Journal of Guidance, Control, and Dynamics
Vol. 30, No. 6, November–December 2007

Nonlinear Semi-Analytic Methods for Trajectory Estimation

Ryan S. Park* and Daniel J. Scheeres†
University of Michigan, Ann Arbor, Michigan 48109

DOI: 10.2514/1.29106

Nonlinear semi-analytic filtering methods to sequentially estimate spacecraft states and their associated uncertainties are presented. We first discuss the state transition tensors that characterize the localized nonlinear behavior of the trajectory statistics and illustrate the importance of higher-order effects on orbit uncertainty propagation. We then present a semi-analytic filtering method by implementing the state transition tensors to sequentially update the filter information with contributions from each measurement, which requires no integration once the tensors are computed. A sun–Earth halo orbit about the L_1 point is considered as an example with realistic orbit uncertainties, and the results are compared with the extended Kalman filter and unscented Kalman filter.

Nomenclature

$\mathbf{g}$, g^i	=	system dynamics vector and its ith component
$\mathbf{h}$, h^i	=	measurement vector and its ith component
$\mathbf{K}_k$	=	Kalman gain computed at time t_k
$\mathbf{m}$, m^i	=	mean vector and its ith component
$\mathbf{m}_k^-$, $(m_k^-)^i$	=	predicted mean vector at time t_k and its ith component
$\mathbf{m}_k^+$, $(m_k^+)^i$	=	updated mean vector at time t_k and its ith component
N	=	system dimension
$\mathbf{n}_k^-$, $(n_k^-)^i$	=	predicted measurement function at time t_k and its ith component
$\mathbf{P}$, P^{ij}	=	covariance matrix and its (i, j) entry
$\mathbf{P}_k^-$, $(P_k^-)^{ij}$	=	predicted covariance matrix at time t_k and its (i, j) entry
$\mathbf{P}_k^+$, $(P_k^+)^{ij}$	=	updated covariance matrix at time t_k and its (i, j) entry
p	=	probability density function
$\mathbf{Q}_k$, Q_k^{ij}	=	diffusion (process noise) matrix at time t_k and its (i, j) entry
$\mathbf{R}_k$, R_k^{ij}	=	measurement noise matrix at time t_k and its (i, j) entry
t	=	time
U	=	potential function for circular restricted three-body problem
u, v, w	=	spacecraft velocity components
$\mathbf{v}$, v^i	=	measurement noise vector and its ith component
$\mathbf{w}$, w^i	=	process noise vector and its ith component
$\mathbf{x}$, x^i	=	state vector and its ith component
x, y, z	=	spacecraft position components
$\mathbf{x}_0$, x_0^i	=	initial state vector and its ith component
$\delta\mathbf{x}$, δx^i	=	relative state vector and its ith component
$\delta\mathbf{x}_0$, δx_0^i	=	initial relative state vector and its ith component
δ_{ij}	=	Dirac delta function
μ_S	=	solar gravitational constant, $1.32712440018 \times 10^{11}\ \mathrm{km}^3/\mathrm{s}^2$
$\mu_\oplus$	=	Earth gravitational constant, $398,600.44\ \mathrm{km}^3/\mathrm{s}^2$
$\boldsymbol{\Phi}$	=	state transition matrix
$\boldsymbol{\phi}$, ϕ^i	=	solution flow and its ith component
$\phi^{i,\gamma_1\cdots\gamma_p}$	=	pth order state transition tensor
$\boldsymbol{\psi}$, ψ^i	=	inverse solution flow and its ith component
$\psi^{i,\gamma_1\cdots\gamma_p}$	=	pth order inverse state transition tensor
ω_E	=	mean motion of the Earth about the sun, $\sqrt{\mathrm{AU}^3/\mu_S} = 1.991 \times 10^{-7}\ \mathrm{s}^{-1}$

I. Introduction

ORBIT uncertainty propagation plays an important role in various space-related applications, such as orbit determination, parameter estimation, correction maneuver design, small-body collision/encounter analysis, etc. In practice, it is usually assumed that the *true* motion (in a statistical sense) of a spacecraft with respect to a nominal trajectory is within a boundary where the linear assumption sufficiently approximates the relative dynamics and the covariance matrix is mapped using the Riccati equations. In some cases, however, the linear assumption fails to provide an accurate realization of the local trajectory motion and in such cases a different method which accounts for the system nonlinearity must be implemented.

The best known technique for nonlinear orbit uncertainty propagation is a Monte Carlo (MC) simulation, which approximates the probability distribution by averaging over a large set of random samples [1]. A Monte Carlo simulation can provide true statistics in the limit, but is computationally intensive and only solves for the statistics of a specific epoch and its associated uncertainties. Hence, for mission operations, these difficulties make Monte Carlo simulations inefficient for practical spacecraft applications. Recently, Park and Scheeres [2–5] have developed a *semi-analytic* method for orbit uncertainty propagation by solving for the higher-order Taylor series terms that describe the localized nonlinear motion and by analytically mapping the initial uncertainties. By considering sufficiently high-order solutions, they have shown that their semi-analytic approach for orbit uncertainty propagation can replicate Monte Carlo simulations with the benefit of added flexibility in initial orbit statistics.

In this paper, we derive nonlinear filters by assuming a Gaussian statistic and by implementing the semi-analytic orbit uncertainty propagation technique developed by Park and Scheeres. A filter is usually composed of two parts, *prediction* and *update*. Orbit uncertainty propagation relates to the prediction problem while the uncertain distribution of a measurement and state influences the update part. In conventional trajectory navigation, one is usually given a reference (nominal) trajectory with precise ephemerides and the extended Kalman filter (EKF) is used for trajectory estimation. The objective of trajectory navigation is then to follow the reference trajectory while minimizing some predefined optimality constraints, such as the number of trajectory correction maneuvers, flight time, fuel, etc. [Note that when the trajectory deviates from the reference trajectory over some error boundary a correction maneuver is applied

Presented as Paper 6399 at the AAS/AIAA 2006 Astrodynamics Specialists Conference, Keystone, CO, 21–24 August 2006; received 30 November 2006; accepted for publication 24 March 2007. Copyright © 2007 by the authors. Published by the American Institute of Aeronautics and Astronautics, Inc., with permission. Copies of this paper may be made for personal or internal use, on condition that the copier pay the $10.00 per-copy fee to the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923; include the code 0731-5090/07 $10.00 in correspondence with the CCC.

*Ph.D. Student, Department of Aerospace Engineering; currently Member, Technical Staff, Outer Planet Navigation, Jet Propulsion Laboratory, Pasadena, CA 91109; Ryan.S.Park@jpl.nasa.gov. Member AIAA.

†Associate Professor, Department of Aerospace Engineering; scheeres@umich.edu. Associate Fellow AIAA.

to recourse the spacecraft to the reference trajectory (or to an alternate trajectory that satisfies mission objectives).] The basic underlying concept of such a process is to stay within the linear region by taking a sufficient number of measurements and linearly map the deviation and statistics via the state transition (fundamental) matrix with respect to the nominal trajectory. However, when the nonlinearity is significant or when only a limited number of measurements are available, it may be necessary to consider a filter that incorporates system nonlinearity.

The sun–Earth halo orbit about the L_1 point based on the circular restricted three-body problem (CR3BP) is chosen as an example because the overall nonlinearity is small, but the trajectory is unstable and when the spacecraft is not sufficiently observed, the nonlinear effect can become significant. The proposed semi-analytic filters are compared with the extended Kalman filter and unscented Kalman filter with realistic orbit uncertainties. The result shows that our higher-order filters provide faster convergence and a superior solution as compared to linear filters.

II. Higher-Order Perturbation Analysis

The motion of a spacecraft can be modeled with first-order ordinary differential equations:

$$\frac{\mathrm{d}\mathbf{x}(t)}{\mathrm{d}t} = \mathbf{g}[t; \mathbf{x}(t)] \tag{1}$$

where $\mathbf{g}[t; \mathbf{x}(t)]$ represents the system dynamics vector with a dimension N and $\mathbf{x} = \{x^i \mid i = 1, \ldots, N\}$ represents the spacecraft state vector with the initial condition $\mathbf{x}(t_0) = \mathbf{x}_0$. The solution flow, which maps the initial state at t_0 to t, is then defined as

$$\mathbf{x}\,(t) = \boldsymbol{\phi}(t; \mathbf{x}_0, t_0) \tag{2}$$

The solution flow is governed by

$$\frac{\mathrm{d}\boldsymbol{\phi}(t; \mathbf{x}_0, t_0)}{\mathrm{d}t} = \mathbf{g}[t, \boldsymbol{\phi}(t; \mathbf{x}_0, t_0)] \tag{3}$$

$$\boldsymbol{\phi}\,(t_0; \mathbf{x}_0, t_0) = \mathbf{x}_0 \tag{4}$$

By considering a similar notation, we define the inverse solution flow that maps the state at t to the initial state as

$$\mathbf{x}_{\,0} = \boldsymbol{\psi}(t, \mathbf{x}; t_0) \tag{5}$$

In this framework we define the local trajectory dynamics $\delta\mathbf{x}$ by applying a Taylor series expansion about the reference (nominal) trajectory, that is, $\delta\mathbf{x}(t) = \boldsymbol{\phi}(t; \mathbf{x}_0 + \delta\mathbf{x}_0, t_0) - \boldsymbol{\phi}(t; \mathbf{x}_0, t_0)$, for some initial deviation $\delta\mathbf{x}_0$. The mth order solution can be stated using the Einstein summation convention as

$$\delta x^i(t) = \sum_{p=1}^{m} \frac{1}{p!} \phi_{(t,t_0)}^{i,\gamma_1\cdots\gamma_p} \delta x_0^{\gamma_1} \cdots \delta x_0^{\gamma_p} \tag{6}$$

where $\gamma_j \in \{1, \ldots, N\}$, superscripts γ_j denote the γ_jth component of the state vector, and

$$\phi_{(t,t_0)}^{i,\gamma_1\cdots\gamma_p}(t; \mathbf{x}_0, t_0) = \left.\frac{\partial^p \phi^i(t; \boldsymbol{\xi}_0, t_0)}{\partial \xi_0^{\gamma_1} \cdots \partial \xi_0^{\gamma_p}}\right|_{\xi_0^j = x_0^j} \tag{7}$$

We call the higher-order partials of the solution flow Eq. (7) the state transition tensors (STTs), which map the initial deviations to the current time. Note that the first-order case ($p = 1$) reduces to the usual state transition matrix (STM). The differential equations up to fourth-order deviation are given in Eqs. (8–11). For more details on how to obtain these differential equations, readers are referred to [4,5].

$$\dot{\phi}^{\,i,a} = g^{i,\alpha}\phi^{\alpha,a} \tag{8}$$

$$\dot{\phi}^{\,i,ab} = g^{i,\alpha}\phi^{\alpha,ab} + g^{i,\alpha\beta}\phi^{\alpha,a}\phi^{\beta,b} \tag{9}$$

$$\begin{aligned}\dot{\phi}^{i,abc} &= g^{i,\alpha}\phi^{\alpha,abc} + g^{i,\alpha\beta}(\phi^{\alpha,a}\phi^{\beta,bc} + \phi^{\alpha,ab}\phi^{\beta,c} + \phi^{\alpha,ac}\phi^{\beta,b}) \\ &\quad + g^{i,\alpha\beta\gamma}\phi^{\alpha,a}\phi^{\beta,b}\phi^{\gamma,c}\end{aligned} \tag{10}$$

$$\begin{aligned}\dot{\phi}^{i,abcd} &= g^{i,\alpha}\phi^{\alpha,abcd} + g^{i,\alpha\beta}(\phi^{\alpha,abc}\phi^{\beta,d} + \phi^{\alpha,abd}\phi^{\beta,c} + \phi^{\alpha,acd}\phi^{\beta,b} \\ &\quad + \phi^{\alpha,ab}\phi^{\beta,cd} + \phi^{\alpha,ac}\phi^{\beta,bd} + \phi^{\alpha,ad}\phi^{\beta,bc} + \phi^{\alpha,a}\phi^{\beta,bcd}) \\ &\quad + g^{i,\alpha\beta\gamma}(\phi^{\alpha,ab}\phi^{\beta,c}\phi^{\gamma,d} + \phi^{\alpha,ac}\phi^{\beta,b}\phi^{\gamma,d} + \phi^{\alpha,ad}\phi^{\beta,b}\phi^{\gamma,c} \\ &\quad + \phi^{\alpha,a}\phi^{\beta,bc}\phi^{\gamma,d} + \phi^{\alpha,a}\phi^{\beta,bd}\phi^{\gamma,c} + \phi^{\alpha,a}\phi^{\beta,b}\phi^{\gamma,cd}) \\ &\quad + g^{i,\alpha\beta\gamma\delta}\phi^{\alpha,a}\phi^{\beta,b}\phi^{\gamma,c}\phi^{\delta,d}\end{aligned} \tag{11}$$

The initial conditions of the STTs are $\phi_{(t_0,t_0)}^{i,a} = 1$ if $i = a$ and zero otherwise. Once these STTs are computed, they serve a role identical to the STM except that higher-order effects are now included, and thus the solution is nonlinear. Therefore, a significance of the STTs is that the local nonlinear motion of a spacecraft trajectory can be mapped analytically and requires no integration.

The inverse series also exists and is defined as

$$\delta x_0^i = \sum_{p=1}^{m} \frac{1}{p!} \psi_{(t_0,t_k)}^{i,\gamma_1\cdots\gamma_p} \delta x^{\gamma_1} \cdots \delta x^{\gamma_p} \tag{12}$$

where $\gamma_j \in \{1, \ldots, N\}$ and

$$\psi_{(t_0,t)}^{i,\gamma_1\cdots\gamma_p}(t, \mathbf{x}; t_0) = \left.\frac{\partial^p \psi^i(t, \boldsymbol{\xi}; t_0)}{\partial \xi^{\gamma_1} \cdots \partial \xi^{\gamma_p}}\right|_{\xi^j = x^j} \tag{13}$$

We call these higher-order partials the inverse state transition tensors (ISTTs). The ISTTs can be computed by using a similar integration approach as in the STT computation; however, it is more convenient to compute them via series reversion because numerical integration can be costly for large m. As functions of the STTs, the ISTTs mapping from t to t_0 are

$$\psi^{i,a} = [\boldsymbol{\Phi}^{-1}(t, t_0)]^{ia} \tag{14}$$

$$\psi^{i,ab} = -\psi^{i,\alpha}\phi^{\alpha,j_1 j_2}\psi^{j_1,a}\psi^{j_2,b} \tag{15}$$

$$\begin{aligned}\psi^{i,abc} &= -[\psi^{i,\alpha}\phi^{\alpha,j_1 j_2 j_3} + \psi^{i,\alpha\beta}(\phi^{\alpha,j_1}\phi^{\beta,j_2 j_3} + \phi^{\alpha,j_1 j_2}\phi^{\beta,j_3} \\ &\quad + \phi^{\alpha,j_1 j_3}\phi^{\beta,j_2})]\psi^{j_1,a}\psi^{j_2,b}\psi^{j_3,c}\end{aligned} \tag{16}$$

$$\begin{aligned}\psi^{i,abcd} &= -\,[\psi^{i,\alpha}\phi^{\alpha,j_1 j_2 j_3 j_4} + \psi^{i,\alpha\beta}(\phi^{\alpha,j_1 j_2 j_3}\phi^{\beta,j_4} + \phi^{\alpha,j_1 j_2 j_4}\phi^{\beta,j_3} \\ &\quad + \phi^{\alpha,j_1 j_3 j_4}\phi^{\beta,j_2} + \phi^{\alpha,j_1 j_2}\phi^{\beta,j_3 j_4} + \phi^{\alpha,j_1 j_3}\phi^{\beta,j_2 j_4} + \phi^{\alpha,j_1 j_4}\phi^{\beta,j_2 j_3} \\ &\quad + \phi^{\alpha,j_1}\phi^{\beta,j_2 j_3 j_4}) + \psi^{i,\alpha\beta\gamma}(\phi^{\alpha,j_1 j_2}\phi^{\beta,j_3}\phi^{\gamma,j_4} + \phi^{\alpha,j_1 j_3}\phi^{\beta,j_2}\phi^{\gamma,j_4} \\ &\quad + \phi^{\alpha,j_1 j_4}\phi^{\beta,j_2}\phi^{\gamma,j_3} + \phi^{\alpha,j_1}\phi^{\beta,j_2 j_3}\phi^{\gamma,j_4} + \phi^{\alpha,j_1}\phi^{\beta,j_2 j_4}\phi^{\gamma,j_3} \\ &\quad + \phi^{\alpha,j_1}\phi^{\beta,j_2}\phi^{\gamma,j_3 j_4})]\psi^{j_1,a}\psi^{j_2,b}\psi^{j_3,c}\psi^{j_4,d}\end{aligned} \tag{17}$$

where all indices are $1, \ldots, N$, $\boldsymbol{\psi} = \boldsymbol{\psi}_{(t_0,t)}$ and $\boldsymbol{\phi} = \boldsymbol{\phi}_{(t,t_0)}$ are used for the concise notations. Note that Eqs. (14–17) are analytic in the STTs and require no integration.

By applying the forward and inverse state transition tensors, the STTs mapping from time t_r to t_s, where $t_r, t_s \in [t_0, t_f]$ for some final time t_f and $t_r \leq t_s$, can be represented as

$$\phi_{(t_s,t_r)}^{i,a} = [\boldsymbol{\Phi}(t_s, t_0)\boldsymbol{\Phi}^{-1}(t_r, t_0)]^{ia} = \phi_s^{i,\alpha}\psi_r^{\alpha,a} \tag{18}$$

$$\phi_{(t_s,t_r)}^{i,ab} = \phi_s^{i,\alpha}\psi_r^{\alpha,ab} + \phi_s^{i,\alpha\beta}\psi_r^{\alpha,a}\psi_r^{\beta,b} \tag{19}$$

$$\begin{aligned}\phi_{(t_s,t_r)}^{i,abc} &= \phi_s^{i,\alpha}\psi_r^{\alpha,abc} + \phi_s^{i,\alpha\beta}\left(\psi_r^{\alpha,a}\psi_r^{\beta,bc} + \psi_r^{\alpha,ab}\psi_r^{\beta,c} + \psi_r^{\alpha,ac}\psi_r^{\beta,b}\right) \\ &\quad + \phi_s^{i,\alpha\beta\gamma}\psi_r^{\alpha,a}\psi_r^{\beta,b}\psi_r^{\gamma,c}\end{aligned} \tag{20}$$

$$\phi^{i,abcd}_{(t_s,t_r)} = \phi^{i,\alpha}_s \psi^{\alpha,abcd}_r + \phi^{i,\alpha\beta}_s \Big(\psi^{\alpha,abc}_r \psi^{\beta,d}_r + \psi^{\alpha,abd}_r \psi^{\beta,c}_r + \psi^{\alpha,acd}_r \psi^{\beta,b}_r + \psi^{\alpha,ab}_r \psi^{\beta,cd}_r + \psi^{\alpha,ac}_r \psi^{\beta,bd}_r + \psi^{\alpha,ad}_r \psi^{\beta,bc}_r + \psi^{\alpha,a}_r \psi^{\beta,bcd}_r \Big) + \phi^{i,\alpha\beta\gamma}_s \Big(\psi^{\alpha,ab}_r \psi^{\beta,c}_r \psi^{\gamma,d}_r + \psi^{\alpha,ac}_r \psi^{\beta,b}_r \psi^{\gamma,d}_r + \psi^{\alpha,ad}_r \psi^{\beta,b}_r \psi^{\gamma,c}_r + \psi^{\alpha,a}_r \psi^{\beta,bc}_r \psi^{\gamma,d}_r + \psi^{\alpha,a}_r \psi^{\beta,bd}_r \psi^{\gamma,c}_r + \psi^{\alpha,a}_r \psi^{\beta,b}_r \psi^{\gamma,cd}_r \Big) + \phi^{i,\alpha\beta\gamma\delta}_s \psi^{\alpha,a}_r \psi^{\beta,b}_r \psi^{\gamma,c}_r \psi^{\delta,d}_r \tag{21}$$

where all indices are $1,\ldots,N$, $\boldsymbol{\psi}_r = \psi_{(t_0,t_r)}$, $\boldsymbol{\phi}_s = \phi_{(t_s,t_0)}$, and the ISTTs are computed by applying Eqs. (14–17). In other words, once the STTs are computed for the entire reference trajectory $[t_0, t_f]$, the map from an arbitrary point in space to some future time becomes a simple algebraic manipulation. Note that $\phi^{i,\gamma_1\cdots\gamma_p}_{(t_s,t_r)}$ can also be computed by integrating the differential equations given in Eqs. (8–11) for each time interval (t_s, t_r).

One concern is numerical consistency when Eqs. (8–11) are integrated over a long duration of time. To address this, we note that the reference trajectory can be segmented arbitrarily to meet the desired numerical accuracy. Another question that may arise is the computational difficulty (or the long integration time) as we consider the higher-order solutions. Specifically, assuming a system with $N = 6$, the mth order analysis requires integration of $\sum_{q=1}^{m+1} 6^q$ equations. For example, when $m = 3$, one must integrate 1554 equations simultaneously. However, the higher-order solutions can be computed offline, and especially when the orbit is periodic (e.g., halo orbit), these only need to be computed once. Last, the computation of the partials of the dynamics may be of concern. Here, note that there are symbolic manipulators available which provide automatic differentiations, and also note that many of these partials vanish to zero for systems of spacecraft navigation interest. Once the partials are computed, these equations can be cast into a first-order differential equation form and can be integrated forward in time.

III. Higher-Order Extended Kalman Filters

Suppose we are given the continuous trajectory model defined in Eq. (3). Because a spacecraft tracking model is usually discrete, consider the following discrete system model:

$$\mathbf{x}_{k+1} = \boldsymbol{\phi}(t_{k+1}; \mathbf{x}_k, t_k) + \mathbf{w}_k \tag{22}$$

$$\mathbf{z}_{k+1} = \mathbf{h}(\mathbf{x}_{k+1}, t_{k+1}) + \mathbf{v}_{k+1} \tag{23}$$

where $\mathbf{x}_k$ is the true spacecraft state, $\mathbf{w}_k$ is the process noise perturbing the spacecraft state, $\mathbf{z}_k$ is the actual measurement, $\mathbf{h}$ is the measurement function, and $\mathbf{v}_k$ is the measurement noise characterizing the observation error. The process noise and measurement noise are assumed to be noncorrelated, that is, $E[\mathbf{v}_i \mathbf{w}_j^T] = 0$, with the autocorrelations:

$$E\Big[\mathbf{w}_i \mathbf{w}_j^T\Big] = \mathbf{Q}_i \delta_{ij} \tag{24}$$

$$E\Big[\mathbf{v}_i \mathbf{v}_j^T\Big] = \mathbf{R}_i \delta_{ij} \tag{25}$$

for all discrete time indexes i and j. Here, $\mathbf{Q}_i$ and $\mathbf{R}_i$ are also known as the diffusion and measurement noise matrices, respectively.

A. Kalman Filter

Although the Kalman filter algorithm can be derived from Bayes' rule of conditional densities, as pointed out by Julier and Uhlmann [6], the Kalman filter can also be derived from estimations of a few expectations involving a state and a measurement [7]. To show this, consider the system model equations (22) and (23) and suppose we are given a state $\mathbf{x}_k$ with mean $\mathbf{m}_k^+ = E[\mathbf{x}_k \mid \mathbf{z}_k]$ and covariance matrix $\mathbf{P}_k^+ = E[(\mathbf{x}_k - \mathbf{m}_k^+)(\mathbf{x}_k - \mathbf{m}_k^+)^T \mid \mathbf{z}_k]$ at time t_k. The general filtering algorithm can be defined as follows:

Prediction equations:

$$\mathbf{m}_{k+1}^- = E[\boldsymbol{\phi}(t_{k+1}; \mathbf{x}_k, t_k) + \mathbf{w}_k \mid \mathbf{z}_k] \tag{26}$$

$$\mathbf{P}_{k+1}^- = E\{[\boldsymbol{\phi}(t_{k+1}; \mathbf{x}_k, t_k) + \mathbf{w}_k][\boldsymbol{\phi}(t_{k+1}; \mathbf{x}_k, t_k) + \mathbf{w}_k]^T \mid \mathbf{z}_k\} - (\mathbf{m}_{k+1}^-)(\mathbf{m}_{k+1}^-)^T \tag{27}$$

$$\mathbf{n}_{k+1}^- = E[\mathbf{h}(\mathbf{x}_{k+1}, t_{k+1}) + \mathbf{v}_{k+1} \mid \mathbf{z}_k] \tag{28}$$

where $\mathbf{n}_{k+1}^- = E[\mathbf{h}_{k+1} \mid \mathbf{z}_k]$ is the expectation of the measurement computed at t_{k+1}.

Update equations:

$$\mathbf{K}_{k+1} = \mathbf{P}_{k+1}^{\mathbf{xz}} \Big(\mathbf{P}_{k+1}^{\mathbf{zz}}\Big)^{-1} \tag{29}$$

$$\mathbf{m}_{k+1}^+ = \mathbf{m}_{k+1}^- + \mathbf{K}_{k+1}\Big(\mathbf{z}_{k+1} - \mathbf{n}_{k+1}^-\Big) \tag{30}$$

$$\mathbf{P}_{k+1}^+ = \mathbf{P}_{k+1}^- - \mathbf{K}_{k+1} \mathbf{P}_{k+1}^{\mathbf{zz}} \mathbf{K}_{k+1}^T \tag{31}$$

where $\mathbf{K}_k$ is known as the Kalman gain matrix, $\mathbf{P}_k^{\mathbf{xz}}$ is the cross-covariance matrix of the state and the measurement, $\mathbf{P}_k^{\mathbf{zz}}$ is the covariance matrix of the measurement, $\mathbf{z}_k$ is the observation, and the difference between the actual and predicted measurement (i.e., $\mathbf{z}_k - \mathbf{n}_k^-$) is called the *residual* or *innovation*. When a linear dynamical system and Gaussian linear measurement function are considered, Eqs. (26–31) simplify to the conventional linear Kalman filter (LKF).

B. Extended Kalman Filter

For estimation problems, the LKF is probably the most well-known filtering technique. The LKF allows one to compute the minimum mean-square-error (MMSE) solution; however, it can only be used for linear systems, and in general, cannot be used for trajectory navigation. In conventional spacecraft trajectory navigation, the EKF is usually implemented. [In practice, the extended Kalman filter is implemented for trajectory navigation often in a square-root information filter (SRIF) or in U-D filter formulation for numerical precision.] The EKF is based on the filter algorithm given in Eqs. (26–31), but assumes the true trajectory is within the boundary where the linear approximation can sufficiently model the trajectory dynamics and its statistics. Under this assumption, the mean trajectory is propagated according to the deterministic solution flow and the covariance matrix is mapped linearly [8,9].

EKF prediction equations:

$$\mathbf{m}_{k+1}^- = \boldsymbol{\phi}\Big(t_{k+1}; \mathbf{m}_k^+, t_k\Big) \tag{32}$$

$$\mathbf{P}_{k+1}^- = \boldsymbol{\Phi}(t_{k+1}, t_k) \mathbf{P}_k^+ \boldsymbol{\Phi}^T(t_{k+1}, t_k) + \mathbf{Q}_k \tag{33}$$

$$\mathbf{n}_{k+1}^- = \mathbf{h}\Big(\mathbf{m}_{k+1}^-, t_{k+1}\Big) \tag{34}$$

EKF update equations:

$$\mathbf{K}_{k+1} = \mathbf{P}_{k+1}^{\mathbf{xz}} \Big(\mathbf{P}_{k+1}^{\mathbf{zz}}\Big)^{-1} = \mathbf{P}_{k+1}^- \mathbf{H}_{k+1}^T \Big(\mathbf{H}_{k+1} \mathbf{P}_{k+1}^- \mathbf{H}_{k+1}^T + \mathbf{R}_{k+1}\Big)^{-1} \tag{35}$$

$$\mathbf{m}_{k+1}^+ = \mathbf{m}_{k+1}^- + \mathbf{K}_{k+1}\Big(\mathbf{z}_{k+1} - \mathbf{n}_{k+1}^-\Big) \tag{36}$$

$$\mathbf{P}_{k+1}^+ = \mathbf{P}_{k+1}^- - \mathbf{K}_{k+1} \mathbf{P}_{k+1}^{\mathbf{zz}} \mathbf{K}_{k+1}^T = \mathbf{P}_{k+1}^- - \mathbf{K}_{k+1} \mathbf{H}_{k+1} \mathbf{P}_{k+1}^- \tag{37}$$

where $\mathbf{h}(\mathbf{m}_{k+1}^-, t_{k+1})$ is the measurement function evaluated at t_{k+1} as a function of $\mathbf{m}_{k+1}^-$ and $\mathbf{H}_{k+1} = \partial \mathbf{h}_{k+1} / \partial \mathbf{x}_{k+1}$ is the measurement partial computed at t_{k+1}. Note that the STM $\Phi(t_{k+1}, t_k)$ in Eq. (33), which is the first of the STTs, is computed along the previously updated mean trajectory, that is, $\mathbf{m}_k^+$.

Among the many important properties of the extended Kalman filter, we point out two which will be discussed in the example section in more detail. Considering the gain Eq. (35) and the mean update Eq. (36), we observe that as the a priori covariance matrix becomes more accurate (i.e., $\mathbf{P}_{k+1}^{-} \to \mathbf{0}$) the filter values the residual less (i.e., the actual measurement is trusted less). On the other hand, as the measurement becomes more accurate (i.e., $\mathbf{R}_{k+1} \to \mathbf{0}$) the filter values the residual more (i.e., the actual measurement is trusted more). Therefore, effective weighting of the residual is a critical component of maximizing the filter performance.

C. Higher-Order Numerical Extended Kalman Filter

In deriving the higher-order numerical extended Kalman filter (HNEKF), we assume that the reference trajectory and its higher-order state transition tensors are integrated for each time interval between the measurements according to Eqs. (8–11). Under this assumption the local trajectory motion can be mapped analytically over this time interval while incorporating nonlinear effects, and the same analogy applies when mapping the trajectory statistics. We note that this process is numerically quite intensive considering higher-order solutions; however, this can yield a more accurate filter solution.

Once the higher-order state transition tensors are available for some time interval $[t_k, t_{k+1}]$, the mean and covariance matrix of the relative dynamics at t_k can be mapped analytically to t_{k+1} as functions of the probability distribution at t_k. From t_k to t_{k+1}, the propagated mean and covariance can be stated as [4,5]

$$\delta m_{k+1}^{i}(\delta\mathbf{x}_k) = E\Big[\delta x_{k+1}^{i}\Big] = \sum_{p=1}^{m} \frac{1}{p!}\phi_{(t_{k+1},t_k)}^{i,\gamma_1\cdots\gamma_p} E\Big[\delta x_k^{\gamma_1}\cdots\delta x_k^{\gamma_p}\Big] \tag{38}$$

$$\begin{aligned} P_{k+1}^{ij}(\delta\mathbf{x}_k) &= E\Big[\Big(\delta x_{k+1}^{i} - \delta m_{k+1}^{i}\Big)\Big(\delta x_{k+1}^{j} - \delta m_{k+1}^{j}\Big)\Big] \\ &= \left(\sum_{p=1}^{m}\sum_{q=1}^{m}\frac{1}{p!q!}\phi_{(t_{k+1},t_k)}^{i,\gamma_1\cdots\gamma_p}\phi_{(t_{k+1},t_k)}^{j,\zeta_1\cdots\zeta_q} E\Big[\delta x_k^{\gamma_1}\cdots\delta x_k^{\gamma_p}\delta x_k^{\zeta_1}\cdots\delta x_k^{\zeta_q}\Big]\right) \\ &\quad - \delta m_{k+1}^{i}\delta m_{k+1}^{j} \end{aligned} \tag{39}$$

where $\{\gamma_j, \zeta_j\} \in \{1,\ldots,N\}$. As the order of the solution increases, that is, $m \to \infty$, the higher-order solution yields the true mean and covariance matrix computed from Monte Carlo simulations. Now, the only unknowns in Eqs. (38) and (39) are the expectations (i.e., moments) of the deviations. Even if the state at time t_k is Gaussian, except for the case $m = 1$, it is obvious that the mapped trajectory distribution is no longer Gaussian due to system nonlinearity, and hence exact computation requires computation of the higher-order moments.

In particle-based filters, this problem is remedied by using an ensemble of sample points to approximate the probability distribution, whereas a more formal approach is to use the Edgeworth/Gram-Chalier [10] or Laplace approximations to approximate the *posterior* density function. In trajectory navigation, however, the Gaussian assumption has shown to provide a sufficiently accurate statistical approximation. Hence, we assume that the updated estimates are Gaussian and we implement the joint characteristic function to compute the higher-order moments up to $2m$th order as apparent from Eq. (39). By assuming the updated state can be approximated with Gaussian statistics, the higher-order moments are functions of the first two moments. Moreover, if we consider a zero initial mean, all the odd moments of the initial conditions vanish and the equations for the propagated mean and covariance matrix simplify a great deal.

Now, suppose at time t_k, the state estimate has mean $\mathbf{m}_k^{+}$ and covariance matrix $\mathbf{P}_k^{+}$. Also, let $\mathbf{x}(t_k) = \mathbf{m}_k^{+} + \delta\mathbf{x}_k$ be the true trajectory we want to estimate. Following the Kalman filter algorithm, the HNEKF algorithm is given as follows:

HNEKF prediction equations:

$$\begin{aligned} \Big(m_{k+1}^{-}\Big)^{i} &= E\Big[\phi^{i}\Big(t_{k+1};\mathbf{m}_k^{+} + \delta\mathbf{x}_k, t_k\Big) + w_k^{i}\Big] = \phi^{i}\Big(t_{k+1};\mathbf{m}_k^{+}, t_k\Big) \\ &\quad + \delta m_{k+1}^{i}(\delta\mathbf{x}_k) = \phi^{i}\Big(t_{k+1};\mathbf{m}_k^{+}, t_k\Big) \\ &\quad + \sum_{p=1}^{m}\frac{1}{p!}\phi_{(t_{k+1},t_k)}^{i,\gamma_1\cdots\gamma_p} E\Big[\delta x_k^{\gamma_1}\cdots\delta x_k^{\gamma_p}\Big] \end{aligned} \tag{40}$$

$$\begin{aligned} \Big(P_{k+1}^{-}\Big)^{ij} &= E\Big\{\Big[\phi^{i}\Big(t_{k+1};\mathbf{m}_k^{+} + \delta\mathbf{x}_k, t_k\Big) + w_k^{i}\Big] \\ &\quad \times \Big[\phi^{j}\Big(t_{k+1};\mathbf{m}_k^{+} + \delta\mathbf{x}_k, t_k\Big) + w_k^{j}\Big]\Big\} - \Big(m_{k+1}^{-}\Big)^{i}\Big(m_{k+1}^{-}\Big)^{j} \end{aligned} \tag{41}$$

$$\begin{aligned} &= \left(\sum_{p=1}^{m}\sum_{q=1}^{m}\frac{1}{p!q!}\phi_{(t_{k+1},t_k)}^{i,\gamma_1\cdots\gamma_p}\phi_{(t_{k+1},t_k)}^{j,\zeta_1\cdots\zeta_q} E\Big[\delta x_k^{\gamma_1}\cdots\delta x_k^{\gamma_p}\delta x_k^{\zeta_1}\cdots\delta x_k^{\zeta_q}\Big]\right) \\ &\quad - \delta m_{k+1}^{i}(\delta\mathbf{x}_k)\delta m_{k+1}^{j}(\delta\mathbf{x}_k) + Q_k^{ij} \end{aligned} \tag{42}$$

$$\begin{aligned} \Big(n_{k+1}^{-}\Big)^{i} &= E\Big[h^{i}\Big(t_{k+1};\mathbf{m}_k^{+} + \delta\mathbf{x}_k, t_k\Big) + \mathbf{v}_{k+1}\Big] \\ &= h^{i}\Big(t_{k+1};\mathbf{m}_k^{+}, t_k\Big) + \delta n_{k+1}^{i}(\delta\mathbf{x}_k) = h^{i}\Big(t_{k+1};\mathbf{m}_k^{+}, t_k\Big) \\ &\quad + \sum_{p=1}^{m}\frac{1}{p!}h_{(t_{k+1},t_k)}^{i,\gamma_1\cdots\gamma_p} E\Big[\delta x_k^{\gamma_1}\cdots\delta x_k^{\gamma_p}\Big] \end{aligned} \tag{43}$$

where the STTs [i.e., $\phi^{i}(t_{k+1};\mathbf{m}_k^{+}, t_k)$] are computed along the solution flow $\phi(t_{k+1};\mathbf{m}_k^{+}, t_k)$ and

$$h_{(t_{k+1},t_k)}^{i,\gamma_1\cdots\gamma_p} = \left.\frac{\partial^{p} h_{k+1}^{i}}{\partial x_k^{\gamma_1}\cdots\partial x_k^{\gamma_p}}\right|_{\mathbf{x}_{k+1}=\phi(t_{k+1};\mathbf{m}_k^{+},t_k)} \tag{44}$$

Note that $h^{i}(t_{k+1};\mathbf{m}_k^{+}, t_k)$ denotes that the measurement function is evaluated at t_{k+1} as a function of the solution flow $\phi(t_{k+1};\mathbf{m}_k^{+}, t_k)$. [It is important to note that $\mathbf{h}(t_{k+1};\mathbf{m}_k^{+}, t_k) \neq \mathbf{h}(\mathbf{m}_{k+1}^{-}, t_{k+1})$ in general because $\mathbf{m}_{k+1}^{-} \neq \phi(t_{k+1};\mathbf{m}_k^{+}, t_k)$ for general nonlinear systems.] The partial derivatives $h_{(t_{k+1},t_k)}^{i,\gamma_1\cdots\gamma_p}$ up to fourth order are defined as

$$h_{(t_{k+1},t_k)}^{i,a} = h_{k+1}^{i,\alpha}\phi_{k+1}^{\alpha,a} \tag{45}$$

$$h_{(t_{k+1},t_k)}^{i,ab} = h_{k+1}^{i,\alpha}\phi_{k+1}^{\alpha,ab} + h_{k+1}^{i,\alpha\beta}\phi_{k+1}^{\alpha,a}\phi_{k+1}^{\beta,b} \tag{46}$$

$$\begin{aligned} h_{(t_{k+1},t_k)}^{i,abc} &= h_{k+1}^{i,\alpha}\phi_{k+1}^{\alpha,abc} + h_{k+1}^{i,\alpha\beta}\Big(\phi_{k+1}^{\alpha,a}\phi_{k+1}^{\beta,bc} + \phi_{k+1}^{\alpha,ab}\phi_{k+1}^{\beta,c} + \phi_{k+1}^{\alpha,ac}\phi_{k+1}^{\beta,b}\Big) \\ &\quad + h_{k+1}^{i,\alpha\beta\gamma}\phi_{k+1}^{\alpha,a}\phi_{k+1}^{\beta,b}\phi_{k+1}^{\gamma,c} \end{aligned} \tag{47}$$

$$\begin{aligned} h_{(t_{k+1},t_k)}^{i,abcd} &= h_{k+1}^{i,\alpha}\phi_{k+1}^{\alpha,abcd} + h_{k+1}^{i,\alpha\beta}\Big(\phi_{k+1}^{\alpha,abc}\phi_{k+1}^{\beta,d} + \phi_{k+1}^{\alpha,abd}\phi_{k+1}^{\beta,c} \\ &\quad + \phi_{k+1}^{\alpha,acd}\phi_{k+1}^{\beta,b} + \phi_{k+1}^{\alpha,ab}\phi_{k+1}^{\beta,cd} + \phi_{k+1}^{\alpha,ac}\phi_{k+1}^{\beta,bd} + \phi_{k+1}^{\alpha,ad}\phi_{k+1}^{\beta,bc} \\ &\quad + \phi_{k+1}^{\alpha,a}\phi_{k+1}^{\beta,bcd}\Big) + h_{k+1}^{i,\alpha\beta\gamma}\Big(\phi_{k+1}^{\alpha,ab}\phi_{k+1}^{\beta,c}\phi_{k+1}^{\gamma,d} + \phi_{k+1}^{\alpha,ac}\phi_{k+1}^{\beta,b}\phi_{k+1}^{\gamma,d} \\ &\quad + \phi_{k+1}^{\alpha,ad}\phi_{k+1}^{\beta,b}\phi_{k+1}^{\gamma,c} + \phi_{k+1}^{\alpha,a}\phi_{k+1}^{\beta,bc}\phi_{k+1}^{\gamma,d} + \phi_{k+1}^{\alpha,a}\phi_{k+1}^{\beta,bd}\phi_{k+1}^{\gamma,c} \\ &\quad + \phi_{k+1}^{\alpha,a}\phi_{k+1}^{\beta,b}\phi_{k+1}^{\gamma,cd}\Big) + h_{k+1}^{i,\alpha\beta\gamma\delta}\phi_{k+1}^{\alpha,a}\phi_{k+1}^{\beta,b}\phi_{k+1}^{\gamma,c}\phi_{k+1}^{\delta,d} \end{aligned} \tag{48}$$

where $\phi_{k+1} = \phi_{(t_{k+1},t_k)}$ is used for a concise notation and that these are similar to the differential equations of the STTs given in Eqs. (8–11). Note that this prediction step is a simple algebraic operation once the STTs are computed for the time interval $[t_k, t_{k+1}]$.

HNEKF update equations:

$$
\begin{aligned}
\left(P^{\mathbf{zz}}_{k+1}\right)^{ij} &= E\left[\left(\mathbf{z}^-_{k+1}-\mathbf{n}^-_{k+1}\right)\left(\mathbf{z}^-_{k+1}-\mathbf{n}^-_{k+1}\right)^T\right]^{ij}\\
&= E\left[\left(z^-_{k+1}\right)^i\left(z^-_{k+1}\right)^j\right]-\left(n^-_{k+1}\right)^i\left(n^-_{k+1}\right)^j\\
&= E\left\{\left[h^i\left(t_{k+1};\mathbf{m}^+_k+\delta\mathbf{x}_k,t_k\right)+v^i_{k+1}\right]\right.\\
&\quad\times\left.\left[h^j\left(t_{k+1};\mathbf{m}^+_k+\delta\mathbf{x}_k,t_k\right)+v^j_{k+1}\right]\right\}-\left(n^-_{k+1}\right)^i\left(n^-_{k+1}\right)^j\\
&= \left(R^{ij}_{k+1}+\sum_{p=1}^m\sum_{q=1}^m\frac{1}{p!q!}h^{i,\gamma_1\cdots\gamma_p}_{(t_{k+1},t_k)}h^{j,\zeta_1\cdots\zeta_q}_{(t_{k+1},t_k)}\right.\\
&\quad\times\left.E\left[\delta x^{\gamma_1}_k\cdots\delta x^{\gamma_p}_k\delta x^{\zeta_1}_k\cdots\delta x^{\zeta_q}_k\right]\right)-\left(\delta n^-_{k+1}\right)^i\left(\delta n^-_{k+1}\right)^j
\end{aligned}\tag{49}
$$

$$
\begin{aligned}
\left(P^{\mathbf{xz}}_{k+1}\right)^{ij} &= E\left[\left(\mathbf{x}^-_{k+1}-\mathbf{m}^-_{k+1}\right)\left(\mathbf{z}^-_{k+1}-\mathbf{n}^-_{k+1}\right)^T\right]^{ij}\\
&= E\left[\left(x^-_{k+1}\right)^i\left(z^-_{k+1}\right)^j\right]-\left(m^-_{k+1}\right)^i\left(n^-_{k+1}\right)^j\\
&= E\left\{\left[\phi^i\left(t_{k+1};\mathbf{m}^+_k+\delta\mathbf{x}_k,t_k\right)+w^i_k\right]\left[h^j\left(t_{k+1};\mathbf{m}^+_k+\delta\mathbf{x}_k,t_k\right)\right.\right.\\
&\quad+\left.\left.v^j_{k+1}\right]\right\}-\left(m^-_{k+1}\right)^i\left(n^-_{k+1}\right)^j\\
&= \left(\sum_{p=1}^m\sum_{q=1}^m\frac{1}{p!q!}\phi^{i,\zeta_1\cdots\zeta_q}_{(t_{k+1},t_k)}h^{j,\gamma_1\cdots\gamma_p}_{(t_{k+1},t_k)}E\left[\delta x^{\gamma_1}_k\cdots\delta x^{\gamma_p}_k\delta x^{\zeta_1}_k\cdots\delta x^{\zeta_q}_k\right]\right)\\
&\quad-\left(\delta m^-_{k+1}\right)^i\left(\delta n^-_{k+1}\right)^j
\end{aligned}\tag{50}
$$

$$
\mathbf{K}_{k+1}=\mathbf{P}^{\mathbf{xz}}_{k+1}\left(\mathbf{P}^{\mathbf{zz}}_{k+1}\right)^{-1}\tag{51}
$$

$$
\mathbf{m}^+_{k+1}=\mathbf{m}^-_{k+1}+\mathbf{K}_{k+1}\left(\mathbf{z}_{k+1}-\mathbf{n}^-_{k+1}\right)\tag{52}
$$

$$
\mathbf{P}^+_{k+1}=\mathbf{P}^-_{k+1}-\mathbf{K}_{k+1}\mathbf{P}^{\mathbf{zz}}_{k+1}\mathbf{K}^T_{k+1}\tag{53}
$$

Note that if we consider the measurement function Eq. (23) to be linear in $\mathbf{x}_k$, Eq. (43) simplifies to

$$
\begin{aligned}
\left(n^-_{k+1}\right)^i &= h^i\left(t_{k+1};\mathbf{m}^+_k,t_k\right)+\left(\delta n^-_{k+1}\right)^i = h^i\left(t_{k+1};\mathbf{m}^+_k,t_k\right)\\
&\quad+h^{i,\alpha}_{k+1}\sum_{p=1}^m\frac{1}{p!}\phi^{\alpha,\gamma_1\cdots\gamma_p}_{(t_{k+1},t_k)}E\left[\delta x^{\gamma_1}_k\cdots\delta x^{\gamma_p}_k\right]=h^i\left(t_{k+1};\mathbf{m}^+_k,t_k\right)\\
&\quad+h^{i,\alpha}_{k+1}\left(\delta m^-_{k+1}\right)^\alpha=h^i\left(\mathbf{m}^-_{k+1},t_{k+1}\right)
\end{aligned}\tag{54}
$$

and gives $(\delta n^-_{k+1})^i=h^{i,\alpha}_{k+1}(\delta m^-_{k+1})^\alpha$. Applying this result, Eqs. (49) and (50) simplify to

$$
\begin{aligned}
\left(P^{\mathbf{zz}}_{k+1}\right)^{ij} &= \left(h^{i,\alpha}_{k+1}h^{j,\beta}_{k+1}\sum_{p=1}^m\sum_{q=1}^m\frac{1}{p!q!}\phi^{\alpha,\gamma_1\cdots\gamma_p}_{(t_{k+1},t_k)}\phi^{\beta,\zeta_1\cdots\zeta_q}_{(t_{k+1},t_k)}\right.\\
&\quad\times\left.E\left[\delta x^{\gamma_1}_k\cdots\delta x^{\gamma_p}_k\delta x^{\zeta_1}_k\cdots\delta x^{\zeta_q}_k\right]\right)+R^{ij}_{k+1}-\left(\delta n^-_{k+1}\right)^i\left(\delta n^-_{k+1}\right)^j\\
&= \left(R^{ij}_{k+1}+h^{i,\alpha}_{k+1}h^{j,\beta}_{k+1}E\left[\delta x^\alpha_{k+1}\delta x^\beta_{k+1}\right]\right)-\left(\delta n^-_{k+1}\right)^i\left(\delta n^-_{k+1}\right)^j\\
&= \left(\mathbf{H}_{k+1}\mathbf{P}^-_{k+1}\mathbf{H}^T_{k+1}+\mathbf{R}_{k+1}\right)^{ij}
\end{aligned}\tag{55}
$$

$$
\begin{aligned}
\left(P^{\mathbf{xz}}_{k+1}\right)^{ij} &= \left(h^{j,\alpha}_{k+1}\sum_{p=1}^m\sum_{q=1}^m\frac{1}{p!q!}\phi^{i,\gamma_1\cdots\gamma_p}_{(t_{k+1},t_k)}\phi^{\alpha,\zeta_1\cdots\zeta_q}_{(t_{k+1},t_k)}\right.\\
&\quad\times\left.E\left[\delta x^{\gamma_1}_k\cdots\delta x^{\gamma_p}_k\delta x^{\zeta_1}_k\cdots\delta x^{\zeta_q}_k\right]\right)-\left(\delta m^-_{k+1}\right)^i\left(\delta n^-_{k+1}\right)^j\\
&= E\left[\delta x^i_{k+1}\delta x^\alpha_{k+1}\right]h^{j,\alpha}_{k+1}-\left(\delta m^-_{k+1}\right)^i\left(\delta n^-_{k+1}\right)^j=\left(\mathbf{P}^-_{k+1}\mathbf{H}^T_{k+1}\right)^{ij}
\end{aligned}\tag{56}
$$

which indicates that the *measurement* prediction and update equations are identical to the EKF algorithm. Also note that when $m=1$, the HNEKF becomes the EKF algorithm as shown in Eqs. (32–37).

D. Higher-Order Analytic Extended Kalman Filter

From the derivation of the HNEKF, it is obvious that we can also derive a higher-order analytic extended Kalman filter (HAEKF) by assuming that the reference trajectory and the higher-order solutions (i.e., STTs) are computed over some time span before filtering. The filter algorithm is similar to the HNEKF except that the point of series expansion is now with respect to the initial reference trajectory, not the updated mean as in the HNEKF algorithm.

Suppose the STTs are computed for the time interval of $[t_0,t_f]$ and let $\bar{\mathbf{x}}_k=\phi(t_k;\mathbf{x}_0,t_0)$ represent the reference trajectory for $t_k\in[t_0,t_f]$, where $\mathbf{x}_0$ has mean $\mathbf{m}^+_0$ and covariance matrix $\mathbf{P}^+_0$. Moreover, let $\mathbf{x}(t_k)=\bar{\mathbf{x}}_k+\delta\mathbf{x}_k$ be the true trajectory we want to estimate. Following the Kalman filter algorithm, the HAEKF algorithm is given as follows:

HAEKF prediction equations:

$$
\left(m^-_{k+1}\right)^i=\phi^i\left(t_{k+1};\bar{\mathbf{x}}_k,t_k\right)+\sum_{p=1}^m\frac{1}{p!}\phi^{i,\gamma_1\cdots\gamma_p}_{(t_{k+1},t_k)}E\left[\delta x^{\gamma_1}_k\cdots\delta x^{\gamma_p}_k\right]\tag{57}
$$

$$
\begin{aligned}
\left(P^-_{k+1}\right)^{ij} &= \left(\sum_{p=1}^m\sum_{q=1}^m\frac{1}{p!q!}\phi^{i,\gamma_1\cdots\gamma_p}_{(t_{k+1},t_k)}\phi^{j,\zeta_1\cdots\zeta_q}_{(t_{k+1},t_k)}\right.\\
&\quad\times\left.E\left[\delta x^{\gamma_1}_k\cdots\delta x^{\gamma_p}_k\delta x^{\zeta_1}_k\cdots\delta x^{\zeta_q}_k\right]\right)-\delta m^i_{k+1}(\delta\mathbf{x}_k)\delta m^j_{k+1}(\delta\mathbf{x}_k)+Q^{ij}_k
\end{aligned}\tag{58}
$$

$$
(n^-_{k+1})^i=h^i(\bar{\mathbf{x}}_{k+1},t_{k+1})+\sum_{p=1}^m\frac{1}{p!}h^{i,\gamma_1\cdots\gamma_p}_{(t_{k+1},t_k)}E\left[\delta x^{\gamma_1}_k\cdots\delta x^{\gamma_p}_k\right]\tag{59}
$$

where the STTs are computed along $\bar{\mathbf{x}}_{k+1}=\phi(t_{k+1};\bar{\mathbf{x}}_k,t_k)$ and

$$
h^{i,\gamma_1\cdots\gamma_p}_{(t_{k+1},t_k)}=\left.\frac{\partial^p h^i_{k+1}}{\partial x^{\gamma_1}_k\cdots\partial x^{\gamma_p}_k}\right|_{\mathbf{x}_{k+1}=\bar{\mathbf{x}}_{k+1}}\tag{60}
$$

HAEKF update equations:

$$
\begin{aligned}
\left(P^{\mathbf{zz}}_{k+1}\right)^{ij} &= \left(R^{ij}_{k+1}+\sum_{p=1}^m\sum_{q=1}^m\frac{1}{p!q!}h^{i,\gamma_1\cdots\gamma_p}_{(t_{k+1},t_k)}h^{j,\zeta_1\cdots\zeta_q}_{(t_{k+1},t_k)}\right.\\
&\quad\times\left.E\left[\delta x^{\gamma_1}_k\cdots\delta x^{\gamma_p}_k\delta x^{\zeta_1}_k\cdots\delta x^{\zeta_q}_k\right]\right)-\left(\delta n^-_{k+1}\right)^i\left(\delta n^-_{k+1}\right)^j
\end{aligned}\tag{61}
$$

$$
\begin{aligned}
\left(P^{\mathbf{xz}}_{k+1}\right)^{ij} &= \left(\sum_{p=1}^m\sum_{q=1}^m\frac{1}{p!q!}\phi^{i,\zeta_1\cdots\zeta_q}_{(t_{k+1},t_k)}h^{j,\gamma_1\cdots\gamma_p}_{(t_{k+1},t_k)}\right.\\
&\quad\times\left.E\left[\delta x^{\gamma_1}_k\cdots\delta x^{\gamma_p}_k\delta x^{\zeta_1}_k\cdots\delta x^{\zeta_q}_k\right]\right)-\left(\delta m^-_{k+1}\right)^i\left(\delta n^-_{k+1}\right)^j
\end{aligned}\tag{62}
$$

$$
\mathbf{K}_{k+1}=\mathbf{P}^{\mathbf{xz}}_{k+1}\left(\mathbf{P}^{\mathbf{zz}}_{k+1}\right)^{-1}\tag{63}
$$

$$
\mathbf{m}^+_{k+1}=\mathbf{m}^-_{k+1}+\mathbf{K}_{k+1}\left(\mathbf{z}_{k+1}-\mathbf{n}^-_{k+1}\right)\tag{64}
$$

$$
\mathbf{P}^+_{k+1}=\mathbf{P}^-_{k+1}-\mathbf{K}_{k+1}\mathbf{P}^{\mathbf{zz}}_{k+1}\mathbf{K}^T_{k+1}\tag{65}
$$

As in the HNEKF case, the update equations for the HAEKF become the same as the EKF when we consider a measurement function that is linear in $\mathbf{x}_k$. Also, note that when $m=1$ (i.e., first order), the HAEKF becomes the *linear* Kalman filter, not the EKF. (We call it the higher-order analytic *extended* Kalman filter, not the higher-order linear Kalman filter, because the prediction equations are nonlinear in general.) The superiority of the EKF over the LKF is clearly demonstrated by Maybeck [10]. However, when the true trajectory is within the convergence radius of the reference trajectory, we shall see later that the HAEKF can provide a more accurate solution and faster convergence than the EKF. An advantage of the HAEKF is that it can be precomputed and used online with no numerical integrations.

E. Unscented Kalman Filter

The unscented Kalman filter (UKF), first introduced by Julier and Uhlmann, is being implemented in a diverse field of engineering, science, and economics due to its simplicity while providing faster convergence and better accuracy than the extended Kalman filter. The UKF is initialized with a set of predetermined sigma points and incorporates the second-order trajectory information in the filter model. Detailed derivation and algorithm can be found in [6,11–14].

IV. Examples

In this section, we present several simulations of a halo orbit, which is a periodic orbit where the in-plane and out-of-plane frequencies are the same, computed based on the CR3BP. The governing equations of motion for CR3BP, in nondimensional form, are given as [15]

$$\ddot{\mathrm{x}} - 2\dot{\mathrm{y}} = \frac{\partial U}{\partial \mathrm{x}} \tag{66}$$

$$\ddot{\mathrm{y}} + 2\dot{\mathrm{x}} = \frac{\partial U}{\partial \mathrm{y}} \tag{67}$$

$$\ddot{\mathrm{z}} = \frac{\partial U}{\partial \mathrm{z}} \tag{68}$$

where

$$U = \frac{(1-\mu)}{r_1} + \frac{\mu}{r_2} + \frac{(\mathrm{x}^2 + \mathrm{y}^2)}{2} \tag{69}$$

$$r_1 = [(\mathrm{x} + \mu)^2 + \mathrm{y}^2 + \mathrm{z}^2]^{1/2} \tag{70}$$

$$r_2 = [(\mathrm{x} - 1 + \mu)^2 + \mathrm{y}^2 + \mathrm{z}^2]^{1/2} \tag{71}$$

Here, U is the CR3BP potential and $\mu = \mu_\oplus/(\mu_S + \mu_\oplus)$.

Consider a halo orbit about the sun–Earth L_1 point in a nondimensionalized frame, which can be dimensionalized by applying the length scale of $\ell = 1$ AU $= 1.49597870691 \times 10^8$ km, where AU stands for "astronomical unit," and the time scale of $\tau = 1/\omega_E$. Figure 1 shows the reference (nominal) trajectory for one orbital period (~177.86 days), which corresponds to case 1 given in Table 1. The initial conditions for these orbits are (in nondimensional units)

$$\mathbf{r}_{\mathrm{case\ 1}}(t_0) = [\,0.988884102845168,\ 0.0,\ 0.000921858528329094\,]^T$$

$$\mathbf{v}_{\mathrm{case\ 1}}(t_0) = [\,0.0,\ 0.00893471471659142,\ 0.0\,]^T$$

$$\mathbf{r}_{\mathrm{case\ 2}}(t_0) = [\,0.98888423093423,\ 0.0,\ 0.000929261736280955\,]^T$$

$$\mathbf{v}_{\mathrm{case\ 2}}(t_0) = [\,0.0,\ 0.00893688204973967,\ 0.0\,]^T$$

For the measurement model, we assume a simple linear model where only the y coordinate is observed, that is,

$$z_k = y_k + v_k \tag{72}$$

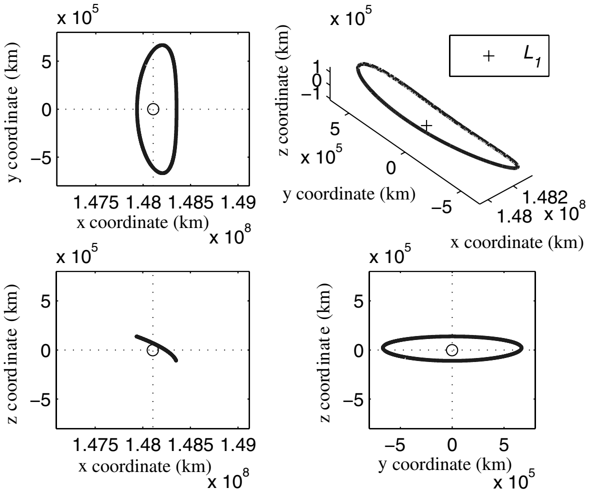


Fig. 1 Nominal halo orbit about the sun–Earth L_1 point.

where y_k represents the vertical position component of the state vector and v_k represents the measurement error. This measurement model can be viewed as a range measurement obtained by optical imaging of the Earth relative to distant stars or a very long baseline interferometry (VLBI) measurement. The measurement noise is assumed to be 0.1 m for each range measurement. This linear assumption simplifies the problem a great deal because the measurement sensitivity does not require the computation of the higher-order partials. This way, it is easier to understand the effect of the nonlinear orbit uncertainty propagation on filter performance.

Initially, the spacecraft state is assumed to be a zero mean Gaussian with position uncertainties of 100 km and velocity uncertainties of 0.1 m/s. [The initial covariance matrix is a diagonal matrix with $(100\ \mathrm{km})^2$ and $(0.1\ \mathrm{m/s})^2$ for position components and velocity components, respectively.] The initial mean and covariance matrix are mapped using the STT approach for $m = \{1, 3\}$, unscented transformation, and Monte Carlo simulations based on 10^6 sample points. Figure 2 shows the mean and the projection of the 1-σ covariance matrix onto the x–y plane after one orbital period. Assuming the Monte Carlo simulation is the true solution, the result shows that the third-order solution is the most accurate approximation, whereas the linear solution fails to characterize the orbit uncertainty distribution.

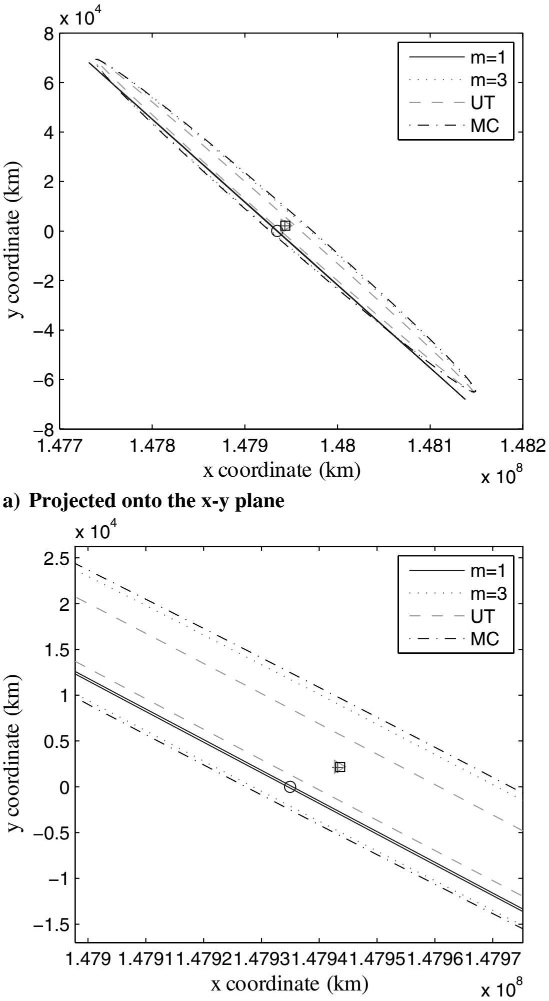
We now consider the same initial uncertainties, but assume the initial guess (mean) is off by 100 km for the position components and 0.1 m/s for the velocity components so that they lie on the boundary of the initial 1-σ ellipsoid. A set of pseudomeasurements are computed based on the reference trajectory with a 20-day increment. Using the *same* measurements, the initial mean and covariance matrix are mapped and solved using the EKF, the UKF, the third-order HNEKF, and the third-order HAEKF. For the HAEKF, because the trajectory is periodic, the STTs are computed and stored for only *one* orbital period, which is divided into two segments for numerical consistency, and reversion of the series is applied to map states analytically. [Note that all filter simulations are based on single runs where the same set of pseudomeasurements are considered (i.e., random noises added to perfect measurements). We have also simulated many different set of pseudomeasurements on the side and have obtained negligible difference in the filter performance given the state and measurement uncertainties considered in this paper.]

Table 1 Halo orbit maximum amplitudes with respect to the sun–Earth L_1 point

Cases	A_x, km	A_y, km	A_z, km
1	245,924	668,228	137,908
2	246,069	668,416	139,015

Figure 3 shows the a priori (predicted) and a posteriori (updated) position and velocity root–sum–square errors, where $\sigma_R = \sqrt{\sigma_{\mathrm{xx}} + \sigma_{\mathrm{yy}} + \sigma_{\mathrm{zz}}}$ and $\sigma_V = \sqrt{\sigma_{\mathrm{uu}} + \sigma_{\mathrm{vv}} + \sigma_{\mathrm{ww}}}$, and σ_{ii} represents the (i, i) component of the covariance matrix. A sudden drop in the uncertainties right after 100 days is due to the fact that the initial covariance matrix is quite large and requires at least six independent measurements to obtain a well-defined (i.e., reduced to the measurement noise level in all directions) a posteriori covariance matrix. The result shows that the EKF overestimates the uncertainties (i.e., assumes they are smaller than they are in actuality) while the UKF, HNEKF, and HAEKF provide conservative uncertainty estimates.

Figure 4 shows the magnitude of the absolute position and velocity errors, that is, the magnitude of the difference between the updated mean and the true state. The result shows that the EKF does not perform well as compared to the higher-order filters. This clearly explains the importance of nonlinear orbit uncertainty propagation. The covariance matrix computed by using the first-order method (i.e., EKF) overestimates the solution, and hence, the residual is trusted less. On the other hand, the UKF and the higher-order filters predict more conservative uncertainties and more effectively balance the a priori uncertainties and the actual measurements (i.e., measurements are valued more than the a priori information in this case). Figures 5 and 6 are based on the same filter setup except that the measurements are updated every 5 days. It shows that there is not much difference in the propagated uncertainties, but the absolute errors are computed more accurately in UKF and higher-order filter runs.



a) Projected onto the x-y plane

b) Larger plot of (a)

Fig. 2 Sun–Earth halo orbit: mean and 1-σ error ellipsoid projected onto the position plane after being propagated for one orbital period.

Figures 7 and 8 show the HNEKF results for cases $m \in \{1, 2, 3\}$. As mentioned earlier, note that the case $m = 1$ is identical to the EKF formulation. The result shows that the higher-order filters, $m \in \{2, 3\}$, provide superior filter performance over the first-order case and it is observed that the second-order effect contains most of the system nonlinearity, indicating that the second-order filter is sufficient for an accurate nonlinear filter in our example. Figures 9 and 10 show the HAEKF uncertainties and absolute error plots, respectively, for $m \in \{1, 2, 3\}$. The uncertainties for $m = 1$ are similar to the EKF solution and for $m = 2$ are similar to the case $m = 3$ as shown in Fig. 3. The absolute error plot shows that all three filters provide good estimation performance even for the case $m = 1$. This is expected because the pseudomeasurements are computed based on the reference trajectory which the STTs are computed based on. In other words, the reference trajectory can be thought of as a regression solution for the simulated measurements.

To analyze the robustness of the higher-order filtering techniques, the pseudomeasurements are now generated from the case 2 halo orbit given in Table 1. Figures 11 and 12 show the simulated filter solutions. The results show that the higher-order solutions are superior over the linear filters, that is, EKF and HAEKF for $m = 1$. As expected, this indicates that the linear Kalman filter is only feasible when the reference trajectory is sufficiently close to the true trajectory. The HAEKFs for $m > 1$, however, have more flexibility in

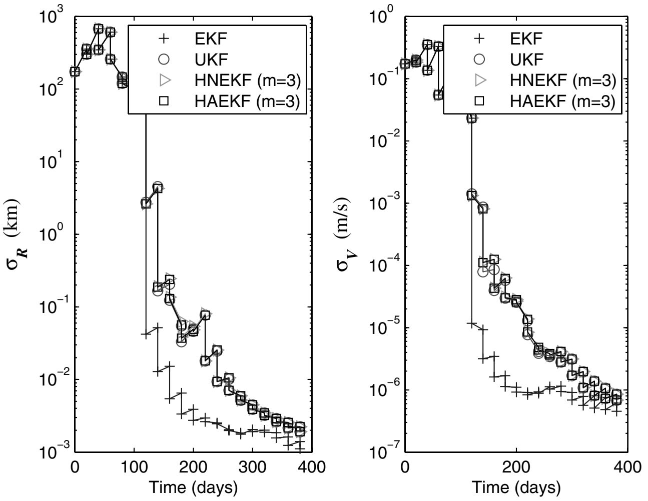


Fig. 3 Comparison of the root–sum–square errors with a 20-day increment measurement update.

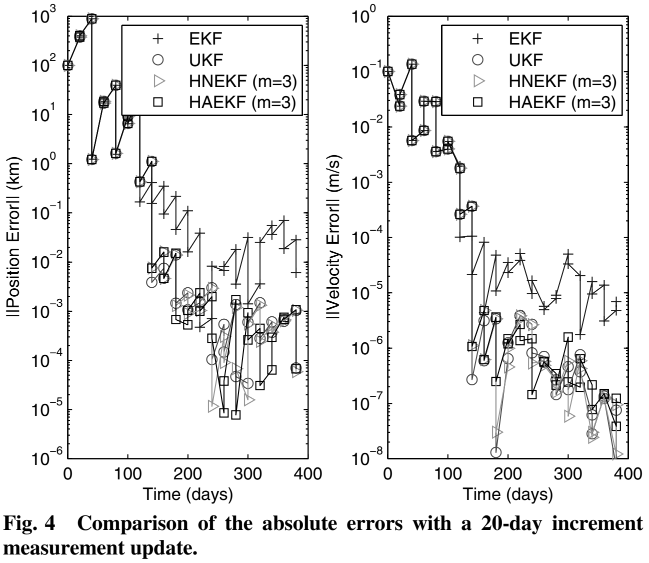


Fig. 4 Comparison of the absolute errors with a 20-day increment measurement update.

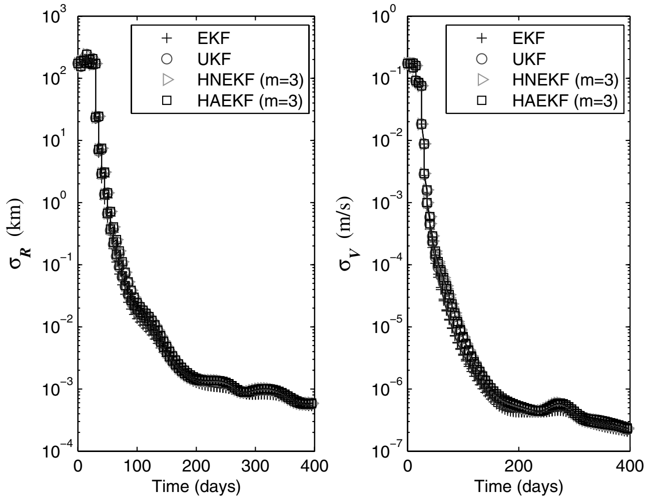


Fig. 5 Comparison of the root–sum–square errors with a 5-day increment measurement update.

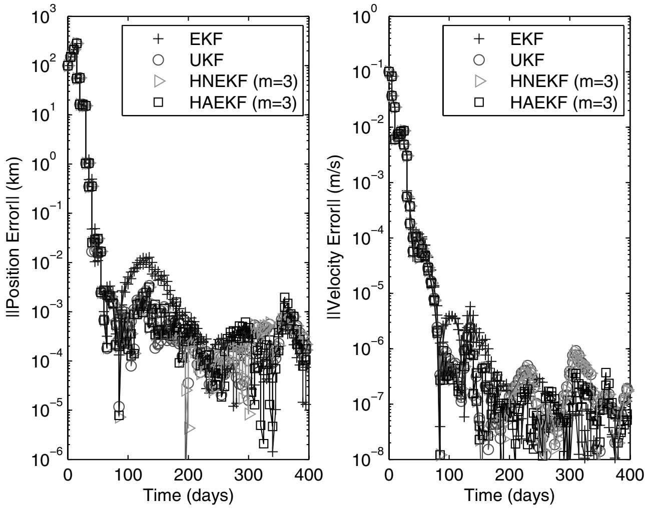


Fig. 6 Comparison of the absolute errors with a 5-day increment measurement update.

the reference trajectory. The overall filter convergence is slightly slower than the previous cases because the initial mean is assumed to be the same as in the previous cases, and thus, it is farther away from the true trajectory (i.e., the trajectory in which the pseudomeasurements are generated).

In this study, the EKF required integration of $N + N^2 = 42$ equations [or $N + N(N + 1)/2 = 27$ equations if the covariance matrix is directly integrated] and the UKF required integration of $(2N + 1)N = 78$ equations between each measurement update, and in the actual filter runs, the EKF was slightly faster than the UKF. The HNEKFs for $m > 1$ provide superior results over the linear filters (even UKF when $m > 2$); however, the computational load increases significantly as m increases. For example, the third-order HNEKF requires integration of 1554 equations. On the other hand, the HAEKF does *not* require any integration in the actual filtering process. The most expensive numerical operation in the HAEKF is the higher-order moment computation; however, there exist various techniques for efficient computation of moments. Hence, for missions with predetermined reference trajectories, the higher-order *analytic* filter may be suitable for the trajectory navigation while obtaining faster convergence and a more accurate filter solution than the EKF.

To be more specific, having the higher-order semi-analytic method onboard a spacecraft or a launch vehicle provides a complete solution space for the given reference trajectory depending on the order of solution. Hence, for problems where the nonlinearity is significant and requires a rapid convergence, such as spacecraft launch, planetary/small-body orbit insertion, or autonomous precision landing, applying the HAEKF can result in a more rapid, accurate, and robust state estimation than the EKF.

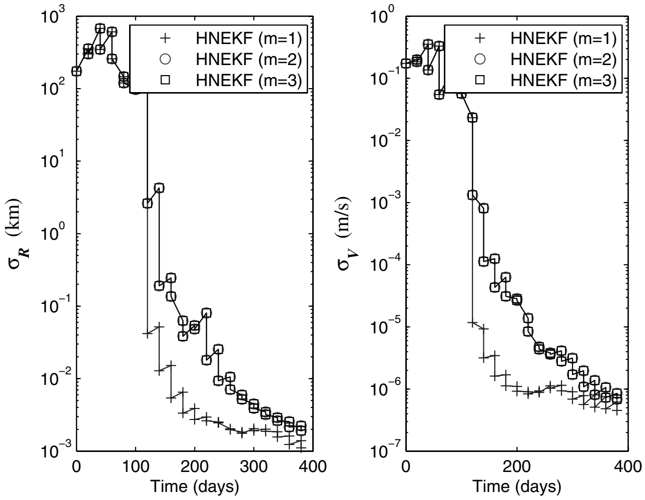


Fig. 7 Comparison of the root–sum–square errors with a 20-day increment measurement update.

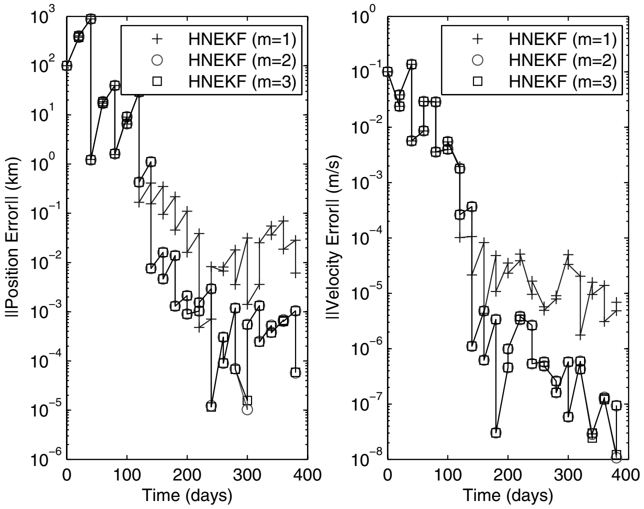


Fig. 8 Comparison of the absolute errors with a 20-day increment measurement update.

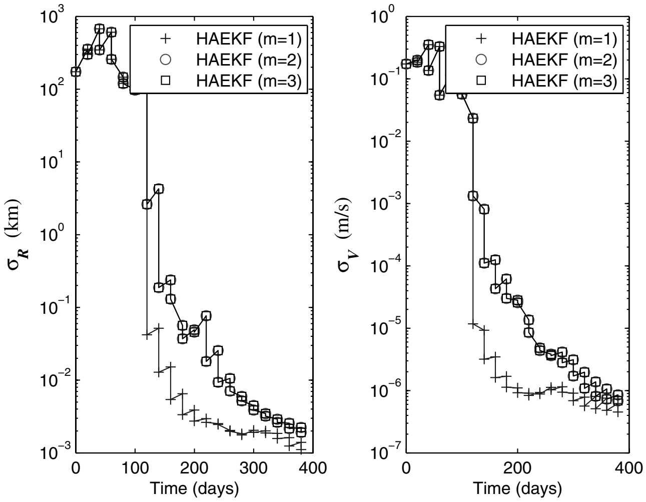


Fig. 9 Comparison of the root–sum–square errors with a 20-day increment measurement update.

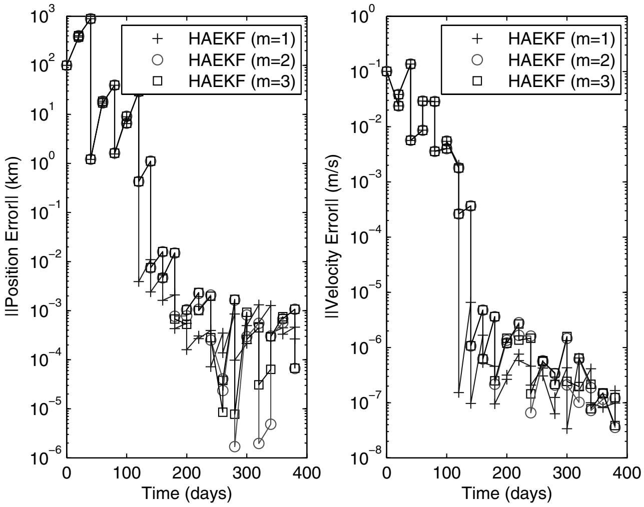


Fig. 10 Comparison of the absolute errors with a 20-day increment measurement update.

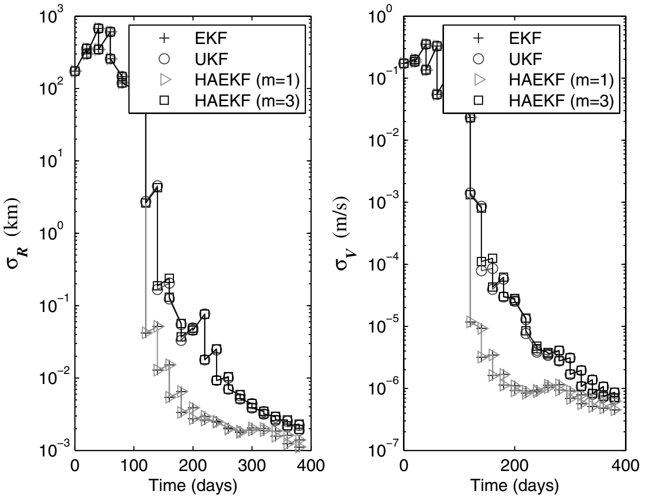


Fig. 11 Comparison of the root–sum–square errors with a 20-day increment measurement update based on the halo orbit case 2.

V. Conclusions

We derived two Kalman-type filters, called higher-order numerical extended Kalman filter and higher-order analytic extended Kalman filter, by directly applying the higher-order solutions to the Kalman filter algorithm. These higher-order filters were compared with the conventional extended Kalman filter and the unscented Kalman filter based on halo orbits computed in a restricted three-body problem frame about the sun–Earth $L1$ point. The filter simulations were carried out assuming the dynamics of the system are perfectly known, but there are errors in the initial state and in the measurements. The results showed that a higher-order filter provides faster convergence, a superior filter solution, and more flexibility in the initial guess over linear filters. Also, the Gaussian assumption of the a posteriori state yielded a sufficient approximation even for nonlinear filters. For the cases where the reference trajectory was relatively close to the true trajectory, the HAEKF provided solutions essentially equivalent to both the UKF and the HNEKF, and yielded a much faster filter process. This indicates that once trajectory solutions are stored on a spacecraft, an autonavigation processor that incorporates trajectory nonlinearity and allows fast convergence may be feasible in practice.

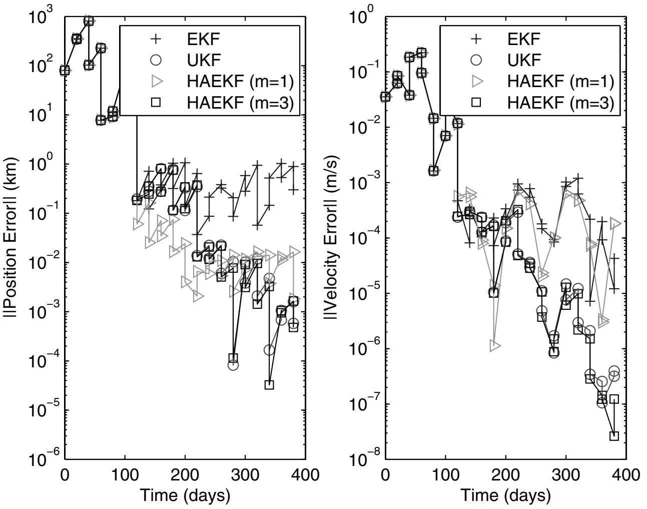


Fig. 12 Comparison of the absolute errors with a 20-day increment measurement update based on the halo orbit case 2.

Acknowledgement

The research described in this paper was sponsored by a grant from the Jet Propulsion Laboratory, California Institute of Technology which is under contract with the National Aeronautics and Space Administration.

References

[1] Doucet, A., de Freitas, N., and Gordon, N., *Sequential Monte Carlo Methods in Practice*, Springer–Verlag, Berlin, 2001.

[2] Park, R., and Scheeres, D., "Nonlinear Mapping of Gaussian State Covariance and Orbit Uncertainties," AAS Paper 05-170, Jan. 2005.

[3] Park, R., and Scheeres, D., "Nonlinear Mapping of Gaussian State Uncertainties: Theory and Application to Spacecraft Control and Navigation," AAS Paper 05-404, Aug. 2005.

[4] Park, R., and Scheeres, D., "Nonlinear Mapping of Gaussian Statistics: Theory and Applications to Spacecraft Trajectory Design," *Journal of Guidance, Control, and Dynamics*, Vol. 29, No. 6, 2006, pp. 1367–1375.

[5] Park, S., "Nonlinear Trajectory Navigation," Ph.D. Thesis, University of Michigan, Ann Arbor, MI, 2007.

[6] Julier, S., and Uhlmann, J., "Unscented Filtering and Nonlinear Estimation," *Proceedings of the IEEE*, Vol. 92, No. 3, March 2004, pp. 401–422.

[7] Kalman, R., "A New Approach to Linear Filtering and Prediction Problems," *Transactions of the ASME, Journal of Basic Engineering, Series D*, Vol. 82, March 1960, pp. 35–45.

[8] Bierman, G., *Factorization Methods for Discrete Sequential Estimation*, Vol. 128, Academic Press, New York, 1977.

[9] Montenbruck, O., and Gill, E., *Satellite Orbits*, 2nd ed., Springer, New York, 2001.

[10] Maybeck, P., *Stochastic Models, Estimation, and Control*, Vol. 2, Academic Press, New York, 1982.

[11] Julier, S., Uhlmann, J., and Durrant-Whyte, H., "A New Approach for Filtering Nonlinear Systems," *Proceedings of the American Control Conference*, June 1995, Vol. 3, pp. 1628–1632.

[12] Julier, S., Uhlmann, J., and Durrant-Whyte, H., "A New Method for the Nonlinear Transformation of Means and Covariances in Filters and Estimators," *IEEE Transactions on Automatic Control*, Vol. 45, No. 3, March 2000, pp. 477–482.

[13] Julier, S., and Uhlmann, J., "The Scaled Unscented Transformation," *Proceedings of the American Control Conference*, 2002, Vol. 6, pp. 4555–4559.

[14] Wan, E., and van der Merwe, R., *Kalman Filtering and Neural Networks: The Unscented Kalman Filter*, Wiley, New York, 2001, chap. 7.

[15] Pollard, H., *Celestial Mechanics*, The Mathematical Association of America, 1976.