JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 28, No. 5, September–October 2005

Backward-Smoothing Extended Kalman Filter

Mark L. Psiaki*
Cornell University, Ithaca, New York 14853-7501

The principle of the iterated extended Kalman filter has been generalized to create a new filter that has superior performance when the estimation problem contains severe nonlinearities. The new filter is useful when nonlinearities might significantly degrade the accuracy or convergence reliability of other filters. The new filter solves a nonlinear smoothing problem for the current and past sample intervals using iterative numerical techniques. This approach retains the nonlinearities of a fixed number of stages that precede the stage of interest, and it processes information from earlier stages in an approximate manner. The algorithm has been simulation tested on a difficult spacecraft attitude estimation problem that includes sensing of fewer than three axes and significant dynamic model uncertainty. The filter compensates for this uncertainty via simultaneous estimation of moment of inertia parameters. The new filter exhibits markedly better convergence reliability and accuracy than an extended Kalman filter and an unscented Kalman filter for estimation problems that start with large initial attitude or attitude rate errors.

I. Introduction

A NEW kind of nonlinear Kalman filter has been developed to solve discrete–time estimation problems approximately. It extends the principle of the iterated extended Kalman filter to include a direct treatment of the nonlinearities in the dynamics and measurement models for a number of recent sample intervals. The goal of the filter is to achieve better convergence reliability and accuracy than existing nonlinear filters, such as the traditional extended Kalman filter (EKF),[1] the iterated extended Kalman filter (IEKF),[1] and the unscented Kalman filter (UKF).[2,3]

The new backward-smoothing extended Kalman filter (BSEKF) treats more of the nonlinearities more directly than do other filters. The EKF uses one- or two-term Taylor series approximations of the nonlinear dynamics and measurement functions. Expansions are performed about the a posteriori or the a priori mean, whichever is available. It uses these expansions and the linear Kalman filter equations to propagate dynamically and update the state estimates and their estimation error covariances.[1] The IEKF attempts to improve its linear approximation of the measurement equation by relinearizing it about an a posteriori state estimate that is based on the preceding linearization. In one option of the IEKF, several relinearizations may be implemented.[1] The UKF propagates a distribution of points through the nonlinear dynamics and measurement functions.[2,3] These points are called sigma points, and they are used to estimate the means and covariances of Gaussian approximations of the nonlinear functions' output distributions. If one is careful about how one chooses the UKF's sigma points, then it retains second-order nonlinear effects that are missed by certain implementations of the EKF,[2,3] which is why it can have improved convergence and accuracy in comparison to many EKFs.

The BSEKF's improved performance comes from its method of dealing with the nonlinearities of the present and past dynamics and measurement functions. This method is an extension of the principle of the IEKF. The idea is to relinearize the current and past measurement functions, along with the current and past dynamics functions, about improved guesses of the current and past state and process noise vectors. Appropriate relinearization points are chosen via iterative nonlinear smoothing over an interval of time that ends at the current sample time. This process results in a state estimation algorithm that treats all of the nonlinearities over a number of stages without any approximation.

The BSEKF is similar in spirit to the filter described in Ref. 4. The principal difference of the current formulation is that it works for a general discrete-time problem, whereas the Ref. 4 filter works directly on a sampled-data continuous-time problem that has been derived for a specific application. The present discrete-time formulation simplifies the mathematics as follows: It eliminates both the need to designate deterministic and nondeterministic parts of the state vector and the need to develop joint probability distributions that do not include the process noise vector.

The BSEKF can be classified as a special kind of causal batch filter. A careful researcher could synthesize the BSEKF approach from various ideas that exist in the literature on batch filtering, but no reference combines all of the necessary components in one place. In Ref. 5, the use of a moving-window batch filter with nonlinear Gauss–Newton updates is mentioned as a means of performing causal nonlinear filtering, but the formulation does not include process noise, nor does it explain important details about how best to incorporate the effects of data from before the batch window. In Ref. 6, a moving-window batch filter is mentioned that has similar shortcomings, and even less detail is given. In Ref. 7, a method is defined for solving a discrete-time batch estimation problem that includes process noise, but it makes no mention of a moving-window implementation as a means of causal nonlinear filtering. In Ref. 1, a nonlinear maximum a posteriori (MAP) smoother and its relationship to dynamic programming are outlined, but no solution algorithm is presented, nor is a moving-window implementation for use in causal filtering discussed.

This paper does not compare the BSEKF to the method known as particle filtering (PF),[8] which uses Monte Carlo techniques. A particle filter is able, in theory, to compute a better estimate than the BSEKF or any competing nonlinear filter, but it can have a large computational cost for challenging problems because of the difficulty of selecting relevant points in the state space. The BSEKF is more similar in spirit to an EKF, an IEKF, and a UKF because each algorithm is an approximate method that uses a bounded number of computations. There seems little point in comparing such a method to a method that is exact in the limit of an infinite number of computations. On a different note, a useful approach for extremely nonlinear problems may be to combine the BSEKF and PF techniques. The former could be used to guide the selection of particles (points in state space) for efficient implementation of the latter.

The remainder of this paper is divided into three main sections, plus conclusions. In Sec. II, the nonlinear discrete-time filtering problem is defined as a MAP estimation problem and aspects of how the EKF, IEKF, and UKF use approximation to solve it are

Received 8 July 2004; revision received 13 December 2004; accepted for publication 26 January 2005.

*Associate Professor, Sibley School of Mechanical and Aerospace Engineering. Associate Fellow AIAA.

discussed. In Sec. III, how the BSEKF makes fewer approximations to the MAP problem than any of the other filters is described, and a detailed description of how to implement the algorithm is presented. In Sec. IV, BSEKF, EKF, and UKF filtering results are presented for an attitude estimation problem with simultaneous parameter estimation. The paper's conclusions are presented in Sec. V.

II. Nonlinear Filtering Problem and Discussion of Several Existing Filters

The goal of the BSEKF is to solve the following discrete-time nonlinear filtering problem: Find

$$\boldsymbol{x}_k \text{ along with } \boldsymbol{x}_i \text{ and } \boldsymbol{w}_i \text{ for } i = 0, 1, 2, \ldots, k-1 \tag{1a}$$

to minimize

$$J = \frac{1}{2}\sum_{i=0}^{k-1}\left\{\boldsymbol{w}_i^T Q_i^{-1}\boldsymbol{w}_i + [\boldsymbol{y}_{i+1} - \boldsymbol{h}_{i+1}(\boldsymbol{x}_{i+1})]^T R_{i+1}^{-1}[\boldsymbol{y}_{i+1} - \boldsymbol{h}_{i+1}(\boldsymbol{x}_{i+1})]\right\} + \frac{1}{2}(\boldsymbol{x}_0 - \hat{\boldsymbol{x}}_0)^T P_0^{-1}(\boldsymbol{x}_0 - \hat{\boldsymbol{x}}_0) \tag{1b}$$

subject to

$$\boldsymbol{x}_{i+1} = \boldsymbol{f}_i(\boldsymbol{x}_i, \boldsymbol{w}_i) \qquad \text{for} \quad i = 0, 1, 2, \ldots, k-1 \tag{1c}$$

where $\boldsymbol{x}_i$ is the system's state vector at time t_i, $\boldsymbol{w}_i$ is the discrete-time process noise vector that acts on the system from time t_i to time t_{i+1}, and $\boldsymbol{y}_{i+1}$ is the sensor measurement vector at time t_{i+1}. Equation (1c) and the vector function $\boldsymbol{f}_i(\boldsymbol{x}_i, \boldsymbol{w}_i)$ give the system's nonlinear dynamics model in the form of a state difference equation. The equation $\boldsymbol{y}_{i+1} = \boldsymbol{h}_{i+1}(\boldsymbol{x}_{i+1}) + \boldsymbol{\nu}_{i+1}$ is its nonlinear measurement model, where $\boldsymbol{h}_{i+1}(\boldsymbol{x}_{i+1})$ is a vector function and where $\boldsymbol{\nu}_{i+1}$ is an additive measurement noise vector. The process noise is modeled as a discrete-time Gaussian white noise with a mean of zero and a covariance of Q_i. The measurement noise is modeled as a zero-mean discrete-time Gaussian white noise with covariance R_{i+1}. The filter starts at time t_0 with the a posteriori state estimate $\hat{\boldsymbol{x}}_0$ and the a posteriori state estimation error covariance matrix P_0.

The filtering problem in Eqs. (1a–1c) is posed in the MAP form.[1] The joint a posteriori probability density of $\boldsymbol{x}_0$ and $\boldsymbol{w}_0$, $\ldots$, $\boldsymbol{w}_{k-1}$ conditioned on $\hat{\boldsymbol{x}}_0, \boldsymbol{y}_1, \boldsymbol{y}_2, \boldsymbol{y}_3, \ldots, \boldsymbol{y}_k$, is proportional to $\exp(-J)$ if $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k$ are generated by iterating Eq. (1c). Minimization of J is equivalent to maximization of this probability density. For linear/Gaussian estimation problems, MAP estimation is equivalent to minimum mean-square error estimation, in which $\hat{\boldsymbol{x}}_k = E\{\boldsymbol{x}_k | \hat{\boldsymbol{x}}_0, \boldsymbol{y}_1 \ldots, \boldsymbol{y}_k\}$, the conditional mean. For nonlinear problems, the MAP estimate of $\boldsymbol{x}_k$ will not necessarily equal the conditional mean, but it is usually a very useful estimate. The reason for choosing a MAP framework for the BSEKF is that the resulting problem in Eqs. (1a–1c) becomes a nonlinear least-squares problem. There exist powerful techniques for solving nonlinear least-squares problems,[9] and they can be exploited to implement the BSEKF.

One normally solves a sequence of filtering problems that is parameterized by an increasing sequence of terminal stage indices, $k = 1, 2, 3, \ldots$. The sequence of solutions to these problems is $\hat{\boldsymbol{x}}_1, \hat{\boldsymbol{x}}_2, \hat{\boldsymbol{x}}_3, \ldots$. The filtered estimate $\hat{\boldsymbol{x}}_k$ is the MAP estimate of $\boldsymbol{x}_k$ conditioned on the data $\boldsymbol{y}_1, \boldsymbol{y}_2, \boldsymbol{y}_3, \ldots, \boldsymbol{y}_k$.

Insight can be gained by rewriting the problem in Eqs. (1a–1c) in a manner that does not contain summations. Suppose that one solves the optimal estimation problem in Eqs. (1a–1c) subject to the additional constraint that $\boldsymbol{x}_k$ must take on a particular fixed value that is not necessarily the optimum. One optimizes $\boldsymbol{x}_i$ and $\boldsymbol{w}_i$ for $i = 0, 1, 2, \ldots, k-1$, and the resulting cost is a function of the chosen fixed value of $\boldsymbol{x}_k$. Call this optimal cost function $J_{\text{opt}[k]}(\boldsymbol{x}_k)$. This cost does not depend on any $\boldsymbol{x}_i$ or $\boldsymbol{w}_i$ for $i < k$ because each of these vectors is constrained to take on its optimal value, which is a function of the given $\boldsymbol{x}_k$ value. It is easy to see that $\hat{\boldsymbol{x}}_k$ is the value that minimizes $J_{\text{opt}[k]}(\boldsymbol{x}_k)$. The function $J_{\text{opt}[k]}(\boldsymbol{x}_k)$ is normally a complicated nonlinear function of $\boldsymbol{x}_k$ for which no analytic expression exists.

There is a $J_{\text{opt}[i]}(\boldsymbol{x}_i)$ function for each index $i = 0, 1, 2, \ldots, k-1, k, \ldots$, and the function for $i = k-1$ can be used to rewrite the MAP estimation problem in a one-step form: Find

$$\boldsymbol{x}_k \text{ along with } \boldsymbol{x}_{k-1} \text{ and } \boldsymbol{w}_{k-1} \tag{2a}$$

to minimize

$$J = \tfrac{1}{2}\boldsymbol{w}_{k-1}^T Q_{k-1}^{-1}\boldsymbol{w}_{k-1} + \tfrac{1}{2}[\boldsymbol{y}_k - \boldsymbol{h}_k(\boldsymbol{x}_k)]^T R_k^{-1}[\boldsymbol{y}_k - \boldsymbol{h}_k(\boldsymbol{x}_k)] + J_{\text{opt}[k-1]}(\boldsymbol{x}_{k-1}) \tag{2b}$$

subject to

$$\boldsymbol{x}_k = \boldsymbol{f}_{k-1}(\boldsymbol{x}_{k-1}, \boldsymbol{w}_{k-1}) \tag{2c}$$

The EKF and the IEKF implicitly use an approximate form of the cost function $J_{\text{opt}[k-1]}(\boldsymbol{x}_{k-1})$ to form their equivalents of problem (2a–2c). The approximation takes the form

$$J_{\text{opt}[k-1]}(\boldsymbol{x}_{k-1}) \cong \tfrac{1}{2}(\boldsymbol{x}_{k-1} - \hat{\boldsymbol{x}}_{k-1})^T P_{k-1}^{-1}(\boldsymbol{x}_{k-1} - \hat{\boldsymbol{x}}_{k-1}) + \text{const} \tag{3}$$

where $\hat{\boldsymbol{x}}_{k-1}$ is the filtered state estimate at time t_{k-1} and P_{k-1} is the corresponding state estimation error covariance matrix. This quadratic approximation of $J_{\text{opt}[k-1]}(\boldsymbol{x}_{k-1})$ is generated using Taylor series techniques that include correct linear terms and correct or approximate quadratic terms for the effects of the functions $\boldsymbol{f}_{k-2}(\boldsymbol{x}_{k-2}, \boldsymbol{w}_{k-2})$ and $\boldsymbol{h}_{k-1}(\boldsymbol{x}_{k-1})$. The approximations of the effects of any nonlinearities in the functions $\boldsymbol{f}_i(\boldsymbol{x}_i, \boldsymbol{w}_i)$ and $\boldsymbol{h}_{i+1}(\boldsymbol{x}_{i+1})$ for $i < k-2$ are even less accurate.

Although the UKF is not based on MAP techniques, one can reinterpret it as a MAP filter that also uses an approximation of $J_{\text{opt}[k-1]}(\boldsymbol{x}_{k-1})$. A strength of the UKF is that its approximation always includes second-order effects for the functions $\boldsymbol{f}_{k-2}(\boldsymbol{x}_{k-2}, \boldsymbol{w}_{k-2})$ and $\boldsymbol{h}_{k-1}(\boldsymbol{x}_{k-1})$.

The EKF, the IEKF, and the UKF make additional approximations. The EKF and IEKF state propagations use linear or quadratic Taylor series approximations of $\boldsymbol{f}_{k-1}(\boldsymbol{x}_{k-1}, \boldsymbol{w}_{k-1})$ expanded about $\boldsymbol{x}_{k-1} = \hat{\boldsymbol{x}}_{k-1}$ and $\boldsymbol{w}_{k-1} = 0$. The IEKF measurement update treats the full nonlinearity of the function $\boldsymbol{h}_k(\boldsymbol{x}_k)$ in a Gauss–Newton minimization, but an EKF uses a linear or quadratic Taylor series expansion of $\boldsymbol{h}_k(\boldsymbol{x}_k)$ about the a priori estimate $\bar{\boldsymbol{x}}_k$. The EKF update is, in effect, a single iteration of a Gauss–Newton or Newton nonlinear least-squares procedure. The UKF evaluates $\boldsymbol{f}_{k-1}(\boldsymbol{x}_{k-1}, \boldsymbol{w}_{k-1})$ and $\boldsymbol{h}_k(\boldsymbol{x}_k)$ at a distribution of sigma points to form its estimate of $\boldsymbol{x}_k$. Its procedures can be interpreted as being describing-function-type statistical linearizations.

Each of these approximations allows its filter to avoid direct solution of the nonlinear least-squares problem in Eqs. (1a–1c). The original problem takes the form of a nonlinear trajectory optimization problem. The problem must be re-solved for each successive value of k if the application calls for real-time filtering to produce the sequence of estimates $\hat{\boldsymbol{x}}_1, \hat{\boldsymbol{x}}_2, \hat{\boldsymbol{x}}_3, \ldots$, and the size of the problem grows with k. The approximate methods work with the problem in Eqs. (2a–2c). Its size does not grow with k.

The weakness of any technique that approximates $J_{\text{opt}[k-1]}(\boldsymbol{x}_{k-1})$ is that its estimate of $\boldsymbol{x}_k$ is implicitly based on suboptimal estimates of $\boldsymbol{x}_i$ and $\boldsymbol{w}_i$ for $i = 0, 1, 2, \ldots, k-2$. To understand this, consider Fig. 1, in which the first six samples of a Kalman filter output time history are plotted for a two-state linear spring–mass–damper oscillator (see upper solid black curve with asterisks). The truth trajectory is shown in Fig. 1 as the lower gray dashed line with the square symbols. Five smoothed trajectories are also plotted for the five fixed smoothing intervals 0–1, 0–2, $\ldots$, 0–5. The Kalman filter at no time supposes that the system state has evolved along the solid black Kalman filter curve. Rather, the Kalman filter estimate at each sample point is based on its implicit supposition that the system has arrived at the point in question by traveling along the smoothed time history that ends at that point. The smoothed curves do not lie exactly on top of each other. This is because the Kalman filter implicitly "changes its mind" about past states whenever a new piece of data becomes available at a new sample time. This property is best illustrated by the progression of smoothed state estimates for

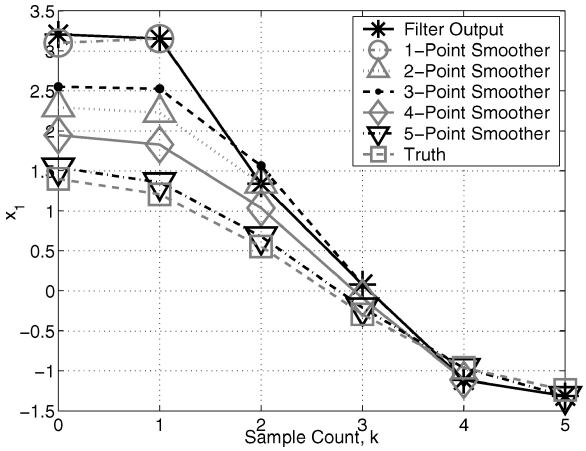

Fig. 1 Relationship between Kalman filter estimates and corresponding implicit smoothed estimates at previous sample times.

sample $k=0$ as the filter progresses forward in time. By the time the filter reaches sample $k=5$, the corresponding smoothed estimate at sample $k=0$ is close to the truth state. (Refer to curve with the ∇ symbols.) An approximate nonlinear filter's implicit smoothed values may not achieve comparable accuracy, and inaccurate assessment of the past can decrease the accuracy of the filter's current state estimate.

III. BSEKF Algorithm

A. Filter That Solves a Sequence of Nonlinear Smoothing Problems

The BSEKF improves its convergence reliability and accuracy by reducing the amount of approximation in the problem that it solves. It computes $\hat{\boldsymbol{x}}_k$ for each k by solving the following problem: Find

$$\boldsymbol{x}_k \text{ along with } \boldsymbol{x}_i \text{ and } \boldsymbol{w}_i \text{ for } i=k-m,\ldots,k-1 \tag{4a}$$

to minimize

$$J=\frac{1}{2}\sum_{i=k-m}^{k-1}\left\{\boldsymbol{w}_i^T Q_i^{-1}\boldsymbol{w}_i+[\boldsymbol{y}_{i+1}-\boldsymbol{h}_{i+1}(\boldsymbol{x}_{i+1})]^T R_{i+1}^{-1}[\boldsymbol{y}_{i+1}-\boldsymbol{h}_{i+1}(\boldsymbol{x}_{i+1})]\right\}+\frac{1}{2}\left(\boldsymbol{x}_{k-m}-\hat{\boldsymbol{x}}_{k-m}^*\right)^T\left(P_{k-m}^*\right)^{-1}\times\left(\boldsymbol{x}_{k-m}-\hat{\boldsymbol{x}}_{k-m}^*\right) \tag{4b}$$

subject to

$$\boldsymbol{x}_{i+1}=\boldsymbol{f}_i(\boldsymbol{x}_i,\boldsymbol{w}_i)\qquad\text{for}\quad i=k-m,\ldots,k-1 \tag{4c}$$

This problem represents an approximation of the problem in Eqs. (1a–1c) to the extent that the cost term $0.5(\boldsymbol{x}_{k-m}-\hat{\boldsymbol{x}}_{k-m}^*)^T(P_{k-m}^*)^{-1}(\boldsymbol{x}_{k-m}-\hat{\boldsymbol{x}}_{k-m}^*)$ is an approximation of the cost function $J_{\text{opt}[k-m]}(\boldsymbol{x}_{k-m})$. It retains all of the nonlinearities of the most recent m stages, but it approximates the nonlinear effects in the functions $\boldsymbol{f}_0(\boldsymbol{x}_0,\boldsymbol{w}_0)$ through $\boldsymbol{f}_{k-m-1}(\boldsymbol{x}_{k-m-1},\boldsymbol{w}_{k-m-1})$ and $\boldsymbol{h}_1(\boldsymbol{x}_1)$ through $\boldsymbol{h}_{k-m}(\boldsymbol{x}_{k-m})$ by its approximation of $J_{\text{opt}[k-m]}(\boldsymbol{x}_{k-m})$. The problem in Eqs. (4a–4c) is an m-stage nonlinear smoothing problem.

Contrary to appearances, the vector $\hat{\boldsymbol{x}}_{k-m}^*$ and the matrix P_{k-m}^* in Eq. (4b) do not constitute the a posteriori state estimate and its estimation error covariance matrix at sample time t_{k-m}. Although these quantities look and act like $\hat{\boldsymbol{x}}_{k-m}$ and P_{k-m} in this smoothing problem, their principal purpose is to cause the last cost term in Eq. (4b) to be a reasonable approximation of $J_{\text{opt}[k-m]}(\boldsymbol{x}_{k-m})$. They are computed by a filtering technique, but they are not the true filtered values because they are based on linearizations of preceding dynamics and measurement functions about smoothed values from smoothing problems that extend beyond time t_{k-m}. The actual computation of $\hat{\boldsymbol{x}}_{k-m}^*$ and P_{k-m}^* will be discussed in Sec. III.D.

Equations (4a–4c) can constitute a very good approximation of the problem in Eqs. (1a–1c) if the time lag m is large enough, that is, if enough of the past stages are treated explicitly by the nonlinear smoother. A large m makes the optimal $\hat{\boldsymbol{x}}_k$ insensitive to the accuracy of the approximation of $J_{\text{opt}[k-m]}(\boldsymbol{x}_{k-m})$. A large m also makes the problem (1a–1c) smoothed estimate of $\boldsymbol{x}_i$ for $i<k-m$ insensitive to the terminal time of the smoothing problem. This implies something important about the linearizations of the functions $\boldsymbol{f}_i(\boldsymbol{x}_i,\boldsymbol{w}_i)$ and $\boldsymbol{h}_{i+1}(\boldsymbol{x}_{i+1})$ for $i<k-m$ that are used to compute $\hat{\boldsymbol{x}}_{k-m}^*$ and P_{k-m}^*: They are close to the linearizations that would have been used if those stages had been explicitly included in the smoothing calculation of Eqs. (4a–4c). The approximate $J_{\text{opt}[k-m]}(\boldsymbol{x}_{k-m})$ is more accurate as a result.

The BSEKF limits the number of stages that it uses to set up the sequence of nonlinear smoothing problems that end at $k=1,2,3,\ldots$. It works with a target number of stages m_{target}. When $k\le m_{\text{target}}$, it uses $m=k$ stages, and it makes no approximations. When $k>m_{\text{target}}$, it uses $m=m_{\text{target}}$ stages, and it uses $\hat{\boldsymbol{x}}_{k-m}^*$ and P_{k-m}^* to approximate $J_{\text{opt}[k-m]}(\boldsymbol{x}_{k-m})$. The number of stages in any given smoother problem can be written compactly as $m(k)=\min\{k,m_{\text{target}}\}$, where the min{, } function returns the minimum of its two input arguments.

This target number of explicitly optimized stages is a design choice. It is wise to choose m_{target} to be as small as possible to minimize the computational burden of the algorithm, but the convergence reliability and accuracy of the BSEKF will degrade if m_{target} is too small because it approaches the IEKF in this case. A designer must perform a tradeoff between computational burden on the one hand and convergence reliability and estimation accuracy on the other hand. It may be practical to modify m_{target} dynamically. Note that m_{target} could be set to a larger value during initialization to ensure convergence. After convergence had been achieved, m_{target} could be lowered to conserve computational resources. In the present study, m_{target} is held fixed in the interest of simplicity.

B. Solution Method for BSEKF Nonlinear Smoothing Problem

The nonlinear least-squares smoothing problem in Eqs. (4a–4c) can be solved using a guarded Gauss–Newton method.[9] The Gauss–Newton method is iterative. It starts with a first guess and computes increments that should produce a converging sequence of guesses. Each guess has a lower least-squares cost than the previous one. The increment to a given guess is computed by solving a linearized least-squares problem. Convergence is guaranteed by a guarding procedure that scales down the computed increment to the guess, if necessary, to force a decrease in the least-squares cost from one guess to the next.

The application of the Gauss–Newton method to the problem in Eqs. (4a–4c) works with guesses and increments of $\boldsymbol{x}_{k-m}$ and of $\boldsymbol{w}_{k-m}$, $\boldsymbol{w}_{k-m+1}$, $\boldsymbol{w}_{k-m+2},\ldots,\boldsymbol{w}_{k-1}$. Suppose that the initial guesses are $\boldsymbol{x}_{k-m}^0$ and $\boldsymbol{w}_{k-m}^0$, $\boldsymbol{w}_{k-m+1}^0$, $\boldsymbol{w}_{k-m+2}^0,\ldots,\boldsymbol{w}_{k-1}^0$. The nonlinear smoothing algorithm starts from these values and iterates as follows:

1) Set the Gauss–Newton iteration counter to $j=0$.

2) Start from $\boldsymbol{x}_{k-m}=\boldsymbol{x}_{k-m}^j$ and iterate Eq. (4c) for $i=k-m,\ldots,k-1$, using the process noise guesses $\boldsymbol{w}_{k-m}^j,\ldots,\boldsymbol{w}_{k-1}^j$, to compute the corresponding state guesses $\boldsymbol{x}_{k-m+1}^j,\ldots,\boldsymbol{x}_k^j$. Compute J^j, the Eq. (4b) cost evaluated at this solution guess.

3) Evaluate the linearized dynamics matrices $\Phi_i=\partial\boldsymbol{f}_i/\partial\boldsymbol{x}_i$ and $\Gamma_i=\partial\boldsymbol{f}_i/\partial\boldsymbol{w}_i$ at the points $(\boldsymbol{x}_i^j,\boldsymbol{w}_i^j)$ for $i=k-m,\ldots,k-1$, and evaluate the measurement error vector $\Delta\boldsymbol{y}_i=\boldsymbol{y}_i-\boldsymbol{h}_i(\boldsymbol{x}_i^j)$ and the linearized measurement matrix $H_i=\partial\boldsymbol{h}_i/\partial\boldsymbol{x}_i$ at the points $\boldsymbol{x}_i^j$ for $i=k-m+1,\ldots,k$.

4) Use the vectors and matrices from step 3 to define the following linearized smoothing problem: Find

$$\Delta\boldsymbol{x}_k \text{ along with } \Delta\boldsymbol{x}_i \text{ and } \Delta\boldsymbol{w}_i \text{ for } i=k-m,\ldots,k-1 \tag{5a}$$

to minimize

$$J=\frac{1}{2}\sum_{i=k-m}^{k-1}\left\{\left[\boldsymbol{w}_i^j+\Delta\boldsymbol{w}_i\right]^T Q_i^{-1}\left[\boldsymbol{w}_i^j+\Delta\boldsymbol{w}_i\right]+[\Delta\boldsymbol{y}_{i+1}-\boldsymbol{H}_{i+1}\Delta\boldsymbol{x}_{i+1}]^T R_{i+1}^{-1}[\Delta\boldsymbol{y}_{i+1}-\boldsymbol{H}_{i+1}\Delta\boldsymbol{x}_{i+1}]\right\}+\frac{1}{2}\left(\boldsymbol{x}_{k-m}^j+\Delta\boldsymbol{x}_{k-m}-\hat{\boldsymbol{x}}_{k-m}^*\right)^T\left(P_{k-m}^*\right)^{-1}\times\left(\boldsymbol{x}_{k-m}^j+\Delta\boldsymbol{x}_{k-m}-\hat{\boldsymbol{x}}_{k-m}^*\right) \tag{5b}$$

subject to

$$\Delta \boldsymbol{x}_{i+1} = \Phi_i \Delta \boldsymbol{x}_i + \Gamma_i \Delta \boldsymbol{w}_i, \qquad i = k-m, \ldots, k-1 \tag{5c}$$

and solve it by using linear smoothing techniques.

5) Set the initial trial search step size to $\gamma = 1$.

6) Compute the candidate next guess of the smoothed solution as follows:

$$\boldsymbol{x}_{k-m}^{j+1} = \boldsymbol{x}_{k-m}^{j} + \gamma \Delta \boldsymbol{x}_{k-m}$$

$$\boldsymbol{w}_i^{j+1} = \boldsymbol{w}_i^{j} + \gamma \Delta \boldsymbol{w}_i \qquad \text{for} \quad i = k-m, \ldots, k-1$$

$$\boldsymbol{x}_{i+1}^{j+1} = \boldsymbol{f}_i\left(\boldsymbol{x}_i^{j+1}, \boldsymbol{w}_i^{j+1}\right) \qquad \text{for} \quad i = k-m, \ldots, k-1$$

and compute the corresponding Eq. (4b) cost, J^{j+1}.

7) If $J^{j+1} \geq J^j$, then set $\gamma = 0.5\gamma$ and return to step 6; otherwise, continue to step 8.

8) Stop if termination criteria are satisfied or if j is too large, and set $\hat{\boldsymbol{x}}_k = \boldsymbol{x}_k^{j+1}$; otherwise, set $j = j+1$ and go to step 3.

This Gauss–Newton implementation is guaranteed to be globally convergent to a local minimum of the nonlinear least-squares cost function unless it inadvertently lands exactly on a saddle point or a local maximum during its search process. This one failure possibility is improbable. The global convergence guarantee derives from the step length adjustment part of the algorithm in steps 6 and 7. There is guaranteed to exist a certain value $\gamma_0 > 0$ such that $J^{j+1} < J^j$ for all step lengths γ in the range $0 < \gamma < \gamma_0$, provided that the current solution guess is not a local minimum, a saddle point, or a local maximum of J. This guarantees that steps 6 and 7 terminate at a finite γ with $J^{j+1} < J^j$. The inequalities $J \geq 0$ and $J^{j+1} < J^j$ guarantee global convergence to at least a local minimum. (Strictly speaking, a more sophisticated line search than step size halving is required to guarantee global convergence.[9] It is the author's experience that step size halving induces global convergence in most situations.) If the first guess of the solution is not too wild or if the cost function only has one local minimum, that is, the global minimum, then the solution will converge to the global minimum.

The robust convergence properties of the guarded Gauss–Newton method ensure that the BSEKF will be at least as likely to converge as almost any other nonlinear filter. The only way to improve on it is to solve its smoothing problem using a nonlinear numerical optimization algorithm that is guaranteed to reach the global minimum, even when there are multiple local minima. Such algorithms are computationally expensive.

It may be acceptable to terminate the Gauss–Newton algorithm early, before it has reached the exact minimum of the nonlinear smoothing problem. The possibility of early termination is provided in step 8 by setting an upper bound on j. This expedient seeks a very good estimate that yields a low Eq. (4b) cost that may not be the exact minimum. It can save a great deal of computation time in some situations while sacrificing little in terms of filter accuracy.

The local rate of convergence of the Gauss–Newton method is dependent on the magnitudes of the errors in the least-squares cost in Eq. (4b) and on the curvature of the nonlinear dynamics and measurement functions. If these quantities are small in some appropriate sense, then the Gauss–Newton method's convergence rate will be quadratic, like that of Newton's method (see Ref. 9). If the quantities are large, then the rate of convergence will be slower.

C. Solution Procedure for Linearized Smoothing Problem and Computation of Filter's Estimation Error Covariance Matrix

Various linear fixed-interval smoothing solution procedures can be used to solve the problem in Eqs. (5a–5c). A suitable algorithm must compute the smoothed process noise estimates in addition to the smoothed state estimates. The method used here is a square-root information matrix method adapted from Ref. 10.

A square-root information matrix is the square root of the inverse of the corresponding covariance matrix. Suppose that one defines the following square-root information matrices: $R_{ww(i)}$ is the square-root information matrix associated with $\boldsymbol{w}_i$, which implies that $Q_i = [R_{ww(i)}]^{-1}[R_{ww(i)}]^{-T}$, where the notation $[\,]^{-T}$ indicates the transpose of the inverse of the matrix in the brackets. Similarly, $R_{vv(i)}$ is the square-root information matrix of the measurement noise in $\boldsymbol{y}_i$, so that $R_i = [R_{vv(i)}]^{-1}[R_{vv(i)}]^{-T}$, and $R_{xx(k-m)}$ is the square-root information matrix associated with the initial state cost function so that $P_{k-m}^{*} = [R_{xx(k-m)}]^{-1}[R_{xx(k-m)}]^{-T}$.

The following algorithm solves the linear smoothing problem in Eqs. (5a–5c) by performing a forward-filtering pass following by a backward-smoothing pass:

1) Start at the sample index $i = k-m$ and set $\Delta \boldsymbol{z}_{x(i)} = R_{xx(k-m)} \times [\hat{\boldsymbol{x}}_{k-m}^{*} - \boldsymbol{x}_{k-m}^{j}]$.

2) Form the block matrix on the right-hand side of the following equation and orthonormal/upper-triangular (QR) factorize that matrix to determine the matrices on the left-hand side:

$$T_i \begin{bmatrix} \bar{R}_{ww(i)} & \bar{R}_{wx(i)} \\ 0 & R_{xx(i+1)} \end{bmatrix} = \begin{bmatrix} R_{ww(i)} & 0 \\ -R_{xx(i)}\Phi_i^{-1}\Gamma_i & R_{xx(i)}\Phi_i^{-1} \\ 0 & R_{vv(i+1)}H_{i+1} \end{bmatrix} \tag{6}$$

where T_i is an orthonormal matrix and where $\bar{R}_{ww(i)}$ and $R_{xx(i+1)}$ are square, nonsingular, upper-triangular matrices.

3) Compute the vectors $\Delta \bar{\boldsymbol{z}}_{w(i)}$, $\Delta \boldsymbol{z}_{x(i+1)}$, and $\Delta \boldsymbol{z}_{r(i)}$, by using the T_i matrix in the following formula:

$$\begin{bmatrix} \Delta \bar{\boldsymbol{z}}_{w(i)} \\ \Delta \boldsymbol{z}_{x(i+1)} \\ \Delta \boldsymbol{z}_{r(i)} \end{bmatrix} = T_i^{T} \begin{bmatrix} -R_{ww(i)}\boldsymbol{w}_i^{j} \\ \Delta \boldsymbol{z}_{x(i)} \\ R_{vv(i+1)}\Delta \boldsymbol{y}_{i+1} \end{bmatrix} \tag{7}$$

4) If $i = k-1$, go to step 5; otherwise, set $i = i+1$ and go to step 2.

5) Compute $\Delta \boldsymbol{x}_k = R_{xx(k)}^{-1}\Delta \boldsymbol{z}_{x(k)}$ and set $i = k-1$.

6) Compute $\Delta \boldsymbol{w}_i = \bar{R}_{ww(i)}^{-1}[\Delta \bar{\boldsymbol{z}}_{w(i)} - \bar{R}_{wx(i)}\Delta \boldsymbol{x}_{i+1}]$.

7) Compute $\Delta \boldsymbol{x}_i = \Phi_i^{-1}[\Delta \boldsymbol{x}_{i+1} - \Gamma_i \Delta \boldsymbol{w}_i]$.

8) If $i = k-m$ stop; otherwise, set $i = i-1$ and go to step 6.

Steps 1–4 of this algorithm constitute the forward square-root information filtering pass, and steps 5–8 constitute the backward-smoothing pass. Steps 2 and 3 combine the state propagation and measurement update of Ref. 10 into a single QR factorization step.

The linearized smoother solution can be used to compute the filter's estimation error covariance matrix. Suppose that the nonlinear smoother iterations of the preceding subsection have terminated. In that case $\hat{\boldsymbol{x}}_k = \boldsymbol{x}_k^{j+1}$, with j being the highest iteration count at which the Gauss–Newton iterations terminated. If the linearized smoothing problem is re-solved to produce an additional increment, then the solution to problem (5a–5c) will be all zeros for the state and process noise vector time-history increments. The corresponding value of $R_{xx(k)}$ from step 5 of the linear solution procedure will be the square-root information matrix for the estimation error in $\hat{\boldsymbol{x}}_k$, and the covariance matrix for this error will be $P_k = [R_{xx(k)}]^{-1}[R_{xx(k)}]^{-T}$.

This formulation of the linear smoother uses some vector and matrix notation without distinguishing the iteration of the Gauss–Newton method to which it applies and without distinguishing the original smoothing problem to which it applies. The quantities the values of which may change, but the notations of which do not change, are m, $R_{xx(i)}$, $\Delta \boldsymbol{z}_{x(i)}$, $\bar{R}_{ww(i)}$, $\bar{R}_{wx(i)}$, $\Delta \bar{\boldsymbol{z}}_{w(i)}$, $\Delta \boldsymbol{z}_{r(i)}$, T_i, Φ_i, Γ_i, H_i, and $\Delta \boldsymbol{y}_i$. For example, $R_{xx(i)}$ is the forward-pass filter's state estimation error square root information matrix at sample i, but its value can vary from iteration to iteration of the Gauss–Newton method due to changes in the linearizations of the nonlinear dynamics and measurement functions. Its value also can change as the smoothing problem under consideration changes due to a change in the terminal sample number k. In other words, for a fixed i, $R_{xx(i)}$ can change as $k-i$ increases during the normal progression of the BSEKF to new smoother end times. The particular Gauss–Newton iteration and smoothing problem to which a given matrix or vector applies will normally be obvious from context.

The vectors $\Delta \boldsymbol{z}_{r(k-m)}, \ldots, \Delta \boldsymbol{z}_{r(k-1)}$, which are computed as part of the linear smoother solution, can be used to predict the new

optimal cost of the nonlinear least-squares problem:

$$J_{\text{newapprox}} = \frac{1}{2}\sum_{i=k-m}^{k-1} \Delta z_{r(i)}^T \Delta z_{r(i)} \tag{8}$$

This linearized cost prediction can be useful for developing termination criteria for the Gauss–Newton iterations. If the new approximate cost is almost as large as the old cost, then the iterations may be near termination. Alternatively, if the actual new cost based on nonlinear dynamics and measurement functions is nearly the same as this linearized prediction, then the algorithm may terminate in very few steps because the linearized model is very accurate.

D. Recursive BSEKF Operation

An important feature of the BSEKF is its ability to operate recursively. Care must be taken to define a sensible recursion that sets up the new smoothing problem with terminal sample $k+1$ based on the solution of the smoothing problem with terminal sample k. The most important points concern how to determine the $\hat{x}^*_{k-m+1}$ and P^*_{k-m+1} values that are used to approximate $J_{\text{opt}[k-m+1]}(x_{k-m+1})$ in the next smoothing problem and how to determine a good first guess of that problem's solution so that the number of Gauss–Newton iterations can be kept small.

If $k \geq m_{\text{target}}$ then the quantities $\hat{x}^*_{k-m+1}$ and P^*_{k-m+1} are computed from the linearized forward filtering pass of the preceding smoother problem. Suppose that $R_{xx(k-m+1)}$ and $\Delta z_{x(k-m+1)}$ are, respectively, the state estimation error square-root information matrix and the corresponding information vector at sample $i = k-m+1$ during the last Gauss–Newton iteration of the smoothing problem that ends at sample k. Suppose, also, that x_{k-m+1} is the final smoothed state estimate at sample $i = k-m+1$ for this same smoothing problem. Then the following formulas set up the needed quantities for the next smoothing problem:

$$\hat{x}^*_{k-m+1} = x_{k-m+1} + R^{-1}_{xx(k-m+1)} \Delta z_{x(k-m+1)} \tag{9a}$$

$$P^*_{k-m+1} = R^{-1}_{xx(k-m+1)} R^{-T}_{xx(k-m+1)} \tag{9b}$$

Equation (9b) and the definition of $R_{xx(i)}$ in the linearized smoothing problem can be used to show that $R_{xx(k-m+1)}$ from the linearized smoothing problem that ends at sample k becomes the initial square-root information matrix in the BSEKF's next linearized smoothing problem, the one that ends at sample $k+1$.

If $k < m_{\text{target}}$, then there is no need to compute the quantities in Eqs. (9a) and (9b). The new smoothing problem that ends at sample $k+1$ still begins at sample 0 in this case. The known quantities $\hat{x}^*_0$ and P^*_0 are used to define the last term in the new problem's Eq. (4b) cost function.

The following procedure explains how to make a good first guess of the solution to the problem in Eqs. (4a–4c). This guess tends to minimize the number of BSEKF Gauss–Newton solution iterations. The approach is to use the solution from the preceding BSEKF smoother as much as possible. This is likely to be a good first guess because measurement data from one additional sample should not make a large change in the smoothed state at any given time. Suppose that the smoother solution with terminal sample k is x^k_{k-m} and w^k_i for $i = k-m, \ldots, k-1$. The smoothed state vector estimate at sample $k-m+1$ associated with this solution is $x^k_{k-m+1} = f_{k-m}(x^k_{k-m}, w^k_{k-m})$. The first guess for the problem with terminal sample $k+1$ is set to $x_{k-m+1} = x^k_{k-m+1}$ and $w_i = w^k_i$ for $i = k-m+1, \ldots, k-1$, and $w_k = 0$. This w_k guess is the only sensible choice because the BSEKF will not yet have processed measurement data that have been affected by w_k.

Slight modifications to this procedure are needed during the startup transient when $k < m_{\text{target}}$. The initial sample of each smoothing problem remains fixed at zero during this phase of operation. The smoothing problem that ends at sample $k+1$ uses the first guesses x^k_0 and $w_i = w^k_i$ for $i = 0, \ldots, k-1$, and $w_k = 0$. The initial BSEKF smoothing problem, the one with terminal sample $k = 1$, uses the first guess $x_0 = \hat{x}_0$ and $w_0 = 0$.

One might be tempted to reach the false conclusion that the BSEKF is noncausal because it uses the solution to a smoothing problem to compute its $\hat{x}_k$ estimate. Causality is guaranteed because the BSEKF computes $\hat{x}_k$ based on the measurement data $y_0, y_1, y_2, \ldots, y_k$, and it computes $\hat{x}_{k+1}$ based on the measurement data $y_0, y_1, y_2, \ldots, y_{k+1}$, and so forth for $\hat{x}_{k+2}, \hat{x}_{k+3}, \hat{x}_{k+4}, \ldots$. It is true that the BSEKF recomputes a new, smoothed estimate of x_k in the process of computing $\hat{x}_{k+1}$, but this new estimate is not used as the filtered, that is, causal, estimate of x_k. Rather, this smoothed x_k estimate and the corresponding smoothed estimates at all preceding sample times are used merely as auxiliary variables that are needed to account properly for past nonlinearities during the computation of $\hat{x}_{k+1}$.

When the terminal sample increments from k to $k+1$, the only new data required by the BSEKF are the quantities Q_k, y_{k+1}, and R_{k+1}, along with the functions $f_k(x_k, w_k)$ and $h_{k+1}(x_{k+1})$. All of the other quantities and functions that are required to set up the smoothing problem in Eqs. (4a–4c) and to initialize the guess of its solution are available from the smoothing problem and solution associated with terminal sample k. This information includes the quantities $Q_{k-m+1}, \ldots, Q_{k-1}$, $y_{k-m+2}, \ldots, y_k$, $R_{k-m+2}, \ldots, R_k$, $\hat{x}^*_{k-m+1}$, P^*_{k-m+1}, x^k_{k-m+1}, and $w^k_{k-m+1}, \ldots, w^k_{k-1}$, along with the functions $f_{k-m+1}[x_{k-m+1}, w_{k-m+1}], \ldots, f_{k-1}(x_{k-1}, w_{k-1})$ and $h_{k-m+2}[x_{k-m+2}], \ldots, h_k(x_k)$. These quantities and functions define a metastate that the BSEKF passes from one sample time to the next.

E. BSEKF Computational Burden and Methods for Speeding Execution

The computational burden of the new algorithm is significantly larger than that of an EKF or a UKF. In comparison to an EKF, the setup and solution of the linearized smoothing problem for one BSEKF Gauss–Newton update requires m times as many $f_i(x_i, w_i)$ and $h_{i+1}(x_{i+1})$ function and derivative evaluations and approximately $2m$ times as much linear algebra. Several Gauss–Newton iterations may be required to reach a reasonable solution of each nonlinear smoothing problem, perhaps $j_{\text{end}} = 4$ or 5 in typical cases, but more for problems with very strong nonlinearities. Thus, the computational expense of this algorithm will be greater than that of a conventional EKF by a factor of $m \times j_{\text{end}}$ to $2m \times j_{\text{end}}$. This factor could be as large as 100–300 for typical problems. The UKF typically requires about twice as much computation as an EKF. Thus, the BSEKF is much more expensive computationally than an EKF or a UKF in many cases. It should be used only when its convergence robustness or accuracy demonstrate enough superiority to justify its greatly increased computational expense.

IV. BSEKF, EKF, and UKF Performance on Truth-Model Simulation Example

A. Spacecraft Attitude Estimation Problem

The BSEKF has been applied to a difficult problem from the field of spacecraft attitude estimation that has been taken from Ref. 11. The problem concerns a tumbling microsatellite in a circular orbit the altitude of which is 823 km and the inclination of which is 82 deg. Its rotation rate starts at 2 deg/s. Its inertia matrix is $I_m = [1.6135, -0.0151, -0.0703; -0.0151, 1.5497, -0.0435; -0.0703, -0.0435, 1.5571]\ \text{kg}\cdot\text{m}^2$. The filter uses only magnetometer measurements sampled once every 10 s to estimate attitude and rate. It must rely on Euler's equation for dynamic propagation of the attitude rates. It must exploit system observability and accurate dynamic propagation to estimate the attitude and rate about the instantaneous magnetic field vector, because measurements of that vector are only sensitive to rotations about the two axes that are orthogonal to it.

The filter starts out with a moment of inertia matrix estimate the principal inertia ratios of which are wrong by as much as 11% and the principal axes directions of which are wrong by between 34 and 44 deg: $I_m = \text{diag}(1.5715, 1.5715, 1.5715)\ \text{kg}\cdot\text{m}^2$. The error in any given element is no more than 4.4% of the maximum principal inertia. This level of inaccuracy is typical in cases in which

flight moments and products of inertia are predicted based on preflight mass properties calculations that rely on design dimensions and component mass measurements. Large principal axis direction uncertainties result from small errors in the inertia matrix, because the design has nearly equal principal inertias. These uncertainties make attitude estimation difficult when the spacecraft is tumbling, because it is hard to predict the axis of nutation.

The filter also estimates moment of inertia matrix parameters with the aim of improving its Euler dynamics model and thereby improving its attitude and rate estimates. This is done by augmenting the state to include six parameters that determine the moment of inertia matrix.[11] The overall scaling of the inertia matrix is unobservable, but it does not need to be estimated because the scaling does not affect the system's attitude motions. The unobservability of the scaling is addressed by assigning a very small a priori variance to one of the inertia matrix magnitude parameters. This acts as a soft constraint on the scaling.

B. Problem State Vector and Dynamics and Measurement Model Functions

The 12-element state vector of the estimator is made up of the spacecraft angular rate vector, a 3-parameter attitude representation, and 6 moment of inertia matrix parameters

$$\boldsymbol{x} = [\boldsymbol{\omega}^T, \alpha_1, \alpha_2, \theta, p_{I1}, \ldots, p_{I6}]^T \tag{10}$$

where $\boldsymbol{\omega}$ is the attitude rate vector of the spacecraft body with respect to inertial coordinates expressed in spacecraft coordinates, α_1 and α_2 are the nonzero elements of the Gibbs vector for an attitude rotation that is perpendicular to the measured magnetic field, θ is an attitude rotation about the measured magnetic field, and p_{I1} through p_{I6} are the six inertia matrix parameters.[11]

The attitude quaternion that parameterizes the transformation from inertial reference coordinates to spacecraft body-fixed coordinates can be constructed from α_1, α_2, θ, $\hat{\boldsymbol{b}}_{\rm sc}$, and $\hat{\boldsymbol{b}}_{\rm in}$. The latter two quantities are magnetic field unit direction vectors, with the first being the measured field in spacecraft coordinates and the second being the modeled field in inertial reference coordinates. The attitude quaternion is

$$\boldsymbol{q}(\alpha_1, \alpha_2, \theta; \hat{\boldsymbol{b}}_{\rm sc}, \hat{\boldsymbol{b}}_{\rm in}, \hat{\boldsymbol{v}}_1, \hat{\boldsymbol{v}}_2) = \frac{1}{\sqrt{1+\alpha_1^2+\alpha_2^2}} \begin{bmatrix} (\alpha_1 \hat{\boldsymbol{v}}_1 + \alpha_2 \hat{\boldsymbol{v}}_2) \\ 1 \end{bmatrix} \otimes \begin{bmatrix} \hat{\boldsymbol{b}}_{\rm sc} \sin(\theta/2) \\ \cos(\theta/2) \end{bmatrix} \otimes \boldsymbol{q}_{\rm min}(\hat{\boldsymbol{b}}_{\rm sc}, \hat{\boldsymbol{b}}_{\rm in}) \tag{11}$$

The symbol $\otimes$ in Eq. (11) stands for quaternion multiplication, and the quantities $\hat{\boldsymbol{v}}_1$ and $\hat{\boldsymbol{v}}_2$ are any two unit direction vectors chosen so that $[\hat{\boldsymbol{v}}_1, \hat{\boldsymbol{v}}_2, \hat{\boldsymbol{b}}_{\rm sc}]$ is a right-handed orthonormal triad. Note that the carat overstrike denotes a unit vector rather than an estimate in the current subsection.The final quaternion on the right-hand side of Eq. (11) is the minimum rotation that satisfies the measurement constraint $\hat{\boldsymbol{b}}_{\rm sc} = A(\boldsymbol{q})\hat{\boldsymbol{b}}_{\rm in}$ (Ref. 12), where $A(\boldsymbol{q})$ is the 3×3 direction cosines matrix associated with the quaternion $\boldsymbol{q}$. This minimum rotation is

$$\boldsymbol{q}_{\rm min}(\hat{\boldsymbol{b}}_{\rm sc}, \hat{\boldsymbol{b}}_{\rm in}) = \begin{cases} \dfrac{1}{\sqrt{2\left(1+\hat{\boldsymbol{b}}_{\rm sc}^T \hat{\boldsymbol{b}}_{\rm in}\right)}} \begin{bmatrix} \hat{\boldsymbol{b}}_{\rm sc} \times \hat{\boldsymbol{b}}_{\rm in} \\ \left(1+\hat{\boldsymbol{b}}_{\rm sc}^T \boldsymbol{b}_{\rm in}\right) \end{bmatrix} & \text{if } -1 < \hat{\boldsymbol{b}}_{\rm sc}^T \hat{\boldsymbol{b}}_{\rm in} \\ \begin{bmatrix} \hat{\boldsymbol{v}} \\ 0 \end{bmatrix} & \text{if } -1 = \hat{\boldsymbol{b}}_{\rm sc}^T \hat{\boldsymbol{b}}_{\rm in} \end{cases} \tag{12}$$

where $\hat{\boldsymbol{v}}$ in the second condition is any unit vector that is perpendicular to both $\hat{\boldsymbol{b}}_{\rm sc}$ and $\hat{\boldsymbol{b}}_{\rm in}$.

This attitude representation isolates the attitude component with large uncertainty θ from the two components whose uncertainty is small, α_1 and α_2. The principal challenge of magnetometer-only spacecraft attitude and rate estimation is to estimate θ and the corresponding attitude rate from a time series of magnetometer measurements.[11] The parameters α_1 and α_2 are known a priori to be nearly zero.

The six inertia matrix parameters can be used to compute the moment of inertia matrix in spacecraft coordinates according to the following formula:

$$I_m = A_1(p_{I4}) A_2(p_{I5}) A_3(p_{I6}) \times \begin{bmatrix} \dfrac{p_{I2}^2 + p_{I3}^2}{12} & 0 & 0 \\ 0 & \dfrac{p_{I1}^2 + p_{I3}^2}{12} & 0 \\ 0 & 0 & \dfrac{p_{I1}^2 + p_{I2}^2}{12} \end{bmatrix} \times A_3^T(p_{I6}) A_2^T(p_{I5}) A_1^T(p_{I4}) \tag{13}$$

In this formula, $A_l(p)$ is the 3×3 direction cosines matrix for a rotation of p radians about the lth axis; p_{I1}, p_{I2}, and p_{I3} are the lengths of the sides of an equivalent uniform rectangular box multiplied by the square root of its mass; and p_{I4}, p_{I5}, and p_{I6} are three Euler angles that parameterize the rotation of this box with respect to spacecraft coordinates. Equation (13) ensures that I_m is a physically realizable moment of inertia matrix.

The $\boldsymbol{f}_i(\boldsymbol{x}_i, \boldsymbol{w}_i)$ dynamics function for this system uses Eq. (11) and its inverse together with Euler's equation of attitude dynamics and the quaternion kinematics equation to propagate the state from sample time t_i to sample time t_{i+1}. It first uses Eq. (11) to compute $\boldsymbol{q}_i = \boldsymbol{q}[(\boldsymbol{x}_i)_4, (\boldsymbol{x}_i)_5, (\boldsymbol{x}_i)_6; \hat{\boldsymbol{b}}_{{\rm sc}i}, \hat{\boldsymbol{b}}_{{\rm in}i}, \hat{\boldsymbol{v}}_{1i}, \hat{\boldsymbol{v}}_{2i}]$, where the notation $(\,)_l$ refers to the lth element of the vector in the parentheses. Next, it uses numerical integration to propagate Euler's equation and the quaternion kinematics equation

$$\dot{\boldsymbol{\omega}} = I_m^{-1}[-\boldsymbol{\omega} \times (I_m \boldsymbol{\omega}) + \boldsymbol{n}_{\rm GG}(\boldsymbol{q}; I_m) + A(\boldsymbol{q})\boldsymbol{w}_i] \tag{14a}$$

$$\dot{\boldsymbol{q}} = \frac{1}{2} \begin{bmatrix} 0 & \omega_3 & -\omega_2 & \omega_1 \\ -\omega_3 & 0 & \omega_1 & \omega_2 \\ \omega_2 & -\omega_1 & 0 & \omega_3 \\ -\omega_1 & -\omega_2 & -\omega_3 & 0 \end{bmatrix} \boldsymbol{q} \tag{14b}$$

to time t_{i+1} starting from the initial conditions $\boldsymbol{\omega}(t_i) = [(\boldsymbol{x}_i)_1; (\boldsymbol{x}_i)_2; (\boldsymbol{x}_i)_3]$ and $\boldsymbol{q}(t_i) = \boldsymbol{q}_i$ at time t_i. The term $\boldsymbol{n}_{\rm GG}(\boldsymbol{q}, I_m)$ on the right-hand side of Eq. (14a) is the gravity-gradient torque vector in spacecraft coordinates, and the term that involves $\boldsymbol{w}_i$ implies that this process noise vector is a 3×1 disturbance torque vector that is measured relative to inertial coordinates and that is constant during the sample interval. The results of the numerical integration are $\boldsymbol{\omega}(t_{i+1})$ and $\boldsymbol{q}(t_{i+1})$. The first three elements of the discrete-time dynamics function output vector $\boldsymbol{f}_i$ equal $\boldsymbol{\omega}(t_{i+1})$.

Elements 4–6 of $\boldsymbol{f}_i$ can be determined by inverting the formula for the attitude quaternion as a function of the three attitude parameters. This done by solving the equations $\boldsymbol{q}(t_{i+1}) = \boldsymbol{q}[(\boldsymbol{f}_i)_4, (\boldsymbol{f}_i)_5, (\boldsymbol{f}_i)_6; \hat{\boldsymbol{b}}_{{\rm sc}i+1}, \hat{\boldsymbol{b}}_{{\rm in}i+1}, \hat{\boldsymbol{v}}_{1i+1}, \hat{\boldsymbol{v}}_{2i+1}]$ for $(\boldsymbol{f}_i)_4$, $(\boldsymbol{f}_i)_5$, and $(\boldsymbol{f}_i)_6$. the derivation of analytic formulas for $(\boldsymbol{f}_i)_4$, $(\boldsymbol{f}_i)_5$, and $(\boldsymbol{f}_i)_6$ as functions of $\boldsymbol{q}(t_{i+1})$ is straightforward but tedious. It has been omitted for the sake of brevity.

This inversion has a singularity whenever the (α_1, α_2) part of the rotation in Eq. (11) has a magnitude of 180 deg. This is to be expected because all three-parameter attitude representations must have a singularity. The filter state is unlikely ever to encounter this singularity because it corresponds to huge magnetometer measurement errors.

The inversion's formula for $(\boldsymbol{f}_i)_6$, which yields θ_{i+1}, equals two times the output of a two-argument arctangent function. Thus, it is restricted to the range $-2\pi \le (\boldsymbol{f}_i)_6 \le 2\pi$. This leads to a discontinuity in $(\boldsymbol{f}_i)_6$ that has important consequences when using a UKF.

The remaining six elements of the vector function $\boldsymbol{f}_i$ characterize the dynamics of the inertia matrix parameters. These parameters are modeled as being constants. Therefore,

$$(\boldsymbol{f}_i)_l = (\boldsymbol{x}_i)_l \qquad \text{for} \quad l = 7, \ldots, 12 \tag{15}$$

The two-dimensional $\boldsymbol{h}_i(\boldsymbol{x}_i)$ measurement function takes the following simple form:

$$\boldsymbol{h}_i(\boldsymbol{x}_i) = \begin{bmatrix} (\boldsymbol{x}_i)_4 \\ (\boldsymbol{x}_i)_5 \end{bmatrix} = \begin{bmatrix} \alpha_{1i} \\ \alpha_{2i} \end{bmatrix} \tag{16}$$

This simple form is the direct result of Eq. (11), which defines α_1 and α_2 to be the rotation errors in the measured magnetic field unit direction vector.[11] This form of the measurement model dictates that each $\boldsymbol{y}_i$ measurement is known a priori: $\boldsymbol{y}_i = 0$. This is reasonable because the actual measurement $\hat{\boldsymbol{b}}_{sci}$ is used to define the attitude representation of Eq. (11).

C. Truth-Model Simulation

The performance of the system has been tested using a truth-model simulation. The simulation has realistic models of the atmospheric drag torque, the solar- and albedo-radiation pressure torque, and the torque due to the Earth's gravity gradient. Magnetometer scale-factor errors, misalignment errors, biases, and thermal noise are also modeled because these error sources act as disturbances to the estimator. They are sized to be typical of the errors that might exist subsequent to an attitude-independent in-flight magnetometer calibration. This is the same truth-model simulation as has been used in Sec. V.C of Ref. 11.

D. Filter Tuning and Initialization

The filter is tuned by selecting the Q_i and R_i covariance matrices. The process noise covariance in all of this paper's cases is $Q_i = I_{3\times 3} \times (9 \times 10^{-13}\mathrm{N}^2 \cdot \mathrm{m}^2 \cdot \mathrm{s})/\Delta t_i$, where $\Delta t_i = t_{i+1} - t_i$. This process noise intensity is between 3 and 10 times larger than the low-frequency limit of the power spectral density of the combined drag and solar/albedo-radiation pressure disturbance torques. This represents a reasonable agreement between actual process noise and the filter's model. The measurement noise covariance is $R_i = I_{2\times 2} \times (1.9 \times 10^{-5})$, which corresponds to a 0.5-deg per-axis measurement error standard deviation for the spacecraft-referenced magnetic field unit direction vector $\hat{\boldsymbol{b}}_{sci}$.

The filter is initialized by selecting the initial state estimate $\hat{\boldsymbol{x}}_0$ and the corresponding estimation error covariance matrix P_0. Three different initializations have been considered in this study. All of them set $\alpha_1 = \alpha_2 = 0$ in $\hat{\boldsymbol{x}}_0$, and the corresponding 2×2 block of P_0 is set equal to R_0. This initialization reflects the fact that these two attitude parameters are measurement error states. The first case starts with relatively small initial attitude and rate errors. The initial θ error is only 0.34 deg, and the corresponding variance in P_0 is $(15\ \mathrm{deg})^2$, which reflects the increased uncertainty normally associated with this attitude parameter. The initial attitude rate error in the direction parallel to the magnetic field vector, that is, in the θ direction, is 0.17 deg/s, and the initial rate error perpendicular to the magnetic field vector is 0.07 deg/s. The magnitude of the initial rate error is only 9.1% of the actual initial attitude rate. The initial rate error uncertainties in P_0 are set to $(2.1\ \mathrm{deg/s})^2$ along the magnetic field direction and to $(0.07\ \mathrm{deg/s})^2$ along the other two directions.

The second initialization case is similar to the first, except that the initial θ error is increased to 180 deg to test the filter's ability to converge from large initial errors. The corresponding variance in P_0 is increased to $(90\ \mathrm{deg})^2$. The initial attitude rate error parallel to the magnetic field also increases for this case to 1.54 deg/s, which is 77% the actual rate.

Like the second case, the third initialization also tests the filter's ability to converge from large initial errors, but its large uncertainty is in attitude rate about the magnetic field vector. The initial error in this component of the rate is set to 18 deg/s, and the variance in P_0 for this component is set to $(9\ \mathrm{deg/s})^2$. In addition, the initial θ error is 32.7 deg and has the same sign as the large initial attitude rate error about this same axis. This initial rate error is very large, nine times as large as the initial truth attitude rate. The initial attitude rate error for this third case causes a 180-deg increase in the attitude error during the filter's 10-s sample interval. Such a large error creates a risk of filter divergence due to an effect that is like aliasing.

All three cases use the same initial estimates and covariance matrix for the moment of inertia parameters $p_{I1}, \ldots, p_{I6}$. Consistent with the initial I_m estimate discussed in Sec. IV.A, $p_{I1} = p_{I2} = p_{I3} = 3.07\ \mathrm{kg}^{0.5} \cdot \mathrm{m}$ and $p_{I4} = p_{I5} = p_{I6} = 0$ deg. The block of P_0 that models the initial uncertainty in p_{I1}, p_{I2}, and p_{I3} is $\mathrm{diag}(6.1 \times 10^{-3}, 6.1 \times 10^{-3}, 6.1 \times 10^{-7})\ \mathrm{kg} \cdot \mathrm{m}^2$. These values reflect a 2.5% one-sigma uncertainty in the initial estimates of p_{I1} and p_{I2} and a 0.025% initial uncertainty in p_{I3}. The latter uncertainty enforces a soft constraint on the unobservable scaling of the spacecraft inertias. The P_0 block that corresponds to p_{I4}, p_{I5}, and p_{I6} is set to $I_{3\times 3} \times (15\ \mathrm{deg})^2$, which reflects the initial uncertainty in the spacecraft's principal axis directions.

E. Comparison Filters

Two nonlinear filters have been applied to the data, in addition to the BSEKF. One is an EKF, and the other is a UKF. The EKF uses the standard first-order technique that linearizes the $\boldsymbol{f}_i(\boldsymbol{x}_i, \boldsymbol{w}_i)$ dynamics function about the a posteriori $\boldsymbol{x}_i$ estimate and $\boldsymbol{w}_i = 0$ and that linearizes the $\boldsymbol{h}_{i+1}(\boldsymbol{x}_{i+1})$ measurement function about the a priori $\boldsymbol{x}_{i+1}$ estimate. It uses standard linear square-root information filter operations to perform each state propagation and measurement update.

The UKF includes two improvements in comparison to the filter defined in Ref. 3. First, it uses a better square-root filtering technique, one like that in Ref. 13, to perform its measurement update. Second, it includes a special fix to deal with the discontinuity at $(\boldsymbol{f}_i)_6 = \pm 2\pi$ because its sigma points may span this discontinuity. The modification to the UKF changes $(\boldsymbol{f}_i)_6$ for each sigma point other than the nominal point by adding or subtracting the multiple of 2π that causes the resulting $(\boldsymbol{f}_i)_6$ value to fall within $\pm\pi$ of the nominal $(\boldsymbol{f}_i)_6$ value. This fix exploits the fact that the dynamics function includes a redundancy within the range of continuous $(\boldsymbol{f}_i)_6$ output of two equivalent $(\boldsymbol{f}_i)_6$ values that differ by 2π. Several alternatives have been tried for unwinding the $(\boldsymbol{f}_i)_6$ wrap-around discontinuity, and this method has produced the best UKF filtering results for difficult cases that have large initial errors. Note that this discontinuity does not cause problems for the BSEKF or the EKF because it does not affect their linearizations.

The UKF parameters α, β, and κ tune the radius of its sigma points.[3] Recommended values are α in the range $10^{-4} \le \alpha \le 1$, $\beta = 2$ for Gaussian distributions, and $\kappa = 0$ or $\kappa = 3 - \dim([\boldsymbol{x}_i; \boldsymbol{w}_i]) = -12$. Eight different reasonable combinations of these parameters have been tried in an attempt to optimize the UKF. Two of these combinations have been used to produce the UKF results presented in this paper. Combination A (UKF A) consists of the values $\alpha = 1$, $\beta = 2$, and $\kappa = -12$, and the combination B (UKF B) values are $\alpha = 0.5$, $\beta = 2$, and $\kappa = -2$.

One might develop an additional comparison filter using multiple-model techniques.[1] A multiple-model filter has not been considered because such filters are inefficient for combined state/parameter estimation problems that have a large number of uncertain parameters. A multiple-model filter includes a separate model and filter for each member of a set of grid points in parameter space. Additional filters are needed if one wants to use multiple-model techniques to deal with large initial condition uncertainties.[11] The required number of filters becomes prohibitive for the present problem.

F. Filtering Results

The BSEKF performed very well on all three attitude estimation cases. An implementation that uses $m_{\mathrm{target}} = 30$ explicit nonlinear smoothing samples converges in all three cases. It reduces its total attitude error to less than 2 deg in less than 0.1 h, and it settles into steady-state performance in less than 1 h. Its steady-state peak per-axis attitude error is 0.95 deg, and its peak total attitude error is 1.34 deg.

The quantity $\|\hat{I}_m \mathrm{tr}(I_m)/\mathrm{tr}(\hat{I}_m) - I_m\|/\|I_m\|$ is a good metric of the accuracy with which the filter's moment of inertia matrix estimate $\hat{I}_m$ approximates its true inertia matrix I_m (Ref. 11). This quantity is the fractional matrix error norm of the estimate. The trace ratio in this expression is included to remove the unobservable (and ignorable) overall scaling. This error starts out at 0.05585, or 5.585%. After

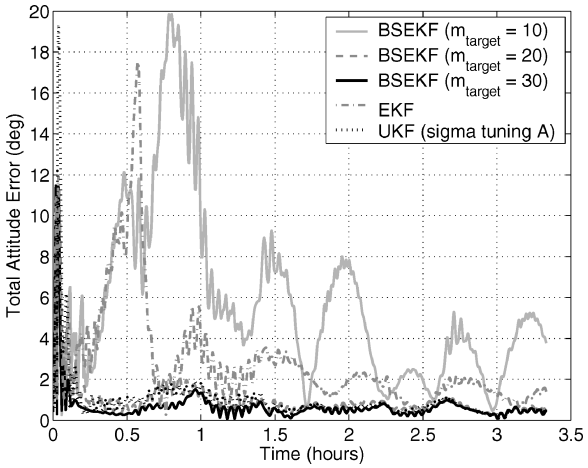

Fig. 2 Total attitude error time histories for three versions of BSEKF, EKF, and UKF A when starting from moderate initial errors.

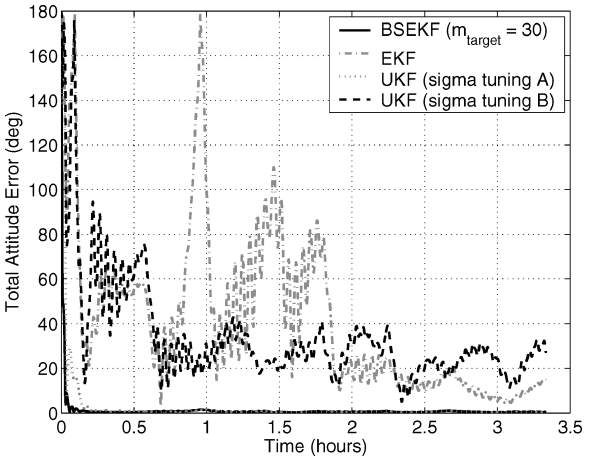

Fig. 3 Total attitude error time histories that result from 180-deg initial attitude error for BSEKF, EKF, UKF A, and UKF B.

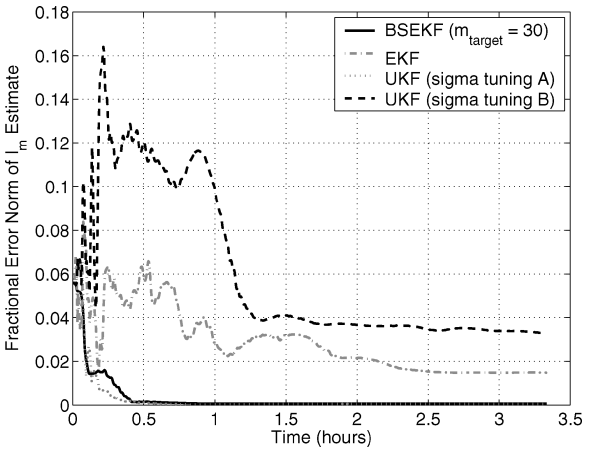

Fig. 4 Fractional error norm time histories of I_m estimates for BSEKF, EKF, UKF A, and UKF B when starting from 180-deg initial attitude error.

an hour of filtering, it settles to 0.00065, which represents a 98.8% decrease in the inertia matrix modeling error.

This performance is far superior to the performance obtained on this same problem using the filter described in Ref. 11. Its filter is a variant of the IEKF that also employs statistical hypothesis testing to enhance its global convergence properties. It has a peak steady-state per-axis attitude error of 18 deg, and it reduces the fractional I_m error only to 0.029 (2.9%). The current investigation was prompted by this poor performance.

The BSEKF also does better than the EKF and the UKF. Consider case 1, which has moderate initial errors. Figure 2 shows a plot of the total attitude estimation error time histories produced by five different filters for this case. The worst two filters are the EKF (darker gray dash–dotted curve) and the BSEKF that uses $m_{\text{target}} = 10$ sample intervals (gray solid curve). Greatly improved performance is obtained using the BSEKF with $m_{\text{target}} = 20$ (darker gray dashed curve). The best performing filters are UKF A (black dotted curve) and the BSEKF with $m_{\text{target}} = 30$ (black solid curve). These latter two curves settle to under 2 deg of total attitude error in less than 0.2 h, and after 2.7 h of operation they fall almost on top of each other. Although it is hard to tell from Fig. 2, the BSEKF with $m_{\text{target}} = 30$ settles faster than UKF A, and its peak total attitude error for all times past 1 h is 0.49 deg smaller than that of UKF A.

The $m_{\text{target}} = 30$ BSEKF also estimates the moment of inertia matrix with slightly more accuracy than does UKF A. The final fractional error norm is 0.00059 for this BSEKF and 0.00078 for UKF A. The fractional moment of inertia estimation error norms for the other filters follow the same trend in their attitude errors: The BSEKF with $m_{\text{target}} = 10$ is 9.5 times less accurate than the BSEKF with $m_{\text{target}} = 30$, the EKF is 3.5 times less accurate, and the BSEKF with $m_{\text{target}} = 20$ is 41% less accurate.

It is interesting that the best BSEKF achieves improved accuracy vs UKF A for case 1. The BSEKF only computes first derivatives of its nonlinear dynamics and measurement functions, but the UKF is specifically tailored to capture the effects of these functions' curvature. The important point is that the BSEKF also captures curvature effects through the Gauss–Newton iterations that it uses to solve its nonlinear smoothing problem.

Another important point is that the BSEKF will be better than other filters only if the number of explicitly optimized sample intervals m_{target} is sufficiently large. The worst of the five filters is the BSEKF with $m_{\text{target}} = 10$, but the best of them is the BSEKF with $m_{\text{target}} = 30$. A rule of thumb is to choose m_{target} large enough to allow the filter to converge nearly to steady-state when $k = m_{\text{target}}$. Figure 2 indicates that this rough convergence time is on the order of 0.15 h. This time span translates into $m_{\text{target}} = 54$. The value $m_{\text{target}} = 30$ is much closer to 54 than is the value $m_{\text{target}} = 10$, which is why the corresponding BSEKF functions well.

It is surprising that the BSEKF with $m_{\text{target}} = 10$ performs worse than the EKF. The EKF can be interpreted as being a BSEKF that uses $m_{\text{target}} = 1$ and only one Gauss–Newton step. Given its lower m_{target} value, one would expect the EKF to have poorer performance than the BSEKF with $m_{\text{target}} = 10$. The counterintuitive superior performance of the EKF in comparison to this BSEKF may be a case of "ignorance is bliss" vs "a little knowledge is dangerous."

The performance gains of the BSEKF are more significant when it must deal with large initial errors. Consider Figs. 3 and 4, in which case two performance results are compared for the following four filters: the $m_{\text{target}} = 30$ BSEKF, the EKF, UKF A, and UKF B. Recall that case 2 starts with a 180-deg initial θ attitude error. Figure 3 shows plots of the total attitude error time histories for the four filters, and Fig. 4 shows plots of the time histories of their fractional moment of inertia estimation error norms. The BSEKF (black solid curve) has the fastest attitude error convergence. It reduces its peak total attitude error from 180 deg to less than 2 deg in less than 0.1 h. UKF A (gray dotted curve) is comparable to the BSEKF. The EKF (gray dash–dotted curve) and UKF B (black dashed curve) both perform poorly. If one considers that all of the filters have reached steady state after 2.5 h, then the peak steady-state total attitude errors for the four filters are 32.8 deg for UKF B, 20.8 deg for the EKF, 1.0 deg for the BSEKF, and 0.9 deg for UKF A.

The Fig. 4 fractional I_m error norm time histories tell a similar story. UKF A and the BSEKF converge much faster than the other two filters, and their steady-state errors are much lower. The I_m estimates of UKF A and of the BSEKF both nearly reach steady state in 0.5 h, whereas UKF B and the EKF take about 2.5 h to converge. The steady-state error norms for the four filters' I_m estimates are 0.00052 for UKF A, 0.00059 for the BSEKF, 0.01483 for the EKF, and 0.03271 for UKF B. These relative accuracies are similar to

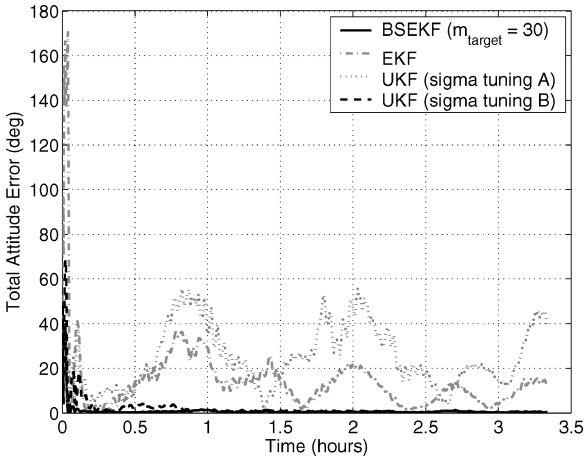

Fig. 5 Effect of 18-deg/s initial attitude rate error on total attitude error time histories for BSEKF, EKF, UKF A, and UKF B.

those of the attitude errors because they are closely related due to the filters' reliance on the Euler attitude dynamics model.

The results in Figs. 2–4 might lead one to conclude that UKF A is comparable to or even better than the BSEKF, but case 3 shows this to be untrue. Recall that this case starts with an initial attitude rate error of 18 deg/s about an axis aligned with the Earth's magnetic field vector. Figure 5 shows plots of the case 3 total attitude error time histories for these same four filters. The BSEKF (black solid curve) performs as well as it does for cases 1 and 2. The EKF (gray dash–dotted curve) exhibits slightly better performance than for case 2; the excursion to a 180-deg error at $t = 1$ h is gone, and its peak error after 2.5 h is reduced to 15.7 deg. UKF B (black dashed curve) shows a more significant performance improvement: It converges to an attitude error of less than 4.5 deg after 0.5 h and to less than 2 deg after 1 h. Seriously degraded, however, is the performance of UKF A (gray dotted curve). Its total attitude error rises to values as high 46 deg even after 3 h of operation.

To summarize the three cases, the BSEKF performs significantly better than the EKF in all three cases, and it performs significantly better than at least one candidate UKF when there are large initial condition errors. One might be tempted to think that the UKF is comparable to the BSEKF because it is possible to tune the UKF's sigma point parameters so that it performs about as well as the BSEKF in each considered case. This is not a fair comparison, however, because it is not possible to know a priori which UKF tunings to use because the best tuning parameters are functions of the unknown initial errors as demonstrated by the wide discrepancies in the performances of UKF A and UKF B for filtering cases 2 and 3 (review Figs. 3–5).

The sensitivity of the UKF to its sigma point tuning parameters is actually worse than is indicated by the results in Figs. 3–5. All of the six other reasonable tuning combinations that have been tried yield a divergent UKF for case 3, and two of the other tunings yield a divergent UKF for case 2. Thus, the performance of the UKF is sensitive to its sigma point tunings, whereas that of the BSEKF is robust if m_{target} is chosen to be sufficiently large.

The superior performance of the EKF in comparison to UKF A for case 3 and the comparable to superior performance of the EKF in comparison to UKF B for case 2 both come as surprises. Perhaps these surprises are the results of the discontinuity of $[\boldsymbol{f}_i(\boldsymbol{x}_i, \boldsymbol{w}_i)]_6$ that has been discussed in Secs. IV.B and IV.E. Another possibility is that the UKF has trouble with the singularity of the $\boldsymbol{f}_i(\boldsymbol{x}_i, \boldsymbol{w}_i)$ function, which is caused by the use of a three-parameter attitude representation. Some of the UKF's sigma points may stray near to the singularity, even when its state estimate is remote from it. These stray points can wreak havoc with the UKF calculations. Discontinuities and singularities in $\boldsymbol{f}_i(\boldsymbol{x}_i, \boldsymbol{w}_i)$ can be avoided by adapting the quaternion-based attitude determination UKF from Ref. 14 to estimate moment of inertia parameters in addition to attitude and rate. Such a UKF may perform better than the EKF in all cases, but it is never expected to be superior to the BSEKF for all reasonable sigma point tunings and all initial errors.

The BSEKF's rms estimation errors have been compared to the standard deviations computed from its calculated estimation error covariance matrix. The ratios of the actual rms errors to the computed standard deviations fall in the range from 2.2 to 2.6 for the attitude errors and in the range from 1.6 to 4.0 for the attitude rate errors. This indicates that the filter is somewhat mistuned, but not drastically so. This level of disagreement between actual rms errors and computed standard deviations is reasonable given that the truth-model's process noise disturbance torques and measurement errors are not the white noise processes assumed by the filter.

The BSEKF's performance improvements come at a large computational cost. When $m_{\text{target}} = 30$, the BSEKF requires between 30 and 60 times as much computation as an EKF for each iteration of its Gauss–Newton nonlinear least-squares solver, and multiple iterations are usually required to solve the problem. The BSEKF has been run with a maximum limit of 10 Gauss–Newton iterations per problem, and the average number of iterations for all three cases was 3.1 per problem. During the first 0.5 h of filtering, the average number of iterations per problem was higher, between 8.4 and 8.6, with many problems terminating prematurely at the maximum of 10 iterations. During the remaining time, the average number of iterations per problem was between 2.1 and 2.2, with the vast majority terminating after just two iterations. This indicates that the smoothing problems are harder to solve when the inertia matrix uncertainties are large. This makes sense because these introduce strong nonlinearities in the estimation problem. A direct comparison of computation time for these cases shows that the BSEKF with $m_{\text{target}} = 30$ requires between 165 and 175 times as much processing as the EKF. These ratios range from $1.77m_{\text{target}}$ to $1.88m_{\text{target}}$ times the average number of Gauss–Newton iterations per problem. They are consistent with the analysis in Sec. III.E. The computation time for the BSEKF is only about 18–19 times that of the UKF for this problem. This particular UKF is slower than a typical UKF due to its use of square-root techniques.

This comparison between the BSEKF and other filters is preliminary in nature. It has been presented to illustrate the BSEKF's possible performance improvements. The most appropriate applications for the BSEKF remain to be determined. Given the large computational expense of this new filter, it should be applied sparingly. It could be used more liberally in off-line studies as a sort of "gold standard" against which to benchmark filters that require less computation. Additional future work should include Monte Carlo simulation studies that more fully explore the BSEKF's performance and relative merits, including the effects of varying degrees of process noise and measurement noise.

V. Conclusions

A new nonlinear Kalman filter has been developed. It retains more of the nonlinearities of the original problem to achieve improved accuracy and convergence reliability for problems with strong nonlinear effects. It is based on Gauss–Newton iterative solution of a nonlinear smoothing problem at each sample time. The nonlinear smoothing problem retains the exact nonlinearities of a fixed number of sample intervals before the time of interest, and it approximates the effects of earlier sample times' dynamics and measurement nonlinearities by using an EKF-like strategy. The globally convergent nature of the Gauss–Newton nonlinear least-squares optimization algorithm yields a filter that is very likely to converge to a nearly optimal solution of the original nonlinear estimation problem if enough sample intervals are treated exactly.

Test results for a simulated estimation problem confirm that the BSEKF has dramatically better performance than an EKF and a representative UKF when using a time series of magnetometer measurements to estimate the attitude, attitude rate, and moment of inertia matrix of a tumbling microsatellite. The BSEKF is able to achieve subdegree accuracy on all three axes, despite the inability to deduce the attitude and rate about the instantaneous magnetic field directly from the current measurement. Its 5.6% initial moment of inertia estimation error converges to 0.065% after 1 h of

filtering. It can achieve this performance even when it starts from an initial attitude error of 180 deg or an initial attitude rate error of 180 deg per sample interval. For cases with large initial attitude or rate errors, the UKF can exhibit comparable performance to the BSEKF, it can produce peak steady-state attitude errors that are 30 to 40 times larger than those of the BSEKF, or it can diverge, depending on the values of its sigma point tuning parameters. Similarly, the EKF's steady-state errors for cases with large initial attitude or rate uncertainty are 16–25 times larger than those of the BSEKF.

Acknowledgments

Helpful comments from Mark Campbell, John Crassidis, Todd Humphreys, Jeremy Kasdin, F. Landis Markley, and the reviewers have contributed to the final form of this paper. Michael Hough's insistence that the backward-smoothing extended Kalman filter is nothing more than a batch filter provided a helpful prompting for the author to clarify the relationship of this paper to the literature on batch filters.

References

[1]Bar-Shalom, Y., Li, X.-R., and Kirubarajan, T., *Estimation with Applications to Tracking and Navigation*, Wiley, New York, 2001, pp. 92–98, 381–395, 404–409, and 441–466.

[2]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. AC-45, No. 3, 2000, pp. 477–482.

[3]Wan, E. A., and van der Merwe, R., "The Unscented Kalman Filter," *Kalman Filtering and Neural Networks*, edited by S. Haykin, Wiley, New York, 2001, pp. 221–280.

[4]Shapiro, C., "New Formulation and Solution to the Navigation State Vector Estimation Problem," *Proceedings of the ION NTM 2004*, Inst. of Navigation, Fairfax, VA, 2004, pp. 840–861.

[5]Brookner, E., *Tracking and Kalman Filtering Made Easy*, Wiley, New York, 1998, pp. 370–374.

[6]Gelb, A. (ed.), *Applied Optimal Estimation*, MIT Press, Cambridge, MA, 1974, pp. 214–216.

[7]Sage, A. P., and Melsa, J. L., *System Identification*, Academic Press, New York, 1971, pp. 57–65 and 92–109.

[8]Gordon, N. J., Salmond, D. J., and Smith, A. F. M., "Novel Approach to Nonlinear/Non-Gaussian Bayesian State Estimation," *IEE Proceedings—F, Radar and Signal Processing*, Vol. 140, No. 2, 1993, pp. 107–113.

[9]Gill, P. E., Murray, W., and Wright, M. H., *Practical Optimization*, Academic Press, New York, 1981, pp. 88–93, 99–105, and 133–141.

[10]Bierman, G. J., *Factorization Methods for Discrete Sequential Estimation*, Academic Press, New York, 1977, pp. 69–76, 115–122, and 214–217.

[11]Psiaki, M. L., "Global Magnetometer-Based Spacecraft Attitude and Rate Estimation," *Journal of Guidance, Control, and Dynamics*, Vol. 27, No. 2, 2004, pp. 240–250.

[12]Reynolds, R. G., "Quaternion Parameterization and a Simple Algorithm for Global Attitude Estimation," *Journal of Guidance, Control, and Dynamics*, Vol. 21, No. 4, 1998, pp. 669–671.

[13]Brunke, S., and Campbell, M. E., "Square Root Sigma Point Filtering for Real-Time, Nonlinear Estimation," *Journal of Guidance, Control, and Dynamics*, Vol. 27, No. 2, 2004, pp. 314–317.

[14]Crassidis, J. L., and Markley, F. L., "Unscented Filtering for Spacecraft Attitude Estimation," *Journal of Guidance, Control, and Dynamics*, Vol. 26, No. 4, 2003, pp. 536–542.