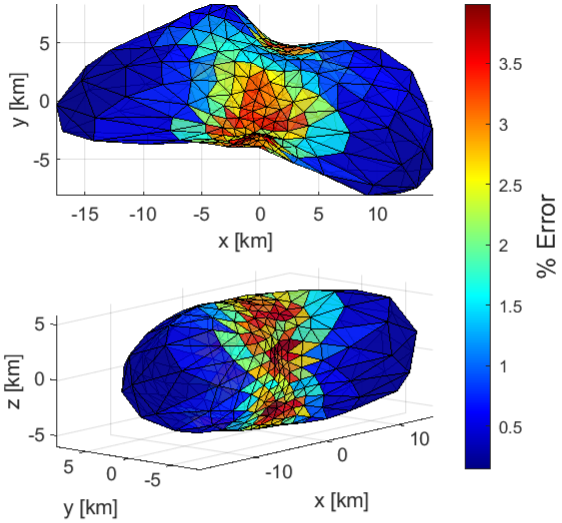


Figure 12. Eros Shape Error Heat Map

The large craters on the top and bottom of Eros have the largest shape errors. This is unsurprising, as craters are difficult to estimate with silhouette-based measurements. However, due to the spatial correlations involved with the GP shape, the craters can be estimated to some degree. It is clear that the extreme ends of Eros (along the x-axis) are easier to estimate. Previous work with the GP batch filter demonstrated that unique geographical features yield improved shape estimation.[18] This pattern holds for Eros' extremities. Large shape errors are also noted along the bottom of Eros (along the negative z-axis). This region unilluminated due to the Sun's relative position with respect to Eros. Furthermore, the concentration of nodes at the center due to the spherical distribution of the basis set leads to higher errors. The shape estimation errors are expected to decrease as more measurements become available.

The results from the Monte Carlo simulations indicate that the GPSF can estimate the state, with the exception of the unobservable prime meridian, for various asteroids (see Appendices B-D). The GPSF is tuned to demonstrate state convergence over a week of measurement incorporation, while also emphasizing filter pessimism. For navigation system design, the GPSF tuning parameters should be tuned on a regular basis to ensure continual state estimation improvement. Otherwise, the measurement underweighting and covariance inflation are expected to cause extreme filter pessimism and limited estimation improvement.

The GPSF about Bennu is particularly sensitive. The results in Appendix D indicate that while the state estimate converges to the truth, the GPSF is frequently overconfident despite using the same tuning parameters and relative initial errors as the other asteroids. Note that a more pessimistic filter can be obtained by increasing the measurement underweighting or the covariance inflation. Estimation about Bennu poses particular difficulties, including the extremely tight position and velocity error tolerances and Bennu's lack of geographical features. It is likely that the GPSF about Bennu, and other similar bodies, will require careful orbit planning to ensure the filter remains viable.

Conclusions. The process of modeling small bodies with a GP and the derivation of the GPSF are provided. The GPSF is demonstrated to estimate the observing spacecraft's relative position and velocity, while simultaneously estimating the central body's pole, spin, and shape with silhouette-based measurements. The various measurement underweighting techniques and the covariance inflation scheme are implemented to ensure the GPSF remains pessimistic by design. A detailed derivation of the measurement partials is provided, marking a significant improvement in run-time performance from numerical techniques. The robustness of the GPSF is exhibited through Monte Carlo studies, where the GPSF is able to estimate hundreds of perturbed states.

The primary goal of future work is to introduce improved realism in test scenarios. The current state is to be augmented to include the small body's mass and spacecraft attitude offsets. Process noise, solar radiation pressure, and more sophisticated gravity models are also to be considered. While the GPSF is desired to operate solely on silhouette-based measurements, additional measurement sources are to be considered for inclusion. Additionally, it will become necessary to study how many basis nodes are necessary to sufficiently capture the true shape of a higher fidelity model.

While the GPSF can estimate the truth, further study is required to determine what level of estimation errors are necessary for mission operations. Potential feasibility metrics include pointing accuracy for Earth communication, pointing accuracy towards the small body for continuing GPSF applications, and station keeping accuracy. These feasibility metrics will highlight areas where the GPSF needs improvement, and potentially provide further evidence that the GPSF is viable for autonomous navigation.

Acknowledgements. This research was carried out at The University of Texas at Austin, funded by the Jet Propulsion Laboratory, California Institute of Technology, under Research Support Agreement 1704630, and at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

References.

[1] R. Gaskell, O. Barnouin-Jha, D. J. Scheeres, A. Konopliv, T. Mukai, S. Abe, J. Saito, M. Ishiguro, T. Kubota, T. Hashimoto, *et al.*, "Characterizing and navigating small bodies with imaging data," *Meteoritics & Planetary Science*, vol. 43, no. 6, pp. 1049–1061, 2008.

[2] O. Barnouin, R. Gaskell, E. Kahn, C. Ernst, M. Daly, E. Bierhaus, C. Johnson, B. Clark, and D. Lauretta, "Assessing the quality of topography from stereophotoclinometry," *Asteroids, Comets, Meteors 2014*, p. 28, 2014.

[3] M. Perry, O. Barnouin, M. Daly, J. Seabrook, E. Palmer, R. Gaskell, K. Craft, J. Roberts, L. Philpott, M. A. Asad, *et al.*, "The global topography of bennu: altimetry, photoclinometry, and processing," in *European Planetary Science Congress*, pp. EPSC2017–952, 2017.

[4] R. Park, A. Vaughan, A. Konopliv, A. Ermakov, N. Mastrodemos, J. Castillo-Rogez, S. Joy, A. Nathues, C. Polanskey, M. Rayman, *et al.*, "High-resolution shape model of ceres from stereophotoclinometry using dawn imaging data," *Icarus*, vol. 319, pp. 812–827, 2019.

[5] R. Gaskell, O. Barnouin, M. Daly, E. Palmer, J. Weirich, C. Ernst, R. Daly, and D. Lauretta, "Stereophotoclinometry on the osiris-rex mission: mathematics and methods," *The Planetary Science Journal*, vol. 4, no. 4, p. 63, 2023.

[6] N. Takeishi, T. Yairi, Y. Tsuda, F. Terui, N. Ogawa, and Y. Mimasu, "Simultaneous estimation of shape and motion of an asteroid for automatic navigation," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2861–2866, IEEE, 2015.

[7] B. Bercovici and J. W. McMahon, "Robust autonomous small-body shape reconstruction and relative navigation using range images," *Journal of Guidance, Control, and Dynamics*, vol. 42, no. 7, pp. 1473–1488, 2019.

[8] K. Kuppa, J. W. McMahon, and A. B. Dietrich, "Initial pole axis and spin direction estimation of asteroids using infrared imagery," *Journal of Guidance, Control, and Dynamics*, vol. 47, no. 6, pp. 1055–1071, 2024.

[9] J. W. McMahon and D. J. Scheeres, "Shape estimation from lightcurves including constraints from orbit determination," in *Advanced Maui Optical and Space Surveillance Technologies Conference*, p. 56, 2016.

[10] D. Baker and J. W. McMahon, "Shape and pole estimation for small-bodies on approach," in *AIAA SCITECH 2022 Forum*, p. 2382, 2022.

[11] E. M. Zucchelli, B. A. Jones, and R. P. Russell, "Pose and shape estimation of a small body via extended target tracking," *American Astronautical Society, Astrodynamics Specialist Conference*, pp. 19–678, 2019.

[12] E. M. Zucchelli, N. Lifset, B. A. Jones, R. P. Russell, and S. Bhaskaran, "Towards limb-based autonomous navigation and mapping of primitive bodies," *American Astronautical Society, Astrodynamics Specialist Conference*, pp. 22–644, 2022.

[13] C. Hollenberg, "Horizon-based autonomous navigation and mapping for small body missions," *Univeristy of Texas Libraries*, 2023.

[14] N. Wahlström and E. Özkan, "Extended target tracking using gaussian processes," *IEEE Transactions on Signal Processing*, vol. 63, no. 16, pp. 4165–4178, 2015.

[15] S. Julier, "The stability of covariance inflation methods for slam," in *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453)*, vol. 3, pp. 2749–2754 vol.3, 2003.

[16] T. Vidal-Calleja, J. Andrade-Cetto, and A. Sanfeliu, "Conditions for suboptimal filter stability in slam," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, vol. 1, pp. 27–32 vol.1, 2004.

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

[18] Q. P. Moon, B. A. Jones, R. P. Russell, C. Hollenberg, D. P. Lubey, and S. Bhaskaran, "Silhouette based gaussian process batch filtering for simultaneous localization and mapping about small bodies," *Denver, CO), Astrodynamics Specialist Conference, AAS-24-196*, vol. 1, 2024.

[19] C. E. Rasmussen and H. Nickisch, "Gaussian processes for machine learning (gpml) toolbox," *The Journal of Machine Learning Research*, vol. 11, pp. 3011–3015, 2010.

[20] S. Panja, A. Chatterjee, and G. Yasmin, "Kernel functions of svm: A comparison and optimal solution," in *Advanced Informatics for Computing Research: Second International Conference, ICAICR 2018, Shimla, India, July 14–15, 2018, Revised Selected Papers, Part I 2*, pp. 88–97, Springer, 2019.

[21] Q. Moon and D. K. Geller, "Batch dilution of precision optimal navigation planning for cislunar environments," *The Journal of the Astronautical Sciences*, vol. 70, no. 6, p. 44, 2023.

[22] B. Clark, A. Sen, X.-D. Zou, D. Dellagiustina, S. Sugita, N. Sakatani, M. Thompson, D. Trang, E. Tatsumi, M. Barucci, *et al.*, "Overview of the search for signs of space weathering on the low-albedo asteroid (101955) bennu," *Icarus*, p. 115563, 2023.

[23] C. W. Hergenrother, M. C. Nolan, R. P. Binzel, E. A. Cloutis, M. A. Barucci, P. Michel, D. J. Scheeres, C. D. d'Aubigny, D. Lazzaro, N. Pinilla-Alonso, *et al.*, "Lightcurve, color and phase function photometry of the osiris-rex target asteroid (101955) bennu," *Icarus*, vol. 226, no. 1, pp. 663–670, 2013.

[24] R. S. Hudson and S. J. Ostro, "Shape and non-principal axis spin state of asteroid 4179 toutatis," *Science*, vol. 270, no. 5233, pp. 84–86, 1995.

[25] R. Gaskell, "Gaskell eros shape model v1. 0," *NASA Planetary Data System*, pp. NEAR–A, 2008.

[26] B. Carry, M. Kaasalainen, W. J. Merline, T. G. Müller, L. Jorda, J. D. Drummond, J. Berthier, L. O'Rourke, J. Ďurech, M. Küppers, *et al.*, "Shape modeling technique koala validated by esa rosetta at (21) lutetia," *Planetary and Space Science*, vol. 66, no. 1, pp. 200–212, 2012.

[27] Blender Online Community, *Blender - a 3D modelling and rendering package.* Blender Foundation, Blender Institute, Amsterdam,

[28] A. McHutchon and C. Rasmussen, "Gaussian process training with input noise," *Advances in neural information processing systems*, vol. 24, 2011.

[29] J. E. Johnson, V. Laparra, and G. Camps-Valls, "Accounting for input noise in gaussian process parameter retrieval," *IEEE Geoscience and Remote Sensing Letters*, vol. 17, no. 3, pp. 391–395, 2019.

[30] A. R. Campbell, "Numerical analysis of complex-step differentiation in spacecraft trajectory optimization problems," *Texas Digital Commons*, 2011.

[31] J. L. Crassidis and J. L. Junkins, *Optimal estimation of dynamic systems.* Chapman and Hall/CRC, 2004.

[32] B. Bell and F. Cathey, "The iterated kalman filter update as a gauss-newton method," *IEEE Transactions on Automatic Control*, vol. 38, no. 2, pp. 294–297, 1993.

[33] W. M. Lear, "Multi-phase navigation program for the space shuttle orbiter," *Johnson Spacecraft Center, Houston, Texas, IN*, 1973.

[34] R. Zanetti, K. J. DeMars, and R. H. Bishop, "Underweighting nonlinear measurements," *Journal of guidance, control, and dynamics*, vol. 33, no. 5, pp. 1670–1675, 2010.

[35] E. Navon and B. Bobrovsky, "An efficient outlier rejection technique for kalman filters," *Signal Processing*, vol. 188, p. 108164, 2021.

[36] M. Roth and F. Gustafsson, "An efficient implementation of the second order extended kalman filter," in *14th International Conference on Information Fusion*, pp. 1–6, IEEE, 2011.

[37] NASA, "Asteroid fact sheet," September 27, 2019.

[38] A. F. Cheng, A. Santo, K. Heeres, J. Landshof, R. Farquhar, R. Gold, and S. Lee, "Near-earth asteroid rendezvous: Mission overview," *Journal of Geophysical Research: Planets*, vol. 102, no. E10, pp. 23695–23708, 1997.

[39] Q. Moon and D. Geller, "Regression techniques for the statistics of correction maneuvers magnitudes," in *Breckenridge, CO), AAS Guidance and Control Conference, AAS-24-026*, vol. 1, 2024.

[40] D. T. Renshaw and J. A. Christian, "Subpixel localization of isolated edges and streaks in digital images," *Journal of Imaging*, vol. 6, no. 5, p. 33, 2020.

[41] 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, pp. 854–864, 2019.

[42] 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," in *AIAA SciTech 2022 Forum*, p. 0745, 2022.

[43] 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, pp. 1231–1248, 2009.

[44] Q. Moon, D. H. Kuettel, D. K. Geller, S. Welsh, S. Stewart, and T. Crain, "Validation of linear covariance analysis for nova-c cislunar trajectory design," in *Proceedings of the 44th Annual American Astronautical Society Guidance, Navigation, and Control Conference, 2022*, pp. 115–133, Springer, 2024.

[45] D. Geller, D. Woffinden, and S. Bieniawski, "Sensitivity of optimal midcourse correction scheduling for robust cislunar trajectory design," in *45th Rocky Mountain AAS GN&C Conference*, no. AAS 23-061, 2023.

[46] D. Lauretta, S. Balram-Knutson, E. Beshore, W. Boynton, C. Drouet d'Aubigny, D. DellaGiustina, H. Enos, D. Golish, C. Hergenrother, E. Howell, *et al.*, "Osirisrex: sample return from asteroid (101955) bennu," *Space Science Reviews*, vol. 212, pp. 925–984, 2017.

Appendix A: Analytic Partials. The pinhole camera with data association is a function of state elements $\boldsymbol{x}_i$, the jth measured radius f_j, node $\boldsymbol{e}_j$, and optimization angle γ_j:

$$\boldsymbol{h}\left[\boldsymbol{x}_i, \boldsymbol{e}_j^B(\boldsymbol{x}_i, \gamma_j(\boldsymbol{x}_i)), f_j(\boldsymbol{x}_i, \boldsymbol{e}_j^B(\boldsymbol{x}_i, \gamma_j(\boldsymbol{x}_i)))\right]. \tag{43}$$

The pinhole camera can be more compactly expressed in terms of Eq. 7, and the partials expressed as:

$$\frac{d\boldsymbol{h}}{d\boldsymbol{x}_i} = \boldsymbol{M}\frac{\partial \boldsymbol{\rho}}{\partial \boldsymbol{s}}\frac{d\boldsymbol{s}}{d\boldsymbol{x}_i}, \tag{44}$$

where

$$\frac{d\boldsymbol{h}}{d\boldsymbol{x}_i} = \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}_i} + \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{e}}\left(\frac{\partial \boldsymbol{e}}{\partial \boldsymbol{x}_i} + \frac{\partial \boldsymbol{e}}{\partial \gamma}\frac{d\gamma}{d\boldsymbol{x}_i}\right) + \frac{\partial \boldsymbol{h}}{\partial f}\left(\frac{\partial f}{\partial \boldsymbol{x}_i} + \frac{\partial f}{\partial \boldsymbol{e}}\left(\frac{\partial \boldsymbol{e}}{\partial \boldsymbol{x}_i} + \frac{\partial \boldsymbol{e}}{\partial \gamma}\frac{d\gamma}{d\boldsymbol{x}_i}\right)\right). \tag{45}$$

The j subscript is dropped for compactness. All nodes, whether basis or measured, are in the body fixed frame unless otherwise noted.

Some partials within the chain rule are required for each state partial. The partials of $\boldsymbol{s}$ with respect to the input radius f and node $\boldsymbol{e}$ are

$$\tfrac{\partial \boldsymbol{s}}{\partial f} = \boldsymbol{T}_C^I \boldsymbol{T}_I^A \boldsymbol{e}, \tag{46}$$

$$\tfrac{\partial \boldsymbol{s}}{\partial \boldsymbol{e}} = \boldsymbol{T}_C^I \boldsymbol{T}_I^A f. \tag{47}$$

The partial of the measured radius f with respect to the input node $\boldsymbol{e}$ is solved according to the GP regression equation in Eq. 2

$$\tfrac{\partial f}{\partial \boldsymbol{e}} = \tfrac{\partial \boldsymbol{C}(\boldsymbol{e}, \boldsymbol{E}')}{\partial \boldsymbol{e}} \boldsymbol{C}^{-1}(\boldsymbol{E}', \boldsymbol{E}')\boldsymbol{f}', \tag{48}$$

$$\boldsymbol{\mathcal{X}}_{nn} = \tfrac{\boldsymbol{C}(\boldsymbol{e}, \boldsymbol{E}'_n)}{l^2\sqrt{1-(\kappa \boldsymbol{e}^T \boldsymbol{E}'_n)^2}}, \tag{49}$$

$$\tfrac{\partial f}{\partial \boldsymbol{e}} = \left(\boldsymbol{E}' \boldsymbol{\mathcal{X}} \boldsymbol{C}^{-1}(\boldsymbol{E}', \boldsymbol{E}')\boldsymbol{f}'\right)^T, \tag{50}$$

where the subscript n denotes the nth node of the basis set. Note that if $\kappa = 1$, Eq. 49 has a singularity when $\boldsymbol{e}^T \boldsymbol{E}'_n = 1$, corresponding to the kernel function in Eq. 3. The partial of the measured node $\boldsymbol{e}$ with respect to the optimization angle γ is

$$\tfrac{\partial \boldsymbol{e}}{\partial \gamma} = \boldsymbol{T}_A^I \boldsymbol{\Gamma}_{\boldsymbol{\zeta}} \left[\boldsymbol{i}_{\boldsymbol{\zeta}} \times\right] \boldsymbol{i}_{\boldsymbol{r}}. \tag{51}$$

The partials of $\boldsymbol{s}$ with respect to each state element are expressed as:

$$\tfrac{\partial \boldsymbol{s}}{\partial \boldsymbol{r}} = -\boldsymbol{T}_C^I, \tag{52}$$

$$\tfrac{\partial \boldsymbol{s}}{\partial \boldsymbol{v}} = \boldsymbol{0}, \tag{53}$$

$$\tfrac{\partial \boldsymbol{s}}{\partial \alpha} = \boldsymbol{T}_C^I \tfrac{\partial \boldsymbol{T}_I^A}{\partial \alpha} f\boldsymbol{e}, \tag{54}$$

$$\tfrac{\partial \boldsymbol{s}}{\partial \delta} = \boldsymbol{T}_C^I \tfrac{\partial \boldsymbol{T}_I^A}{\partial \delta} f\boldsymbol{e}, \tag{55}$$

$$\tfrac{\partial \boldsymbol{s}}{\partial \dot{\theta}} = \boldsymbol{T}_C^I \tfrac{\partial \boldsymbol{T}_I^A}{\partial \dot{\theta}} f\boldsymbol{e}, \tag{56}$$

$$\tfrac{\partial \boldsymbol{s}}{\partial \theta_0} = \boldsymbol{T}_C^I \tfrac{\partial \boldsymbol{T}_I^A}{\partial \theta_0} f\boldsymbol{e}, \tag{57}$$

$$\tfrac{\partial \boldsymbol{s}}{\partial \boldsymbol{f}'} = \boldsymbol{0}, \tag{58}$$

where the partials of $\boldsymbol{T}_I^A$ with respect to the pole elements follow simply from differentiation of the sine and cosine terms. Likewise, the partials of $\frac{\partial \boldsymbol{e}}{\partial \boldsymbol{x}_i}$ with respect to the pole states are simply

$$\tfrac{\partial \boldsymbol{e}}{\partial \alpha} = \tfrac{\partial \boldsymbol{T}_A^I}{\partial \alpha} \boldsymbol{\Gamma}_{\boldsymbol{\zeta}} \boldsymbol{i}_{\boldsymbol{r}}, \tag{59}$$

$$\tfrac{\partial \boldsymbol{e}}{\partial \delta} = \tfrac{\partial \boldsymbol{T}_A^I}{\partial \delta} \boldsymbol{\Gamma}_{\boldsymbol{\zeta}} \boldsymbol{i}_{\boldsymbol{r}}, \tag{60}$$

$$\tfrac{\partial \boldsymbol{e}}{\partial \dot{\theta}} = \tfrac{\partial \boldsymbol{T}_A^I}{\partial \dot{\theta}} \boldsymbol{\Gamma}_{\boldsymbol{\zeta}} \boldsymbol{i}_{\boldsymbol{r}}, \tag{61}$$

$$\tfrac{\partial \boldsymbol{e}}{\partial \theta_0} = \tfrac{\partial \boldsymbol{T}_A^I}{\partial \theta_0} \boldsymbol{\Gamma}_{\boldsymbol{\zeta}} \boldsymbol{i}_{\boldsymbol{r}}. \tag{62}$$

The partials $\frac{\partial \boldsymbol{e}}{\partial \boldsymbol{v}}$ and $\frac{\partial \boldsymbol{e}}{\partial \boldsymbol{f}'}$ are zero. The partial of $\boldsymbol{e}$ with respect to $\boldsymbol{r}$ is much more involved, as the rotation matrix $\boldsymbol{\Gamma}_{\boldsymbol{\zeta}}$ is dependent on $\boldsymbol{r}$. The partial $\frac{\partial \boldsymbol{e}}{\partial \boldsymbol{r}}$ is expressed as

$$\frac{\partial \boldsymbol{e}}{\partial \boldsymbol{r}} = \boldsymbol{T}_A^I \frac{\partial \boldsymbol{e}^I}{\partial \boldsymbol{r}}, \tag{63}$$

where

$$\frac{\partial \boldsymbol{e}^I}{\partial \boldsymbol{r}} = \frac{1}{r}\boldsymbol{\Gamma}_{\boldsymbol{\zeta}}\Big[\boldsymbol{I} - \boldsymbol{i}_{\boldsymbol{r}}\boldsymbol{i}_{\boldsymbol{r}}{}^T\Big] + \left[\frac{\partial \boldsymbol{\Gamma}_{\boldsymbol{\zeta}}}{\partial r_1}\boldsymbol{i}_{\boldsymbol{r}}, \frac{\partial \boldsymbol{\Gamma}_{\boldsymbol{\zeta}}}{\partial r_2}\boldsymbol{i}_{\boldsymbol{r}}, \frac{\partial \boldsymbol{\Gamma}_{\boldsymbol{\zeta}}}{\partial r_3}\boldsymbol{i}_{\boldsymbol{r}}\right]. \tag{64}$$

The partials of the rotation matrix $\boldsymbol{\Gamma}_{\boldsymbol{\zeta}}$ with respect to each element of the position vector are

$$\frac{\partial \boldsymbol{\Gamma}_{\boldsymbol{\zeta}}}{\partial r_i} = \sin(\gamma)\frac{\partial\left[\boldsymbol{i}_{\boldsymbol{\zeta}}\times\right]}{\partial r_i} + (1-\cos(\gamma))\left(\frac{\partial\left[\boldsymbol{i}_{\boldsymbol{\zeta}}\times\right]}{\partial r_i}\left[\boldsymbol{i}_{\boldsymbol{\zeta}}\times\right] + \left[\boldsymbol{i}_{\boldsymbol{\zeta}}\times\right]\frac{\partial\left[\boldsymbol{i}_{\boldsymbol{\zeta}}\times\right]}{\partial r_i}\right), \tag{65}$$

where

$$\tfrac{\partial[\boldsymbol{i}_{\boldsymbol{\zeta}}\times]}{\partial r_i} = \left[\tfrac{\partial \boldsymbol{i}_{\boldsymbol{\zeta}}}{\partial r_i}\right]\times, \tag{66}$$

$$\tfrac{\partial \boldsymbol{i}_{\boldsymbol{\zeta}}}{\partial r_i} = \left(-\tfrac{1}{\zeta}\left(\boldsymbol{I} - \boldsymbol{i}_{\boldsymbol{\zeta}}\boldsymbol{i}_{\boldsymbol{\zeta}}{}^T\right)\left[\boldsymbol{i}_{\boldsymbol{a}}\times\right]\right)_{:,i}. \tag{67}$$

The partial $\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}_i}$ is zero for all states except for the shape

$$\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{f}'} = \boldsymbol{C}(\boldsymbol{E}, \boldsymbol{E}')\boldsymbol{C}^{-1}(\boldsymbol{E}', \boldsymbol{E}'). \tag{68}$$

Root-Solved Partials

There is no explicit term for γ, as it is obtained through optimizing the objective function in the data association routine. Instead, $\frac{d\gamma}{d\boldsymbol{x}_i}$ is found through the partials of a root solved process:

$$\frac{d\gamma}{d\boldsymbol{x}_i} = -\left(\frac{dq}{d\gamma}\right)^{-1}\frac{dq}{d\boldsymbol{x}_i}, \tag{69}$$

where the expression q is defined by the first order necessary conditions of the objective function:

$$\tfrac{\partial J}{\partial \gamma} = \tfrac{(r - f\cos(\gamma)\left[\frac{\partial f}{\partial \gamma} r\sin(\gamma) + fr\cos(\gamma) - f^2\right]}{\sqrt{f^2 + r^2 - 2fr\cos(\gamma)}^3}, \tag{70}$$

$$q = \tfrac{\partial f}{\partial \gamma} r\sin(\gamma) + fr\cos(\gamma) - f^2. \tag{71}$$

Note that in order for the first order necessary condition to be satisfied, the numerator must equal zero. Furthermore, the term $r - f\cos(\gamma)$ in every mission design scenario will be nonzero, as the orbit is greater than the asteroid's radius.

The partials needed for q with respect to γ are decomposed with the chain rule:

$$\frac{dq}{d\gamma} = \frac{\partial q}{\partial \gamma} + \frac{\partial q}{\partial f}\frac{\partial f}{\partial \boldsymbol{e}}\frac{\partial \boldsymbol{e}}{\partial \gamma} + \frac{\partial q}{\partial f_\gamma}\frac{df_\gamma}{d\gamma}, \tag{72}$$

where

$$f_\gamma = \frac{\partial f}{\partial \boldsymbol{e}}\frac{\partial \boldsymbol{e}}{\partial \gamma}. \tag{73}$$

The partials required for all state elements are

$$\tfrac{\partial q}{\partial \gamma} = f_\gamma r\cos(\gamma) - fr\sin(\gamma), \tag{74}$$

$$\tfrac{\partial q}{\partial f} = r\cos(\gamma) - 2f, \tag{75}$$

$$\tfrac{\partial q}{\partial f_\gamma} = r\sin(\gamma). \tag{76}$$

Solving for $\frac{df_\gamma}{d\gamma}$ requires additional chain rule implementation, but for compactness is expressed as

$$\frac{df_\gamma}{d\gamma} = \frac{\partial \boldsymbol{e}}{\partial \gamma}^T\frac{\partial^2 f}{\partial \boldsymbol{e}^2}\frac{\partial \boldsymbol{e}}{\partial \gamma} + \frac{\partial f}{\partial \boldsymbol{e}}\frac{\partial^2 \boldsymbol{e}}{\partial \gamma^2}, \tag{77}$$

where

$$\tfrac{\partial^2 \boldsymbol{e}}{\partial \gamma^2} = \boldsymbol{T}_A^I\boldsymbol{\Gamma}_{\boldsymbol{\zeta}}\left[\boldsymbol{i}_{\boldsymbol{\zeta}}\times\right]^2\boldsymbol{i}_{\boldsymbol{r}}, \tag{78}$$

$$\boldsymbol{\mathcal{Y}}_{nn} = \tfrac{1}{l^2}\left[\tfrac{1}{l^2}\tfrac{\boldsymbol{C}(\boldsymbol{e},\boldsymbol{E}'_n)}{1-(\kappa\boldsymbol{e}^T\boldsymbol{E}'_n)^2} + \tfrac{\boldsymbol{C}(\boldsymbol{e},\boldsymbol{E}'_n)(\kappa\boldsymbol{e}^T\boldsymbol{E}'_n)}{\sqrt{1-(\kappa\boldsymbol{e}^T\boldsymbol{E}'_n)^2}^3}\right], \tag{79}$$

$$\tfrac{\partial^2 f}{\partial \boldsymbol{e}^2} = \boldsymbol{E}'\mathrm{diag}\left[\boldsymbol{\mathcal{Y}}\boldsymbol{C}^{-1}(\boldsymbol{E}',\boldsymbol{E}')\boldsymbol{f}'\right]\boldsymbol{E}'^T. \tag{80}$$

Note again that singularity issues arise when $\kappa = 1$ for Eq. 79.

The partials of q with respect to the state elements are decomposed with the chain rule according to

$$\frac{dq}{d\boldsymbol{x}_i} = \frac{\partial q}{\partial \boldsymbol{x}_i} + \frac{\partial q}{\partial f}\left[\frac{\partial f}{\partial \boldsymbol{e}}\frac{\partial \boldsymbol{e}}{\partial \boldsymbol{x}_i} + \frac{\partial f}{\partial \boldsymbol{x}_i}\right] + \frac{\partial q}{\partial f_\gamma}\frac{df_\gamma}{d\boldsymbol{x}_i}. \tag{81}$$

Fortunately, $\frac{\partial q}{\partial \boldsymbol{x}_i}$ is zero for all state elements except position

$$\frac{\partial q}{\partial \boldsymbol{r}} = \boldsymbol{i_r}^T(f_\gamma\sin(\gamma) + f\cos(\gamma)). \tag{82}$$

Lastly, solving for $\frac{df_\gamma}{d\boldsymbol{x}_i}$, which again requires a chain rule, but is simplified as

$$\frac{df_\gamma}{d\boldsymbol{x}_i} = \frac{\partial \boldsymbol{e}}{\partial \gamma}^T\left[\frac{\partial^2 f}{\partial \boldsymbol{e}^2}\frac{\partial \boldsymbol{e}}{\partial \boldsymbol{x}_i} + \frac{\partial^2 f}{\partial \boldsymbol{x}_i\partial \boldsymbol{e}}\right] + \frac{\partial f}{\partial \boldsymbol{e}}\frac{\partial^2 \boldsymbol{e}}{\partial \boldsymbol{x}_i\partial \gamma}. \tag{83}$$

The new second derivative term $\frac{\partial^2 f}{\partial \boldsymbol{x}_i\partial \boldsymbol{e}}$ is easily solved with respect to the shape using Eq. 50, and is zero for all other state elements.

The second derivative $\frac{\partial^2 \boldsymbol{e}}{\partial \boldsymbol{x}_i\partial \gamma}$ with respect to the pole elements only depends on the rotation matrix and are solved easily for using Eq. 51. The partials with respect to position $\frac{\partial \boldsymbol{e}}{\partial \gamma\partial r_i}$ are again not trivial, but are solved with the product rule using terms already derived

$$\frac{\partial \boldsymbol{e}}{\partial \gamma\partial r_i} = \boldsymbol{T}_A^I\left[\frac{\partial \boldsymbol{\Gamma}_{\boldsymbol{\zeta}}}{\partial r_i}\left[\boldsymbol{i}_{\boldsymbol{\zeta}}\times\right]\boldsymbol{i}_{\boldsymbol{r}} + \boldsymbol{\Gamma}_{\boldsymbol{\zeta}}\frac{\partial\left[\boldsymbol{i}_{\boldsymbol{\zeta}}\times\right]}{\partial r_i}\boldsymbol{i}_{\boldsymbol{r}} + \boldsymbol{\Gamma}_{\boldsymbol{\zeta}}\left[\boldsymbol{i}_{\boldsymbol{\zeta}}\times\right]\frac{\partial \boldsymbol{i}_{\boldsymbol{r}}}{\partial r_i}\right]. \tag{84}$$

All intermediate partial required are verified against a complex step derivative.[30] Furthermore, the complete partials are checked against a multi-step numerical measurement partial, which include the data association step.

Appendix B: Monte Carlo Results: Lutetia. Figure 13 depicts the position navigation errors for 500 Monte Carlo simulations about Lutetia. The plots are again zoomed in to better visualize the navigation errors near the end of the simulations. Figure 14 depicts the velocity navigation errors. The analytical covariance tends to be larger than the empirical covariance, particularly as time progresses.

The position and velocity navigation errors demonstrate notable skewness initially, pointing to initial position and velocity errors potentially beyond the region of linearization. The position estimation errors decrease from 6.5 kilometers per axis to less than 33 meters per axis, while the velocity errors decrease from 8.82 cm/s per axis to 0.61 mm/s per axis.

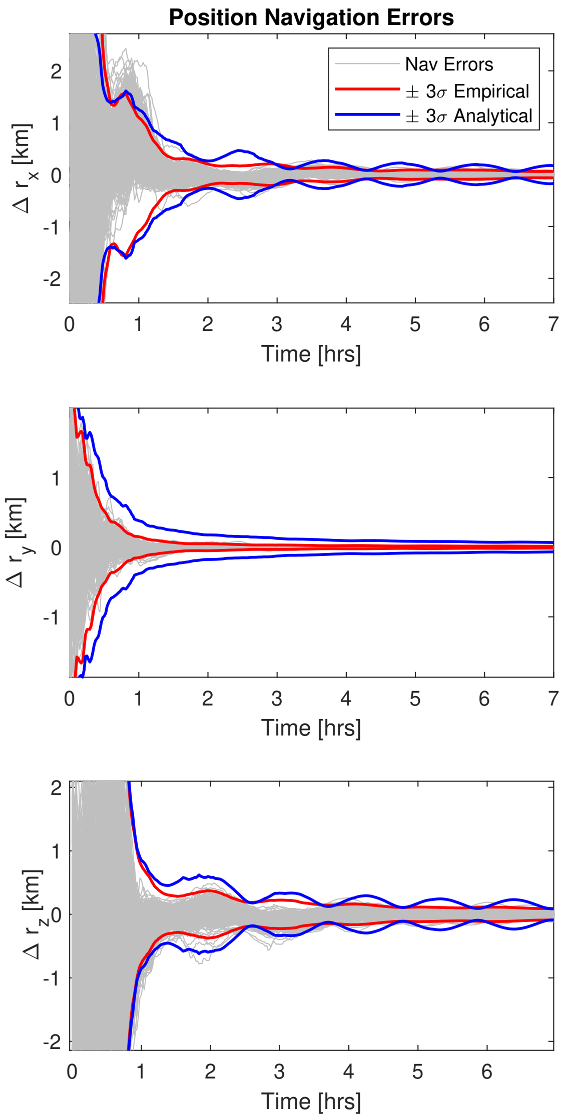


Figure 13. Lutetia Position Navigation Errors

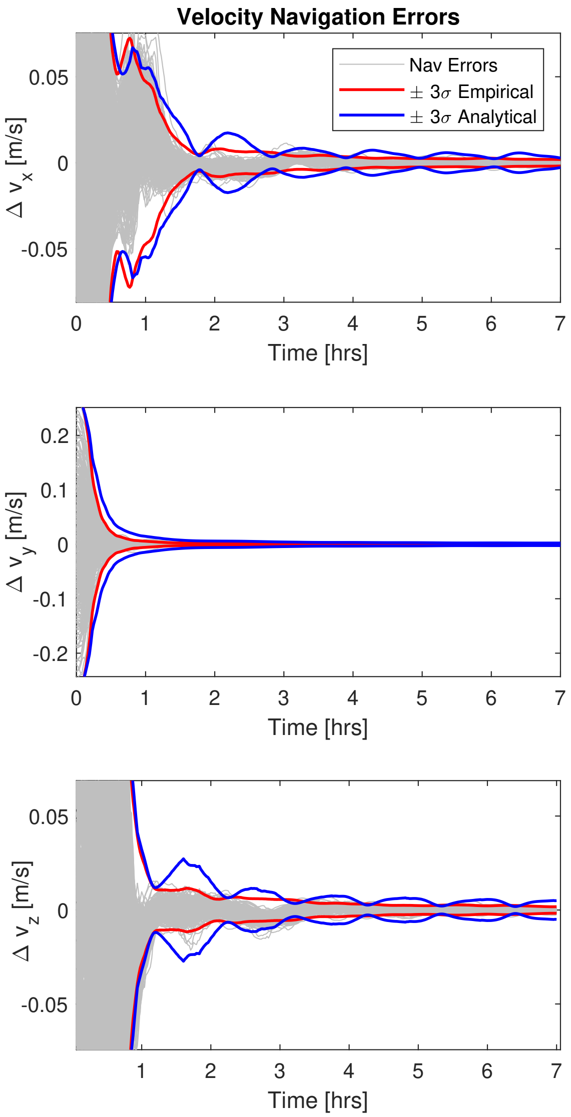


Figure 14. Lutetia Velocity Navigation Errors

Figure 15 depicts the navigation errors for the right ascension, declination, and spin rate. A few cases notably lie outsize the empirical 3σ bounds for the right ascension, but still converge to the truth. Over a week of measurements, the pole errors decrease from a standard deviation of 7.5 degrees to less than 0.02 degrees. The spin rate decreases from 1.32 degrees per hour to 0.001 degrees per hour (or from 3% to 0.0023%).

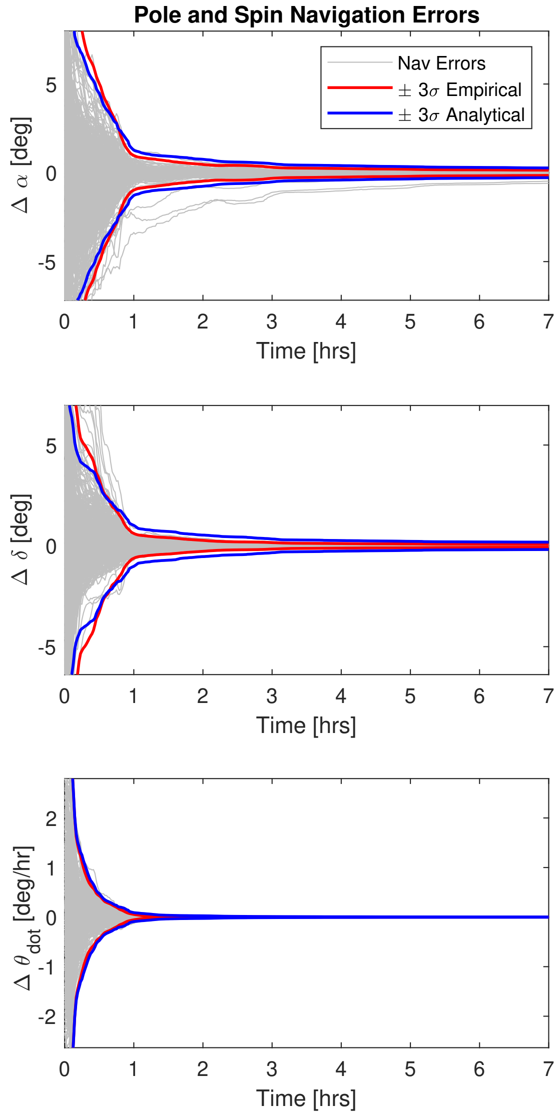


Figure 15. Lutetia Spin/Pole Navigation Errors

Figure 16 depicts the shape navigation errors for the 500 Monte Carlo simulations. While the shape errors quickly converge, as they did for Eros, the rate of convergence is less dramatic initially. Recall Lutetia spins more slowly than Eros, and has fewer distinctive shape features.

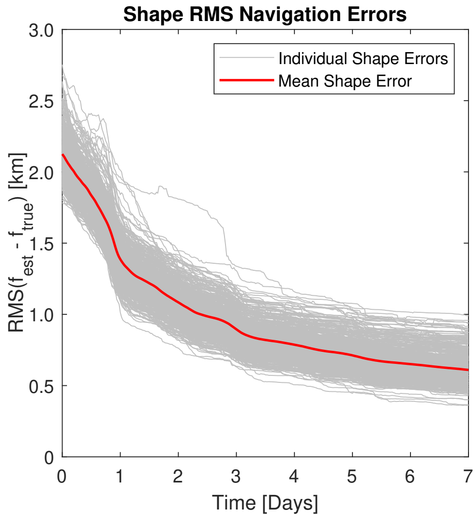


Figure 16. Lutetia Navigation Shape Errors

Figure 17 depicts the NEES for each Monte Carlo run. As designed, there is a clear bias towards filter pessimism due to the measurement underweighting and covariance inflation. Notably, some runs are over-confident with their estimation near the beginning of the simulation. These cases are the same as those with outlying right ascension estimation, demonstrating that filter overconfidence can lead to poorer state estimation. All cases eventually achieve the desired pessimism.

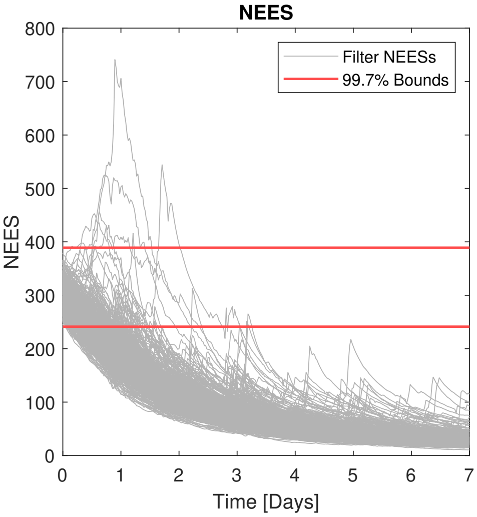


Figure 17. Lutetia NEES Values

Figure 18 depicts the heat map of Lutetia, capturing the shape errors at the final time. As with Eros, craters experience a higher level of estimation errors, but still improve from the original 5% errors. The largest errors are on the bottom (negative z-axis direction), which combines poor illumination and crater geometry. The concentration of nodes due to the spherical distribution does not cause higher regions of error for Lutetia. Thus while the spherical distribution of nodes may be poorly suited for elongated bodies like Eros, they are well suited for bodies like Lutetia that are near-spherical.

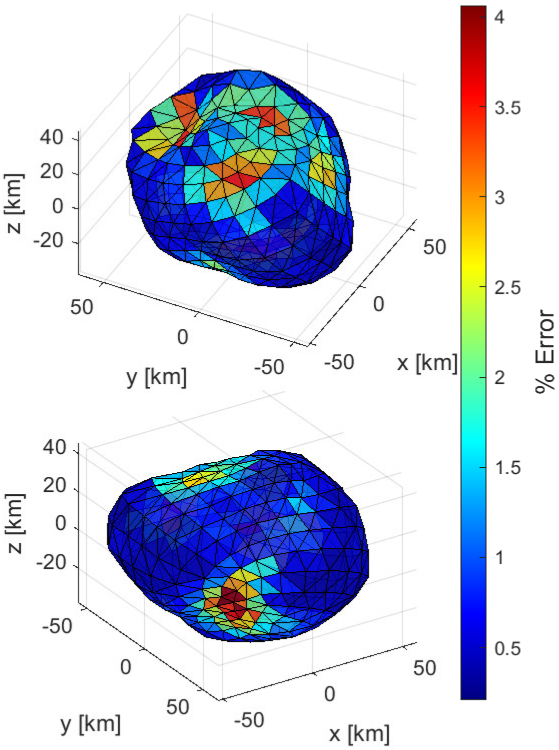


Figure 18. Lutetia Shape Error Heat Map

Appendix C: Monte Carlo Results: Toutatis. Figure 19 depicts the position navigation errors for 500 Monte Carlo simulations about Toutatis. The plots are again zoomed in to better visualize the navigation errors near the end of the simulations. Figure 20 depicts the velocity navigation errors. The analytical covariance and empirical covariances here are similar over time, but remain inconsistent with each other due to the filter pessimism.

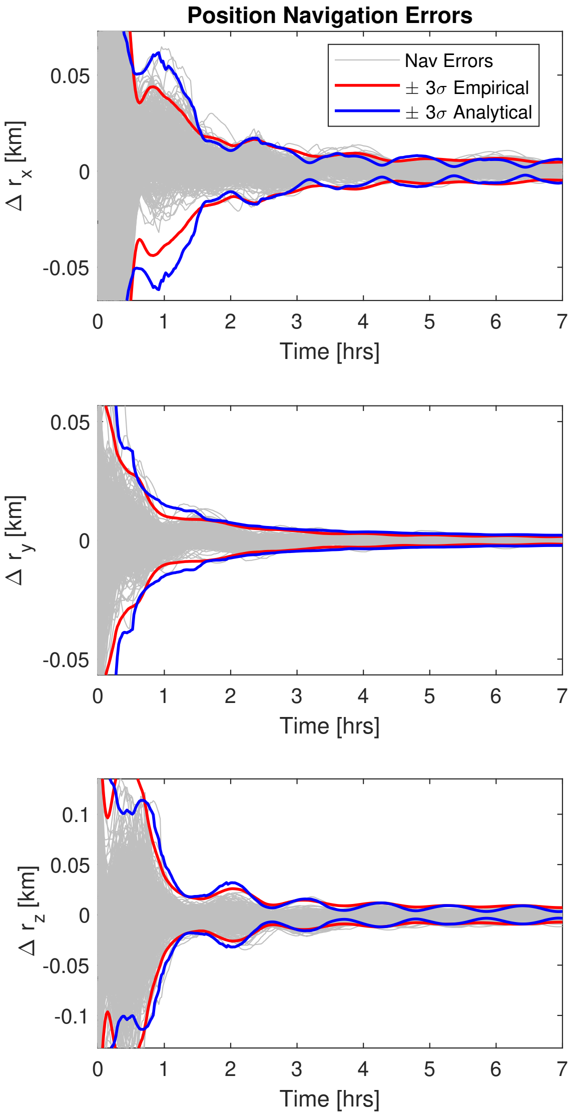


Figure 19. Toutatis Position Navigation Errors

The position and velocity navigation errors demonstrate the GPSF's ability to estimate the truth about smaller bodies. The position estimation errors decrease from 150 meters per axis to less than 1.5 meters per axis, while the velocity errors decrease from 2.00 mm/s per axis to 0.05 mm/s per axis. The results about Toutatis demonstrate that smaller estimation errors in position and velocity are attainable by the filter. Thus, similar estimation performance can be obtained about larger bodies such as Lutetia and Eros.

Figure 21 depicts the navigation errors for the right ascension, declination, and spin rate. A few cases notably lie outsize the empirical 3σ bounds for the right ascension, but still converge to the truth. Over a week of measurements, the pole errors decrease from a standard deviation of 7.5 degrees to less than 0.8 degrees. The spin rate decreases from 0.08 degrees per hour to 0.001 degrees per hour (or from 3% to 0.0375%). The pole and spin estimation performance is far worse for Toutatis compared to the estimation errors about Lutetia and Eros.

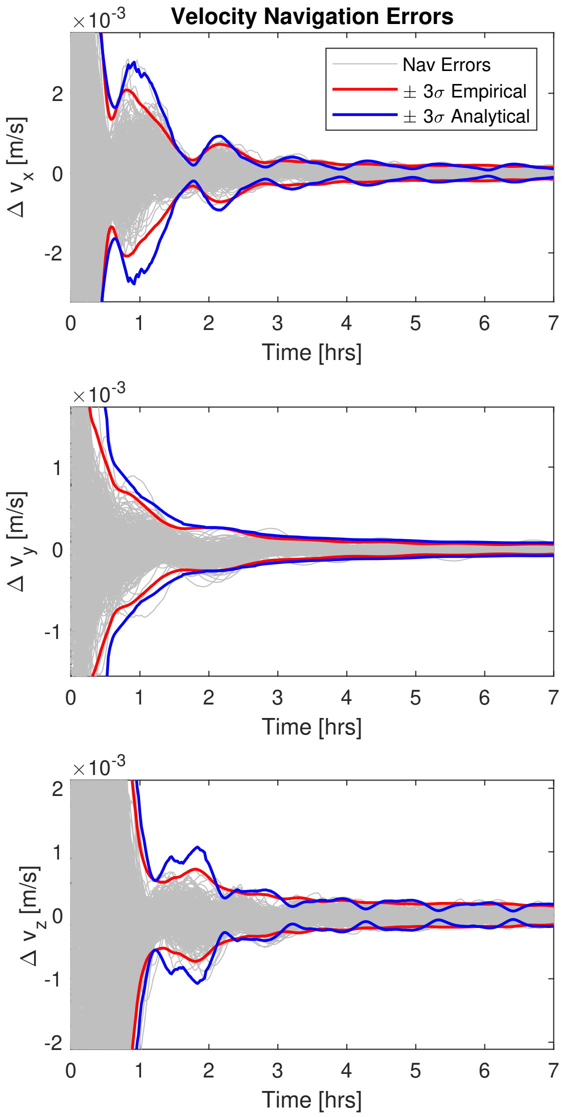


Figure 20. Toutatis Velocity Navigation Errors

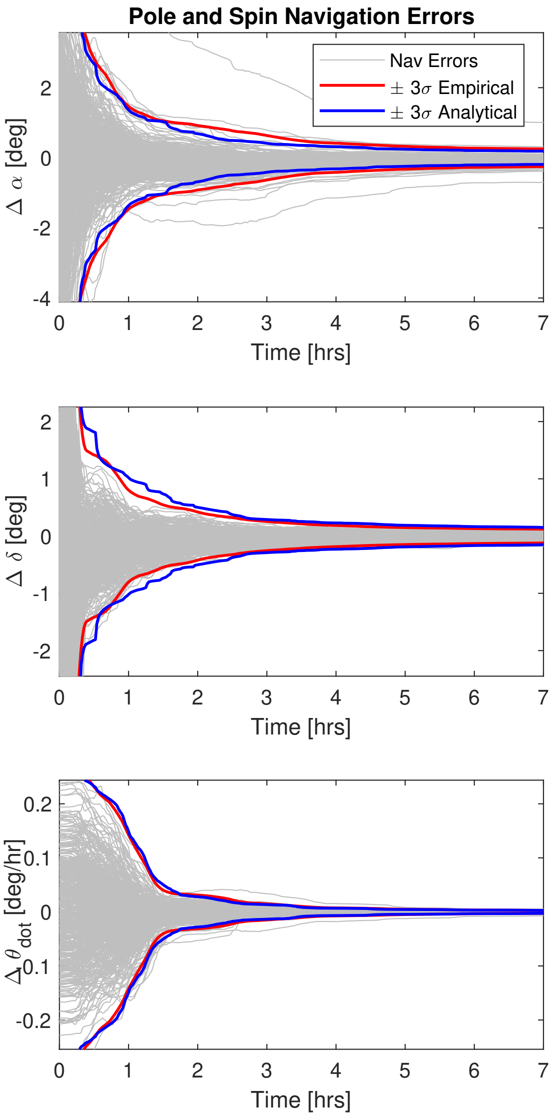


Figure 21. Toutatis Spin/Pole Navigation Errors

Figure 22 depicts the shape navigation errors for the 500 Monte Carlo simulations. While shape estimation is clearly demonstrated, the rate and level of convergence is significantly worse for Toutatis than for Lutetia and Eros. The slow spin of Toutatis leads to worse estimation performance, as the spacecraft observes less of the shape.

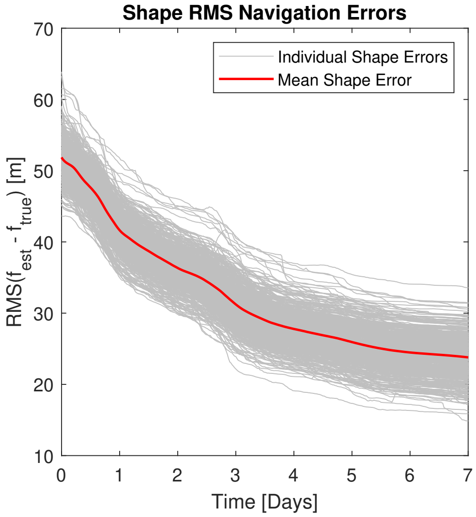


Figure 22. Toutatis Navigation Shape Errors

Figure 23 depicts the NEES for each Monte Carlo run. As designed, there is a clear bias towards filter pessimism due to the measurement underweighting and covariance inflation.

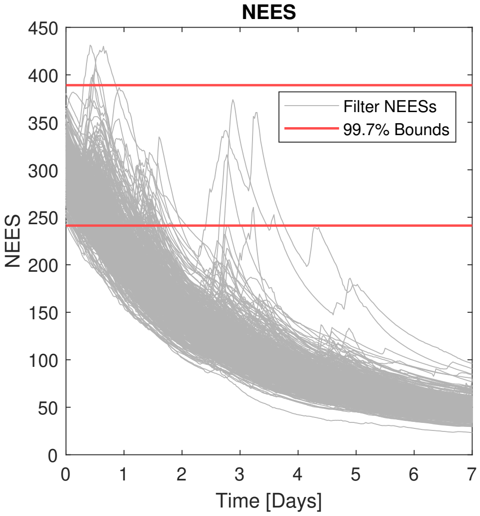


Figure 23. Toutatis NEES Values

Figure 24 depicts the heat map of Toutatis, capturing the shape errors at the final time. Consistent with the shape estimation performance depicted in Fig. 22, there are far more regions on Toutatis that are poorly estimated. Like Eros, the extreme ends of Toutatis (along the z-axis) experience the best estimation, while regions near the center have higher errors. The negative impact of spherically distributed basis nodes is particularly evident with Toutatis. Densely packed nodes yield more unobservable craters when perturbed, leading to higher estimation errors.

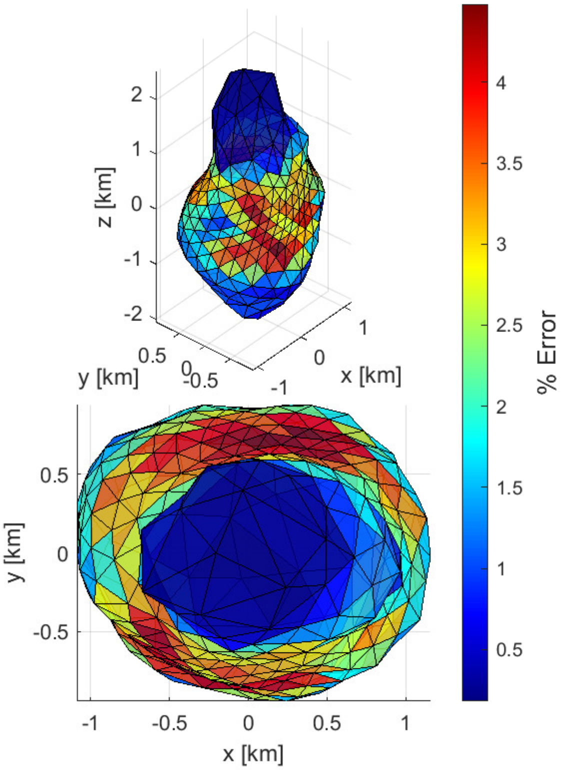


Figure 24. Toutatis Shape Error Heat Map

The slow spin of Toutatis is a driving factor of estimation performance. A week of silhouette-based measurements is likely insufficient for mission scenarios, as the majority of Toutatis remains unobserved. However, the shape estimation errors improve across the entire body, demonstrated by the heat map. Thus, the GPSF is able to estimate the entire shape, but would benefit from longer simulations to observe more of Toutatis.

Appendix D: Monte Carlo Results: Bennu. Figure 25 depicts the position navigation errors for 500 Monte Carlo simulations about Bennu. The plots are again zoomed in to better visualize the navigation errors near the end of the simulations. Figure 26 depicts the velocity navigation errors. The analytical covariance tends to be larger than the empirical covariance, particularly as time progresses.

The position and velocity navigation errors demonstrate notable skewness initially, pointing to initial position and velocity errors potentially beyond the region of linearization. The position estimation errors decrease from 20 meters per axis to less than 0.01 meters per axis, while the velocity errors decrease from 0.3 mm/s per axis to 0.004 mm/s per axis. As Toutatis, the results indicate the GPSF can obtain extreme precision.

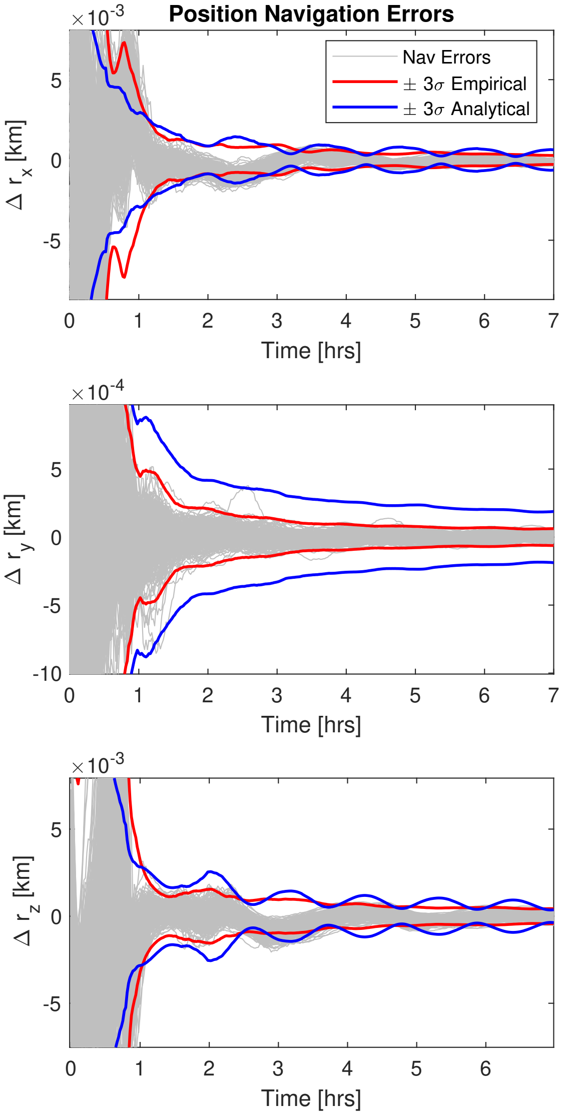


Figure 25. Bennu Position Navigation Errors

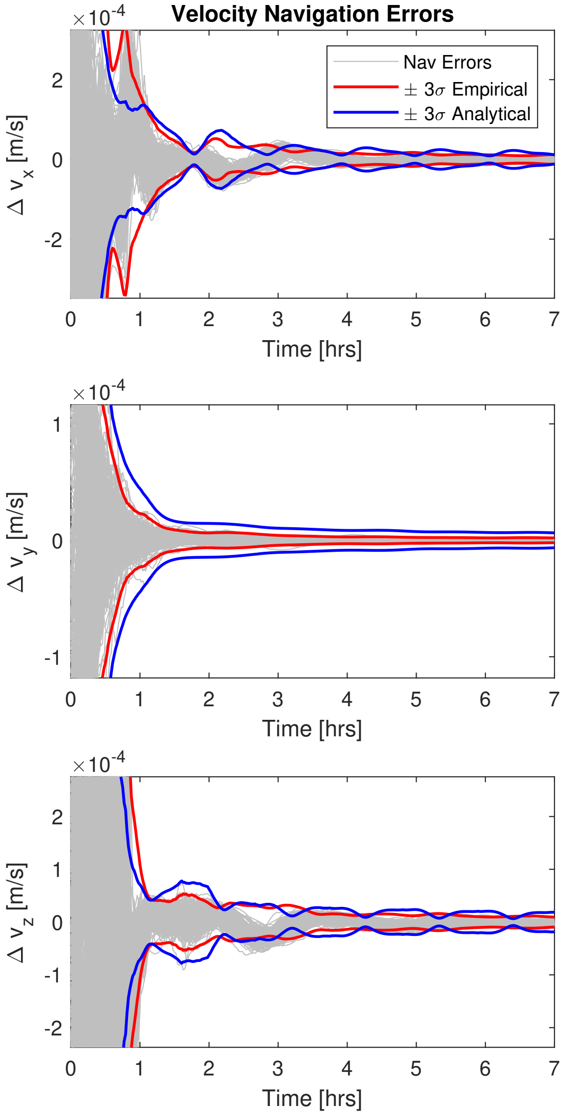


Figure 26. Bennu Velocity Navigation Errors

Figure 27 depicts the navigation errors for the right ascension, declination, and spin rate. Over a week of measurements, the pole errors decrease from a standard deviation of 7.5 degrees to less than 0.8 degrees. The spin rate decreases from 2.53 degrees per hour to 0.002 degrees per hour (or from 3% to 0.0024%).

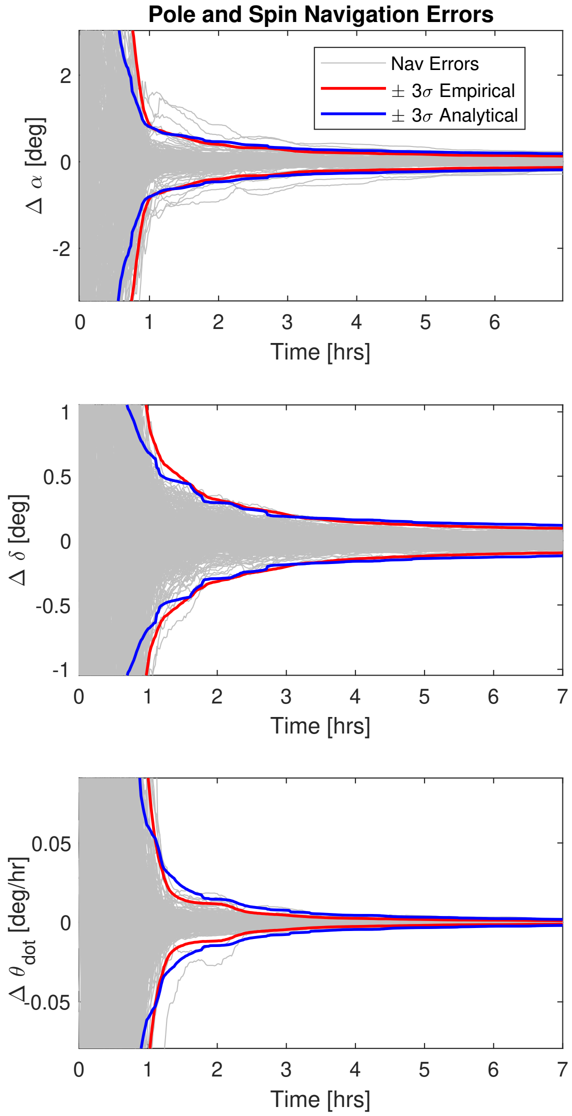


Figure 27. Bennu Spin/Pole Navigation Errors

Figure 28 depicts the shape navigation errors for the 500 Monte Carlo simulations. The shape errors converge rapidly and achieve the best estimation performance out of all the asteroid simulations.

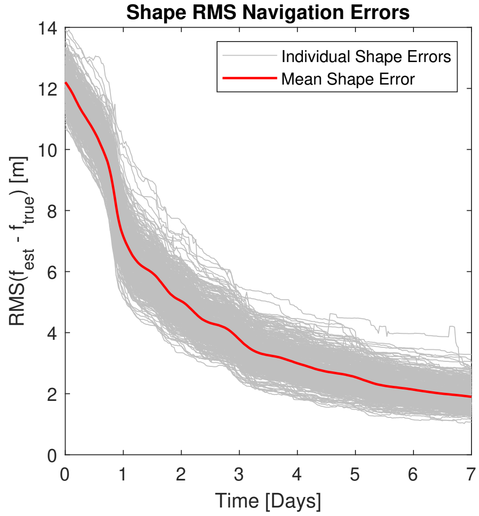


Figure 28. Bennu Navigation Shape Errors

Figure 29 depicts the NEES for each Monte Carlo run. As designed, there is a clear bias towards filter pessimism due to the measurement underweighting and covariance inflation. However, there are a large number of cases that experience filter over-confidence initially. Efforts to reduce the filter over-confidence demonstrated that the GPSF may be ill-suited for this particular simulation. Estimation about Bennu is particularly difficult due to it's low mass, corresponding small orbit, and near-spherical shape.

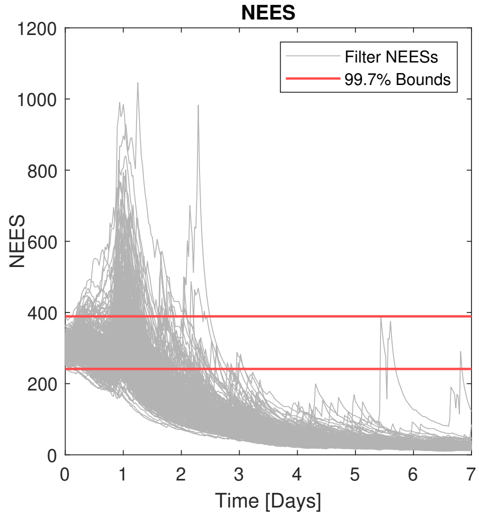


Figure 29. Bennu NEES Values

Figure 30 depicts the heat map of Bennu, capturing the shape errors at the final time. The GPSF is able to estimate the entirety of Bennu, as there are no craters or unobservable regions. Note that across the shape, the largest estimation errors are approximately 1.7%. Due to Bennu's shape and spin, the GPSF is able to achieve a greater level of estimation than the other asteroids.

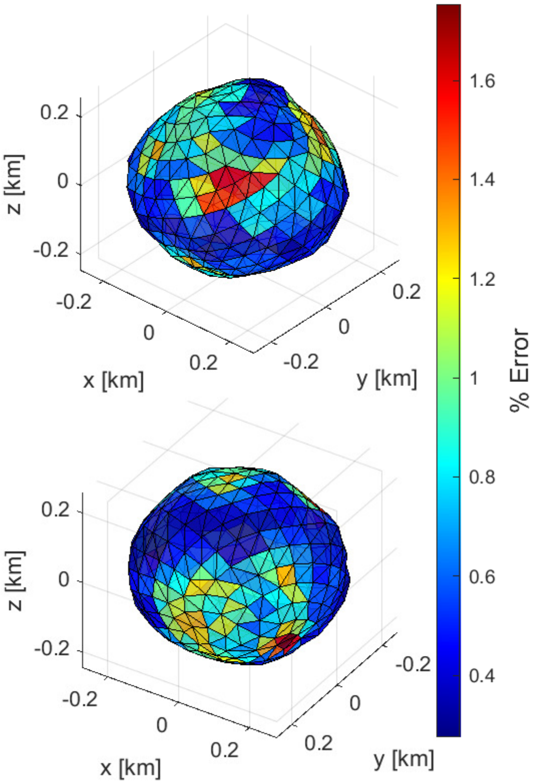


Figure 30. Bennu Shape Error Heat Map