JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 31, No. 4, July–August 2008

Nonlinear Geometric Estimation for Satellite Attitude

James M. Valpiani* and Philip L. Palmer†
Surrey Space Centre, Guildford, GU2 7XH, England

DOI: 10.2514/1.32715

A novel algorithm for efficient high-accuracy satellite attitude estimation is presented to address the increasing performance requirements of resource-constrained small satellites. The algorithm results from an investigation of the Bayesian nonlinear estimation problem based on the phase-space geometry of Hamiltonian systems. Probability density functions are shown to be conserved properties of deterministic Hamiltonian systems, and appropriate geometric integrators exactly preserve the functions as they evolve in time. Based on these insights, a new iterative filter is derived that conserves the geometric structure and invariant properties of the dynamics and exactly preserves the nonlinear a posteriori probability density function when solving for the state estimate. Comparisons with a benchmark iterative filter demonstrate significantly reduced computational burden, improved state estimation accuracy, and improved constants of motion estimation, particularly in the presence of high nonlinearity and low noise inputs. Based on numerous simulations, the authors conclude that this new method shows promise for improved attitude estimation onboard high-performance, resource-constrained small satellites.

I. Introduction

INCREASING satellite attitude requirements demand high-accuracy estimation methods that are capable of operating under significant constraints. To meet these demands, dynamic modeling has been used as an effective alternative to rate-sensing hardware for satellite missions. This is particularly true for small satellites, though satellites of all sizes are increasingly using dynamic modeling to reduce costs and provide contingency capabilities in case of hardware failure [1,2]. This research aims to improve attitude estimation by exploiting the Hamiltonian nature of attitude dynamics, which confers unique geometric properties to the system phase space.

In 1968, Athans et al. [3] identified two main methods that were being used to solve the nonlinear filtering problem. The first method approximated the nonlinear dynamic and observation equations to use linear filtering theory. The second method used approximations to the exact nonlinear filtering solutions, such as those found in [4,5]. Broadly speaking, estimation solutions continue to fall into one of these two categories [6–8]. Primarily because of its low computational burden, a majority of the nonlinear estimation methods used on satellites have fallen into the former category, including the ubiquitous extended Kalman filter (EKF) [4,9–11]. Although this approach has been used extensively, it renders estimation methods poorly suited for highly nonlinear dynamic systems, leading to an increased demand on satellites to continually update attitude-orientation knowledge [12]. On the other hand, approximating optimal nonlinear filtering is particularly difficult because the solution to the Fokker–Planck equation is a probability density function extending over an infinite domain. Many efforts have been made to approximate or solve numerically for exact solutions; among these methods are Monte Carlo methods, finite difference methods, and Fourier series representations, all of which have been documented as computationally expensive or difficult to implement and therefore unsuitable for online implementation [13–15]. The qualities of both these nonlinear estimation methods are troublesome for small satellites with constrained resources and ever-increasing attitude requirements. Recent advancements in small-satellite attitude determination and control system capabilities such as slewing rates of $3\,\mathrm{deg\ s^{-1}}$ or better [16] highlight the need for high-accuracy efficient attitude estimation methods, which this paper seeks to address.

Recent research in the fields of dynamic modeling and estimation has focused on preserving the geometric properties of Hamiltonian systems [17–20]. In particular, geometry-conserving maps have been applied to Kalman filters for satellite attitude, resulting in substantial improvements in state accuracies and constants of motion preservation [21]. In this paper, a more general investigation of geometric estimation is conducted. Beginning with the Bayesian formulation of the nonlinear estimation problem, the geometric nature of Hamiltonian systems is exploited to show that for the deterministic estimation problem (i.e., estimation problem with stochastic initial conditions, deterministic dynamics, and no process noise), the Fokker–Planck equation, which governs the evolution of the state probability density function (pdf), is zero. This leaves the probability density function associated with the system state invariant over time [22]. Therefore, an appropriate geometric integrator may be used to solve for the evolution of the pdf exactly, with the additional benefit of conserving the underlying geometric structure of the system dynamics.

Generally, the state pdf is not invariant under stochastic flow (i.e., in the presence of process noise); to exploit the pdf invariance property of deterministic systems while incorporating process noise, the initial pdf is adjusted such that under deterministic flow, it approximates the unadjusted initial pdf under stochastic flow. The nonlinearly propagated pdf is then combined with a system measurement via Bayes's rule. The resulting nonlinear pdf's maximum-likelihood state may be formulated as a nonlinear least-squares problem, and the Gauss–Newton method is used to derive an iterative algorithm called the nonlinear geometric filter (NGF). In addition to conserved properties of the Hamiltonian dynamics, this filter preserves conserved properties that are inherent in the estimation process. This filter is then specialized to the case of satellite attitude in a similar fashion to the well-known multiplicative extended Kalman filter (MEKF) [23,24], resulting in the multiplicative geometric filter (MGF). Comparisons with the multiplicative version of the benchmark iterative filter, the iterated Kalman filter smoother (IKFS), reveal significant advantages to using the MGF for attitude estimation.

In this paper, a new nonlinear solution to the general estimation problem for Hamiltonian systems is presented according to the following structure. In Sec. II, the equations of motion governing satellite attitude are defined, followed by a brief review of geometric

Presented as Paper 6159 at the 2006 AIAA/AAS Astrodynamics Specialist Conference and Exhibit, Keystone, CO, 21–24 August 2006; received 21 July 2007; revision received 3 December 2007; accepted for publication 4 December 2007. This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States. 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 0731-5090/08 $10.00 in correspondence with the CCC.

*Ph.D., Surrey Space Centre; james.valpiani@gmail.com.
†Reader, Surrey Space Centre; palmer@surrey.ac.uk. Member AIAA.

integrators. Then the nonlinear estimation problem is summarized in Sec. III, followed by the theory underlying the NGF and the explicit NGF algorithm in Sec. IV. The NGF's attitude variant, the MGF, is derived in Sec. V. The IKFS is presented in Sec. VI and similarities between the IKFS and NGF algorithms are discussed. Comparisons between the MGF and IKFS for a typical small-satellite scenario are presented in Sec. VII, and Sec. VIII concludes.

II. Equations of Motion

A. Coordinate Systems

Fundamentally, three-axis attitude parameterization specifies the orientation of one reference frame with respect to another reference frame. In this research, three reference frames are used. The first of these is the body-axis coordinate system, which is defined by a right-handed set of three orthogonal unit vectors $\hat{\mathbf{x}}_b$, $\hat{\mathbf{y}}_b$, and $\hat{\mathbf{z}}_b$ cooriginated at the center of mass of the satellite.

The second is the local orbital coordinate system defined by a right-handed set of three orthogonal unit vectors $\hat{\mathbf{x}}_o$, $\hat{\mathbf{y}}_o$, and $\hat{\mathbf{z}}_o$ cooriginated at the center of mass of the satellite (thus coinciding with the origin of the body frame). The $\hat{\mathbf{z}}_o$ axis points toward the gravitating center (nadir direction) and the $\hat{\mathbf{y}}_o$ axis points opposite the orbital angular momentum vector $\mathbf{h}_o$ (orbital antinormal direction). The $\hat{\mathbf{x}}_o$ axis completes the right-handed orthogonal set.

The final frame is the inertial coordinate system with its origin at the center of the Earth and axes defined by a right-handed set of three orthogonal unit vectors $\hat{\mathbf{x}}_i$, $\hat{\mathbf{y}}_i$, and $\hat{\mathbf{z}}_i$. The direction of the inertial $\hat{\mathbf{y}}_i$ axis is chosen to be opposite the direction of the orbital angular momentum vector $\mathbf{h}_o$ and hence points in the same direction as the local orbital $\hat{\mathbf{y}}_o$ axis. The directions of the $\hat{\mathbf{x}}_i$ and $\hat{\mathbf{z}}_i$ axes are chosen to be the same as the $\hat{\mathbf{x}}_o$ and $\hat{\mathbf{z}}_o$ axes as the satellite passes over the Earth's equator at the ascending node. Note that this system is not truly inertial due to precession of the orbital plane about the Earth. However, comparing the precession's relatively slow period to the typical period of rotation for an Earth-orbiting satellite, the frame is assumed to be sufficiently inertial.

B. Kinematic Parameterizations

The four-component unit quaternion is the method of choice for representing satellite attitude [25] because it is the minimum-parameter globally nonsingular attitude parameter set. In this section, relevant direction cosine matrix and quaternion definitions are given.

Quaternions are subject to noncommutative algebra, which is briefly reviewed here. Consider a unit quaternion

$$\mathbf{q} = [\, q_1 \quad q_2 \quad q_3 \quad q_4 \,]^T = \begin{bmatrix} \boldsymbol{q} \\ q_4 \end{bmatrix} \tag{1}$$

subject to the constraint

$$\mathbf{q}^T\mathbf{q} = 1 \tag{2}$$

where $\boldsymbol{q}$ is the imaginary or vector part and q_4 is the real part of the quaternion. Unit quaternions can be thought of as four-dimensional vectors confined to the surface of the unit sphere S^3. The composition of two quaternions $\mathbf{p}$ and $\mathbf{q}$ is defined here according to [23]

$$\mathbf{p} \otimes \mathbf{q} = \begin{bmatrix} p_4\boldsymbol{q} + q_4\boldsymbol{p} - \boldsymbol{p} \times \boldsymbol{q} \\ p_4 q_4 - \boldsymbol{p} \cdot \boldsymbol{q} \end{bmatrix} \tag{3}$$

The composition of quaternions is bilinear, so that they also may be written as

$$\mathbf{p} \otimes \mathbf{q} = \Omega(\mathbf{p})\mathbf{q} = \Omega^*(\mathbf{q})\mathbf{p} \tag{4}$$

where

$$\Omega(\mathbf{p}) = [\, \Xi(\mathbf{p}) \quad \mathbf{p} \,] \tag{5a}$$

$$\Omega^*(\mathbf{q}) = [\, \Xi^*(\mathbf{q}) \quad \mathbf{q} \,] \tag{5b}$$

and

$$\Xi(\mathbf{p}) = \begin{bmatrix} p_4\mathrm{I}_{3\times 3} - \boldsymbol{p}^\times \\ -\boldsymbol{p}^T \end{bmatrix} \tag{6a}$$

$$\Xi^*(\mathbf{q}) = \begin{bmatrix} q_4\mathrm{I}_{3\times 3} + \boldsymbol{q}^\times \\ -\boldsymbol{q}^T \end{bmatrix} \tag{6b}$$

The superscript $\times$ is used to denote the cross-product matrix

$$\boldsymbol{q}^\times = \begin{bmatrix} 0 & -q_3 & q_2 \\ q_3 & 0 & -q_1 \\ -q_2 & q_1 & 0 \end{bmatrix} \tag{7}$$

where

$$\boldsymbol{q}^\times \boldsymbol{p} = \boldsymbol{q} \times \boldsymbol{p} \tag{8}$$

Note that the composition of two quaternions results in a quaternion, so that if

$$\mathbf{m} = \mathbf{p} \otimes \mathbf{q} \tag{9}$$

then the inverse relationship is

$$\mathbf{p} = \mathbf{m} \otimes \mathbf{q}^* \tag{10}$$

where $\mathbf{m}$ is a quaternion and the * superscript in $\mathbf{q}^*$ is the conjugate operator of the quaternion, defined as

$$\mathbf{q}^* = \begin{bmatrix} -\boldsymbol{q} \\ q_4 \end{bmatrix} \tag{11}$$

Additionally, the identity quaternion is defined as

$$\mathbf{q}_{\mathrm{id}} = [\, 0 \quad 0 \quad 0 \quad 1 \,]^T \tag{12}$$

The nine-parameter direction cosine matrix $A \in \mathbb{R}^{3\times 3}$ is a proper real orthogonal matrix that maps vectors from one coordinate frame to another and is regarded as the fundamental attitude representation for this research. The identity rotation is defined as

$$A_{\mathrm{id}} = [\, \hat{\mathbf{i}} \quad \hat{\mathbf{j}} \quad \hat{\mathbf{k}} \,] = \mathrm{I}_{3\times 3} \tag{13}$$

The direction cosine matrix can be represented in terms of the unit quaternion via

$$A(\mathbf{q}) = \Xi^{*T}(\mathbf{q})\Xi(\mathbf{q}) = \left(q_4^2 - |\boldsymbol{q}|^2\right)\mathrm{I}_{3\times 3} - 2q_4\boldsymbol{q}^\times + 2\boldsymbol{q}\boldsymbol{q}^T \tag{14}$$

C. Exact Attitude System

The exact equations of motion [12] describe the attitude motion of a triaxial rigid body possessing internal angular momentum devices and subject to gravity gradient torque in circular orbit about a spherical Earth. The satellite's attitude is defined by a quaternion that relates its body coordinate frame to the local orbital frame. The time evolution of this quaternion is

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{q} = \frac{1}{2}[\Omega(\boldsymbol{\omega}^\dagger) + \Omega^*(\mathbf{n}^\dagger)]\mathbf{q} \tag{15}$$

where

$$\mathbf{n} = [\, 0 \quad n \quad 0 \,]^T \tag{16a}$$

$$n = \sqrt{\frac{\mu}{a^3}} \tag{16b}$$

where $\mathbf{n}$ is the angular rate of the orbital frame with respect to the inertial frame expressed in orbital frame coordinates, μ is the gravitational constant for Earth, a is the circular orbital radius, and the superscript † indicates the quaternion form of a three-vector;

that is,

$$\boldsymbol{\omega}^{\dagger} = \begin{bmatrix} \boldsymbol{\omega} \\ 0 \end{bmatrix} \tag{17}$$

and $\boldsymbol{\omega}$ is the angular rate of the body frame with respect to the inertial frame expressed in body-frame coordinates.

In addition, the exact system's dynamics incorporate gravity gradient torque, which is particularly important in modeling and stabilizing small-satellite attitude. Approximating the gradient torque to second order and describing the rotational motion of the satellite with respect to an inertial frame gives [26]

$$I\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{\omega} = (I\boldsymbol{\omega} + \mathbf{h}_{\text{int}}) \times \boldsymbol{\omega} + 3n^2(A(\mathbf{q})\hat{\mathbf{k}}) \times I(A(\mathbf{q})\hat{\mathbf{k}}) \tag{18}$$

where I is the moment-of-inertia tensor of the satellite and $\mathbf{h}_{\text{int}}$ is the satellite's internal angular momentum vector, which is assumed constant. Without loss of generality, the principal moment-of-inertia axes are assumed to be along the body axes and, as a result, its off-diagonal terms are zero.

The exact system possesses the following constants of motion:

$$\|\mathbf{q}\| = 1 \tag{19a}$$

$$H_{\text{exact}} = \frac{1}{2}\boldsymbol{\omega} \cdot I\boldsymbol{\omega} + (A(\mathbf{q})\mathbf{n}) \cdot (I\boldsymbol{\omega} + \mathbf{h}_{\text{int}}) + \frac{3n^2}{2}(A(\mathbf{q})\hat{\mathbf{k}}) \cdot I(A(\mathbf{q})\hat{\mathbf{k}}) \tag{19b}$$

where H_{exact} is the system Hamiltonian. In the absence of gravity gradient torque, rotational energy and the angular momentum vector magnitude are the conserved quantities. Using angular momentum vectors in place of angular rate vectors found in the system equations, in which the two vectors are trivially related via

$$\mathbf{h} = I\boldsymbol{\omega} \tag{20}$$

then the exact satellite attitude system has state vector

$$\mathbf{x} = \begin{bmatrix} \mathbf{q} \\ \mathbf{h} \end{bmatrix} \tag{21}$$

All Hamiltonian systems' equations of motion may be expressed as

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{x} = J(\mathbf{x})\nabla_{\mathbf{x}}H(\mathbf{x}) \tag{22}$$

where $J(\mathbf{x})$ is called the structure matrix, which defines a Poisson bracket that is well known to satisfy the properties of bilinearity, skew symmetry, Leibniz's rule, and the Jacobi identity [27]. For the exact attitude system, then, the equations of motion take the form

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{x} = J_{\text{exact}}\nabla_{\mathbf{x}}H_{\text{exact}} \tag{23}$$

where

$$J_{\text{exact}} = \begin{bmatrix} 0 & \frac{1}{2}\Xi^*(\mathbf{q}) \\ -\frac{1}{2}\Xi^{*T}(\mathbf{q}) & \mathbf{h}^{\times} + \mathbf{h}_{\text{int}}^{\times} \end{bmatrix} \tag{24}$$

Because J_{exact} depends linearly on $\mathbf{x}$, the exact attitude system is called Lie–Poisson [28]. Its Hamiltonian nature confers geometric constraints on its phase space, and these constraints imply the preservation of phase-space volume [29]. In other words, the volume in phase space occupied by a set of initial conditions in a neighborhood around a given state of interest remains constant as all the initial conditions evolve in time. This integral invariant implies strong restrictions on possible Hamiltonian system solutions.

D. Geometric Integration

For Hamiltonian systems, a state at any time completely determines the state at all later times, thus forming a trajectory in phase space. So, given an initial value, the system solution is a function of time. To reflect this, a mapping is defined that evolves initial conditions along phase-space trajectories:

$$\mathbf{x}_{t_0+\tau} = \phi_{\tau}(\mathbf{x}_{t_0}) \tag{25}$$

where $\phi_{\tau}: \mathbb{R}^m \to \mathbb{R}^m$ is called the flow map of the system and, in this context, it evolves an initial state at time t_0 forward by the interval τ. Often, it is convenient to denote the state at a particular time along a given phase-space trajectory, and this may be represented in map notation via $\mathbf{x}_t = \phi_t(\mathbf{x}_{t_0})$. For the many cases when the flow map ϕ is not known explicitly, then numerical maps ψ are used to evolve the system state. If ψ satisfies the same geometric constraints as the flow it approximates, then the numerical method is called geometric [30].

More specifically, geometric integrators are "methods that exactly (i.e., up to rounding errors) conserve qualitative properties associated to the solutions of the dynamic system under study" [29]. A "qualitative property" is a physical property such as conserved angular momentum for rigid-body motion or a more abstract property such as phase-space geometry. This is in contrast to the standard approach to numerical integration, which involves changing a set of initial conditions by moving them in the direction specified by the differential equations, and to do so in such a way as to minimize the error between the true solution and the approximated solution over a given time interval. In general, nongeometric methods will introduce secular errors into conserved properties over time and therefore will misrepresent conservative systems as being dissipative [31].

It is not necessarily the case that geometric integrators are preferable to nongeometric integrators of the same order in all cases; according to conventional wisdom [29,32], standard methods are generally more appropriate when seeking small local errors and geometric methods are more appropriate when seeking reliable modeling over long periods of time. This implies that for the attitude estimation problem, which operates on short time scales, standard methods are more appropriate for attitude modeling. However, it is known [33] that a geometric integrator's error can be dramatically smaller than a standard integrator's when the system solution is quasi-periodic, which was exploited to good effect in previous geometric integration research [12,34,35]. As Karasopoulos [36] pointed out, a rigid body under the influence of gravity gradient torque in a circular orbit undergoes periodic and quasi-periodic motion. In addition, satellite motion is inherently based on rigid-body motion, which has a particularly appealing property when modeled by geometric integrators. In the case of the rigid body, it is possible for a geometric integrator to preserve not only the structure of the flow map, but also the exact energy of the system and the first integral of angular momentum magnitude. These additional benefits are a consequence of structure-preserving mappings and come "free of charge" when using appropriate geometric methods.

The nonlinear filter developed in this paper is not dependent on a particular geometric integrator, it only requires that the integrators used be geometric. Like the EKF, it requires a nonlinear propagator and linearized propagator [i.e., a state transition matrix (STM)]. Note that a geometric STM has the same meaning as a standard (nongeometric) STM. However, it has the additional advantage of preserving phase-space volume within uncertainty ellipsoids as they evolve linearly in time, whereas nongeometric STMs will generally introduce unphysical growth or decay in the phase-space volume. Because phase-space volume is a conserved property for Hamiltonian systems, geometric STMs more closely correspond to physical motion than nongeometric STMs. References [12,21] detail the development and performance of a number of nonlinear and linearized geometric integrators that are suitable for use in the NGF, including those employed in this paper.

To simplify notation when discussing numerical integrators, $\boldsymbol{\psi}$ is defined as a function that geometrically evolves both a system state and a STM. In other words, $\boldsymbol{\psi}$ encompasses two separate geometric

numerical maps ψ and Ψ. As already noted, ψ nonlinearly evolves an initial system state from time t_0 to time t in such a way as to preserve the geometric properties of the flow it approximates. Similarly, Ψ linearly evolves an initial STM such that it relates state deviations at t_0 to deviations at t:

$$\begin{bmatrix} \mathbf{x}_t \\ \Phi(t,t_0) \end{bmatrix} = \boldsymbol{\psi}_t\left(\begin{bmatrix} \mathbf{x}_{t_0} \\ \Phi(t_0,t_0) \end{bmatrix}\right) = \begin{bmatrix} \psi_t(\mathbf{x}_{t_0}) \\ \Psi_t(\Phi(t_0,t_0)) \end{bmatrix} \tag{26}$$

Whereas ψ evolves a state along its nonlinear phase-space trajectory, Ψ evolves STMs along linearizations of the nonlinear phase-space trajectory. The initial STM $\Phi(t_0,t_0)$ is the identity matrix by definition, and the final STM $\Phi(t,t_0)$ linearly relates state deviations from $\mathbf{x}_{t_0}$ at time t_0 to state deviations from $\mathbf{x}_t$ at time t. The notation introduced here will prove useful in NGF development.

III. Nonlinear Estimation Theory

Fundamentally, estimation is the determination of a system's state in the presence of noisy observations, uncertain initial conditions, and random disturbances too complex to model. This research effort takes the Bayesian approach in solving the estimation problem. Consider the continuous-time Itô stochastic differential equation [4]

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{x}_t = f(\mathbf{x}_t) + G(\mathbf{x}_t)\boldsymbol{\eta}_t, \qquad t \geq t_0 \tag{27}$$

where $\mathbf{x}_t = [x_1 \quad x_2 \quad \ldots \quad x_m]_t^T \in \mathbb{R}^m$ is the system state at time t, $f(\mathbf{x}_t) \in \mathbb{R}^m$ describes the deterministic behavior of the state, $G(\mathbf{x}_t) \in \mathbb{R}^{m\times k}$ characterizes the diffusion, and $\{\boldsymbol{\eta}_t \in \mathbb{R}^k, t \geq t_0\}$ is a zero-mean white Gaussian noise process with $\mathcal{E}\{\boldsymbol{\eta}_t, \boldsymbol{\eta}_{t'}\} = Q(t)\delta(t-t')$. Observations of the system are taken at discrete time instants t_n,

$$\mathbf{z}_n = h(\mathbf{x}_n) + \mathbf{v}_n \qquad n = 1,2,\ldots, \qquad t_0 \leq t_n < t_{n+1} \tag{28}$$

where $\mathbf{z}_n \in \mathbb{R}^r$ is a vector of observations, $h(\mathbf{x}_n) \in \mathbb{R}^r$ relates the system states to the observations, and $\{\mathbf{v}_n \in \mathbb{R}^r, n = 1,\ldots\}$ is a white Gaussian noise process with $\mathbf{v}_n \sim N(0, R_n)$. For convenience, the state $\mathbf{x}$ at time t will generally be written as $\mathbf{x}_t$, and at discrete time instants t_n, it will be written as $\mathbf{x}_n$; also, functions of the state are generally functions of time t (or t_n), though this will not be explicitly stated, for convenience. It is assumed that the initial state, the process noise, and the measurement noise are independent, and that some initial state probability density $\wp(\mathbf{x}_0)$ is known.

The nonlinear estimation problem is to find the state estimates conditioned on the observations. This is accomplished by determining the evolution of the conditional probability density function $\wp(\mathbf{x}_n|Z_n)$, where $Z_n = \{\mathbf{z}_d : t_0 < t_d \leq t_n\}$ and $\wp(\mathbf{x}_0|Z_0) = \wp(\mathbf{x}_0)$. Knowing the conditional density function, descriptive statistics can be used to estimate the state.

A. Fokker–Planck Equation

Between observations at times t_{n-1} and t_n, the conditional density $\wp(\mathbf{x}_t|Z_{n-1})$ satisfies the Fokker–Planck equation (also known as Kolmogorov's forward equation) [14]:

$$\frac{\partial}{\partial t}\wp = -\sum_{i=1}^{m}\frac{\partial}{\partial x_i}[\wp f_i] + \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\frac{\partial^2}{\partial x_i \partial x_j}[\wp[GQG^T]_{ij}], \qquad t_{n-1} \leq t < t_n \tag{29}$$

where $\wp(\mathbf{x}_t|Z_{n-1})$, $f_i(\mathbf{x}_t)$, $G(\mathbf{x}_t)$, and Q_t were replaced by $\wp$, f_i, G, and Q, respectively, for simplicity.

B. Bayes's Rule

Using Eq. (29), some conditional density $\wp(\mathbf{x}_{n-1}|Z_{n-1})$ can be evolved in time up to an observation at time t_n, giving the a priori density $\wp(\mathbf{x}_n|Z_{n-1})$. Then the a posteriori conditional density $\wp(\mathbf{x}_n|Z_n)$ can be determined via Bayes's rule [4]:

$$\wp(\mathbf{x}_n|Z_n) = \frac{\wp(\mathbf{z}_n|\mathbf{x}_n)\wp(\mathbf{x}_n|Z_{n-1})}{\int \wp(\mathbf{z}_n|\xi,t_n)\wp(\xi,t_n|Z_{n-1})\,\mathrm{d}\xi} \tag{30}$$

Using transformation of variables, $\wp(\mathbf{z}_n|\mathbf{x}_n)$ can be rewritten as $\wp_{\mathbf{v}_n}(\mathbf{z}_n - h(\mathbf{x}_n))$, and because $\mathbf{v}_n \sim N(0, R_n)$,

$$\wp(\mathbf{z}_n|\mathbf{x}_n) = \frac{1}{|2\pi R_n|^{\frac{1}{2}}} e^{-\frac{1}{2}([\mathbf{z}_n - h(\mathbf{x}_n)]^T R_n^{-1}[\mathbf{z}_n - h(\mathbf{x}_n)])} \tag{31}$$

where $|\cdot|$ is the matrix determinant. The nonlinear estimation solution is represented in block diagram form in Fig. 1 [14].

The a posteriori conditional density $\wp(\mathbf{x}_n|Z_n)$ is the complete solution of the nonlinear estimation problem because it embodies all statistical information about $\mathbf{x}$ at t_n which is contained in the observations and the initial condition $\wp(\mathbf{x}_0)$. However, Jazwinski [4] demonstrated that for the general nonlinear case, the solutions to the mode, the mean, and other higher-order moments are infinite-dimensional. As a result, approximations must be made to develop a realizable nonlinear filter.

IV. Nonlinear Geometric Filtering Theory Development

Consider the deterministic portion of the general Fokker–Planck Eq. (29), where Q is ignored for the time being. Under this assumption, the Fokker–Planck equation reduces to a deterministic Hamiltonian system:

$$\frac{\partial}{\partial t}\wp = -\sum_{i=1}^{m}\frac{\partial}{\partial x_i}[\wp f_i] \tag{32}$$

Recalling that Hamiltonian systems may be expressed in structure matrix form as in Eq. (22), then Eq. (32) may be written as [37]

$$\frac{\partial}{\partial t}\wp = -\left[\frac{\partial}{\partial \mathbf{x}}\wp\right]^T f - \wp\sum_{i=1}^{m}\frac{\partial}{\partial x_i}[J(\mathbf{x})\nabla_{\mathbf{x}}H(\mathbf{x})]_i \tag{33}$$

The second term on the right-hand side of Eq. (33) may be expressed as

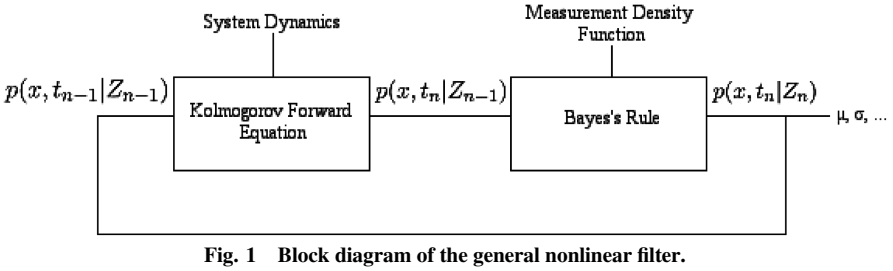

Fig. 1 Block diagram of the general nonlinear filter.

$$\sum_{i=1}^{m}\frac{\partial}{\partial x_i}[J(\mathbf{x})\nabla_{\mathbf{x}}H(\mathbf{x})]_i=\sum_{i=1}^{m}\left[\left[\frac{\partial}{\partial x_i}J(\mathbf{x})\right]\nabla_{\mathbf{x}}H(\mathbf{x})\right]_i + \sum_{i=1}^{m}\left[J(\mathbf{x})\left[\frac{\partial}{\partial x_i}\nabla_{\mathbf{x}}H(\mathbf{x})\right]\right]_i \tag{34}$$

where $\wp$ was ignored, for convenience. Using the antisymmetric property of any Hamiltonian structure matrix in addition to the property of second-order mixed partial derivatives that

$$\frac{\partial^2}{\partial x_a\partial x_b}H(\mathbf{x})=\frac{\partial^2}{\partial x_b\partial x_a}H(\mathbf{x}) \tag{35}$$

it is straightforward to show that the second term of Eq. (34) vanishes. The first term of Eq. (34) defines the phase-space compressibility [38]:

$$\kappa=\sum_{i=1}^{m}\left[\left[\frac{\partial}{\partial x_i}J(\mathbf{x})\right]\nabla_{\mathbf{x}}H(\mathbf{x})\right]_i \tag{36}$$

For Hamiltonians with state-independent structure matrices (such as canonical systems), it is trivial to verify that $\kappa=0$. Similarly, a brief inspection reveals that $\kappa=0$ for the exact attitude system's structure matrix in Eq. (24). Flow maps with this property are incompressible; in other words, a volume of initial states in phase space will remain constant as the states evolve along their trajectories, imposing strict conditions on the evolution of system solutions. In this case, Eq. (33) reduces to

$$\frac{\mathrm{d}}{\mathrm{d}t}\wp=0 \tag{37}$$

Therefore, the probability density function for these systems is an integral invariant of the system. With this result, a general solution for the evolution of the probability density function may be derived assuming that its functional form is known at a certain time, based on the work of Scheeres et al. [22]. Using the definition of an integral invariant and Eq. (37),

$$\int_{B_{t_0}}\wp(\mathbf{x}_{t_0})\,\mathrm{d}\mathbf{x}_{t_0}=\int_{B_t}\wp(\mathbf{x}_t)\,\mathrm{d}\mathbf{x}_t \tag{38}$$

The system flow $\mathbf{x}_t=\phi_t(\mathbf{x}_{t_0})$ may be used to transform the second integral:

$$\int_{B_{t_0}}\wp(\mathbf{x}_{t_0})\,\mathrm{d}\mathbf{x}_{t_0}=\int_{B_{t_0}}\wp(\phi_t(\mathbf{x}_{t_0}))\left|\frac{\partial\mathbf{x}_t}{\partial\mathbf{x}_{t_0}}\right|\mathrm{d}\mathbf{x}_{t_0} \tag{39}$$

However, the geometric nature of Hamiltonian systems implies that [27,39] $|\partial\mathbf{x}_t/\partial\mathbf{x}_{t_0}|=1$, and because Eq. (39) is independent of B_{t_0}, then

$$\wp(\mathbf{x}_{t_0})=\wp(\phi_t(\mathbf{x}_{t_0})) \tag{40}$$

The insight from this result is that if the functional form of a probability density function is known at time t_0, then it can be characterized at any other time t. Using the inverse flow from time t to t_0, $\mathbf{x}_{t_0}=\phi_{t_0}(\mathbf{x}_t)$, in Eq. (40) and assuming the explicit form of $\wp$ is known at time t_0, then the probability density function at time t is

$$\wp(\mathbf{x}_t)=\wp(\phi_{t_0}(\mathbf{x}_t)) \tag{41}$$

For the satellite attitude problem, the true solution flow ϕ is nonlinear and requires the use of a numerical integrator to solve for the system state. Although structure-preserving geometric integrators by definition satisfy $|\partial\mathbf{x}_t/\partial\mathbf{x}_{t_0}|=1$, the most commonly applied integrators such as standard Runge–Kutta methods [40] do not preserve the structure of Hamiltonian systems and therefore do not satisfy this flow condition.

Considering these results in the context of the nonlinear estimation problem, this means that a probability density function of a deterministic Hamiltonian system can be completely characterized as it transforms forward in time. The accuracy of the solution is limited only by the numerical method employed for state propagation. An appropriate geometric method preserves qualitative properties of both the state evolution and its associated density function so that the properties of Eq. (41) may be exploited. In contrast, nongeometric integrators violate the physical properties of the state evolution and do not strictly preserve the conditions necessary for conservative density functions. In the following sections, the NGF is developed to exploit the ability of geometric integrators to accurately preserve Hamiltonian density functions as they evolve in time.

A. Density Function Approximations

Applying the result from Sec. IV to the Hamiltonian estimation problem, assume an initial conditional density function at time t_n:

$$\wp(\mathbf{x}_n|Z_n) \tag{42}$$

Equation (41) gives the evolution of this initial conditional density function between observations. Therefore, at the next observation time t_{n+1}, the transformed density function is

$$\wp(\mathbf{x}_{n+1}|Z_n)=\wp(\phi_{t_n}(\mathbf{x}_{n+1})|Z_n) \tag{43}$$

Note that this entails backward propagation of the state $\mathbf{x}_{n+1}$. Given this a priori density function, the new observation $\mathbf{z}_{n+1}$, and its density function, Bayes's rule can be used to determine the a posteriori density function $\wp(\mathbf{x}_{n+1}|Z_{n+1})$.

At this point, no assumptions have been made about the types of density functions used; to realize an actual filter, these functions must be defined. Assume that the initial state density $\wp(\mathbf{x}_n|Z_n)$ is Gaussian:

$$\mathbf{x}_n\sim N\left(\hat{\mathbf{x}}_n^+,P_n^+\right) \tag{44}$$

Assuming no diffusion, the a priori density is given by the deterministic Fokker–Planck equation, and the exact solution for Hamiltonian systems follows in Eq. (43) with

$$\phi_{t_n}(\mathbf{x}_{n+1})\sim N\left(\hat{\mathbf{x}}_n^+,P_n^+\right) \tag{45}$$

Assume also that observations have additive white Gaussian noise, as in Eq. (28). Then using Bayes's rule from Eqs. (30) and (31) and the a priori density in Eq. (43) gives the a posteriori density:

$$\wp(\mathbf{x}_{n+1}|Z_{n+1})=ce^{-\frac{1}{2}(\cdot)} \tag{46a}$$

$$(\cdot)=\left[\phi_{t_n}(\mathbf{x}_{n+1})-\hat{\mathbf{x}}_n^+\right]^TP_n^{+^{-1}}\left[\phi_{t_n}(\mathbf{x}_{n+1})-\hat{\mathbf{x}}_n^+\right] + [\mathbf{z}_{n+1}-h(\mathbf{x}_{n+1})]^TR_{n+1}^{-1}[\mathbf{z}_{n+1}-h(\mathbf{x}_{n+1})] \tag{46b}$$

where $\mathbf{x}_{n+1}$ is the independent variable, $\hat{\mathbf{x}}_n^+$ is the (fixed) state from time t_n, and P_n^+ is the (fixed) covariance estimate from time t_n. Note that the constant

$$c=\frac{1}{\int\wp(\mathbf{z}_{n+1}|\xi,t_{n+1})\wp(\xi,t_{n+1}|Z_n)\,\mathrm{d}\xi} \tag{47}$$

is independent of $\mathbf{x}$ and that the normalizing constants $1/|2\pi R_{n+1}|^{\frac{1}{2}}$ in $\wp(\mathbf{z}_{n+1}|\mathbf{x}_{n+1})$ and $1/|2\pi P_n^+|^{\frac{1}{2}}$ in $\wp(\mathbf{x}_n|Z_n)$ are independent of $\mathbf{x}$ and therefore cancel between the numerator and denominator when applying Eq. (30).

The density $\wp(\mathbf{x}_{n+1}|Z_{n+1})$ is the full solution to the diffusionless nonlinear estimation problem. Generally, this is not a satisfactory approach; diffusion is an integral part of the estimation process because it allows the filter to account for unmodeled disturbances. However, the stochastic Fokker–Planck equation (29) is generally unsolvable for nonlinear functions, and so an approximation must be made to obtain an a priori density. To this end, denote the true a priori density as $\wp_{\text{true}}(\mathbf{x}_{n+1}|Z_n)$. Its covariance $P_{n+1,\text{true}}^-$ is a function of the

initial covariance P_n^+ and the cumulative diffusion between t_n and t_{n+1}, denoted as Q_{n+1}.

Though the stochastic Fokker–Planck equation cannot be solved, in general, the deterministic one can be and the solution was shown to be very simple for Hamiltonian systems [see Eq. (43)]. To exploit this, the NGF seeks a modified initial covariance P_n^{*+} that gives the true covariance $P_{n+1,\text{true}}^-$ when evolved by the deterministic Fokker–Planck solution. In other words, the NGF uses a modified initial covariance P_n^{*+} that, under deterministic evolution, results in approximately the same final covariance $P_{n+1,\text{true}}^-$ given by stochastic evolution of the initial covariance P_n^+. Modifying the initial covariance with cumulative diffusion, rather than the usual practice of adding it to the a priori density function, allows the NGF to incorporate process noise within the deterministic solution to the Fokker–Planck equation. The deterministic solution in Eq. (46a) defines the density function at time t_{n+1} based on the density function at time t_n; therefore, to account for cumulative diffusion, it must be defined at time t_n and incorporated into the density function at time t_n.

In the presence of diffusion, the following approximation is made during the NGF state estimation phase:

$$\wp(\mathbf{x}_{n+1}|Z_n) \approx \wp_{\text{True}}(\mathbf{x}_{n+1}|Z_n) \tag{48}$$

where

$$\wp(\mathbf{x}_{n+1}|Z_n) = \wp(\phi_{t_n}(\mathbf{x}_{n+1})|Z_n) \tag{49a}$$

$$\phi_{t_n}(\mathbf{x}_{n+1}) \sim N\left(\hat{\mathbf{x}}_n^+, P_n^+ + \tilde{Q}_n\right) \tag{49b}$$

Note the contrast between this a priori density characterized by Eq. (49b) and the original diffusionless a priori density characterized by Eq. (45). Here, $\tilde{Q}_n$ represents the modification to the initial covariance such that $P_n^+ + \tilde{Q}_n \approx P_n^{*+}$. In other words, the initial covariance P_n^+ is modified such that, under deterministic evolution, it approximates the initial covariance under stochastic evolution, $P_{n+1,\text{true}}^-$. The result is a nonlinear approximation to the true a priori density.

It has been shown previously that in the case of zero diffusion, the NGF exactly preserves the a priori density function as it evolves in time (i.e., exactly solves the deterministic Fokker–Planck equation). Generally, it is desirable to keep diffusion small because large diffusion terms indicate significant unmodeled disturbances that detract from the accuracy of the estimation process. Therefore, the motivation for this approach is clear: as diffusion becomes smaller, the NGF a priori density more accurately approximates the true a priori density, and in the limiting case of $Q_{n+1} = \tilde{Q}_n = 0$, it is exact.

The key problem then is to solve for $\tilde{Q}_n$, and manipulation of the standard Kalman filter affords an elegant first-order solution. In each iteration of the Kalman filter, the process noise covariance matrix Q_{n+1} is added to the propagated state covariance matrix to determine the predicted covariance

$$P_{n+1}^-(\mathbf{x}_{n+1}) = \Phi(\mathbf{x}_{n+1})P_n^+\Phi^T(\mathbf{x}_{n+1}) + Q_{n+1} \tag{50}$$

where the state transition matrix notation $\Phi(\mathbf{x}_{n+1})$ is shorthand for $\Phi(\mathbf{x}_{n+1}, \phi_n(\mathbf{x}_{n+1}))$ and is used to emphasize the trajectory on which the matrix is dependent. Note that this solution incorporates a deterministic part $\Phi(\mathbf{x}_{n+1})P_n^+\Phi^T(\mathbf{x}_{n+1})$ and a stochastic part Q_{n+1}. Therefore, $\tilde{Q}_n$ is the covariance matrix that, when added to P_n^+, evolves deterministically to the predicted covariance in Eq. (50):

$$\Phi(\mathbf{x}_{n+1})\left[P_n^+ + \tilde{Q}_n\right]\Phi^T(\mathbf{x}_{n+1}) = \Phi(\mathbf{x}_{n+1})P_n^+\Phi^T(\mathbf{x}_{n+1}) + Q_{n+1} \tag{51}$$

This readily reduces to

$$\tilde{Q}_n(\mathbf{x}_{n+1}) = \Phi^{-1}(\mathbf{x}_{n+1})Q_{n+1}\Phi^{T^{-1}}(\mathbf{x}_{n+1}) \tag{52}$$

Clearly, $\tilde{Q}_n(\mathbf{x}_{n+1})$ is dependent on the trajectory that defines the state transition matrix, and this was made explicit in the notation. As will be seen shortly, $\tilde{Q}_n(\mathbf{x}_{n+1})$ is never explicitly calculated in the NGF algorithm.

With $\tilde{Q}_n(\mathbf{x}_{n+1})$ defined, the a priori density in Eq. (49) may be used to solve for the a posteriori density $\wp(\mathbf{x}_{n+1}|Z_{n+1})$, thereby incorporating diffusion in the estimation solution. Applying Bayes's rule to Eq. (49), the a posteriori density in Eq. (46) then becomes

$$\wp(\mathbf{x}_{n+1}|Z_{n+1}) = ce^{-\frac{1}{2}(\cdot)} \tag{53a}$$

$$(\cdot) = \left[\phi_{t_n}(\mathbf{x}_{n+1}) - \hat{\mathbf{x}}_n^+\right]^T\left[P_n^+ + \tilde{Q}_n(\mathbf{x}_{n+1})\right]^{-1}\left[\phi_{t_n}(\mathbf{x}_{n+1}) - \hat{\mathbf{x}}_n^+\right] + [\mathbf{z}_{n+1} - h(\mathbf{x}_{n+1})]^T R_{n+1}^{-1}[\mathbf{z}_{n+1} - h(\mathbf{x}_{n+1})] \tag{53b}$$

B. State Estimation

The density $\wp(\mathbf{x}_{n+1}|Z_{n+1})$ is the full solution to the stochastic nonlinear estimation problem, and its mean is commonly used as the state estimate. However, because of the constraint imposed by the nonlinear backward map ϕ_{t_n} in Eq. (53), $\wp(\mathbf{x}_{n+1}|Z_{n+1})$ is generally not a Gaussian density function and determining its mean is nontrivial. Most methods used to do so are prohibitively computationally expensive. If $\wp(\mathbf{x}_{n+1}|Z_{n+1})$ is unimodal and concentrated about the mean, then there is a negligible difference between the mean and the mode (or peak) of the density function [6]. In addition to being the maximum-likelihood estimator, the mode represents a possible system state, whereas the mean is not always guaranteed to do so (the mean of a quaternion distribution will not generally be of unit norm, for instance). For these reasons, the mode is used as the state estimate for this filter.

Determining the mode of the a posteriori density function is equivalent to finding the maximum of Eq. (53). Because the maximum of $\wp(\mathbf{x}_{n+1}|Z_{n+1})$ is the maximum of any monotonic function of $\wp(\mathbf{x}_{n+1}|Z_{n+1})$, and because the natural logarithm is monotonic, then the mode of the pdf can be stated as [6]

$$\hat{\mathbf{x}}_{n+1}^+ = \underset{\mathbf{x}_{n+1}}{\arg\max}\,(\cdot) \tag{54}$$

where $(\cdot)$ is defined in Eq. (53b). This problem can be reformulated in to a nonlinear least-squares problem

$$\hat{\mathbf{x}}_{n+1}^+ = \underset{\mathbf{x}_{n+1}}{\arg\min}\,\frac{1}{2}\Lambda^T(\mathbf{x}_{n+1})\Lambda(\mathbf{x}_{n+1}) \tag{55}$$

where

$$\Lambda(\mathbf{x}_{n+1}) = S^{-\frac{1}{2}}(\mathbf{x}_{n+1})V(\mathbf{x}_{n+1}) \tag{56a}$$

$$S(\mathbf{x}_{n+1}) = \begin{bmatrix} P_n^+ + \tilde{Q}_n(\mathbf{x}_{n+1}) & 0 \\ 0 & R_{n+1} \end{bmatrix} \tag{56b}$$

$$V(\mathbf{x}_{n+1}) = \begin{bmatrix} \psi_{t_n}(\mathbf{x}_{n+1}) \\ \mathbf{z}_{n+1} \end{bmatrix} - \begin{bmatrix} \hat{\mathbf{x}}_n^+ \\ h(\mathbf{x}_{n+1}) \end{bmatrix} \tag{56c}$$

where $[S^{\frac{1}{2}}(\mathbf{x}_{n+1})]^T S^{\frac{1}{2}}(\mathbf{x}_{n+1}) = S(\mathbf{x}_{n+1})$, and ψ was used in place of the flow ϕ to reflect the use of an appropriate geometric numerical method.

This least-squares problem may be solved using the well-known iterative Gauss–Newton method [6]:

$$\mathbf{x}_{n+1}^{k+1} = \mathbf{x}_{n+1}^k - \left[\Lambda'^T\left(\mathbf{x}_{n+1}^k\right)\Lambda'\left(\mathbf{x}_{n+1}^k\right)\right]^{-1}\Lambda'^T\left(\mathbf{x}_{n+1}^k\right)\Lambda\left(\mathbf{x}_{n+1}^k\right) \tag{57}$$

where the superscript k indicates the iteration of the Gauss–Newton algorithm, and $\Lambda'(\cdot)$ is the Jacobian matrix of $\Lambda(\cdot)$:

$$\Lambda'\left(\mathbf{x}_{n+1}^k\right) = \begin{bmatrix} P_n^+ + \tilde{Q}_n\left(\mathbf{x}_{n+1}^k\right) & 0 \\ 0 & R_{n+1} \end{bmatrix}^{-\frac{1}{2}} \begin{bmatrix} \Phi^{-1}\left(\mathbf{x}_{n+1}^k\right) \\ -H\left(\mathbf{x}_{n+1}^k\right) \end{bmatrix} \quad (58)$$

where $H(\mathbf{x}_{n+1}^k)$ is the Jacobian matrix of the observation function $h(\mathbf{x}_{n+1}^k)$ (not to be confused with the Hamiltonian function), and as before, $\Phi(\mathbf{x}_{n+1}^k)$ is shorthand for $\Phi(\mathbf{x}_{n+1}^k, \phi_n(\mathbf{x}_{n+1}^k))$. After considerable matrix algebra (detailed in [21]), this simplifies to

$$\mathbf{x}_{n+1}^{k+1} = \mathbf{x}_{n+1}^k - \Phi\left(\mathbf{x}_{n+1}^k\right)\left(\psi_{t_n}\left(\mathbf{x}_{n+1}^k\right) - \hat{\mathbf{x}}_n^+\right) + K\left(\mathbf{x}_{n+1}^k\right) \times \left(\mathbf{z}_{n+1} - h\left(\mathbf{x}_{n+1}^k\right) + H\left(\mathbf{x}_{n+1}^k\right)\Phi\left(\mathbf{x}_{n+1}^k\right)\left(\psi_{t_n}\left(\mathbf{x}_{n+1}^k\right) - \hat{\mathbf{x}}_n^+\right)\right) \quad (59)$$

where $K(\mathbf{x}_{n+1}^k)$ is

$$K\left(\mathbf{x}_{n+1}^k\right) = P_{n+1}^-\left(\mathbf{x}_{n+1}^k\right)H^T\left(\mathbf{x}_{n+1}^k\right) \times \left[H\left(\mathbf{x}_{n+1}^k\right)P_{n+1}^-\left(\mathbf{x}_{n+1}^k\right)H^T\left(\mathbf{x}_{n+1}^k\right) + R_{n+1}\right]^{-1} \quad (60)$$

Equation (59) is the final form of the NGF state update. In addition to its geometric properties, one appealing aspect of the NGF is its nonlinear treatment of system observations. As a result, recursive algorithms such as this one have shown significant decreases in errors due to linear approximations to the observation model [6]. However, observation nonlinearities are outside the scope of this paper and will not be explored further.

C. Covariance Update

The covariance update of the NGF may be derived by first considering the least-squares cost function at the heart of the NGF state update and implicit in Eq. (55):

$$C(\mathbf{x}_{n+1}) = \tfrac{1}{2}\Lambda^T(\mathbf{x}_{n+1})\Lambda(\mathbf{x}_{n+1}) \quad (61)$$

The least-squares solution is found by minimizing the gradient of the cost function:

$$\nabla C(\mathbf{x}_{n+1}) = \Lambda'^T(\mathbf{x}_{n+1})\Lambda(\mathbf{x}_{n+1}) \quad (62)$$

The Gauss–Newton algorithm makes the following approximation to the cost function gradient [6]:

$$\nabla C(\mathbf{x}_{n+1}) \approx \Lambda'^T\left(\hat{\mathbf{x}}_{n+1}^+\right)\Lambda\left(\hat{\mathbf{x}}_{n+1}^+\right) + \Lambda'^T\left(\hat{\mathbf{x}}_{n+1}^+\right)\Lambda'\left(\hat{\mathbf{x}}_{n+1}^+\right)\left(\mathbf{x}_{n+1} - \hat{\mathbf{x}}_{n+1}^+\right) \quad (63)$$

By definition, $\hat{\mathbf{x}}_{n+1}^+$ minimizes the cost function so that

$$\nabla C\left(\hat{\mathbf{x}}_{n+1}^+\right) = \Lambda'^T\left(\hat{\mathbf{x}}_{n+1}^+\right)\Lambda\left(\hat{\mathbf{x}}_{n+1}^+\right) = 0 \quad (64)$$

Equating the true gradient in Eq. (62) with the Gauss–Newton approximation in Eq. (63) and evaluating at $\hat{\mathbf{x}}_{n+1}^+$ [i.e., substituting the result from Eq. (64)] gives

$$\Delta\hat{\mathbf{x}}_{n+1}^+ = \mathbf{x}_{n+1} - \hat{\mathbf{x}}_{n+1}^+ = \left[\Lambda'^T\left(\hat{\mathbf{x}}_{n+1}^+\right)\Lambda'\left(\hat{\mathbf{x}}_{n+1}^+\right)\right]^{-1}\Lambda'^T(\mathbf{x}_{n+1})\Lambda(\mathbf{x}_{n+1}) \quad (65)$$

The NGF state update recursively solves for $\hat{\mathbf{x}}_{n+1}^+$, and once this state is known, its covariance may be solved via

$$P_{n+1}^+ = E\left(\Delta\hat{\mathbf{x}}_{n+1}^+\Delta\hat{\mathbf{x}}_{n+1}^{+T}\right) \quad (66)$$

where $E(\cdot)$ is the expectation operator. To obtain a closed-form expression for the covariance, note that the component of $H(\mathbf{x}_{n+1})$ in row j, column h may be approximated as [6]

$$[H(\mathbf{x}_{n+1})]_{jh} \approx \left[H\left(\hat{\mathbf{x}}_{n+1}^+\right)\right]_{jh} + \left[H'\left(\hat{\mathbf{x}}_{n+1}^+\right)\right]_{jh}\left(\mathbf{x}_{n+1} - \hat{\mathbf{x}}_{n+1}^+\right) + \text{HOT} \quad (67)$$

where $[H'(\hat{\mathbf{x}}_{n+1}^+)]_{jh}$ is the Jacobian matrix of $[H(\hat{\mathbf{x}}_{n+1}^+)]_{jh}$ and higher-order terms (HOT) are ignored. Noting that $P_{n+1}^-(\mathbf{x}_{n+1})$ and $\Phi(\mathbf{x}_{n+1})$ may be similarly approximated, then a considerable amount of matrix algebra leads to the familiar covariance update rule:

$$P_{n+1}^+ = \left[I - K\left(\hat{\mathbf{x}}_{n+1}^+\right)H\left(\hat{\mathbf{x}}_{n+1}^+\right)\right]P_{n+1}^-\left(\hat{\mathbf{x}}_{n+1}^+\right) \quad (68)$$

(more numerically stable forms of this update rule exist; see, for instance, [4]). As with the state update rule, the full derivation of this expression may be found in [21].

D. Nonlinear Geometric Filter Summary

The nonlinear geometric filter is summarized next. Assuming an initial state $\hat{\mathbf{x}}_n^+$, an initial covariance matrix P_n^+, a system observation $\mathbf{z}_{n+1}$ with covariance R_{n+1}, and a geometric map $\boldsymbol{\psi}$, the NGF is governed by the following equations:

Begin loop.

If k=0,

$$\begin{bmatrix} \mathbf{x}_{n+1}^0 \\ \Phi\left(\mathbf{x}_{n+1}^0, \hat{\mathbf{x}}_n^+\right) \end{bmatrix} = \boldsymbol{\psi}_{t_{n+1}}\left(\begin{bmatrix} \hat{\mathbf{x}}_n^+ \\ \Phi\left(\hat{\mathbf{x}}_n^+, \hat{\mathbf{x}}_n^+\right) \end{bmatrix}\right) \quad (69a)$$

$$\Phi\left(\mathbf{x}_{n+1}^0\right) = \Phi\left(\mathbf{x}_{n+1}^0, \hat{\mathbf{x}}_n^+\right) \quad (69b)$$

by definition,

$$\psi_{t_n}\left(\mathbf{x}_{n+1}^0\right) = \hat{\mathbf{x}}_n^+ \quad (69c)$$

Else,

$$\begin{bmatrix} \psi_{t_n}\left(\mathbf{x}_{n+1}^k\right) \\ \Phi\left(\psi_{t_n}\left(\mathbf{x}_{n+1}^k\right), \mathbf{x}_{n+1}^k\right) \end{bmatrix} = \boldsymbol{\psi}_{t_n}\left(\begin{bmatrix} \mathbf{x}_{n+1}^k \\ \Phi\left(\mathbf{x}_{n+1}^k, \mathbf{x}_{n+1}^k\right) \end{bmatrix}\right) \quad (69d)$$

$$\Phi\left(\mathbf{x}_{n+1}^k\right) = \Phi^{-1}\left(\psi_{t_n}\left(\mathbf{x}_{n+1}^k\right), \mathbf{x}_{n+1}^k\right) \quad (69e)$$

End if.

$$P_{n+1}^-\left(\mathbf{x}_{n+1}^k\right) = \Phi\left(\mathbf{x}_{n+1}^k\right)P_n^+\Phi^T\left(\mathbf{x}_{n+1}^k\right) + Q_{n+1} \quad (69f)$$

$$H\left(\mathbf{x}_{n+1}^k\right) = \left[\frac{\partial}{\partial\mathbf{x}}h(\mathbf{x})\right]\Big|_{\mathbf{x}_{n+1}^k} \quad (69g)$$

$$K\left(\mathbf{x}_{n+1}^k\right) = P_{n+1}^-\left(\mathbf{x}_{n+1}^k\right)H^T\left(\mathbf{x}_{n+1}^k\right)\left[H\left(\mathbf{x}_{n+1}^k\right)P_{n+1}^-\left(\mathbf{x}_{n+1}^k\right) \times H^T\left(\mathbf{x}_{n+1}^k\right) + R_{n+1}\right]^{-1} \quad (69h)$$

$$\mathbf{x}_{n+1}^{k+1} = \mathbf{x}_{n+1}^k - \Phi\left(\mathbf{x}_{n+1}^k\right)\left(\psi_{t_n}\left(\mathbf{x}_{n+1}^k\right) - \hat{\mathbf{x}}_n^+\right) + K\left(\mathbf{x}_{n+1}^k\right)\left(\mathbf{z}_{n+1} - h\left(\mathbf{x}_{n+1}^k\right)\right) + H\left(\mathbf{x}_{n+1}^k\right)\Phi\left(\mathbf{x}_{n+1}^k\right)\left(\psi_{t_n}\left(\mathbf{x}_{n+1}^k\right) - \hat{\mathbf{x}}_n^+\right)\Big) \quad (69i)$$

$$k = k + 1 \quad (69j)$$

End loop.

$$\hat{\mathbf{x}}_{n+1}^+ = \mathbf{x}_{n+1}^k \quad (69k)$$

$$P_{n+1}^{+} = \left[I - K\left(\hat{\mathbf{x}}_{n+1}^{+}\right)H\left(\hat{\mathbf{x}}_{n+1}^{+}\right)\right]P_{n+1}^{-}\left(\hat{\mathbf{x}}_{n+1}^{+}\right) \tag{69l}$$

The first iteration of the NGF uses the forward map $\boldsymbol{\psi}_{t_{n+1}}$ in Eq. (69a) to predict the state at time t_{n+1}. In all other iterations, the NGF uses the backward map $\boldsymbol{\psi}_{t_n}$ in Eq. (69d) to backward propagate the state. Inspection of the preceding equations reveals that the first iteration of the NGF is equivalent to the extended Kalman filter.

A few notes on implementation are in order. Note that Eqs. (69b) and (69c) are provided to clarify notation and do not require calculations. From an algorithmic point of view, the forward map $\boldsymbol{\psi}_{t_{n+1}}$ and backward map $\boldsymbol{\psi}_{t_n}$ differ only by the sign of the time step used in their calculations. Therefore, the forward and backward maps do not require separate algorithms. Similarly, it is straightforward to modify the backward map such that it outputs the previous state $\psi_{t_n}(\mathbf{x}_{n+1}^{k})$ and the forward-mapping state transition matrix $\Phi(\mathbf{x}_{n+1}^{k})$ with no more calculation than is required for the standard backward map. In other words, it is possible to modify the backward map such that Eq. (69e) is unnecessary. If the process noise matrix Q_{n+1} is trajectory-dependent, then strictly speaking, it must be recalculated with each iteration of the NGF (as is done in all simulations in this paper) for use in Eq. (69f). However, because Q_{n+1} is primarily used as a tuning parameter, it is often adequate to calculate it once per time step. Finally, there are a number of suitable termination conditions for the iterative portion of the NGF, including a maximum number of iterations or a minimum user-defined difference between state estimates produced by successive iterations. Reference [41] discusses termination conditions for the Gauss–Newton method in detail.

V. Multiplicative Geometric Filter

The NGF is applicable to general Hamiltonian problems, but estimation of unit quaternions for the satellite attitude problem poses well-documented challenges, including near-singular covariance matrices and improper operators for quaternion composition [23,24,42]. The most successful solution to these problems is embodied in the MEKF [23,24]. This section aims to apply the strengths of the MEKF to the NGF; the result is the MGF.

The following sections deal exclusively with the MGF as it differs from the NGF. A brief overview of the MGF follows. Then a new geometric map for the state deviation is defined. Using this map, the MGF algorithm is presented. Finally, the filter is adapted to the case of a satellite with quaternion attitude observations only.

A. Overview

Like the MEKF, the MGF represents true attitude as a quaternion product of two unit quaternions:

$$\mathbf{q}_t = \delta\mathbf{q}(\mathbf{a}_t) \otimes \bar{\mathbf{q}}_t \tag{70}$$

where $\bar{\mathbf{q}}_t$ is a deterministic reference quaternion. In this case, $\mathbf{q}_t$ is the true rotation from the body frame to the orbital frame, and $\bar{\mathbf{q}}_t$ is the reference rotation from the body frame to the orbital frame. The rotation from $\bar{\mathbf{q}}_t$ to $\mathbf{q}_t$ is represented as a quaternion deviation $\delta\mathbf{q}(\mathbf{a}_t)$, where $\mathbf{a}_t$ is a three-parameter attitude representation. Here, $\mathbf{a}_t$ is used to represent the vector portion of a quaternion deviation; that is, $\mathbf{a}_t = \delta\boldsymbol{q}_t$, where

$$\delta\mathbf{q}(\delta\boldsymbol{q}_t) = \begin{bmatrix} \delta\boldsymbol{q}_t \\ \sqrt{1 - \|\delta\boldsymbol{q}_t\|^2} \end{bmatrix} \tag{71}$$

Taking $\delta\boldsymbol{q}_t$ to be a random vector with expectation $\delta\hat{\boldsymbol{q}}_t$, the MGF attitude estimate is

$$\hat{\mathbf{q}}_t = \delta\mathbf{q}(\delta\hat{\boldsymbol{q}}_t) \otimes \bar{\mathbf{q}}_t \tag{72}$$

Similarly, the MGF represents the true rate as a sum of a deterministic reference rate vector and a rate deviation vector:

$$\boldsymbol{\omega}_t = \delta\boldsymbol{\omega}_t + \bar{\boldsymbol{\omega}}_t \tag{73}$$

where $\delta\boldsymbol{\omega}_t$ is a random variable. Using an estimate $\delta\hat{\boldsymbol{\omega}}_t$, the MGF rate estimate is

$$\hat{\boldsymbol{\omega}}_t = \delta\hat{\boldsymbol{\omega}}_t + \bar{\boldsymbol{\omega}}_t \tag{74}$$

The deviation vector is defined as $\delta\mathbf{x}_t = [\,\delta\hat{\boldsymbol{q}}_t^T \quad \delta\boldsymbol{\omega}_t^T\,]^T$ and the reference vector is defined as $\bar{\mathbf{x}}_t = [\,\bar{\mathbf{q}}_t^T \quad \bar{\boldsymbol{\omega}}_t^T\,]^T$. Rather than directly estimating the satellite attitude and rate as straightforward application of the NGF requires, the MGF estimates the deviation vector and, using the reference vector, provides estimates for the satellite attitude and rate via Eqs. (72) and (74). In doing so, it avoids the difficulties of unit quaternion estimation mentioned previously.

With these relationships in hand, a brief overview of the MGF follows. The MGF begins at time t_n with three components: the initial state estimate $\hat{\mathbf{x}}_n^{+}$, the initial state references $\bar{\mathbf{x}}_n$, and the initial deviation estimate $\delta\hat{\mathbf{x}}_n^{+}$. Like the MEKF, the initial state reference is set equal to the initial state estimate so that the initial deviation is identically zero.

The MGF maps the reference attitude and rate vector forward to an observation time t_{n+1} to determine the predicted reference values $\hat{\mathbf{x}}_{n+1}^{-}$. The state deviation prediction $\delta\hat{\mathbf{x}}_{n+1}^{-}$ is trivially zero. At this stage, the MEKF and the MGF become distinguishable. The MEKF propagates the deviation covariance matrix to t_{n+1} and applies the linear Kalman filter to derive a state deviation estimate $\delta\hat{\mathbf{x}}_{n+1}^{+}$. This estimate represents the minimum variance estimate, assuming a Gaussian a priori density function. In contrast, the MGF (like the NGF) makes no assumptions about the a priori density function during the state estimate step. Instead, it represents the true nonlinear a priori density as a function of the initial density and the system dynamics. Using Bayes's rule, it combines this nonlinear density with the Gaussian observation density to give the nonlinear a posteriori density function. Using an appropriate nonlinear solver, it then seeks the deviation estimate $\delta\hat{\mathbf{x}}_{n+1}^{+}$ that maximizes the a posteriori density.

From here, the MEKF and the MGF use the same rule to update the nonlinear state estimate $\hat{\mathbf{x}}_{n+1}^{+}$. Then the reference state is set equal to the estimated state so that $\delta\hat{\mathbf{x}}_{n+1}^{+}$ is reset to zero, returning the MGF to the same condition in which it began at time t_n.

B. State Mapping

The MGF state update entails nonlinear backward propagation of the deviation vector $\delta\mathbf{x}$. This section defines a geometric mapping for the deviation vector using already-defined mappings for the exact nonlinear state $\mathbf{x}$. The approach is straightforward. An initial state prediction $\bar{\mathbf{x}}_n$ and its value $\bar{\mathbf{x}}_{n+1} = \psi_{t_{n+1}}(\bar{\mathbf{x}}_n)$ at the current time t_{n+1} are known. Given a deviation $\delta\mathbf{x}_{n+1}$ at t_{n+1}, this map seeks to evolve it backward in time to give $\delta\mathbf{x}_n$. First, it composes the deviation $\delta\mathbf{x}_{n+1}$ with the state prediction $\bar{\mathbf{x}}_{n+1}$ to obtain a new deviated state $\check{\mathbf{x}}_{n+1}$. Then it backward propagates the deviated state using a geometric map for the nonlinear attitude state to give $\check{\mathbf{x}}_n$. Finally, it composes $\check{\mathbf{x}}_n$ with the state prediction's initial value $\bar{\mathbf{x}}_n$ to derive the backward-propagated deviation $\delta\mathbf{x}_n$. A more detailed description follows.

To begin with, note that the different operations used to combine a state deviation with a nonlinear state (i.e., composition for quaternions and summation for rates) can be cumbersome. To simplify notation, define a new operator $\oplus$ that "adds" a state deviation to a nonlinear state, such that

$$\delta\mathbf{x} \oplus \mathbf{x} = \begin{bmatrix} \delta\mathbf{q}(\delta\boldsymbol{q}) \otimes \mathbf{q} \\ \delta\boldsymbol{\omega} + \boldsymbol{\omega} \end{bmatrix} \tag{75}$$

and a new operator $\ominus$ that "differences" two nonlinear states $\mathbf{x}_a$ and $\mathbf{x}_b$ to give a state deviation such that

$$\mathbf{x}_a \ominus \mathbf{x}_b = \begin{bmatrix} \delta\boldsymbol{q}(\mathbf{q}_a \otimes \mathbf{q}_b^{*}) \\ \boldsymbol{\omega}_a - \boldsymbol{\omega}_b \end{bmatrix} \tag{76}$$

where $\delta\boldsymbol{q}(\cdot)$ denotes the three-vector portion of the quaternion deviation $(\cdot)$. These operators are related by the identity

$$\delta\mathbf{x} = (\delta\mathbf{x} \oplus \mathbf{x}) \ominus \mathbf{x} \tag{77}$$

Using these operators, a deviation $\delta\mathbf{x}_{n+1}$ and state prediction $\bar{\mathbf{x}}_{n+1}$ at time t_{n+1} may be combined to give a deviated state:

$$\check{\mathbf{x}}_{n+1} = \begin{bmatrix} \check{\mathbf{q}}_{n+1} \\ \check{\boldsymbol{\omega}}_{n+1} \end{bmatrix} = \delta\mathbf{x}_{n+1} \oplus \bar{\mathbf{x}}_{n+1} \tag{78}$$

Then the deviated state may be backward-propagated using a geometric map ψ via

$$\check{\mathbf{x}}_{n} = \psi_{t_n}(\check{\mathbf{x}}_{n+1}) \tag{79}$$

Finally, the backward-propagated deviation may be recovered using the state prediction's initial value $\bar{\mathbf{x}}_n$:

$$\delta\mathbf{x}_n = \begin{bmatrix} \delta\boldsymbol{q}_n \\ \delta\boldsymbol{\omega}_n \end{bmatrix} = \check{\mathbf{x}}_n \ominus \bar{\mathbf{x}}_n \tag{80}$$

To express this evolution in simplified form, define the deviation vector map $\delta\psi$ as

$$\delta\mathbf{x}_n = \delta\psi_{t_n}(\delta\mathbf{x}_{n+1}, \bar{\mathbf{x}}_{n+1}) = \psi_{t_n}(\delta\mathbf{x}_{n+1} \oplus \bar{\mathbf{x}}_{n+1}) \ominus \psi_{t_n}(\bar{\mathbf{x}}_{n+1}) \tag{81}$$

where $\bar{\mathbf{x}}_{n+1}$ is fixed and $\delta\mathbf{x}_{n+1}$ is the state upon which the map operates. Clearly, the map $\delta\psi$ is nothing more than the standard map ψ with slightly modified initial conditions and a subtraction operation to recover the propagated deviation. Using this notation, the backward-propagating geometric map $\delta\boldsymbol{\psi}_{t_n}$ for the MGF may also be defined as

$$\begin{aligned} &\begin{bmatrix} \delta\mathbf{x}_n \\ \Phi(\delta\mathbf{x}_n \oplus \psi_{t_n}(\bar{\mathbf{x}}_{n+1}), \delta\mathbf{x}_{n+1} \oplus \bar{\mathbf{x}}_{n+1}) \end{bmatrix} \\ &\quad = \delta\boldsymbol{\psi}_{t_n}\left(\begin{bmatrix} \delta\mathbf{x}_{n+1} \\ \bar{\mathbf{x}}_{n+1} \\ \Phi(\delta\mathbf{x}_{n+1} \oplus \bar{\mathbf{x}}_{n+1}, \delta\mathbf{x}_{n+1} \oplus \bar{\mathbf{x}}_{n+1}) \end{bmatrix}\right) \\ &\quad = \begin{bmatrix} \delta\psi_{t_n}(\delta\mathbf{x}_{n+1}, \bar{\mathbf{x}}_{n+1}) \\ \Psi_{t_n}(\Phi(\delta\mathbf{x}_{n+1} \oplus \bar{\mathbf{x}}_{n+1}, \delta\mathbf{x}_{n+1} \oplus \bar{\mathbf{x}}_{n+1})) \end{bmatrix} \end{aligned} \tag{82}$$

It is emphasized that $\delta\boldsymbol{\psi}$ does not fundamentally differ from $\boldsymbol{\psi}$ and that they rely on the same underlying nonlinear and first-order maps.

C. State Estimation

The MGF makes the same assumptions as the NGF regarding the Gaussianity of initial and observation pdfs. So the initial state pdf is defined as

$$\wp(\delta\mathbf{x}_n | Z_n) \tag{83a}$$

$$\delta\mathbf{x}_n \sim N\left(0, P_n^+\right) \tag{83b}$$

where $\delta\hat{\mathbf{x}}_n^+ = 0$ because the reference state is set equal to the estimated state after each observation. As with the NGF, the MGF uses the deterministic Fokker–Planck equation to determine the a priori pdf. To do so, it accounts for cumulative diffusion by modifying the initial pdf using $\tilde{Q}_n(\delta\mathbf{x}_{n+1}, \hat{\mathbf{x}}_{n+1}^-)$ as in Eq. (52), where $\tilde{Q}_n$ is dependent on both the predicted state and the deviation state. Then the a priori pdf may be defined in terms of the initial pdf via

$$\wp(\delta\mathbf{x}_{n+1} | Z_n) = \wp\left(\delta\psi_{t_n}\left(\delta\mathbf{x}_{n+1}, \hat{\mathbf{x}}_{n+1}^-\right) | Z_n\right) \tag{84a}$$

$$\delta\psi_{t_n}\left(\delta\mathbf{x}_{n+1}, \hat{\mathbf{x}}_{n+1}^-\right) \sim N\left(0, P_n^+ + \tilde{Q}_n\left(\delta\mathbf{x}_{n+1}, \hat{\mathbf{x}}_{n+1}^-\right)\right) \tag{84b}$$

Given an observation $\mathbf{z}_{n+1}$ and its covariance R_{n+1} and applying Bayes's rule, the a posteriori density takes the form

$$\wp(\delta\mathbf{x}_{n+1} | Z_{n+1}) = ce^{-\frac{1}{2}(\cdot)} \tag{85a}$$

$$\begin{aligned} (\cdot) = &\left[\delta\psi_{t_n}\left(\delta\mathbf{x}_{n+1}, \hat{\mathbf{x}}_{n+1}^-\right)\right]^T \\ &\times \left[P_n^+ + \tilde{Q}_n\left(\delta\mathbf{x}_{n+1}, \hat{\mathbf{x}}_{n+1}^-\right)\right]^{-1}\left[\delta\psi_{t_n}\left(\delta\mathbf{x}_{n+1}, \hat{\mathbf{x}}_{n+1}^-\right)\right] \\ &+ [\mathbf{z}_{n+1} - h(\delta\mathbf{x}_{n+1})]^T R_{n+1}^{-1}[\mathbf{z}_{n+1} - h(\delta\mathbf{x}_{n+1})] \end{aligned} \tag{85b}$$

where c is a constant and may be ignored. Note that this pdf is a function of both $\delta\hat{\mathbf{x}}_n^+$ and $\delta\hat{\mathbf{x}}_{n+1}^-$, but because these are zero by definition, they are excluded for notational convenience. As in the NGF, the maximum-likelihood estimate of the a posteriori density may be solved using the following nonlinear least-squares formulation:

$$\delta\hat{\mathbf{x}}_{n+1}^+ = \underset{\delta\mathbf{x}_{n+1}}{\arg\ \min} \frac{1}{2}\Lambda^T(\delta\mathbf{x}_{n+1})\Lambda(\delta\mathbf{x}_{n+1}) \tag{86}$$

where

$$\Lambda(\delta\mathbf{x}_{n+1}) = S^{-\frac{1}{2}}(\delta\mathbf{x}_{n+1})V(\delta\mathbf{x}_{n+1}) \tag{87a}$$

$$S(\delta\mathbf{x}_{n+1}) = \begin{bmatrix} P_n^+ + \tilde{Q}_n\left(\delta\mathbf{x}_{n+1}, \hat{\mathbf{x}}_{n+1}^-\right) & 0 \\ 0 & R_{n+1} \end{bmatrix} \tag{87b}$$

$$V(\delta\mathbf{x}_{n+1}) = \begin{bmatrix} \delta\psi_{t_n}\left(\delta\mathbf{x}_{n+1}, \hat{\mathbf{x}}_{n+1}^-\right) \\ \mathbf{z}_{n+1} - h(\delta\mathbf{x}_{n+1}) \end{bmatrix} \tag{87c}$$

This least-squares problem may be solved using the Gauss–Newton method, as outlined previously. Once the deviation estimate $\delta\hat{\mathbf{x}}_{n+1}^+$ is known, the nonlinear state may be estimated via

$$\hat{\mathbf{x}}_{n+1}^+ = \delta\hat{\mathbf{x}}_{n+1}^+ \oplus \hat{\mathbf{x}}_{n+1}^- \tag{88}$$

Then the reference state is set equal to $\hat{\mathbf{x}}_{n+1}^+$ and the deviation is reset:

$$\delta\hat{\mathbf{x}}_{n+1}^+ = 0 \tag{89}$$

Covariance propagation is governed by the same equations as in the NGF.

D. Multiplicative Geometric Filter Summary

The multiplicative geometric filter is summarized next. Setting the reference trajectory $\bar{\mathbf{x}}_n$ equal to the initial state estimate $\hat{\mathbf{x}}_n^+$ so that the initial deviation is zero, and given an initial deviation covariance matrix P_n^+, a system observation $\mathbf{z}_{n+1}$ with covariance R_{n+1}, and a geometric map $\boldsymbol{\psi}$, the MGF is governed by the following equations:

Begin loop,

If k=0,

$$\begin{bmatrix} \hat{\mathbf{x}}_{n+1}^- \\ \Phi\left(\hat{\mathbf{x}}_{n+1}^-, \hat{\mathbf{x}}_n^+\right) \end{bmatrix} = \boldsymbol{\psi}_{t_{n+1}}\left(\begin{bmatrix} \hat{\mathbf{x}}_n^+ \\ \Phi\left(\hat{\mathbf{x}}_n^+, \hat{\mathbf{x}}_n^+\right) \end{bmatrix}\right) \tag{90a}$$

$$\Phi\left(\delta\mathbf{x}_{n+1}^0 \oplus \hat{\mathbf{x}}_{n+1}^-\right) = \Phi\left(\hat{\mathbf{x}}_{n+1}^-, \hat{\mathbf{x}}_n^+\right) \tag{90b}$$

by definition,

$$\delta\mathbf{x}_{n+1}^0 = \delta\hat{\mathbf{x}}_n^+ = 0 \tag{90c}$$

and

$$\delta\psi_{t_n}\left(\delta\mathbf{x}_{n+1}^0, \hat{\mathbf{x}}_{n+1}^-\right) = \delta\hat{\mathbf{x}}_n^+ \tag{90d}$$

Else,

$$\begin{bmatrix} \delta\psi_{t_n}(\delta\mathbf{x}^k_{n+1}, \hat{\mathbf{x}}^-_{n+1}) \\ \Phi(\delta\psi_{t_n}(\delta\mathbf{x}^k_{n+1}, \hat{\mathbf{x}}^-_{n+1}) \oplus \hat{\mathbf{x}}^+_n, \delta\mathbf{x}^k_{n+1} \oplus \hat{\mathbf{x}}^-_{n+1}) \end{bmatrix} = \delta\boldsymbol{\psi}_{t_n}\left(\begin{bmatrix} \delta\mathbf{x}^k_{n+1} \\ \hat{\mathbf{x}}^-_{n+1} \\ \Phi(\delta\mathbf{x}^k_{n+1} \oplus \hat{\mathbf{x}}^-_{n+1}, \delta\mathbf{x}^k_{n+1} \oplus \hat{\mathbf{x}}^-_{n+1}) \end{bmatrix}\right) \tag{90e}$$

$$\Phi\left(\delta\mathbf{x}^k_{n+1} \oplus \hat{\mathbf{x}}^-_{n+1}\right) = \Phi^{-1}\left(\delta\psi_{t_n}\left(\delta\mathbf{x}^k_{n+1}, \hat{\mathbf{x}}^-_{n+1}\right) \oplus \hat{\mathbf{x}}^+_n, \delta\mathbf{x}^k_{n+1} \oplus \hat{\mathbf{x}}^-_{n+1}\right) \tag{90f}$$

End if.

$$P^-_{n+1}\left(\delta\mathbf{x}^k_{n+1}\right) = \Phi\left(\delta\mathbf{x}^k_{n+1} \oplus \hat{\mathbf{x}}^-_{n+1}\right)P^+_n\Phi^T\left(\delta\mathbf{x}^k_{n+1} \oplus \hat{\mathbf{x}}^-_{n+1}\right) + Q_{n+1} \tag{90g}$$

$$H\left(\delta\mathbf{x}^k_{n+1}\right) = \left[\frac{\partial}{\partial\mathbf{x}}h(\mathbf{x})\right]\Big|_{\delta\mathbf{x}^k_{n+1}\oplus\hat{\mathbf{x}}^-_{n+1}} \tag{90h}$$

$$K\left(\delta\mathbf{x}^k_{n+1}\right) = P^-_{n+1}\left(\delta\mathbf{x}^k_{n+1}\right)H^T\left(\delta\mathbf{x}^k_{n+1}\right)\Big[H\left(\delta\mathbf{x}^k_{n+1}\right) \times P^-_{n+1}\left(\delta\mathbf{x}^k_{n+1}\right)H^T\left(\delta\mathbf{x}^k_{n+1}\right) + R_{n+1}\Big]^{-1} \tag{90i}$$

$$\delta\mathbf{x}^{k+1}_{n+1} = \delta\mathbf{x}^k_{n+1} - \Phi\left(\delta\mathbf{x}^k_{n+1} \oplus \hat{\mathbf{x}}^-_{n+1}\right)\delta\psi_{t_n}\left(\delta\mathbf{x}^k_{n+1}, \hat{\mathbf{x}}^-_{n+1}\right) + K\left(\delta\mathbf{x}^k_{n+1}\right)\left(\mathbf{z}_{n+1} - h\left(\delta\mathbf{x}^k_{n+1}\right) + H\left(\delta\mathbf{x}^k_{n+1}\right) \times \Phi\left(\delta\mathbf{x}^k_{n+1} \oplus \hat{\mathbf{x}}^-_{n+1}\right)\delta\psi_{t_n}\left(\delta\mathbf{x}^k_{n+1}, \hat{\mathbf{x}}^-_{n+1}\right)\right) \tag{90j}$$

$$k = k + 1 \tag{90k}$$

End loop.

$$\delta\hat{\mathbf{x}}^+_{n+1} = \delta\mathbf{x}^k_{n+1} \tag{90l}$$

$$P^+_{n+1} = \left[I - K\left(\delta\hat{\mathbf{x}}^+_{n+1}\right)H\left(\delta\hat{\mathbf{x}}^+_{n+1}\right)\right]P^-_{n+1}\left(\delta\hat{\mathbf{x}}^+_{n+1}\right) \tag{90m}$$

$$\hat{\mathbf{x}}^+_{n+1} = \delta\hat{\mathbf{x}}^+_{n+1} \oplus \hat{\mathbf{x}}^-_{n+1} \tag{90n}$$

$$\bar{\mathbf{x}}_{n+1} = \hat{\mathbf{x}}^+_{n+1} \tag{90o}$$

$$\therefore \delta\hat{\mathbf{x}}^+_{n+1} = 0 \tag{90p}$$

Although seemingly more complicated than the NGF, the MGF is actually straightforward. However, the simplicity of the algorithm is obscured by cumbersome notation, and so a more thorough explanation is given here. Just as the first iteration of the NGF is equivalent to the EKF, the first iteration of the MGF is equivalent to the MEKF. Also like the NGF, the MGF uses a forward map $\boldsymbol{\psi}_{t_{n+1}}$ in its first iteration and a backward map $\delta\boldsymbol{\psi}_{t_n}$ in all subsequent iterations. Equations (90a–90d) represent the state prediction stage of the MGF's first iteration. This stage simply forward propagates the previous nonlinear state $\hat{\mathbf{x}}^+_n$ to $\hat{\mathbf{x}}^-_{n+1}$ to provide a reference trajectory. Equation (90b) provides a standard simplified notation in place of the full state transition matrix notation, which is overly cumbersome; it does not require any computation. Equations (90c) and (90d) make explicit that in the first MGF iteration, the initial and predicted deviations are zero by definition and that they belong to the same trajectory; as with Eq. (90b), these equations are provided for notational convenience and do not require computation.

Equations (90e) and (90f) represent the state prediction stage of MGF iterations subsequent to the first iteration. The first iteration and subsequent iterations differ in two important ways: the first iteration *forward propagates* the *nonlinear reference state* $\mathbf{x}^+_n$, whereas subsequent iterations *backward propagate* the *deviation state* $\delta\mathbf{x}^k_{n+1}$. On the surface, these may seem to require two completely separate algorithms, but, in fact, they do not. The backward deviation map $\delta\boldsymbol{\psi}_{t_n}$ in Eq. (90e) is based fundamentally on the backward map $\boldsymbol{\psi}_{t_n}$ [see Eq. (82)]; as noted for the NGF, the backward map $\boldsymbol{\psi}_{t_n}$ and the forward map $\boldsymbol{\psi}_{t_{n+1}}$ [central to the MGF's first iteration in Eq. (90a)] are distinguished simply by the sign of the time step used. Therefore, thoughtful implementation of the MGF equations requires only a forward map $\boldsymbol{\psi}_{t_{n+1}}$ and algorithms for the $\oplus$ and $\ominus$ operators; the backward deviation map $\delta\boldsymbol{\psi}_{t_n}$ may be derived from these algorithms alone.

Similar to the NGF, MGF iterations subsequent to the first iteration require a backward-propagated deviation state $\delta\psi_{t_n}(\delta\mathbf{x}^k_{n+1}, \hat{\mathbf{x}}^-_{n+1})$ in conjunction with a forward-mapping state transition matrix $\Phi(\delta\mathbf{x}^k_{n+1} \oplus \hat{\mathbf{x}}^-_{n+1})$ [see, for instance, Eq. (90j)]. However, standard implementation of a backward map $\delta\boldsymbol{\psi}_{t_n}$ provides a backward-mapping state transition matrix, as seen in Eq. (90e). Therefore, Eq. (90f) provides the desired state transition matrix via matrix inversion. However, as with the NGF, the backward map $\delta\boldsymbol{\psi}_{t_n}$ can be easily modified to provide a backward-propagated state and a forward-mapping state transition matrix with no additional computations, thereby eliminating the need for Eq. (90f). In either case, Eq. (90f) is useful in that it simplifies the state transition matrix notation in the same way as Eq. (90b).

Equations (90g–90k) represent the state-update stage of a given MGF iteration and are generally self-explanatory. Equations (90l–90p) are the final steps of the MGF. In particular, Eq. (90o) makes explicit the trajectory reset of the MGF, from which Eq. (90p) follows. Neither of these last two equations require explicit computations.

E. Quaternion Observation Model

To incorporate quaternion observations, recall the MGF's a posteriori density function from Eq. (85b) and, in particular, its observation error term:

$$\Delta\mathbf{z} = \mathbf{z} - h(\delta\mathbf{x}) \tag{91}$$

This formulation is problematic when $\mathbf{z} = \mathbf{z}_\mathbf{q}$ is a quaternion observation, because subtraction is not an appropriate quaternion function. Therefore, an observation deviation is derived and presented to the MGF for filtering. Define the quaternion observation as a composition of a quaternion deviation and the state prediction:

$$\mathbf{z}_\mathbf{q} = \delta\mathbf{q}(\mathbf{z}_{\delta\boldsymbol{q}}) \otimes \hat{\mathbf{q}}^- \tag{92}$$

Then the three-vector of the quaternion deviation may be expressed as

$$\mathbf{z}_{\delta\boldsymbol{q}} = \delta\boldsymbol{q}(\mathbf{z}_\mathbf{q} \otimes (\hat{\mathbf{q}}^-)^*) \tag{93}$$

Returning to the MGF observation error term and using $\mathbf{z}_{\delta\boldsymbol{q}}$ as the system observation,

$$\Delta\mathbf{z} = \mathbf{z}_{\delta\boldsymbol{q}} - \delta\boldsymbol{q} \tag{94}$$

where

$$h(\delta\mathbf{x}) = H\delta\mathbf{x} = \delta\boldsymbol{q} \tag{95}$$

and $H = [\,\mathrm{I}_{3\times3} \quad 0_{3\times3}\,]$. In MGF implementation, Eqs. (93) and (94) are used in place of the observation error term from Eq. (91) found in the MGF state update [Eq. (90j)]. Explicitly, the MGF state update becomes

$$\delta\mathbf{x}_{n+1}^{k+1} = \delta\mathbf{x}_{n+1}^{k} - \Phi\left(\delta\mathbf{x}_{n+1}^{k} \oplus \hat{\mathbf{x}}_{n+1}^{-}\right)\delta\psi_{t_n}\left(\delta\mathbf{x}_{n+1}^{k}, \hat{\mathbf{x}}_{n+1}^{-}\right) + K\left(\delta\mathbf{x}_{n+1}^{k}\right)\left(\delta\boldsymbol{q}\left(\mathbf{z}_{\mathbf{q}_{n+1}} \otimes \left(\hat{\mathbf{q}}_{n+1}^{-}\right)^{*}\right) - \delta\boldsymbol{q}_{n+1}^{k} + [\mathrm{I}_{3\times 3} \quad 0_{3\times 3}]\Phi\left(\delta\mathbf{x}_{n+1}^{k} \oplus \hat{\mathbf{x}}_{n+1}^{-}\right)\delta\psi_{t_n}\left(\delta\mathbf{x}_{n+1}^{k}, \hat{\mathbf{x}}_{n+1}^{-}\right)\right) \quad (96)$$

VI. Iterated Kalman Filter Smoother

In the next section, comparisons are made between multiplicative versions of the NGF and the IKFS [4]. The IKFS is an appropriate benchmark because both the NGF and IKFS seek to reduce errors resulting from the dynamic model. Both accomplish this by iteratively solving for the mode of the a posteriori density as the system's state estimate. In contrast to the IKFS, however, the NGF does not smooth backward per se, but incorporates backward propagation as a fundamental part of the state update.

Like the NGF, the IKFS may be thought of as nonlinear least-squares problem solved iteratively by the Gauss–Newton algorithm [4]. The least-squares formulation of the IKFS is [43]

$$\left(\hat{\mathbf{x}}_n^{+}, \hat{\mathbf{x}}_{n+1}^{+}\right) = \underset{(\mathbf{x}_n, \mathbf{x}_{n+1})}{\arg\min} \frac{1}{2}\Lambda^{T}(\mathbf{x}_n, \mathbf{x}_{n+1})\Lambda(\mathbf{x}_n, \mathbf{x}_{n+1}) \quad (97)$$

where

$$\Lambda(\mathbf{x}_n, \mathbf{x}_{n+1}) = S_{n+1}^{-\frac{1}{2}}V(\mathbf{x}_n, \mathbf{x}_{n+1}) \quad (98a)$$

$$S_{n+1} = \begin{bmatrix} P_n^{+} & 0 & 0 \\ 0 & Q_{n+1} & 0 \\ 0 & 0 & R_{n+1} \end{bmatrix} \quad (98b)$$

$$V(\mathbf{x}_n, \mathbf{x}_{n+1}) = \begin{bmatrix} \mathbf{x}_n \\ \mathbf{x}_{n+1} \\ \mathbf{z}_{n+1} \end{bmatrix} - \begin{bmatrix} \hat{\mathbf{x}}_n^{+} \\ \mathbf{x}_n + \int_{t_n}^{t_{n+1}} f(\mathbf{x}_t)\,\mathrm{d}t \\ h(\mathbf{x}_{n+1}) \end{bmatrix} \quad (98c)$$

Note that in standard form, the IKFS uses nongeometric maps to propagate the state. As a result, state propagation is represented by an integral in Eq. (98c), in contrast to the map notation used in the NGF and MGF least-squares formulations.

Application of the Gauss–Newton solver to this least-squares problem gives the final form of the IKFS [4]:

Begin loop,

If k=0,

$$\mathbf{x}_n^{\,0} = \hat{\mathbf{x}}_n^{+} \quad (99a)$$

$$\mathbf{x}_{n+1}^{\,0} = \hat{\mathbf{x}}_n^{+} + \int_{t_n}^{t_{n+1}} f\left(\hat{\mathbf{x}}_t^{+}\right)\mathrm{d}t \quad (99b)$$

$$\bar{\mathbf{x}}_{n+1} = \mathbf{x}_{n+1}^{0} \quad (99c)$$

Else,

$$\bar{\mathbf{x}}_{n+1} = \mathbf{x}_n^{k} + \int_{t_n}^{t_{n+1}} f\left(\mathbf{x}_t^{k}\right)\mathrm{d}t \quad (99d)$$

End if.

$$F(\bar{\mathbf{x}}_t) = \left[\frac{\partial}{\partial\mathbf{x}}f(\mathbf{x})\right]\Big|_{\bar{\mathbf{x}}_t} \quad (99e)$$

$$\frac{\mathrm{d}\Phi\left(\bar{\mathbf{x}}_t, \mathbf{x}_n^{k}\right)}{\mathrm{d}t} = F(\bar{\mathbf{x}}_t)\Phi\left(\bar{\mathbf{x}}_t, \mathbf{x}_n^{k}\right) \quad (99f)$$

$$\Phi(\bar{\mathbf{x}}_{n+1}) = \Phi\left(\bar{\mathbf{x}}_{n+1}, \mathbf{x}_n^{k}\right) \quad (99g)$$

$$P_{n+1}^{-}(\bar{\mathbf{x}}_{n+1}) = \Phi(\bar{\mathbf{x}}_{n+1})P_n^{+}\Phi^{T}(\bar{\mathbf{x}}_{n+1}) + Q_{n+1} \quad (99h)$$

$$H\left(\mathbf{x}_{n+1}^{k}\right) = \left[\frac{\partial}{\partial\mathbf{x}}h(\mathbf{x})\right]\Big|_{\mathbf{x}_{n+1}^{k}} \quad (99i)$$

$$K\left(\mathbf{x}_{n+1}^{k}\right) = P_{n+1}^{-}\left(\bar{\mathbf{x}}_{n+1}\right)H^{T}\left(\mathbf{x}_{n+1}^{k}\right) \times \left[H\left(\mathbf{x}_{n+1}^{k}\right)P_{n+1}^{-}(\bar{\mathbf{x}}_{n+1})H^{T}\left(\mathbf{x}_{n+1}^{k}\right) + R_{n+1}\right]^{-1} \quad (99j)$$

$$S(\bar{\mathbf{x}}_{n+1}) = P_n^{+}\Phi^{T}(\bar{\mathbf{x}}_{n+1})P_{n+1}^{-1}(\bar{\mathbf{x}}_{n+1}) \quad (99k)$$

$$\mathbf{x}_{n+1}^{k+1} = \bar{\mathbf{x}}_{n+1} + \Phi(\bar{\mathbf{x}}_{n+1})\left(\mathbf{x}_n^{0} - \mathbf{x}_n^{k}\right) + K\left(\mathbf{x}_{n+1}^{k}\right)\left(\mathbf{z}_{n+1} - h\left(\mathbf{x}_{n+1}^{k}\right) - H\left(\mathbf{x}_{n+1}^{k}\right)\left(\bar{\mathbf{x}}_{n+1} + \Phi(\bar{\mathbf{x}}_{n+1})\left(\mathbf{x}_n^{0} - \mathbf{x}_n^{k}\right) - \mathbf{x}_{n+1}^{k}\right)\right) \quad (99l)$$

$$\mathbf{x}_n^{\,k+1} = \mathbf{x}_n^{0} + S(\bar{\mathbf{x}}_{n+1})\left(\mathbf{x}_{n+1}^{k+1} - \bar{\mathbf{x}}_{n+1} - \Phi(\bar{\mathbf{x}}_{n+1})\left(\mathbf{x}_n^{0} - \mathbf{x}_n^{k}\right)\right) \quad (99m)$$

$$k = k + 1 \quad (99n)$$

End loop.

$$\hat{\mathbf{x}}_n^{\,+} = \mathbf{x}_n^{k} \quad (99o)$$

$$\hat{\mathbf{x}}_{n+1}^{\,+} = \mathbf{x}_{n+1}^{k} \quad (99p)$$

$$P_{n+1}^{+} = \left[I - K\left(\hat{\mathbf{x}}_{n+1}^{+}\right)H\left(\hat{\mathbf{x}}_{n+1}^{+}\right)\right]P_{n+1}^{-}\left(\hat{\mathbf{x}}_{n+1}^{+}\right) \quad (99q)$$

Note that the first iteration of the IKFS is equivalent to the EKF. Unlike the NGF, the IKFS explicitly calculates a new value for the previous state estimate $\hat{\mathbf{x}}_n^{+}$. This requires the calculation of a smoothing matrix S in Eq. (99k), which will be shown in the next section to substantially increase the computational burden of the IKFS relative to the NGF.

To place the NGF in context, compare the state updates of the NGF in Eq. (59) and the IKFS in Eq. (99l). They are equivalent when it holds that

$$\Phi\left(\mathbf{x}_{n+1}^{k}\right)\left(\psi_{t_n}\left(\mathbf{x}_{n+1}^{k}\right) - \hat{\mathbf{x}}_n^{+}\right) = -\bar{\mathbf{x}}_{n+1} - \Phi(\bar{\mathbf{x}}_{n+1})\left(\mathbf{x}_n^{0} - \mathbf{x}_n^{k}\right) + \mathbf{x}_{n+1}^{k} \quad (100)$$

In the absence of smoothing, that is,

$$\mathbf{x}_n^{\,k} = \hat{\mathbf{x}}_n^{+} \quad (101)$$

then the IKFS is equivalent to the well-known iterated extended Kalman filter (IEKF). The IEKF and the NGF state updates are equivalent when it holds that

$$\Phi\left(\mathbf{x}_{n+1}^{k}\right)\left(\psi_{t_n}\left(\mathbf{x}_{n+1}^{k}\right) - \hat{\mathbf{x}}_n^{+}\right) = \mathbf{x}_{n+1}^{k} - \bar{\mathbf{x}}_{n+1} \quad (102)$$

where

$$\bar{\mathbf{x}}_{n+1} = \hat{\mathbf{x}}_n^{+} + \int_{t_n}^{t_{n+1}} f(\mathbf{x}_t)\,\mathrm{d}t \quad (103)$$

Both sides of Eqs. (100) and (102) are approximations to the predicted state estimation error at t_{n+1}. The fact that the NGF state

update is in good agreement with the IKFS and IEKF supports the theoretical underpinnings of the NGF. However, the algorithmic similarities of the NGF, the IEKF, and the IKFS do not necessarily translate to similarities in performance. Geometric integrators provide a good case in point. On initial inspection, geometric and nongeometric integrators of the same order are often difficult to distinguish from one another; however, when applied to Hamiltonian systems, they have the potential to behave in substantially different ways, due to the geometric integrators' preservation of conserved properties. Likewise, the similarities between the geometric and nongeometric filters developed in this paper belie their potential performance disparities when applied to Hamiltonian systems.

VII. Multiplicative Geometric Filter Results

In this section, comparisons are made between the MGF and the IKFS. In these comparisons, the IKFS was adapted to estimate the vector portion of a stochastic quaternion deviation like the MEKF and the MGF. The first iteration of this multiplicative IKFS is precisely the MEKF algorithm. The IKFS uses a second-order Runge–Kutta integrator for state propagation, and the MGF uses a second-order implicit midpoint rule integrator that is geometric in nature. Both of these integrators are detailed in [12,21].

To compare the MGF and IKFS, realistic satellite configurations and mission profiles were selected for simulation. A representative case is presented here based on the recently launched U.S. Air Force Academy FalconSAT-3 small-satellite mission. This is modeled as a 50-kg satellite in a 560-km, moderately inclined, circular orbit with a high-accuracy observation device substituted for the low-accuracy magnetometer onboard the actual satellite. Instead, the advanced stellar compass is modeled, due to its successful implementation as a rate-gyro replacement onboard an agile resource-constrained small satellite [2]. It has the ability to provide frequent high-accuracy quaternion-based attitude observations at fast rotation rates, and so it is ideal for this investigation. The satellite inertia tensor has diagonal entries of 5, 4, and 2 $\mathrm{kg\,m^2}$, in the roll, pitch, and yaw axes, respectively; the internal angular momentum is zero; and all disturbance torques save for gravity gradient torque are ignored, for simplicity.

The initial satellite attitude is 60, 30, and 30 deg from the orbital frame in the roll, pitch, and yaw axes, respectively, (i.e., a 2-1-3 Euler angle set), and the initial rate is set to 3, 5, and -4 deg $\mathrm{s^{-1}}$ with respect to the inertial frame in the roll, pitch, and yaw axes, respectively. The initial filter attitude estimate is coincident with the orbital frame (i.e., 0 deg in all axes) and the initial filter rate estimate is 0.1 deg $\mathrm{s^{-1}}$ with respect to the inertial frame in the yaw axis. Initial filter estimates of attitude standard deviations are set to 4 deg in each axis, and initial filter estimates of rate standard deviations are set to 3 deg $\mathrm{s^{-1}}$ in each axis. Both filters use a state propagation step of 0.1 s. Observation noise is set to 100 arcseconds (0.028 deg, 1σ) at a frequency of 0.2 Hz. Process noise is tuned separately in each filter to achieve optimal performance, defined here as minimum steady-state attitude error. The resulting process noise values are $\sigma_Q = (1.1 \times 10^{-6})$ deg $\mathrm{s^{-2}}$ for the MGF and $\sigma_Q = (3.6 \times 10^{-6})$ deg $\mathrm{s^{-2}}$ for the IKFS. This represents a worst-case scenario in which each filter is confident in its attitude estimates though they are significantly biased from the true values. In addition, the nonlinearity of the dynamics is significant considering that the satellite is rotating with a magnitude of 7 deg $\mathrm{s^{-1}}$ so that between observations, the satellite changes angular position by roughly 35 deg. To reflect the constrained computational capabilities of small satellites, the number of filter recursions per observation was limited to three.

To compare the MGF and IKFS, 30 simulation runs were conducted using the preceding initial conditions and stochastically determining the attitude observation values for each run. Figure 2 shows the normed root mean square (rms) attitude errors from each filter, and Fig. 3 shows the normed rms rate errors from each filter. As expected, both filters provide state estimates that are more accurate than the observations they receive. However, in the steady state, the MGF attitude and rate errors are approximately 3 times smaller than the IKFS attitude and rate errors. Whereas the IKFS attitude errors

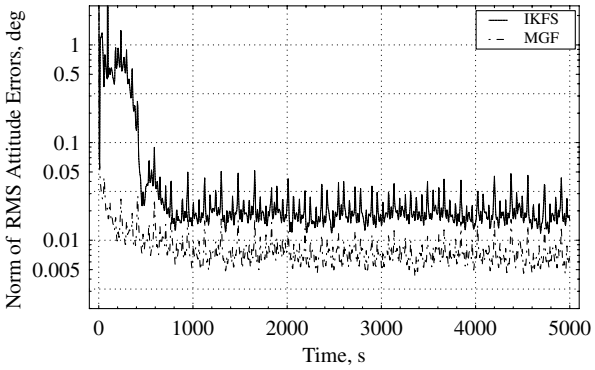

Fig. 2 Comparison of IKFS and MGF normed rms attitude errors.

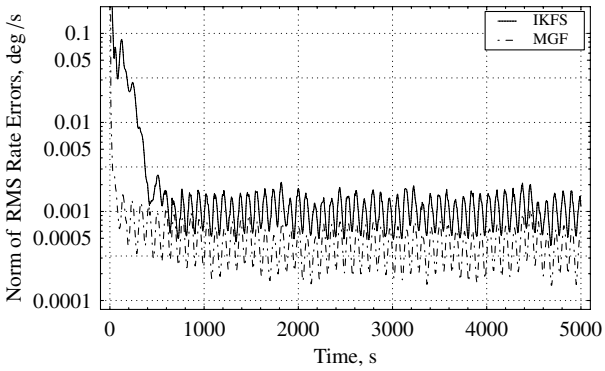

Fig. 3 Comparison of IKFS and MGF normed rms rate errors.

are only slightly less than one standard deviation of observation noise, the MGF affords a significant reduction in noise using the same number of filter iterations. The Hamiltonian errors in Fig. 4 reflect trends that are similar to the attitude and rate results. In this case, the MGF's relative Hamiltonian errors are a magnitude of order smaller than the IKFS' errors, despite using state propagators of the same order. As this simulation demonstrates, one of geometric estimation's strengths is its ability to conserve the same system properties that its dynamic models are designed to preserve[21].

Figure 5 shows each filter's estimated 3σ covariance bounds and rms state errors for the yaw axis. In the steady state, both filters correctly estimate their own covariance bounds. In the convergent phase, however, the IKFS is overconfident in its state estimates, with its actual error regularly an order of magnitude larger than its estimated 3σ covariance bounds. This results in a prolonged convergence period relative to the MGF. At the heart of this behavior is the process noise, which is well known to play an integral role in filter tuning. To achieve minimal steady-state attitude errors, the IKFS process noise was reduced to a minimum, leading the filter to be more confident in its own state estimates at the expense of attitude observations. In the presence of unmodeled torques, the convergence

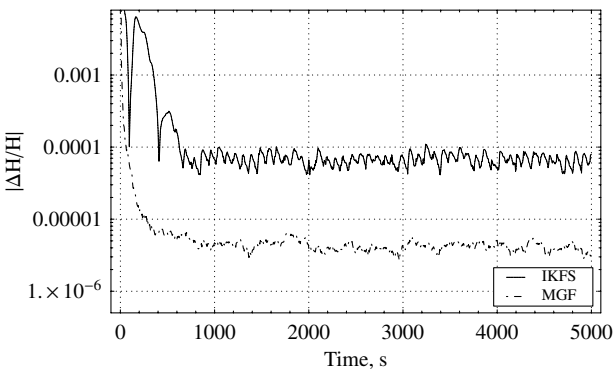

Fig. 4 Comparison of IKFS and MGF relative Hamiltonian errors.

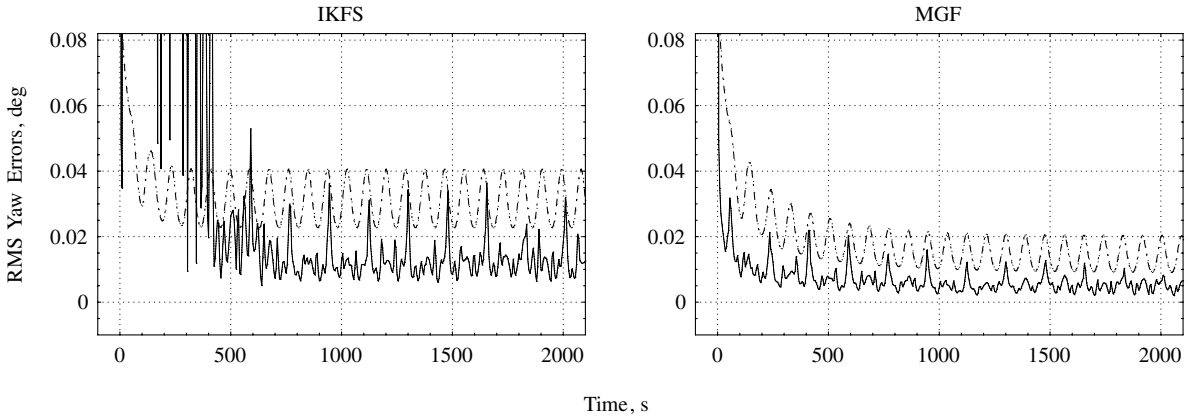

Fig. 5 Comparison of IKFS and MGF estimated covariance bounds.

behavior demonstrated by the IKFS in this simulation could lead to filter divergence, due to an overconfidence in its own state model and a rejection of attitude observations to the contrary. To make the IKFS more robust, one would increase the process noise value, thereby decreasing confidence in the state estimate. However, the steady-state accuracy of the filter will generally suffer as a result. In comparison, the MGF's performance in this case is encouraging: even when tuned for minimal steady-state attitude errors (i.e., when its process noise value is at a minimum), it is capable of correctly estimating its own error bounds. As a result, the MGF requires less process noise (and hence, less degradation of steady-state performance) to ensure convergence.

To conclude this section, consider the computational burden of the IKFS and the MGF. Table 1 presents the floating-point operation (FLOP) count for each algorithm, ignoring the calculation of constant values and using FLOP counting rules in accordance with NASA's Advanced Supercomputing Division (NAS) standards [44], found in Table 2.

Table 1 accounts for one full evaluation of each algorithm, where h is the number of recursions and C_{ψ} is the FLOP count for the particular nonlinear map and STM used in the estimation algorithm. In the case of $h = 1$, both the IKFS and the MGF algorithms are equivalent to the MEKF; therefore, aside from the FLOP count of their respective state maps, the one-recursion IKFS and MGF have the same FLOP count (the apparent disparity in Table 1 is due to the fact that the one-recursion IKFS does not smooth backward). Given multiple recursions, the IKFS requires significantly more calculations because, in addition to calculating a Kalman gain matrix, it calculates a smoothing gain matrix for each recursion. This incurs both a 6×6 matrix multiplication and inversion of the P matrix, which the MGF does not require. In addition, given high-accuracy observations, P values become quite small and inversion can lead to significant numerical complications. Ignoring the nonlinear map and STM calculations, the MGF uses roughly 35% fewer computations per recursion than the IKFS and avoids difficulties associated with covariance matrix inversion.

Table 1 FLOP count per h recursions of the attitude estimators

Algorithm	Total FLOP count
IKFS	$237 + (2370 + C_{\psi})h$
MGF	$237 + (1499 + C_{\psi})h$

Table 2 NAS rules for counting FLOP

Operation	FLOP count
$+, -, *, =$	1
$/, \sqrt{}$	4
sin, cos, exp	8

VIII. Conclusions

In this paper, a new estimation algorithm for Hamiltonian systems was derived. This nonlinear geometric filter is based on the insight that all higher-order moments of a deterministic Hamiltonian system's state probability density function can be preserved as it evolves in time with the use of a geometric integrator. This insight enables the NGF to accurately estimate the state and mitigate the effects of dynamic nonlinearity. In addition, the use of a geometric integrator enables the NGF to preserve physical properties of Hamiltonian motion during state propagation. The NGF was adapted to the case of satellite attitude and, in particular, to the estimation of attitude quaternions, resulting in the multiplicative geometric filter. In a Monte Carlo-based small-satellite simulation, the MGF produced significantly reduced state errors, more accurate covariance bound estimation, and reduced computational burden when compared with the iterated Kalman filter smoother. Considering these advantages, the authors conclude that this new method shows promise for improved attitude estimation onboard high-performance, resource-constrained small satellites.

Acknowledgments

This material is based upon work supported under a National Science Foundation Graduate Research Fellowship, a George C. Marshall Scholarship, and a Harry S. Truman Scholarship. The views expressed in this article are those of the authors and do not reflect the official policy or position of the U.S. Air Force, U.S. Department of Defense, or the U.S. Government. The authors would like to thank F. Markley and M. Roberts for their thorough review of this work and Y. Hashida of Surrey Satellite Technology Ltd. (SSTL) for his many contributions throughout the research effort.

References

[1] Robertson, B., and Stoneking, E., "Satellite GN&C Anomaly Trends," AAS Guidance and Control Conference, Breckenridge, CO, American Astronautical Society Paper AAS 03-071, 2003.

[2] Jorgensen, J. L., Denver, T., Betto, M., and den Braembussche, P. V., "The PROBA Satellite Star Tracker Performance," *Acta Astronautica*, Vol. 56, Nos. 1–2, Jan. 2005, 153–159. doi:10.1016/j.actaastro.2004.09.011

[3] Athans, M., Wishner, R. P., and Bertolini, A., "Suboptimal State Estimation for Continuous-Time Nonlinear Systems from Discrete Noisy Measurements," *IEEE Transactions on Automatic Control*, Vol. 13, No. 5, 1968, pp. 504–514. doi:10.1109/TAC.1968.1098986

[4] Jazwinski, A. H., *Stochastic Processes and Filtering Theory*, Mathematics in Science and Engineering, 1st ed., Vol. 64, Academic Press, London, 1970.

[5] Kushner, H. J., "Nonlinear Filtering: The Exact Dynamical Equations Satisfied by the Conditional Mode," *IEEE Transactions on Automatic Control*, Vol. 12, No. 3, 1967, pp. 262–267. doi:10.1109/TAC.1967.1098582

[6] Bellaire, R. L., "Nonlinear Estimation with Applications to Target Tracking," Ph.D. Thesis, Georgia Inst. of Technology, Atlanta, GA, 1996.

[7] Haykin, S., and Freitas, N. D., "Special Issue on Sequential State Estimation," *Proceedings of the IEEE*, Vol. 92, No. 3, 2004, pp. 399–400. doi:10.1109/JPROC.2003.823140

[8] Julier, S., Uhlmann, J., and Durrant-Whyte, H. F., "A New Method for the Nonlinear Transformation of Means and Covariances in Filters and Estimators," *IEEE Transactions on Automatic Control*, Vol. 45, No. 3, 2000, pp. 477–482. doi:10.1109/9.847726

[9] Gelb, A., (ed.), *Applied Optimal Estimation*, M.I.T. Press, Cambridge, MA, 1974.

[10] Brown, R. G., and Hwang, P. Y. C., *Introduction to Random Signals and Applied Kalman Filtering*, 3rd ed., Wiley, New York, 1997.

[11] Vallado, D. A., *Fundamentals of Astrodynamics and Applications*, 2nd ed., Microcosm Press, London, 2001.

[12] Palmer, P., Mikkola, S., and Hashida, Y., "A Simple High Accuracy Integrator for Spacecraft Attitude Systems," AIAA Guidance, Navigation, and Control Conference, Providence, RI, AIAA Paper 2004-5339, 2004.

[13] Arulampalam, M. S., Maskell, S., Gordon, N., and Clapp, T., "A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking," *IEEE Transactions on Signal Processing*, Vol. 50, No. 2, 2002, pp. 174–188. doi:10.1109/78.978374

[14] Challa, S., and Bar-Shalom, Y., "Nonlinear Filter Design Using Fokker–Planck-Kolmogorov Probability Density Evolutions," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 36, No. 1, 2000, pp. 309–315. doi:10.1109/7.826335

[15] Vathsal, S., "Spacecraft Attitude Determination Using a Second-Order Nonlinear Filter," *Journal of Guidance, Control, and Dynamics*, Vol. 10, No. 6, 1987, pp. 559–566.

[16] Lappas, V. J., Steyn, W. H., and Underwood, C. I., "Experimental Testing of a CMG Cluster for Agile Microsatellites," 11th IEEE Mediterranean Conference on Control and Automation, Rhodes, Greece, Inst. of Electrical and Electronics Engineers Paper T7-143, 2003.

[17] Mikkola, S., Palmer, P., and Hashida, Y., "A Symplectic Orbital Estimator for Direct Tracking on Satellites," *Journal of the Astronautical Sciences*, Vol. 48, No. 1, 1999, pp. 109–125.

[18] Imre, E., and Palmer, P., "High Precision Symplectic Numerical Relative Orbit Propagation," *Journal of Guidance, Control, and Dynamics*, Vol. 30, No. 4, 2007, pp. 965–973. doi:10.2514/1.26846

[19] Lee, T., Sanyal, A., Leok, M., and McClamroch, N. H., "Global Optimal Attitude Estimation Using Uncertainty Ellipsoids," *Systems and Control Letters*, Vol. 57, No. 3, 2008, pp. 236–245. doi:10.1016/j.sysconle.2007.08.014

[20] Lee, T., Sanyal, A., Leok, M., and McClamroch, N. H., "Deterministic Global Attitude Estimation," *45th IEEE Conference on Decision and Control*, Inst. of Electrical and Electronics Engineers, Piscataway, NJ, 2006, pp. 3174–3179.

[21] Valpiani, J., "Geometric Attitude Estimation for Small Satellites," Ph.D. Thesis, Univ. of Surrey, Guildford, Surrey, England, U.K., 2007.

[22] Scheeres, D. J., Hsiao, F. Y., Parrk, R. S., Villac, B. F., and Maruskin, J. M., "Fundamental Limits on Spacecraft Orbit Uncertainty and Distribution Propagation," The Malcolm D. Shuster Astronautics Symposium, American Astronautical Society Paper 05-471, 2005.

[23] Lefferts, E. J., Markley, F. L., and Shuster, M. D., "Kalman Filtering for Spacecraft Attitude Estimation," *Journal of Guidance, Control, and Dynamics*, Vol. 5, No. 5, 1982, pp. 417–429.

[24] Markley, F. L., "Attitude Error Representations for Kalman Filtering," *Journal of Guidance, Control, and Dynamics*, Vol. 26, No. 2, 2003, pp. 311–317.

[25] Crassidis, J. L., Markley, F. L., and Cheng, Y., "Survey of Nonlinear Attitude Estimation Methods," *Journal of Guidance, Control, and Dynamics*, Vol. 30, No. 1, 2007, pp. 12–28. doi:10.2514/1.22452

[26] Beck, J. A., and Hall, C. D., "Relative Equilibria of a Rigid Satellite in a Circular Keplerian Orbit," *Journal of the Astronautical Sciences*, Vol. 46, No. 3, 1998, pp. 215–247.

[27] Arnold, V. I., *Mathematical Models of Classical Methods*, 2nd ed., Graduate Texts in Mathematics, Springer, London, 1974.

[28] McLachlan, R. I., "Explicit Lie-Poisson Integration and the Euler Equations," *Physical Review Letters*, Vol. 71, No. 19, 1993, pp. 3043–3046. doi:10.1103/PhysRevLett.71.3043

[29] Leimkuhler, B., and Reich, S., *Simulating Hamiltonian Dynamics*, Cambridge Monographs on Applied and Computational Mathematics, Cambridge Univ. Press, Cambridge, England, U.K., 2004.

[30] Donnelly, D., and Rogers, E., "Symplectic Integrators: An Introduction," *American Journal of Physics*, Vol. 73, No. 10, 2005, pp. 938–945. doi:10.1119/1.2034523

[31] Hayes, W., "A Brief Survey of Issues Relating to the Reliability of Simulation of the Large Gravitational *N*-Body Problem," Ph.D. Thesis, Univ. of Toronto, Toronto, Ontario, Canada, 1996.

[32] McLachlan, R. I., and Quispel, G. R., "Six Lectures on the Geometric Integration of ODEs," *Foundations of Computational Mathematics*, Cambridge Univ. Press, Cambridge, England, U.K., 1999.

[33] McLachlan, R. I., and Quispel, G. R., "Geometric Integrators for ODEs," *Journal of Physics A*, Vol. 39, May 2006, pp. 5251–5285. doi:10.1088/0305-4470/39/19/S01

[34] Imre, E., and Palmer, P., "A Numerical Approach to High Precision Numerical Relative Orbit Propagation," AAS/AIAA Astrodynamics Specialist Conference, Lake Tahoe, CA, American Astronautical Society Paper 05-296, 2005.

[35] O'Donnell, K., "Satellite Orbits in Resonance with Tesseral Harmonics: Absolute and Relative Orbit Analysis," Ph.D. Thesis, Univ. of Surrey, Guildford, Surrey, England, U.K., 2006.

[36] Karasopoulos, H., and Richardson, D. L., "Chaos in the Pitch Equation of Motion for the Gravity-Gradient Satellite," *AIAA/AAS Astrodynamics Conference*, AIAA, Washington, DC, 1992, pp. 53–65.

[37] Park, R. S., and Scheeres, D. J., "Nonlinear Mapping of Gaussian State Uncertainties: Theory and Applications to Spacecraft Control and Navigation," AAS/AIAA Astrodynamics Specialist Conference, Lake Tahoe, CA, American Astronautical Society Paper 05-404, 2005.

[38] Sergi, A., "Variational Principle and Phase Space Measure in Non-Canonical Coordinates," *Atti Della Accademia Peloritana Dei Pericolanti*, Vol. 83, Nov. 2005.

[39] Sergi, A., and Giaquinta, P., "On the Geometry and Entropy of Non-Hamiltonian Phase Space," *Journal of Statistical Mechanics*, Vol. 2007, Feb. 2007, pp. P02013. doi:10.1088/1742-5468/2007/02/P02013

[40] Wertz, J. R. (ed.), *Spacecraft Attitude Determination and Control*, Kluwer Academic, London, 1978.

[41] Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P., *Numerical Recipes in C*, 2nd ed., Cambridge Univ. Press, Cambridge, England, U.K., 1992.

[42] Markley, F. L., "Attitude Estimation or Quaternion Estimation?" *Journal of the Astronautical Sciences*, Vol. 52, No. 1, 2004, pp. 221–238.

[43] Poore, A. B., Slocumb, B. J., Suchomel, B. J., Obermeyer, F. H., Herman, S. M., and Gadaleta, S. M., "Batch Maximum Likelihood (ML) and Maximum a Posteriori (MAP) Estimation with Process Noise for Tracking Applications," *Signal and Data Processing of Small Targets*, edited by O. E. Drummond, Vol. 5204, Society for Optical Engineering, San Diego, CA, 2003, pp. 188–199.

[44] Centre for Development of Advanced Computing, "Performance Metrics and Scalability Analysis," *Parallel Computing Workshop*, Indian Inst. of Technology, Delhi, India, 2002.