

SMALL BODY OPTICAL NAVIGATION USING THE ADDITIVE DIVIDED DIFFERENCE SIGMA POINT FILTER

Corwin Olson,^{*} Ryan P. Russell[†] and James R. Carpenter[‡]

Sequential methods of state estimation are evaluated for small body autonomous navigation using only optical landmark measurements. Sequential techniques are employed due to the inherently sequential nature of onboard real-time navigation and the limited computational resources of onboard processors. All spacecraft state parameters (position, velocity, and attitude) are directly estimated from the optical landmark measurements to minimize measurement information loss. The standard Extended Kalman Filter (EKF) and the Additive Divided-difference sigma point Filter (ADF) are employed in various mission scenarios, with Monte Carlo analyses to compare the different techniques. The two-level nested Batch Least Squares (BLS) method used in previous studies is transformed into a sequential technique by implementing a “sliding window”, and this method is compared to the EKF and ADF. The ADF performs favorably versus the EKF and nested sliding BLS after an initial convergence period, especially for scenarios with sparse measurements and for attitude estimation. Planners and operators for current and future missions that may use landmark optical navigation about small bodies could benefit from the sequential single-filter architecture and the tuning parameters used in this study.

INTRODUCTION

Navigation about small bodies such as asteroids and comets remains a challenging problem for missions designed to orbit and land on these bodies. Conventionally, this navigation has been done on the ground in an open loop fashion using traditional filters such as the Extended Kalman Filter (EKF) and Batch Least Squares (BLS) that use Earth-based tracking signals, pictures of landmarks on the surface, and range and range-rate measurements to the surface.^{1,2} In recent years researchers and mission planners have sought to evaluate autonomous navigation technologies for small body missions in order to enable new capabilities and reduce dependence on (and the amount needed of) ground resources.^{3,4}

The AutoNav system developed and flown by JPL on five separate missions⁵ shows that autonomous navigation can be achieved for small bodies. AutoNav has not yet been used for orbiting or soft landing missions for small bodies, but the developers of AutoNav are pursuing improvements to enable these activities.⁶ Note that the algorithms used for autonomous landings through the Martian atmosphere of recent Mars rovers and landers are very different than the optical navigation algorithms needed for small body orbiting and landing scenarios, due to the presence of the atmosphere and the speed of the approach. Other researchers have compared sigma point filters such as the Unscented Kalman Filter (UKF) to the EKF for planetary Entry Descent and Landing,⁷ and Rao-Blackwellized Particle Filters for small body landings.⁸

In the studies performed by Bhaskaran, et al.³ and Olson,⁴ a two-level nested BLS approach was used to evaluate the feasibility of autonomous navigation about small bodies. However, this nesting approach results in measurement information loss, and the limited computational resources of onboard processors may lend better to sequential methods. Therein lies a main motivation for the present study: Extend the

^{*}Graduate Student, Department of Aerospace Engineering and Engineering Mechanics, 1 University Station, University of Texas at Austin, Austin, Texas 78712-0235.

[†]Assistant Professor, Department of Aerospace Engineering and Engineering Mechanics, 1 University Station, C0600, University of Texas at Austin, Austin, Texas 78712-0235.

[‡]Aerospace Engineer, Navigation and Mission Design Branch, NASA Goddard Space Flight Center, Greenbelt, MD 20771.

results of References 3 and 4 to evaluate sequential estimation techniques in a single filter architecture. In particular, the conventional EKF is employed as a benchmark, and the more recently developed higher order Additive Divided-difference sigma point Filter^{9,10} (ADF) is evaluated for use in optical navigation at small bodies. Also, the two-level nested BLS method is transformed into a sequential technique by using a “sliding window” to limit the number of measurements processed as the simulation goes forward in time, and this method is compared to the EKF and ADF. All three methods are employed in various mission scenarios to estimate the inertial position, velocity, and attitude of the spacecraft directly from the pixel and line optical measurements of the surface landmarks, with Monte Carlo analyses to compare the different techniques. These mission scenarios primarily include low circular orbits about the asteroid Eros and the much smaller comet Wirtanen, with few landmarks available in order to further stress the filters and reveal performance differences.

The ADF is chosen for evaluation over other modern sigma point filters such as the Unscented Kalman Filter (UKF) because it requires only one additional tuning parameter, versus three for the UKF. This parameter is directly related to the kurtosis of the random variables used for error generation, which simplifies the tuning. The ADF is also in a square root form by default, enhancing numerical stability and ensuring the positive definiteness of the covariance matrix.⁹

OPTICAL NAVIGATION USING LANDMARKS

Many types of measurements can be used for autonomous navigation, including radar, lidar, and optical. The navigation algorithms proposed use only optical measurements, which has strong advantages in terms of cost and ease of implementation. The camera used in the simulations is assumed to be a relatively wide-angle Charge-Coupled Device (CCD) with a focal length of 10 mm, a sensor array of 512 by 512 pixels, and a total field-of-view (FOV) of approximately 35 degrees, based on heritage values.⁴ The results of this paper are generally applicable to any CCD with a similar focal length, pixel array size, and FOV.

The optical measurements are the line-of-sight direction vectors from the spacecraft to the landmarks on the surface of the small body, whose inertial and body-fixed locations are assumed to be known. The process of using these landmark measurements to estimate the spacecraft position, velocity, and attitude are described in detail in previous studies,^{3,4} but a brief overview of the fundamentals is given below.

It is assumed that a shape model, gravity field, and set of landmarks on the surface has been previously determined. The estimation of these items, along with landmark identification, are challenging topics of their own, and will be considered simultaneously in future works. Also a triaxial ellipsoid is used instead of a polyhedron shape model. This simplified shape is not as representative as a full polyhedron shape model, but for navigation purposes the number of landmarks can be reduced to account for the shape model blocking the line-of-sight (LOS) for some landmarks. Night shading has also not been implemented in the current analysis, but the visible landmarks are sparse enough that night shading is not expected to significantly affect the performance. For a simulation with more landmarks, night shading will significantly reduce the number of visible landmarks, especially for lower orbits. Also not included are camera distortion effects on the images that are typically calibrated in flight.

A simplified version of the landmarks observation computation is as follows. To start, the vector from the spacecraft to the i^{th} landmark is given by the equation

$$O_i = L_i - S \quad (1)$$

where L_i is the body-fixed landmark position, and S is the Cartesian spacecraft position in the body-fixed frame. The term “body-fixed” refers to the frame fixed to and centered on the small body. This vector can be transformed to inertial coordinates using the known rotation matrix R_{b2i} from the small body body-fixed frame to the inertial frame.

The camera is assumed to be hard-mounted to the spacecraft bus, with the camera boresight along the spacecraft z-axis for simplicity. Thus the camera frame is synonymous with the spacecraft body-fixed frame, which is defined by the transformation

$$O_c = R_{i2c}R_{b2i}O_i \quad (2)$$

where R_{i2c} is defined by

$$R_{i2c} = R_3(\phi)R_2\left(\frac{\pi}{2} - \delta\right)R_3(\alpha) \quad (3)$$

In equation 3, $R_3(\phi)$ is a rotation about the Z-axis by the twist angle ϕ , $R_2(\frac{\pi}{2} - \delta)$ is a rotation about the Y-axis by the declination value δ subtracted from 90 degrees, and $R_3(\alpha)$ is a rotation about the Z-axis by the right ascension angle α . The declination is subtracted from 90 degrees to avoid singularities when orbiting about the equator of the body.

This LOS vector O_c in camera coordinates is then transformed into the two-dimensional camera focal plane using the gnomonic projection (which maps points on a sphere onto a plane)

$$\begin{bmatrix} x \\ y \end{bmatrix} = \frac{f}{O_c(1)} \begin{bmatrix} O_c(2) \\ O_c(3) \end{bmatrix} \quad (4)$$

where f is the camera focal length. These focal plane coordinates x and y are transformed into pixel and line values, i.e. the horizontal and vertical pixel location within the picture, with the equation

$$\begin{bmatrix} p \\ l \end{bmatrix} = \begin{bmatrix} K_x & K_{xy} & K_{xxy} \\ K_{yx} & K_y & K_{yyx} \end{bmatrix} \begin{bmatrix} x \\ y \\ xy \end{bmatrix} + \begin{bmatrix} p_0 \\ l_0 \end{bmatrix} \quad (5)$$

where the elements of the K matrix are calibrated using star fields in a real mission and p_0 and l_0 are the center pixel and line values. Note that this will produce values in increments smaller than individual pixels, as modern image processing capabilities can generate landmark center values at sub-pixel precision. In this simulation, the diagonal terms K_x and K_y are set to heritage values of 83.333⁴ and the off-diagonal terms are set to zero.

Dual Kinematic and Dynamical Filters

In previous analyses,^{3,4} a set of two filters is used in the estimation process. First, a BLS process is used in a Kinematic filter to estimate the position and attitude corrections of the spacecraft using visual landmark observations. These observations consist of the differences between the actual landmark locations in the image versus the computed locations in the image. The position fix from this Kinematic filter is used as an observation in a BLS Dynamical filter that estimates updates to the nominal epoch position and velocity of the spacecraft.

The weights of the pixel and line values used in the weighted least squares (WLS) method within the Kinematic filter are chosen based on the expected performance of the CCD. The weight matrix of the position fix observations used within the Dynamical filter is set equal to the inverse of the portion of the Kinematic filter post-fit state covariance associated with the position.

Some enhancements have been made to improve the performance of this dual filter. First, the full postfit state covariance from the Kinematic filter is used in the Dynamical filter as the measurement noise covariance. Previously only the diagonal elements were used. Second, a feature called the ‘‘sliding window’’ is implemented, as described in the next section. Third, analytical partials of the pixel and line observations with respect to the attitude portion of the state are computed using Mathematica,¹¹ and verified using finite differencing. Previously, these partials were computed using finite differencing.

Sliding Window

A sliding window is implemented within the Dynamical filter to prevent an excessive number of position fixes from getting used at the same time over extended simulations. This limit on the number of position fixes is important when simulations extend over days and multiple orbits, especially for onboard considerations.

At the beginning of the simulation, the Dynamical filter processes more and more position fixes until it reaches the maximum number of position fixes specified by the sliding window, as described by the equation

$$\hat{x}_{1|1:n} = \left(P_{1|0}^{-1} + \sum_{i=1}^n \Phi(t_i, t_1)^T H_i^T R_i^{-1} H_i \Phi(t_i, t_1) \right)^{-1} \left(\sum_{i=1}^n \Phi(t_i, t_1)^T H_i^T y_i \right) \quad (6)$$

where n is the number of times with position fixes, which is less than or equal to the maximum number of position fixes specified by the sliding window; $\hat{x}_{1|1:n}$ is the estimated correction to the nominal state at time 1, given position fixes from time 1 to time n ; $P_{1|0} = \Phi(t_1, t_0)P_{0|0}\Phi(t_1, t_0)^T$ is the apriori covariance $P_{0|0}$ mapped forward to time 1 using state transition matrix $\Phi(t_1, t_0)$; H_i is the matrix consisting of the partials of the measurements with respect to the state, which in this case is $\begin{bmatrix} I_{3 \times 3} \\ 0_{3 \times 3} \end{bmatrix}$; R_i is the postfit covariance related to the position from the Kinematic filter; and y_i is the observed position fix at time t_i minus the computed position fix at the same time, known as the residual. The efficient LAPACK routine, GESV, is implemented to solve the normal equations (using an iterative refinement technique with LU decomposition).

Once the maximum number of position fixes has been reached, the sliding window is activated. The state estimate correction is then

$$\hat{x}_{2|2:n} = \left((P_{2|1} + P_N)^{-1} + \sum_{i=2}^n \Phi(t_i, t_2)^T H_i^T R_i^{-1} H_i \Phi(t_i, t_2) \right)^{-1} * \left(\sum_{i=2}^n \Phi(t_i, t_2)^T H_i^T y_i \right) \quad (7)$$

where $x_{2|2:n}$ is the estimated state at new epoch time 2, and $P_{2|1}$ and is the postfit covariance from when $n = 1$ (or the propagated covariance state if the Dynamical filter was not activated due to a minimum number of position fixes required) mapped forward to time 2. It is important to avoid “double-counting” the position fixes as the window slides forward, so the new apriori state and covariance must be mapped from the latest postfit state and covariance that did not use any of the position fixes in the current window. The updated epoch state and covariance after each execution of the Dynamical filter can be mapped forward to the current time, and these mapped quantities are saved for use by the sliding window. When another position fix is added, the epoch shifts from time 2 to time 3, and this process continues for the rest of the simulation.

To avoid the issue of the covariance shrinking to unrealistic levels as the window slides forward, process noise P_N is added to the mapped covariance matrix $P_{2|1}$ at each time to keep the covariance open. This technique of adding process noise is similar to what is needed for the EKF (see equation 9).

Single Filter Using Extended Kalman Filter

It is well known that estimation processes benefit from using measurements in their most raw state possible (e.g. smoothing of measurements can lead to information loss). The dual-filter approach described above uses the results of one estimation process (Kinematic filter) as measurements in another (Dynamical filter). Therefore, valuable information may be lost in the process of nesting the two filters, as opposed to using a single filter to simultaneously estimate all of the states. To avoid this issue, the current position, velocity, and attitude are estimated directly from the landmark observations using a standard Extended Kalman Filter, as described by Tapley, et. al.¹² and many other texts. Note that this is only advantageous if there is strong coupling between the filters, which is the case in this analysis as the position is estimated in both filters, and these position values are strongly tied to the dynamics. Thus, it makes sense to use raw landmark measurements, just as in a tightly coupled inertial navigation system that uses GPS pseudorange measurements rather than GPS position fixes.¹³

The partials of the pixel and line values with respect to the state parameters (position, velocity, and attitude) needed for this single filter implementation are fortunately already available from the dual filter approach above, with the position partials and attitude partials the same, and the velocity partials still equal to zero.

Additive Divided Difference Sigma Point Filter

To determine if improved performance can be achieved using a sigma point filter, the ADF is implemented in the same single filter architecture mentioned above for the EKF. The ADF is a second-order divided-difference sigma point filter, using second-order numerical differencing equations to approximate nonlinear dynamical and measurement models.¹⁴ Discrete time propagation and measurement functions are used in this formulation.

The discrete time state propagation, or transition function, is described by

$$\mathbf{X}_{k+1} = \delta(\mathbf{X}_k, \mathbf{w}_k) \quad (8)$$

where k is the previous time and $k + 1$ is the current time. For this implementation of the ADF, the transition function is a numerical integration of the nine element state vector \mathbf{X}_k consisting of the position, velocity, and attitude to time t_{k+1} (versus the three Modified Rodriguez Parameters and gyro drift-rate bias vector estimated in reference 9, or the position and velocity estimated in reference 10), and the \mathbf{w}_k process noise is added to the final state. The mean process noise $\bar{\mathbf{w}}_k$ is assumed to be equal to zero for this implementation. In the ADF, the process noise values are provided as part of the sigma points described below.

The process noise matrix for the position and velocity is computed using the equation

$$\begin{aligned} Q_{k,pv}(t_{k+1}, t_k) &= \int_{t_k}^{t_{k+1}} \Phi(t_{k+1}, \tau) B Q B^T \Phi^T(t_{k+1}, \tau) d\tau \\ &= q \begin{bmatrix} I_3(t_{k+1} - t_k)^3 & I_3(t_{k+1} - t_k)^2 \\ I_3(t_{k+1} - t_k)^2 & I_3(t_{k+1} - t_k) \end{bmatrix} \end{aligned} \quad (9)$$

where I_3 is a 3-by-3 identity matrix, $Q = qI_3$, $B = \begin{bmatrix} 0_{3 \times 3} \\ I_3 \end{bmatrix}$, and q is a tunable parameter with units of length cubed per time squared.¹² The state transition matrix Φ is $\begin{bmatrix} I_3 & I_3(t_{k+1} - t_k) \\ G(t_{k+1} - t_k) & I_3 \end{bmatrix}$, which is derived from the well known property $\dot{\Phi} = A\Phi$, with A equal to $\begin{bmatrix} 0 & I_3 \\ G & 0 \end{bmatrix}$, and G equal to the partial matrix of the acceleration terms with respect to the position. This form of A assumes that the dynamics are not dependent on the velocity. Note that the G term is not used in the process noise due to the form of B . The process noise for the attitude is set to a diagonal matrix with constant user-set values on the diagonal, $Q_{k,a}$. Thus, the total process noise matrix is

$$Q_k(t_{k+1}, t_k) = \begin{bmatrix} Q_{k,pv} & 0_{6 \times 3} \\ 0_{3 \times 6} & Q_{k,a} \end{bmatrix} \quad (10)$$

The measurement model is described by

$$\tilde{\mathbf{Y}}_{k+1} = \mathbf{h}(\mathbf{X}_{k+1}, \mathbf{v}_{k+1}) \quad (11)$$

where \mathbf{v}_{k+1} is the measurement noise, which in this implementation is added to the measurement computed by the measurement model. In this simulation, the measurement is the computed pixel and line value of each visible landmark, with the measurement noise added. Note that, like for the state process noise, this measurement noise is not a direct realization of a random number, but is selected deterministically from the sigma point.

State Prediction The augmented state vector \mathbf{X}_k^{xw} combines the state and mean process noise, as shown in the equation

$$\mathbf{X}_k^{xw} = \begin{bmatrix} \mathbf{X}_k^+ \\ \bar{\mathbf{w}}_k \end{bmatrix} = \begin{bmatrix} \mathbf{X}_k^+ \\ 0_{n_w \times 1} \end{bmatrix}_{n_{xw} \times 1} \quad (12)$$

where \mathbf{X}_k^+ is the state after processing measurements at t_k , n_w is the number of process noise parameters, which is equal to the number of state parameters n_x . Both n_x and n_w equal nine in this implementation, and n_{xw} is the combination of n_x and n_w , equal to eighteen. The augmented square root matrix S_k^{xw} is defined by

$$S_k^{xw} = \begin{bmatrix} \sqrt{P_k^+} & 0 \\ 0 & \sqrt{Q_k} \end{bmatrix}_{n_{xw} \times n_{xw}} \quad (13)$$

where $\sqrt{P_k^+}$ is the Cholesky decomposition of the postfit state covariance at the previous time k , and $\sqrt{Q_k}$ is the Cholesky decomposition of the process noise matrix computed in equation 10.

Using these augmented quantities, the following sigma points are spawned, as described in Lee and Alfriend:⁹

$$\begin{aligned}\mathcal{X}_{1,k}^{xw} &= \mathbf{X}_k^{xw} \\ \mathcal{X}_{i,k}^{xw} &= \mathbf{X}_k^{xw} + hS_{i-1,k}^{xw}, \quad i = 2, \dots, n_{xw} + 1 = 2, \dots, 19 \\ \mathcal{X}_{i,k}^{xw} &= \mathbf{X}_k^{xw} - hS_{i-1-n_{xw},k}^{xw}, \quad i = n_{xw} + 2, \dots, 2n_{xw} + 1 = 20, \dots, 37\end{aligned}\quad (14)$$

where $S_{i-1,k}^{xw}$ is the $(i-1)^{\text{th}}$ column of S_k^{xw} , h is a tuning parameter and is assumed to be $\sqrt{3}$ to start, and n_{xw} is the dimension of the augmented state vector \mathbf{X}_k^{xw} , which is eighteen in this case. Thus, the overall dimensions of \mathcal{X}_k^{xw} are $n_{xw} \times (2n_{xw} + 1) = 18 \times 37$. Each column of \mathcal{X}_k^{xw} can be broken into a 9×1 state and noise part,

$$\mathcal{X}_{i,k}^{xw} = \left[(\mathcal{X}_{i,k}^x)^T (\mathcal{X}_{i,k}^w)^T \right]^T \quad (15)$$

These sigma points are propagated using the state propagation function,

$$\begin{aligned}\mathcal{X}_{i,k+1|k}^x &= \mathbf{f}(\mathcal{X}_{i,k}^x, \mathcal{X}_{i,k}^w, k) \\ &= \mathcal{X}_i^x(t_{k+1}) + \mathcal{X}_{i,k}^w\end{aligned}\quad (16)$$

and the predicted state vector is computed using these propagated sigma points:

$$\mathbf{X}_{k+1}^- = \sum_{i=1}^{2n_{xw}+1=37} \omega_{x,i}^{(m)} \mathcal{X}_{i,k+1|k}^x \quad (17)$$

where the $\omega_{x,i}^{(m)}$ weights are computed as

$$\begin{aligned}\omega_{x,1}^{(m)} &= \frac{h^2 - n_{xw}}{h^2} = \frac{3 - 18}{3} = -5 \\ \omega_{x,i}^{(m)} &= \frac{1}{2h^2} = \frac{1}{6} \quad i = 2, \dots, 2n_{xw} + 1 = 37\end{aligned}\quad (18)$$

The predicted state error covariance can be decomposed using Cholesky decomposition:

$$P_{k+1}^- = \sqrt[{}^c]{P_{k+1}^-} \bullet \sqrt[{}^c]{P_{k+1}^-}^T \quad (19)$$

where each term is $n_x \times n_x = 9 \times 9$. $\sqrt[{}^c]{P_{k+1}^-}$ is computed using a Thin QR subroutine (dgeqr2 in Fortran):

$$Q \begin{bmatrix} \sqrt[{}^c]{P_{k+1}^-}^T \\ 0_{(n_{xw}+n_w) \times n_x} \end{bmatrix}_{2n_{xw} \times n_x} = \begin{bmatrix} \mathcal{S}_{x,k+1}^{(1)} & \mathcal{S}_{x,k+1}^{(2)} \end{bmatrix}^T \quad (20)$$

where Q is a orthogonal $2n_{xw} \times 2n_{xw} = 36 \times 36$ rotation matrix (and not related to the Q_k mentioned previously), $\mathcal{S}_{x,k+1}^{(1)}$ is the $(n_x \times n_{xw})$ forward difference term, and $\mathcal{S}_{x,k+1}^{(2)}$ is the $(n_x \times n_{xw})$ central difference term. These terms are defined by

$$\begin{aligned}\mathcal{S}_{x,k+1(i,i-1)}^{(1)} &= \omega_{d,i}^{(c)} \left[\mathcal{X}_{i,k+1|k}^x - \mathcal{X}_{i+n_{xw},k+1|k}^x \right], \quad i = 2, \dots, n_{xw} + 1 = 19 \\ \mathcal{S}_{x,k+1(i,i-1)}^{(2)} &= \omega_{d,1}^{(c)} \left[\mathcal{X}_{i,k+1|k}^x + \mathcal{X}_{i+n_{xw},k+1|k}^x - 2\mathcal{X}_{1,k+1|k}^x \right], \quad i = 2, \dots, n_{xw} + 1 = 19\end{aligned}\quad (21)$$

where the $\omega_{d,i}^{(c)}$ weights are computed as in the equation

$$\begin{aligned}\omega_{d,1}^{(c)} &= \frac{\sqrt{h^2 - 1}}{2h^2} = \frac{\sqrt{2}}{6} \\ \omega_{d,i}^{(c)} &= \frac{1}{2h} = \frac{1}{2\sqrt{3}} = \frac{\sqrt{3}}{6}, \quad i \geq 2\end{aligned}\quad (22)$$

Measurement Update The measurement update also uses augmented state vectors, though they are defined differently than those used for state prediction, combining the predicted state vector with the measurement noise vector:

$$\mathbf{X}_{k+1|k}^{xv} = \begin{bmatrix} \mathbf{X}_{k+1}^- \\ \bar{\mathbf{v}}_{k+1} \end{bmatrix} = \begin{bmatrix} \mathbf{X}_{k+1}^- \\ 0_{2N_V \times 1} \end{bmatrix}_{n_{xv} \times 1 = (9+2N_V) \times 1} \quad (23)$$

where $2N_V$ is two times the number of visible landmarks at that time, and \mathbf{X}_{k+1}^- is the predicted state from equation 17. The augmented square root matrix $S_{k+1|k}^{xv}$ is defined by

$$S_{k+1|k}^{xv} = \begin{bmatrix} \sqrt[3]{P_{k+1}^-} & 0 \\ 0 & \sqrt[3]{R_{k+1}} \end{bmatrix}_{n_{xv} \times n_{xv} = (9+2N_V) \times (9+2N_V)} \quad (24)$$

where $\sqrt[3]{P_{k+1}^-}$ is computed in equation 20 and $\sqrt[3]{R_{k+1}}$ is the cholesky decomposition of the measurement noise covariance.

Using these augmented quantities, the following measurement update sigma points are spawned:

$$\begin{aligned} \mathcal{X}_{1,k+1}^{xv} &= \mathbf{X}_{k+1|k}^{xv} \\ \mathcal{X}_{i,k+1}^{xv} &= \mathbf{X}_{k+1|k}^{xv} + hS_{i-1,k+1|k}^{xv}, \quad i = 2, \dots, n_{xv} + 1 = 2, \dots, 10 + 2N_V \\ \mathcal{X}_{i,k+1}^{xv} &= \mathbf{X}_{k+1|k}^{xv} - hS_{i-1-n_{xv},k+1|k}^{xv}, \quad i = n_{xv} + 2, \dots, 2n_{xv} + 1 = 11 + 2N_V, \dots, 19 + 4N_V \end{aligned} \quad (25)$$

where $S_{i-1,k+1|k}^{xv}$ is the $(i-1)^{\text{th}}$ column of $S_{k+1|k}^{xv}$, h is a tuning parameter and is assumed to be $\sqrt{3}$ to start, and n_{xv} is the dimension of the augmented state vector $\mathbf{X}_{k+1|k}^{xv}$, which is $9 + 2N_V$ in this case. Thus, the overall dimensions of \mathcal{X}_{k+1}^{xv} are $n_{xv} \times (2n_{xv} + 1) = (9 + 2N_V) \times (19 + 4N_V)$. Each column of \mathcal{X}_{k+1}^{xv} can be broken into a 9×1 state and $2N_V \times 1$ noise part,

$$\mathcal{X}_{i,k+1}^{xv} = \left[(\mathcal{X}_{i,k+1}^{*x})^T (\mathcal{X}_{i,k+1}^v)^T \right]^T \quad (26)$$

These sigma points are fed into the measurement function,

$$\mathcal{Y}_{i,k+1} = \mathbf{h}(\mathcal{X}_{i,k+1}^{*x}, \mathcal{X}_{i,k+1}^v) \quad (27)$$

where $\mathcal{X}_{i,k+1}^v$ is added to the pixel and line values output from the measurement model. The predicted state vector is computed using these measurement function outputs:

$$\bar{\mathbf{Y}}_{k+1} = \sum_{i=1}^{2n_{xv}+1=19+4N_V} \omega_{y,i}^{(m)} \mathcal{Y}_{i,k+1} \quad (28)$$

where the $\omega_{y,i}^{(m)}$ weights are computed as

$$\begin{aligned} \omega_{y,1}^{(m)} &= \frac{h^2 - n_{xv}}{h^2} = \frac{3 - (9 + 2N_V)}{3} \\ \omega_{y,i}^{(m)} &= \frac{1}{2h^2} = \frac{1}{6} \quad i \geq 2 \end{aligned} \quad (29)$$

The innovation covariance matrix can be decomposed using Cholesky decomposition:

$$P_{k+1}^{vv} = \sqrt[3]{P_{k+1}^{vv}} \bullet \sqrt[3]{P_{k+1}^{vv}}^T \quad (30)$$

where each term is $n_v \times n_v = 2N_V \times 2N_V$. $\sqrt[3]{P_{k+1}^{vv}}$ is computed using a Thin QR subroutine (dgeqr2 in Fortran):

$$Q \begin{bmatrix} \sqrt[3]{P_{k+1}^{vv}}^T \\ 0_{(n_{xv}+n_x) \times n_v} \end{bmatrix}_{2n_{xv} \times n_v} = \begin{bmatrix} \mathcal{S}_{v,k+1}^{(1)} & \mathcal{S}_{v,k+1}^{(2)} \end{bmatrix}^T \quad (31)$$

where Q is a orthogonal $2n_{xv} \times 2n_{xv} = (18 + 4N_V) \times (18 + 4N_V)$ rotation matrix (and not related to the Q_k mentioned previously), $\mathcal{S}_{v,k+1}^{(1)}$ is the $(n_v \times n_{xv})$ forward difference term, and $\mathcal{S}_{v,k+1}^{(2)}$ is the $(n_v \times n_{xv})$ central difference term. These terms are defined by

$$\begin{aligned}\mathcal{S}_{v,k+1(\cdot,i-1)}^{(1)} &= \omega_{d,i}^{(c)} [\mathcal{Y}_{i,k+1} - \mathcal{Y}_{i+n_{xv},k+1}], \quad i = 2, \dots, n_{xv} + 1 = 10 + 2N_V \\ \mathcal{S}_{v,k+1(\cdot,i-1)}^{(2)} &= \omega_{d,1}^{(c)} [\mathcal{Y}_{i,k+1} + \mathcal{Y}_{i+n_{xv},k+1} - 2\mathcal{Y}_{1,k+1}], \quad i = 2, \dots, n_{xv} + 1 = 10 + 2N_V\end{aligned}\quad (32)$$

where the $\omega_{d,i}^{(c)}$ weights are the same as those computed in equation 22.

The cross-correlation matrix is computed using the equation

$$P_{k+1}^{xy} = \sqrt[{}^c]{P_{k+1}^-} \left[\mathcal{S}_{v,k+1(\cdot,1:n_x)}^{(1)} \right]^T \quad (33)$$

where the $n_x \times n_x$ term $\sqrt[{}^c]{P_{k+1}^-}$ is computed in equation 20 and the $n_v \times n_x$ term $\mathcal{S}_{v,k+1(\cdot,1:n_x)}^{(1)}$ consists of the first $n_x = 9$ columns of the $\mathcal{S}_{v,k+1}^{(1)}$ term computed above.

The filter gain matrix is generated with the equation

$$K_{k+1} = P_{k+1}^{xy} \left[P_{k+1}^{vv} \right]^{-1} \quad (34)$$

where P_{k+1}^{xy} is a $n_x \times n_v$ matrix, $\left[P_{k+1}^{vv} \right]^{-1}$ is a $n_v \times n_v$ matrix, and thus K_{k+1} is a $n_x \times n_v$ matrix. This gain matrix is analogous to the Kalman gain matrix in the EKF. Thus, the estimate state vector is

$$\mathbf{X}_{k+1}^+ = \mathbf{X}_{k+1}^- + K_{k+1} \nu_{k+1} \quad (35)$$

where ν_{k+1} is the innovation vector defined by

$$\nu_{k+1} = \mathbf{Y}_{k+1} - \bar{\mathbf{Y}}_{k+1} \quad (36)$$

In equation 36, \mathbf{Y}_{k+1} is the actual observation and $\bar{\mathbf{Y}}_{k+1}$ is the predicted observation computed in equation 28.

The updated state covariance matrix can be decomposed using Cholesky decomposition, as shown in the equation

$$P_{k+1}^+ = \sqrt[{}^c]{P_{k+1}^+} \bullet \sqrt[{}^c]{P_{k+1}^+}^T \quad (37)$$

where each term is $n_x \times n_x = 9 \times 9$. $\sqrt[{}^c]{P_{k+1}^+}$ is computed using a Thin QR subroutine (dgeqr2 in Fortran):

$$Q \begin{bmatrix} \sqrt[{}^c]{P_{k+1}^+}^T \\ \mathbf{0}_{(n_{xv}+n_v) \times n_x} \end{bmatrix}_{2n_{xv} \times n_x} = \begin{bmatrix} \sqrt[{}^c]{P_{k+1}^-} - K_{k+1} \mathcal{S}_{k+1,x}^{vv} & K_{k+1} \mathcal{S}_{k+1,x}^{vv} \end{bmatrix}^T \quad (38)$$

where Q is a orthogonal $2n_{xv} \times 2n_{xv}$ rotation matrix (and not related to the Q_k mentioned previously), the $n_x \times n_x$ term $\sqrt[{}^c]{P_{k+1}^-}$ is computed in equation 20, and K_{k+1} is the filter gain matrix computed in equation 34. The remaining two terms are computed as:

$$\begin{aligned}\mathcal{S}_{k+1,x}^{vv} &= \mathcal{S}_{v,k+1(\cdot,1:n_x)}^{(1)} \\ \mathcal{S}_{k+1,xv}^{vv} &= \left[\mathcal{S}_{v,k+1(\cdot,n_x+1)}^{(1)}, \dots, \mathcal{S}_{v,k+1(\cdot,n_{xv})}^{(1)}, \mathcal{S}_{v,k+1(\cdot,1:n_{xv})}^{(2)} \right]\end{aligned}\quad (39)$$

where $\mathcal{S}_{k+1,x}^{vv}$ is a $n_v \times n_x = 2N_V \times 9$ term and $\mathcal{S}_{k+1,xv}^{vv}$ is a $n_v \times (n_x + 2n_v) = 2N_V \times (9 + 4N_V)$ term.

SIMULATIONS

Simulations are performed to assess the performance of the filters described above, especially in a relative sense. All code is written in Fortran, using the latest 1995 and 2003 standards, and compiled using 2011 Intel Fortran. All simulations are run on a single processor of a quad-core Intel Xeon CPU running at 3.60 GHz. It is not the intent of this paper to provide detailed timing comparisons between methods, but the approximate timing values described below can give the reader a sense of the relative performance of each filter. The models and parameters used are described below. Various timing parameters of the simulation are listed in Table 1.

Table 1: Simulation Timing

Simulation Parameter	Nominal Values
Epoch Time	24-Nov-2017 09:55:00.00
End Time	26-Nov-2017 09:55:00.00
Minimum Minutes After Epoch Before First Update	60
Minimum Minutes Before End Time For Last Update	0
Seconds Between Measurements	1000
Seconds Between Attitude Error Realizations	100

Table 2: Small Body Properties

Simulation Parameter	Nominal Values
Initial Pole Right Ascension	30 deg
Initial Pole Declination	40 deg
Pole Right Ascension Rate	2 deg/Julian century
Pole Declination Rate	3 deg/Julian century
Longitude of the Prime Meridian at Epoch	50 deg
Rotation Rate	30 deg/day
GM	$4.4627547 \times 10^{-4} \text{ km}^3/\text{s}^2$
Reference Radius for Gravitational Harmonics	16 km
Max Radius	17.2 km
Intermediate Radius	5.6 km
Min Radius	5.6 km
Number of Landmarks on surface	20
Degree and Order of Gravity Harmonics	4 (Eros)
Normalized Gravity Harmonics Coefficients	Yes

Table 3: Spacecraft Properties

Simulation Parameter	Nominal Values
Spacecraft Mass	100 kg
Spacecraft Area	2 m ²
Spacecraft Initial Position	[20, 0, 0] km
Spacecraft Initial Velocity	[0, 4.723746×10^{-3} , 0] km/s
Propagation Tolerance	1×10^{-13}

The attitude error at intervals defined by “Time Between Attitude Error Realizations” is computed according to the attitude error model, and these error values are then interpolated to the picture times. Thus, the “Time Between Attitude Error Realizations” must be shorter than the “Seconds Between Measurements”.

The spacecraft is assumed to be pointed directly at the center of mass of the small body at every picture time using control moment gyroscopes (CMG) or reaction wheels, and the control error associated with these gyroscopes is the attitude error at every picture time.

Dynamical Trajectory Model

The simulation includes central body acceleration and perturbations from a spherical harmonic gravity field. Accelerations from third body perturbations and solar radiation pressure are deactivated because they do not significantly affect the results and conclusions of the analysis, especially when comparing the different filters. This simplification serves to greatly increase the speed of the simulation.

One scenario tested is a low circular orbit about a small body similar to the asteroid Eros, which is modeled as a triaxial ellipsoid with the same major, intermediate, and minor axes dimensions as Eros. There are twenty predetermined and randomly located landmarks available for navigation, as shown in Table 2. Other properties of the spacecraft are listed in Table 3. Figure 1 shows the orbit element evolution of the truth trajectory for the simulation. A tight circular orbit about the body is chosen to examine how well the filters handle 1) highly non-linear motion and 2) significant drops in the number of visible landmarks. However, the orbit is also chosen such that it remains outside the Brillouin (circumscribing) sphere of the body, so that spherical harmonics can be used for gravity perturbations. Figure 2 shows the truth inertial and body-fixed trajectories for this particular scenario.

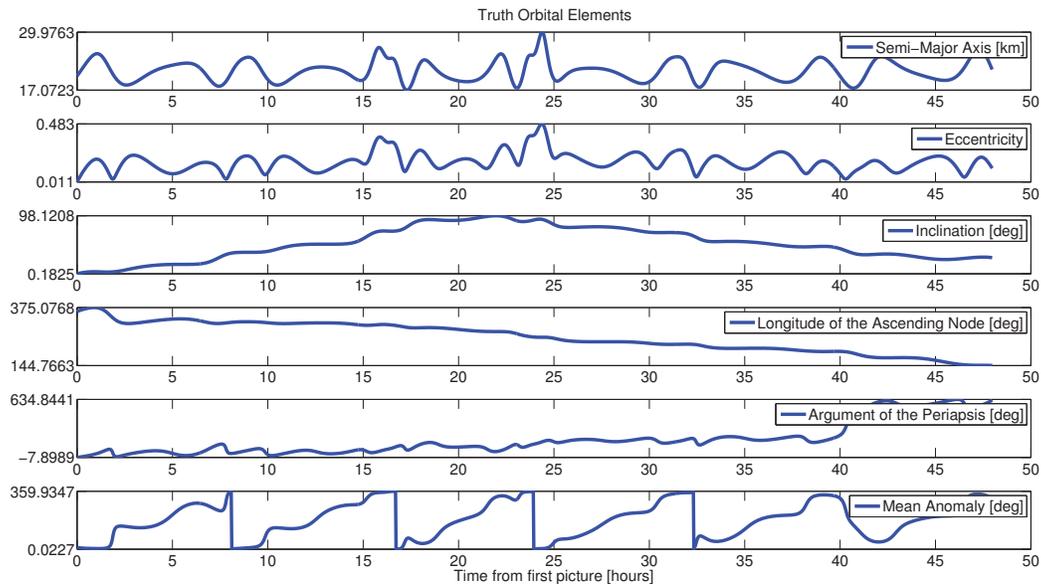


Figure 1: Truth Orbital Elements for simulation scenario

Filter Tuning Parameters

Each of the filters evaluated uses different tuning parameters in order to optimize performance. These parameters are listed in Tables 4, 5, and 6. Note that the measurement noise covariance sigmas are higher than one might expect given the measurement error of 1 pixel (1-sigma), but this measurement underweighting produced the best results after extensive tuning.

Monte Carlo Error Parameters

If the forces modeled in the filters described above completely match those of the truth model, the dynamics of the motion are linear, and the noise in the data is truly random and Gaussian, then the covariance obtained from the filter completely and accurately represents the uncertainty of the estimate. Of course, the dynamics

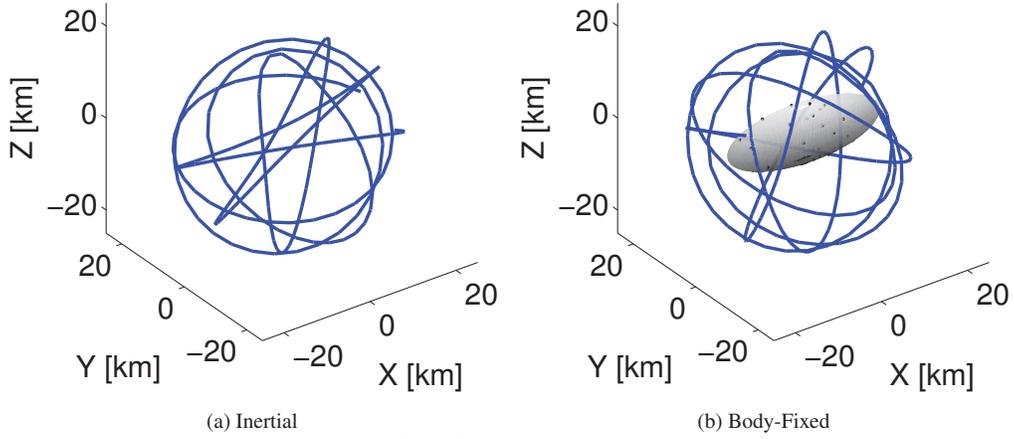


Figure 2: Truth Trajectory

Table 4: Single Filter EKF Tuning Parameters

Simulation Parameter	Nominal Values
Initial Filter Position Covariance Sigmas	[0.5, 0.5, 0.5] km
Initial Filter Velocity Covariance Sigmas	[1×10^{-4} , 1×10^{-4} , 1×10^{-4}] km/s
Initial Filter Attitude Covariance Sigmas	[10, 10, 10] deg
Measurement Noise Covariance Sigmas	[10, 10] pixels
Process Noise q for Position and Velocity, eqn. 9	1×10^{-12} km ² /s ³
Process Noise for Attitude	10 deg

Table 5: Single Filter ADF Tuning Parameters

Simulation Parameter	Nominal Values
Initial Filter Position Covariance Sigmas	[0.5, 0.5, 0.5] km
Initial Filter Velocity Covariance Sigmas	[1×10^{-4} , 1×10^{-4} , 1×10^{-4}] km/s
Initial Filter Attitude Covariance Sigmas	[10, 10, 10] deg
Measurement Noise Covariance Sigmas	[10, 10] pixels
Process Noise q for Position and Velocity, eqn. 9	1×10^{-14} km ² /s ³
Process Noise for Attitude	10 deg
Sigma Point Filter Tuning Parameter h^2	3

are nonlinear and the other conditions are never met in a real mission, nor in the current simulation. Thus Monte Carlo simulations are used to obtain a realistic assessment of the filter accuracy. The 1- σ simulation error parameters sampled in the Monte Carlo simulations are listed in Table 7.

Table 6: Dual Filter BLS Tuning Parameters

Simulation Parameter	Nominal Values
Minimum Number of Position Fixes	2
Maximum Number of Position Fixes	5, 10
Maximum Number of Kinematic Filter Iterations	10
Maximum Number of Dynamical Filter Iterations	10
Kinematic Filter Position Correction Norm Stop Condition	1×10^{-3} km
Kinematic Filter Attitude Correction Norm Stop Condition	1×10^{-2} deg
Dynamical Filter Position Correction Norm Stop Condition	1×10^{-5} km
Initial Kinematic Filter Position Covariance Sigmas	[10, 10, 10] km
Initial Kinematic Filter Attitude Covariance Sigmas	[20, 20, 20] deg
Initial Dynamical Filter Position Covariance Sigmas	[5, 5, 5] km
Initial Dynamical Filter Velocity Covariance Sigmas	$[1 \times 10^{-4}, 1 \times 10^{-4}, 1 \times 10^{-4}]$ km/s
Measurement Noise Covariance Sigmas	[10, 10] pixels
Process Noise q for Position and Velocity, eqn. 9	1×10^{-11} km ² /s ³

Table 7: 1- σ Monte Carlo Error Parameters

Simulation Parameter	Nominal Values
Spacecraft Mass	2 kg
Spacecraft Area	0.1 m ²
Spacecraft Initial Position	[0.005, 0.005, 0.005] km
Spacecraft Initial Velocity	$[5 \times 10^{-6}, 5 \times 10^{-6}, 5 \times 10^{-6}]$ km/s
Initial Attitude	1 deg
Attitude Noise	1.89×10^{-4} deg
Attitude Drift	0.01/3 deg/hr
Attitude Random Walk	0.025 deg/sqrt(hr)
Observation	[1, 1] pixel

RESULTS

Figure 3 shows the error magnitude and covariance for 100 monte carlo trials using the EKF for this scenario, as well as the number of landmarks visible at each picture time. The simulation took approximately 4.6 minutes. An optimum (within an order of magnitude) process noise multiplier value of 1.d-12 was used for the EKF.

Figure 4 shows the error magnitude and covariance for 100 monte carlo trials using the ADF for this scenario, as well as the number of landmarks visible at each picture time. The simulation took approximately 5.5 minutes. This time is longer than the 4.6 minutes needed for the EKF, but not substantially longer. An optimum (within an order of magnitude) process noise multiplier value of 1.d-14 was used for the ADF.

Comparing the performance of the ADF in Figure 4 to the EKF in Figure 3, it is seen that the EKF does appear to have better performance in the beginning of the simulation, as the ADF takes some time to converge for all reasonable turning parameters attempted. However, after this initial convergence period the ADF better handles the sparseness of visible landmarks that occurs shortly before and after hour 20. And in general the ADF error following this middle section is lower than the EKF error, especially for the attitude. The attitude results for the ADF are superior to those from the EKF also because the 3-sigma covariance plot remains above the error and is thus a better representation of the attitude state uncertainty. When the EKF is run with the same lower process noise as the ADF, the sparse measurements around hour 20 result in severely degraded EKF performance for the remainder of the simulation.

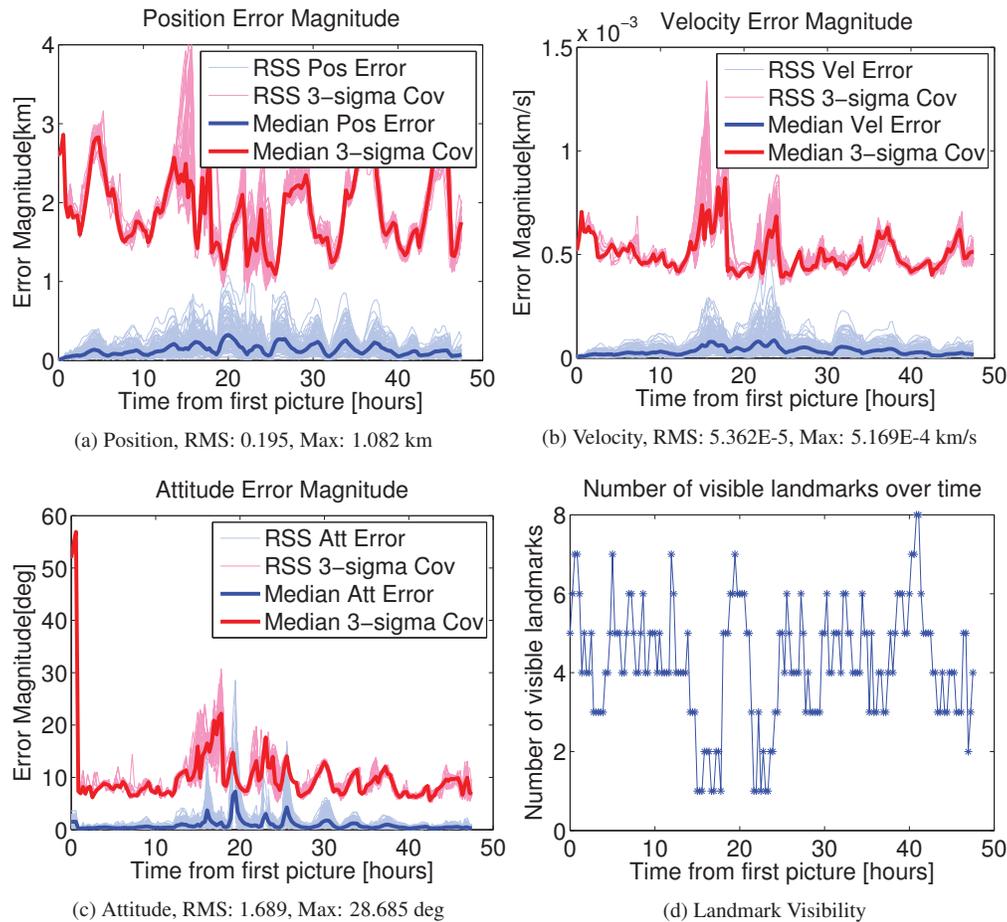


Figure 3: RSS Error, 3x Root-Variance for the EKF, 100 Monte Carlo Trials.

For close circular orbits about smaller bodies with far less mass, such as the 46P/Wirtanen comet that has a mass roughly four orders of magnitude less than Eros, the performance of the EKF and ADF is nearly identical. This lack of difference between the two filters is as expected, because much lower velocities are required to maintain a close circular orbit. This slower orbital motion is far more linear over a given time increment, and thus the propagation of sigma points is not beneficial.

Figure 5 shows the error magnitude and covariance for 100 monte carlo trials using the dual-filter BLS for the Eros scenario, with a sliding window of 5 position fixes. The simulation took approximately 2.5 minutes. Figure 6 shows the error magnitude and covariance for 100 monte carlo trials using the dual-filter BLS with a sliding window of 10 position fixes for this scenario. The simulation took approximately 7.2 minutes.

Comparing Figures 5 and 6, increasing the size of the sliding window from 5 to 10 does not have a substantial affect on the median performance, but the 10 position fix scenario has a number of error spikes beyond all covariance plots. At the beginning of the simulation, the 10 position fix filter performs better than 5 position fix filter, as expected. However, when the simulated spacecraft reaches an area where far fewer landmarks are visible, which occurs around hour 15 as seen in the landmark visibility plots, the number and quality of the position fixes drops significantly. This drop results in lower accuracy state estimates at this time, and when these higher error states become the epoch state used for the sliding window, they are propagated further

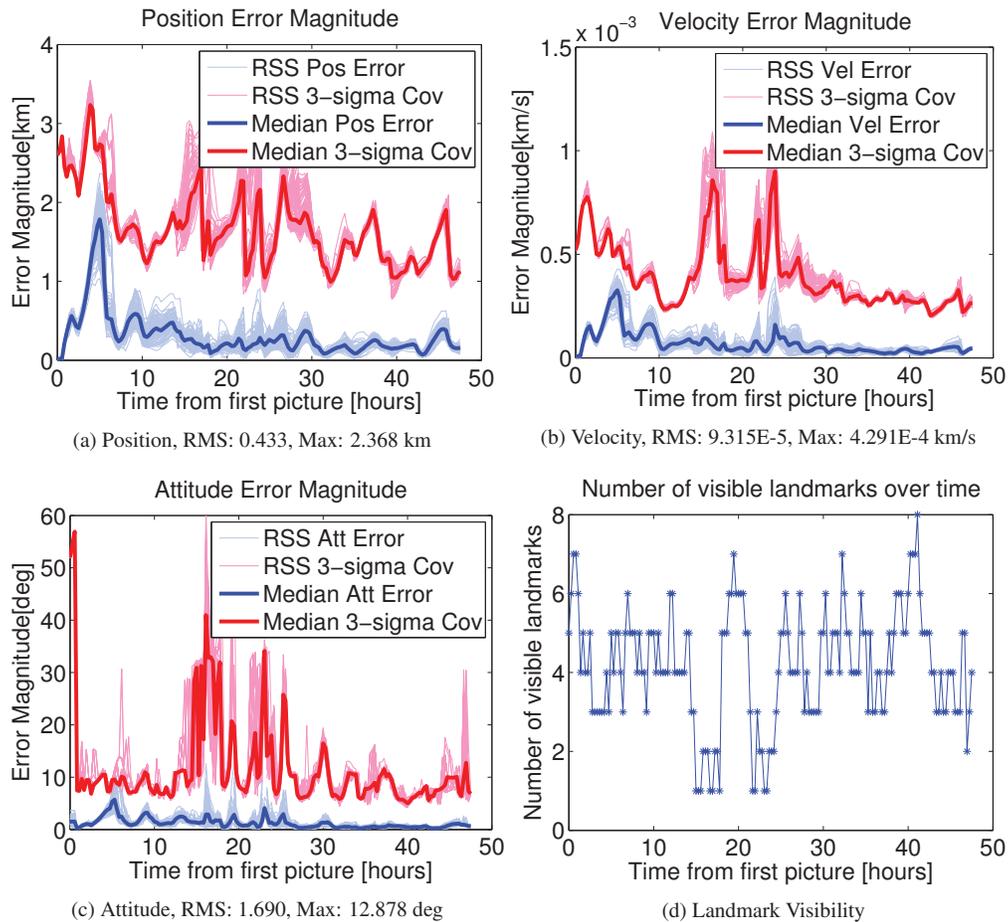


Figure 4: RSS Error, 3x Root-Variance for the ADF, 100 Monte Carlo Trials.

in a wider sliding window, and the dynamical filter is far less successful in fitting the position fixes to this worse propagation. This trend is verified with more position fixes in the sliding window, with 15, 20, and 30 position fix performance increasingly worse.

This greater capability of the shorter sliding window to handle intermittent times of sparse data, as well as the superior performance of the EKF and ADF, suggests that a sequential approach other than this sliding window BLS is best in this scenario. If plentiful landmarks are available, as is the case in previous studies,^{3,4} a sliding BLS approach is likely feasible, but this abundance of landmarks is not likely to exist until significant mapping campaigns have taken place in a mission. The BLS approach also works well in terms of navigation accuracy when all data is considered globally (i.e. when the sliding window feature is off), but the run time rapidly increases as the number of position fix observations increases, which makes the algorithm ill-suited for onboard use.

The runtime of the BLS with a sliding window of 5 position fixes is about half the runtime of the EKF, performing 100 monte carlo trials in 2.5 minutes versus the 4.6 minutes taken by the EKF. However, the EKF is far more accurate given the tuning parameters used (which were adjusted to prevent divergence of any runs, versus optimum performance of the median).

Table 8 summarizes the results given above. The covariance realism test is marked as yes if the covariance properly reflects the uncertainty in the estimate, as judged by the observed errors.

Table 8: Results Summary

Simulation	Pos Error [km]		Vel Error [km/s]		Att Error [deg]		Time [sec]	Cov Realism
	RMS	Max	RMS	Max	RMS	Max		
EKF	0.195	1.082	5.362E-5	5.169E-4	1.689	28.685	276	No
ADF	0.433	2.368	9.315E-5	4.291E-4	1.690	12.878	329	Yes
EKF, after Hour 20	0.204	1.082	5.719E-5	5.169E-4	1.425	16.932		No
ADF, after Hour 20	0.226	0.908	5.492E-5	3.912E-4	1.051	9.237		Yes
BLS, 5 Pos Fix	0.447	9.057	1.519E-4	4.161E-3	N/A	N/A	151	Yes
BLS, 10 Pos Fix	1.684	136.141	4.048E-4	2.657E-2	N/A	N/A	432	No

Many future analyses and simulation enhancements are possible, to both increase the fidelity of the simulation and match more modern specifications for the cameras that are planned for future missions. One possible analysis is a comparison of results from running with many other small bodies of different sizes and shapes, as well as different mission scenarios such as highly elliptical orbits. This would allow us to further test the relative abilities of the ADF versus the EKF in handling non-linear motion. Possible modeling improvements include adding night shading, using a polyhedron shape model rather than a triaxial ellipsoid, including camera optical distortion effects, and increasing the camera resolution to 1024x1024 or greater to match the resolution of cameras used on upcoming missions such as OSIRIS-REx. Adding off-nadir pointing measurements to improve measurement geometry would also be useful to investigate, in order to determine if improved geometry results in improved navigation performance. Finally, an assessment of the different filters with scenarios involving maneuvers and landing scenarios would be very useful, as scenarios involving landings or frequent maneuvers are most likely to need autonomous navigation.

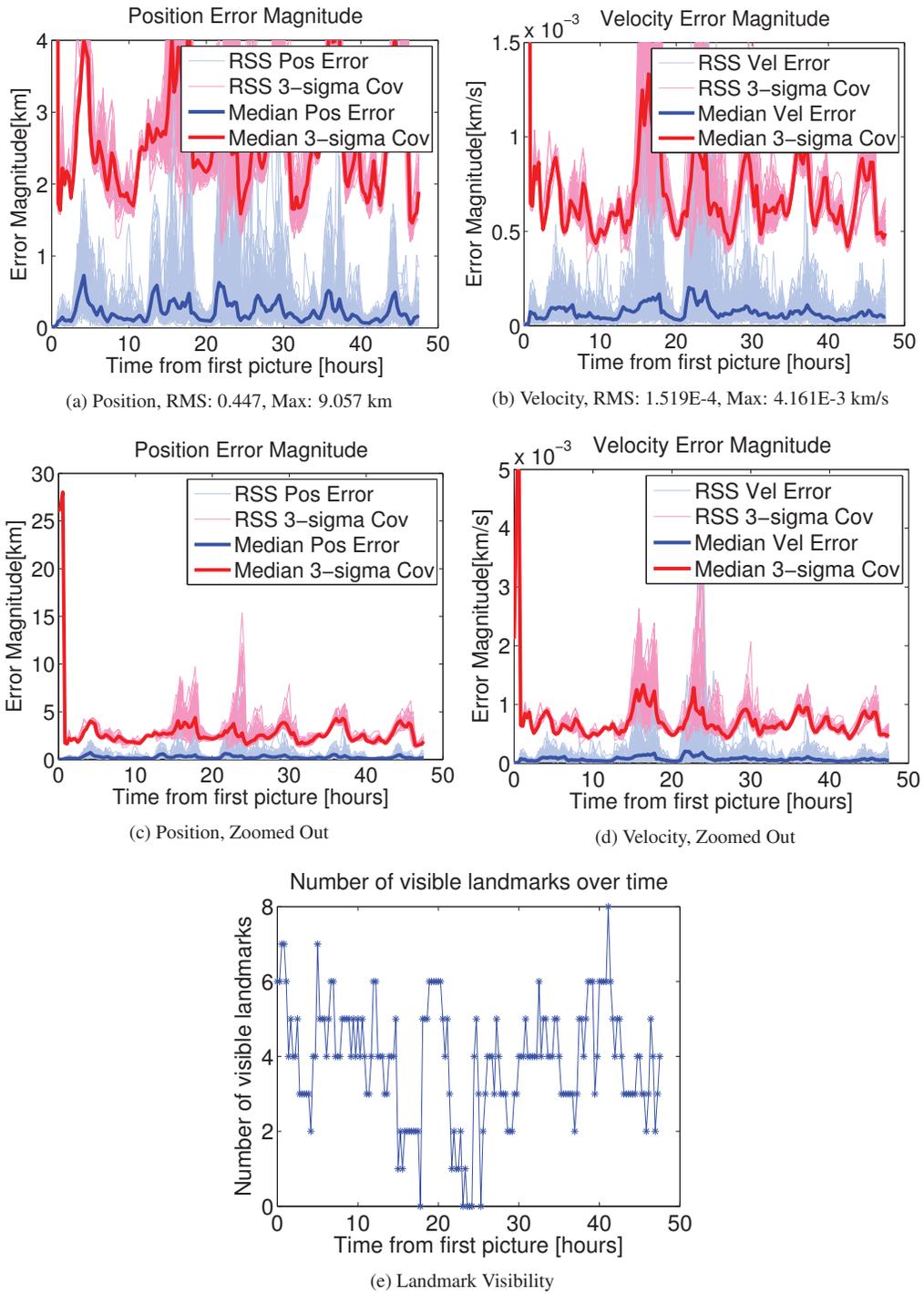


Figure 5: RSS Error, 3x Root-Variance for the BLS, 5 position fix sliding window, 100 Monte Carlo Trials.

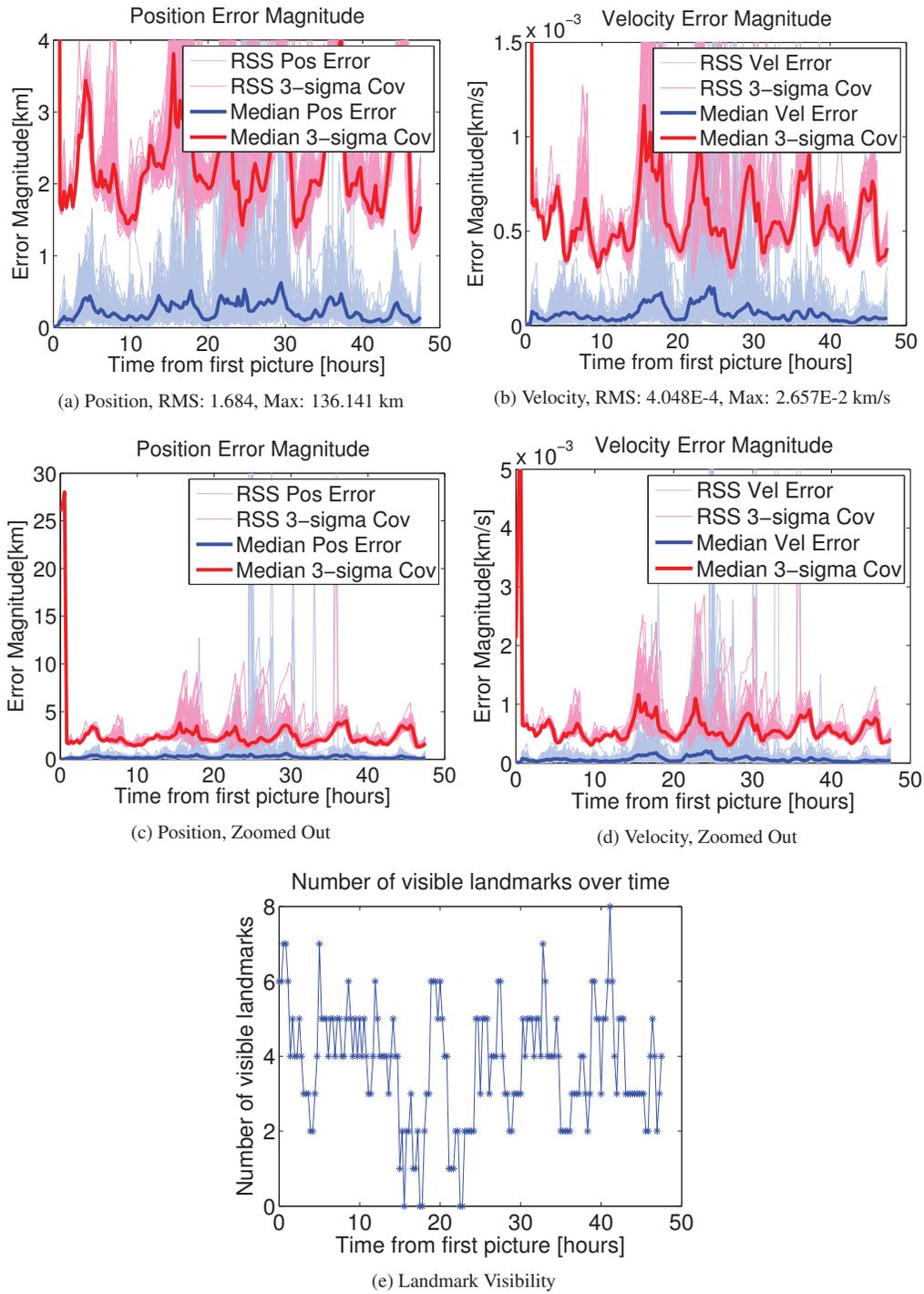


Figure 6: RSS Error, 3x Root-Variance for the BLS, 10 position fix sliding window, 100 Monte Carlo Trials.

CONCLUSIONS

In this paper, we extend the work previously done for small body autonomous optical navigation by evaluating sequential methods of state estimation with only optical landmark measurements. The standard EKF and the ADF are used to estimate the inertial spacecraft position, velocity, and attitude directly from the pixel and line optical measurements, with Monte Carlo analyses to compare the different techniques. The two-level nested BLS method is transformed into a sequential technique by using a “sliding window”, and this method is compared to the EKF and ADF.

The ADF performs favorably versus the EKF and nested BLS with a sliding window after an initialization period, given optimized tuning parameters for each case. It should be noted that the performance of each filter is highly sensitive to the tuning parameters used in the mission scenarios used. The covariance generated by the ADF also better represents the uncertainty in the state estimate, especially for the attitude state, and particularly during times of sparse measurements. This superior performance occurs because the ADF better captures the nonlinear motion and measurements of a spacecraft orbiting a small body using optical landmarks to navigate, particularly for extended simulations and when fewer landmarks are available. However, the EKF achieves better results initially than the ADF. Thus, if it is critical to obtain good estimates as quickly as possible using a rough initial state estimate, the EKF is better suited to the task. Otherwise the ADF provides better long term performance and covariance values, especially for the attitude state.

The ADF is also easier to tune than the traditional UKF, as there is only one additional tunable parameter compared to the EKF. This tuning parameter has a known relationship to the kurtosis of the error distributions, which enables a better first guess and subsequent easier tuning. However, for much smaller bodies, such as Wirtanen, the significantly lower orbital velocities and thus more linear motion lead to almost identical performance for the EKF versus the ADF. Thus, for bodies on the scale of Wirtanen, an ADF provides no significant advantage over the EKF. But for bodies on the scale of Eros or larger, the ADF does provide superior steady state performance for lower and faster orbits, especially when measurements are sparse.

The BLS with a sliding window improves the run time of the dual-filter BLS from previous studies when running over extended simulation times. However, only windows with smaller numbers of position fixes are able to handle segments of sparse measurements within the simulation, which is due to the lower amount of time to propagate forward any epoch states with higher error. In general the dual-filter BLS (even with a shorter sliding window) does not perform as well as the EKF and ADF.

The sequential estimation architecture described, along with the tuning parameters used, may prove useful to future mission planners and operators. In particular, the ADF filter is employed for the first time in small body autonomous optical navigation, which provides an alternative higher order approach to the Unscented Kalman Filter that has proven popular in astrodynamics research recently.

ACKNOWLEDGMENT

The work described in this paper was funded by NASA’s Chief Technology Office through a NASA Space Technology Research Fellowship.

REFERENCES

- [1] B. Williams, P. Antreasian, J. Bordi, E. Carranza, S. Chesley, C. Helfrich, J. Miller, M. Owen, and T. Wang, “Navigation for NEAR Shoemaker - The first spacecraft to orbit an asteroid,” *Advances in the Astronautical Sciences*, 2002, pp. 973–988.
- [2] C. Russel, C. Raymond, T. Fraschetti, M. Rayman, C. Polanskey, K. Schimmels, and S. Joy, “Dawn mission and operations,” *Proceedings of the International Astronomical Union*, Vol. 1, 2005.
- [3] S. Bhaskaran, S. Nandi, S. Broschart, M. Wallace, L. Cangahuala, and C. Olson, “Small Body Landings Using Autonomous Onboard Optical Navigation,” *The Journal of the Astronautical Sciences*, Vol. 58, No. 3, 2011, pp. 409–427.
- [4] C. Olson, “Analysis of Asteroid Landing Capabilities Using Autonomous Optical Navigation,” Master’s thesis, The University of Texas at Austin, Austin, TX, 2009.
- [5] S. Bhaskaran, “Autonomous Navigation for Deep Space Missions,” *SpaceOps 2012 Conference*, 2012.
- [6] Y. Cheng, D. Clouse, A. Johnson, W. Owen, and A. Vaughan, “Evaluation and Improvement of Passive Optical Terrain Relative Navigation Algorithms for Pinpoint Landing,” *AAS/AIAA Spaceflight Mechanics Meeting*, February 2011. Paper AAS 11-221.

- [7] S. Dutta, R. Braun, R. Russell, S. Striepe, and I. Clark, "Comparison of Statistical Estimation Techniques for Mars Entry, Descent, and Landing Reconstruction," *Spacecraft and Rockets*, 2013. Pre-print as of September 2013.
- [8] C. Cocaud and T. Kubota, "Autonomous navigation near asteroid based on visual SLAM," *23rd International Symposium on Space Flight Dynamics*, October-November 2012.
- [9] D. Lee and K. Alfriend, "Additive Divided Difference Filtering for Attitude Estimation using Modified Rodrigues Parameters," *Astronautical Sciences*, Vol. 57, No. 1-2, 2009, pp. 93–111.
- [10] J. Carpenter, S. Hur-Diaz, and F. Markley, "Generalized Covariance Analysis of Additive Divided-Difference Sigma-Point Filters," *AAS/AIAA Spaceflight Mechanics Meeting*, Feb 2011. AAS 11-201.
- [11] I. Wolfram Research, *Mathematica*. Champaign, Illinois: Wolfram Research, Inc., 8.0 ed., 2010.
- [12] B. Tapley, B. Schutz, and G. Born, *Statistical Orbit Determination*. Elsevier Academic Press, 2004.
- [13] J. D. Weiss and D. S. Kee, "A direct performance comparison between loosely coupled and tightly coupled GPS/INS integration techniques," 04 1995. TY: CPAPER; SuppNotes - Proceedings (A96-14001 02-32; Last updated - 2011-11-11).
- [14] M. Nørgaard, N. Poulsen, and O. Ravn, "Advances in Derivative-Free State Estimation for Nonlinear Systems," tech. rep., Department of Mathematical Modelling, DTU, 1998. Revised April 2000.