JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 30, No. 1, January–February 2007

Derivation and Simulation Testing of a Sigma-Points Smoother

Mark L. Psiaki*
Cornell University, Ithaca, New York 14853-7501
and
Massaki Wada
Saila System Inc., Tokyo 141-0022, Japan

DOI: 10.2514/1.22875

A sigma-points fixed-interval smoothing algorithm has been derived from first principles and tested using data from a truth-model simulation. This smoothing algorithm extends the use of sigma-points estimation methods, which have benefited the practice of discrete-time nonlinear filtering, into the realm of discrete-time nonlinear smoothing, where similar benefits are expected. Two equivalent forms of the algorithm have been developed based on Bayesian analysis of smoothing. One form is easier to derive and gives rise to the simplest equations, whereas the other form develops insight into the smoothing calculations by deriving them as a nonlinear pseudomeasurement update that is executed using sigma-points methods. Both forms of the algorithm use data from a forward pass of a sigma-points filter, and the result is a sigma-points adaptation of the Rauch–Tung–Striebel smoother. The algorithm has been tested using a truth-model simulation of a difficult gyroless attitude determination problem that uses only magnetometer data and that simultaneously estimates moment-of-inertia parameters to improve its Euler dynamics propagation of its attitude-rate estimates. The sigma-points smoother produces smaller estimation errors than an extended Kalman smoother, and its accuracy is comparable to that of a nonlinear batch smoother on some test cases.

I. Introduction

THE problem of estimating the states of a dynamic system based on noisy measurement data has been solved through the use of Kalman filters and related filters for real-time systems and through the use of smoothers for offline, noncausal applications. Linear filtering and smoothing problems with Gaussian noise have exact solutions using the Kalman filter and any one of several related smoothers [1,2]. For nonlinear problems, however, the situation is less settled. Exact solutions are possible only for special classes of systems or if one implements a maximum a posteriori estimator by performing iterative numerical optimization of a large batch-type problem [3].

The difficulty for nonlinear problems is that the underlying a posteriori state probability distributions are non-Gaussian and often cannot be represented by a finite set of parameters. Therefore, nonlinear estimators must work with approximations of these distributions. If the nonlinearities are strong enough, then errors in the approximate distributions can seriously degrade accuracy or cause divergence of the estimation error.

Various approximate nonlinear filtering techniques have been developed, the most popular being the extended Kalman filter (EKF) [1,2], the unscented Kalman filter (UKF), also known as the sigma-points filter [4,5], and the particle filter (PF) [6,7]. The EKF and the UKF use Gaussian approximations that reproduce the first and second moments of the true distributions with varying degrees of accuracy. The PF uses Monte Carlo techniques to represent the distribution by samples, possibly with weightings.

The UKF sigma-points technique is gaining in popularity for several reasons. It is simpler to implement than an EKF because it does not require the computation of partial derivatives of the dynamics and measurements functions. It can approximate the mean and covariance of the true a posteriori probability distribution better than the EKF because its calculations can match higher-order terms in the Taylor series of these quantities [5]. This improved approximation can lead to significant accuracy improvements in comparison to an EKF [3,8]. Although a UKF often requires more computation than an EKF, perhaps about two to three times as much, it can be orders of magnitude faster than a PF on high-dimensional problems. Its required number of sigma points scales linearly with the dimension of the state and the process noise, but the required number of points in a PF seems to grow exponentially so that a PF's Monte Carlo techniques can become impractical even for systems with only three or four state dimensions [9].

Given the success and increasing popularity of the sigma-points technique for discrete-time filtering problems, it seems sensible to apply this same technique to discrete-time smoothing problems. Smoothing algorithms are important because they offer superior accuracy in situations where one can use noncausal calculations, as in offline processing of science mission data [10,11]. Backward nonlinear smoothing over past sample intervals also offers the potential to improve causal nonlinear filters because the smoothed results can be used to improve the approximation of the conditional probability density that applies at the current sample [3].

Very little work has been done on sigma-points smoothing algorithms. Wan and van der Merwe [5] and Venkatasubramanian and Leung [12] develop unscented smoothers that rely on having an inverse dynamics function. This is impractical in many situations. Ypma and Heskes [13] present filtering and smoothing algorithms that are based on message passing concepts and that use sigma-points techniques. Their sigma-points smoothing algorithm does not need an inverse dynamics function, but this algorithm is not easily identifiable as being similar to the UKF algorithms of [4,5]. It is difficult even to distill the concepts of [13] into a specific smoothing algorithm that can be encoded in software.

Mao et al. [14] appears to be the only work that presents a sigma-points smoothing algorithm in terms similar to the UKF algorithms of [4,5] without requiring inversion of the dynamics model. Their discrete-time smoothing algorithm is presented in the context of an application, not as the central result of the paper. It is easy to prove

Presented as Paper 6349 at the AIAA Guidance, Navigation, and Control Conference, Keystone, CO, 21–24 August 2006; received 31 January 2006; revision received 7 August 2006; accepted for publication 11 August 2006.

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

that this algorithm is equivalent to the Rauch–Tung–Striebel (RTS) smoother [15] when solving a linear/Gaussian smoothing problem, but Mao et al. [14] offer no demonstration that their algorithm is a sensible sigma-points approximation to the solution of a nonlinear smoothing problem.

The present paper makes three contributions to the subject of sigma-points smoothing. First, it derives the fixed-interval discrete-time smoothing algorithm of [14] using Bayesian analysis and standard sigma-points techniques. The algorithm developed here is an augmentation of the [14] algorithm that also computes process noise estimates and their error covariances. It uses a forward filtering pass followed by a backward smoothing pass, similar to the RTS smoother [15] and to the nonlinear smoother of [13]. The second contribution is the development of an alternate, but equivalent, version of this algorithm that provides insight into the nature of sigma-points smoothing. The third contribution is the application of the sigma-points smoothing algorithm to a significant nonlinear smoothing problem from the field of attitude determination. The sigma-points solution is compared to the solution of an extended Kalman smoother and to the solution of a nonlinear batch smoother that uses the iterative Gauss–Newton method.

The derivations, simulation results, and conclusions of this paper are contained in its five remaining sections. Section II briefly reviews the sigma-points filtering algorithm (i.e., the UKF) because the outputs of a forward-pass sigma-points filter are needed as inputs to the backward-pass of the smoothing algorithm. The smoothing algorithm of [14] is derived in Sec. III. Section IV derives an alternate form of the algorithm based on the idea of a pseudomeasurement. Section V uses a truth-model simulation of a gyroless satellite attitude determination problem to evaluate the smoother and compare it to competing algorithms. Section VI summarizes this paper's results and presents its conclusions.

II. Review of Sigma-Points Filtering to Define Forward Filtering Pass

The sigma-points filter, also known as the UKF, is thoroughly described in [4,5]. The formulation of [5] is repeated in this section to define the forward filtering pass that is needed to prepare for the backward-pass of the fixed-interval smoother. The present formulation has several notational differences from that of [5].

A. Problem Model

The sigma-points filter and the sigma-points fixed-interval smoother both form estimates of the state of the following nonlinear discrete-time stochastic system:

$$\boldsymbol{x}_{k+1} = \boldsymbol{f}_k(\boldsymbol{x}_k, \boldsymbol{w}_k) \quad \text{for } k = 0, 1, 2, \ldots N-1 \tag{1a}$$

$$\boldsymbol{y}_{k+1} = \boldsymbol{h}_{k+1}(\boldsymbol{x}_{k+1}, \boldsymbol{v}_{k+1}) \quad \text{for } k = 0, 1, 2, \ldots N-1 \tag{1b}$$

where $\boldsymbol{x}_k$ is the n-dimensional system state vector at sample k, $\boldsymbol{w}_k$ is the m-dimensional process noise vector that acts from sample k to sample $k+1$, $\boldsymbol{f}_k(\boldsymbol{x}_k, \boldsymbol{w}_k)$ is the n-dimensional discrete-time dynamics nonlinear transition function, N is the terminal sample of the fixed-interval smoothing problem, $\boldsymbol{y}_{k+1}$ is the p-dimensional sensor measurement vector at sample $k+1$, $\boldsymbol{h}_{k+1}(\boldsymbol{x}_{k+1}, \boldsymbol{v}_{k+1})$ is the p-dimensional nonlinear measurement function, and $\boldsymbol{v}_{k+1}$ is the p-dimensional measurement error vector.

The estimator uses the following statistical models: The noise vectors $\boldsymbol{w}_k$ and $\boldsymbol{v}_{k+1}$ are zero-mean, uncorrelated, discrete-time Gaussian white-noise processes with covariances $E\{\boldsymbol{w}_k\boldsymbol{w}_k^T\} = Q_k$ and $E\{\boldsymbol{v}_{k+1}\boldsymbol{v}_{k+1}^T\} = R_{k+1}$. The true state at $k = 0$ is a sample from the Gaussian distribution with known mean $E\{\boldsymbol{x}_0\} = \hat{\boldsymbol{x}}_0$ and covariance $E\{(\boldsymbol{x}_0 - \hat{\boldsymbol{x}}_0)(\boldsymbol{x}_0 - \hat{\boldsymbol{x}}_0)^T\} = \hat{P}_0$.

B. Notation to Distinguish A Priori, A Posteriori, and Smoothed Estimates

The derivations of this paper are developed using mean values and estimation error covariances for the a priori, a posteriori, and smoothed state estimates. The a priori state estimate, i.e., the a priori mean, and its estimation error covariance at sample k are denoted by $\bar{\boldsymbol{x}}_k$ and $\bar{P}_k$. The qualifier a priori indicates that these quantities are conditioned on the data $\{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_{k-1}\}$. The a posteriori state estimate and its error covariance are conditioned on the data $\{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_k\}$, and they are defined to be $\hat{\boldsymbol{x}}_k$ and $\hat{P}_k$. The smoothed state estimate and its estimation error covariance are $\boldsymbol{x}_k^*$ and P_k^*, and they are conditioned on the entire data span, i.e., on $\{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N\}$.

C. Sigma-Points Filter Calculations for One Sample Interval

A sigma-points filter starts with $\hat{\boldsymbol{x}}_k$ and $\hat{P}_k$, and it performs dynamic propagation and measurement update calculations to determine $\hat{\boldsymbol{x}}_{k+1}$ and $\hat{P}_{k+1}$. Its calculations approximate all relevant probability distributions using Gaussians, and it uses the sigma-points approach of [4,5] to approximately compute the means and covariances of nonlinear function outputs when the functions have Gaussian random inputs.

These calculations use matrix square roots of $\hat{P}_k$, Q_k, and R_{k+1} to generate a set of samples in $[\boldsymbol{x}_k; \boldsymbol{w}_k; \boldsymbol{v}_{k+1}]$ space. These square roots are S_{xk}, S_{wk}, and $S_{v(k+1)}$, and they obey the relationships

$$S_{xk}^T S_{xk} = \hat{P}_k, \qquad S_{wk}^T S_{wk} = Q_k, \quad \text{and} \quad S_{v(k+1)}^T S_{v(k+1)} = R_{k+1} \tag{2}$$

These square roots can be computed by using Cholesky factorization [16]. They are not unique because S_{xk}, S_{wk}, and $S_{v(k+1)}$ can be modified through premultiplication by any orthonormal matrix without violating the relationships in Eq. (2).

The matrix square roots are used to generate $2(n + m + p) + 1$ sigma-points according to the following formulas:

$$\mathcal{X}_k^0 = \hat{\boldsymbol{x}}_k, \qquad \mathcal{W}_k^0 = 0, \quad \text{and} \quad \mathcal{V}_{k+1}^0 = 0 \tag{3a}$$

$$\begin{gathered} \mathcal{X}_k^i = \hat{\boldsymbol{x}}_k + \alpha\sqrt{(n+m+p+\kappa)}S_{xk}^T\boldsymbol{e}^i \\ \mathcal{W}_k^i = 0, \quad \text{and} \quad \mathcal{V}_{k+1}^i = 0 \quad \text{for } i = 1, \ldots, n \end{gathered} \tag{3b}$$

$$\begin{gathered} \mathcal{X}_k^i = \hat{\boldsymbol{x}}_k - \alpha\sqrt{(n+m+p+\kappa)}S_{xk}^T\boldsymbol{e}^{i-n} \\ \mathcal{W}_k^i = 0, \quad \text{and} \quad \mathcal{V}_{k+1}^i = 0 \quad \text{for } i = (n+1), \ldots, (2n) \end{gathered} \tag{3c}$$

$$\begin{gathered} \mathcal{X}_k^i = \hat{\boldsymbol{x}}_k \\ \mathcal{W}_k^i = \alpha\sqrt{(n+m+p+\kappa)}S_{wk}^T\boldsymbol{e}^{i-2n}, \quad \text{and} \quad \mathcal{V}_{k+1}^i = 0 \\ \text{for } i = (2n+1), \ldots, (2n+m) \end{gathered} \tag{3d}$$

$$\begin{gathered} \mathcal{X}_k^i = \hat{\boldsymbol{x}}_k \\ \mathcal{W}_k^i = -\alpha\sqrt{(n+m+p+\kappa)}S_{wk}^T\boldsymbol{e}^{i-2n-m}, \quad \text{and} \quad \mathcal{V}_{k+1}^i = 0 \\ \text{for } i = (2n+m+1), \ldots, 2(n+m) \end{gathered} \tag{3e}$$

$$\begin{gathered} \mathcal{X}_k^i = \hat{\boldsymbol{x}}_k \\ \mathcal{W}_k^i = 0, \quad \text{and} \quad \mathcal{V}_{k+1}^i = \alpha\sqrt{(n+m+p+\kappa)}S_{v(k+1)}^T\boldsymbol{e}^{i-2(n+m)} \\ \text{for } i = (2n+2m+1), \ldots, (2n+2m+p) \end{gathered} \tag{3f}$$

$$\begin{gathered} \mathcal{X}_k^i = \hat{\boldsymbol{x}}_k \\ \mathcal{W}_k^i = 0, \quad \text{and} \\ \mathcal{V}_{k+1}^i = -\alpha\sqrt{(n+m+p+\kappa)}S_{v(k+1)}^T\boldsymbol{e}^{i-2(n+m)-p} \\ \text{for } i = (2n+2m+p+1), \ldots, 2(n+m+p) \end{gathered} \tag{3g}$$

where $[\mathcal{X}^i_k; \mathcal{W}^i_k; \mathcal{V}^i_{k+1}]$ for $i = 0, \ldots 2(n+m+p)$ are the sigma-points and $\boldsymbol{e}^i$ is a unit column vector of appropriate dimensions with a 1 in the ith row and zeros in all other rows. The quantities α and κ are scaling parameters of the sigma points. They determine the extent of the deterministic point distribution that is defined by Eqs. (3a–3g). Typical values of α lie in the range $10^{-4} \leq \alpha \leq 1$, and κ is usually set to equal 0 or $3 - (n + m + p)$ [5]. Small values of α cause the points to be grouped tightly around $[\hat{\boldsymbol{x}}_k;0;0]$; in this case, the UKF tends to behave like an EKF that uses central numerical differentiation to compute its Jacobian matrices.

The sigma points from Eqs. (3a–3g) are used in conjunction with weights to estimate the a priori mean and covariance of the combined state and measurement vector at sample $k + 1$. They are first used to compute a priori state and measurement sigma points of the next sample:

$$\tilde{\mathcal{X}}^i_{k+1} = \boldsymbol{f}_k(\mathcal{X}^i_k, \mathcal{W}^i_k) \quad \text{and} \quad \mathcal{Y}^i_{k+1} = \boldsymbol{h}_k(\tilde{\mathcal{X}}^i_{k+1}, \mathcal{V}^i_{k+1}) \quad \text{for } i = 0, \ldots, 2(n+m+p) \tag{4}$$

where the tilde on $\tilde{\mathcal{X}}^i_{k+1}$ has been added to distinguish these propagated state sigma points from the new sigma points that will be generated when the UKF algorithm starts its processing for the interval from sample $k + 1$ to sample $k + 2$. These propagated sigma points are used to estimate the required means and covariances according to the following formulas:

$$\bar{\boldsymbol{x}}_{k+1} = \sum_{i=0}^{2(n+m+p)} W^i_a \tilde{\mathcal{X}}^i_{k+1}, \qquad \bar{\boldsymbol{y}}_{k+1} = \sum_{i=0}^{2(n+m+p)} W^i_a \mathcal{Y}^i_{k+1} \tag{5a}$$

$$\bar{P}_{k+1} = \sum_{i=0}^{2(n+m+p)} W^i_c \left(\tilde{\mathcal{X}}^i_{k+1} - \bar{\boldsymbol{x}}_{k+1}\right)\left(\tilde{\mathcal{X}}^i_{k+1} - \bar{\boldsymbol{x}}_{k+1}\right)^T \tag{5b}$$

$$\bar{P}_{xy(k+1)} = \sum_{i=0}^{2(n+m+p)} W^i_c \left(\tilde{\mathcal{X}}^i_{k+1} - \bar{\boldsymbol{x}}_{k+1}\right)\left(\mathcal{Y}^i_{k+1} - \bar{\boldsymbol{y}}_{k+1}\right)^T \tag{5c}$$

$$\bar{P}_{yy(k+1)} = \sum_{i=0}^{2(n+m+p)} W^i_c \left(\mathcal{Y}^i_{k+1} - \bar{\boldsymbol{y}}_{k+1}\right)\left(\mathcal{Y}^i_{k+1} - \bar{\boldsymbol{y}}_{k+1}\right)^T \tag{5d}$$

where the weights W^i_a and W^i_c for $i = 0, \ldots, 2(n+m+p)$ act like probability masses. The W^i_a weights are for the mean calculations, and the W^i_c weights are for the covariance calculations. They are determined from the formulas [5]

$$\begin{gathered} W^0_a = \frac{\alpha^2(n+m+p+\kappa) - (n+m+p)}{\alpha^2(n+m+p+\kappa)} \\ W^0_c = \frac{\alpha^2(n+m+p+\kappa) - (n+m+p)}{\alpha^2(n+m+p+\kappa)} + (1 - \alpha^2 + \beta), \quad \text{and} \\ W^i_a = W^i_c = \frac{1}{2\alpha^2(n+m+p+\kappa)} \\ \text{for } i = 1, \ldots, 2(n+m+p) \end{gathered} \tag{6}$$

The quantity β is yet another scaling parameter of the sigma-points calculation. It is chosen based on prior knowledge of the $[\boldsymbol{x}_k; \boldsymbol{w}_k; \boldsymbol{v}_{k+1}]$ distribution. $\beta = 2$ is the optimal value for a Gaussian distribution [5].

Although not standard for a UKF algorithm, it is helpful for purposes of deriving the smoothing algorithm to normalize the measurement innovation by the square root of $\bar{P}_{yy(k+1)}$. Suppose that this square root is $\bar{S}_{y(k+1)}$ and that it obeys the relationship

$$\bar{S}^T_{y(k+1)}\bar{S}_{y(k+1)} = \bar{P}_{yy(k+1)} \tag{7}$$

This square root can be determined by performing a Cholesky factorization [16]. The normalized measurement innovation is then

$$\boldsymbol{\nu}_{y(k+1)} = \bar{S}^{-T}_{y(k+1)}[\boldsymbol{y}_{k+1} - \bar{\boldsymbol{y}}_{k+1}] \tag{8}$$

where the notation $(\,)^{-T}$ indicates the inverse of the transpose of the matrix in question.

The square root matrix $\bar{S}_{y(k+1)}$ and the normalized measurement innovation $\boldsymbol{\nu}_{y(k+1)}$ can be used to compute a filter gain and to complete the UKF measurement update. The normalized filter gain matrix is

$$\mathcal{K}_{k+1} = \bar{P}_{xy(k+1)}\bar{S}^{-1}_{y(k+1)} \tag{9}$$

and the state estimate and state error covariance updates are

$$\hat{\boldsymbol{x}}_{k+1} = \bar{\boldsymbol{x}}_{k+1} + \mathcal{K}_{k+1}\boldsymbol{\nu}_{y(k+1)} \quad \text{and} \quad \hat{P}_{k+1} = \bar{P}_{k+1} - \mathcal{K}_{k+1}\mathcal{K}^T_{k+1} \tag{10}$$

The first phase of the fixed-interval smoothing algorithm applies the dynamic propagation and measurement update equations of the UKF recursively starting at sample $k = 0$ and finishing at sample $k = N - 1$. Equations (2–10) are evaluated to perform the propagation and measurement update for each sample interval. Data from each sample are retained for later use in the smoother's backward pass. These data include $\hat{\boldsymbol{x}}_k$, $\hat{P}_k$, $\bar{\boldsymbol{x}}_{k+1}$, $\bar{P}_{k+1}$, and quantities that can be computed using the sigma points.

This brief review of the UKF does not constitute a thorough introduction to the subject. Rather, it is included to define needed notation for the sigma-points smoother and to develop the forward filtering pass that must be performed before the backward pass of the smoother. For a more thorough exposition of the UKF, the reader is referred to [4,5].

D. Cautions about Sigma-Points Filters

Although the sigma-points filtering technique has a number of useful properties, it is not guaranteed to perform well in all situations. Like the EKF, it can only provide local guarantees of convergence. Sigma-points techniques can run into serious difficulties on problems that have singularities or discontinuities. In such situations, a UKF may perform less well than an EKF [3]. An additional difficulty is that a UKF's estimates are not unique. It is the first author's experience that two UKF algorithms operating on the exact same problem sometimes produce significantly different results merely because they use different square root factorizations of the $\hat{P}_k$, Q_k, or R_{k+1} matrices in Eq. (2): recall that the factorizations of Eq. (2) are not unique.

III. Derivation of a Sigma-Points Fixed-Interval Smoothing Algorithm

The goal of this section is to derive the RTS-type sigma-points smoothing algorithm of [14] based on first principles. At a given sample index k, the RTS-like smoother uses information from the data $\boldsymbol{y}_{k+1}, \ldots, \boldsymbol{y}_N$ to develop a smoothed estimate of $\boldsymbol{x}_k$ as an update to the filtered estimate. Although not required, it also develops a smoothed estimate of $\boldsymbol{w}_k$. These estimates are generated by propagating the smoothed estimate of $\boldsymbol{x}_{k+1}$ backward using quantities that have been computed during the forward-pass UKF. This architecture allows the algorithm to compute its smoothed estimates recursively. The present section derives the equations necessary to perform the smoothing backward pass.

A. Bayesian Analysis

The sigma-points smoother of [14] can be derived by performing a Bayesian analysis. A fixed-interval smoother develops its estimates of the state vector $\boldsymbol{x}^*_k$ and the process noise vector $\boldsymbol{w}^*_k$ from the smoothed conditional probability distribution $p(\boldsymbol{x}_k, \boldsymbol{w}_k | \boldsymbol{y}_1, \ldots, \boldsymbol{y}_N)$. This probability distribution can be expressed in terms of the conditioned a posteriori distribution $p(\boldsymbol{x}_k, \boldsymbol{w}_k | \boldsymbol{x}_{k+1}, \boldsymbol{y}_1, \ldots, \boldsymbol{y}_k)$ and the smoothed distribution $p(\boldsymbol{x}_{k+1}|\boldsymbol{y}_1,\ldots,\boldsymbol{y}_N)$. The analysis proceeds as follows:

$$
\begin{aligned}
&p(\boldsymbol{x}_k,\boldsymbol{w}_k|\boldsymbol{y}_1,\ldots,\boldsymbol{y}_N)\\
&\quad=\int p(\boldsymbol{x}_k,\boldsymbol{w}_k|\boldsymbol{x}_{k+1},\boldsymbol{y}_1,\ldots,\boldsymbol{y}_N)p(\boldsymbol{x}_{k+1}|\boldsymbol{y}_1,\ldots,\boldsymbol{y}_N)\,\mathrm{d}\boldsymbol{x}_{k+1}\\
&\quad=\int p(\boldsymbol{x}_k,\boldsymbol{w}_k|\boldsymbol{x}_{k+1},\boldsymbol{y}_1,\ldots,\boldsymbol{y}_k)p(\boldsymbol{x}_{k+1}|\boldsymbol{y}_1,\ldots,\boldsymbol{y}_N)\,\mathrm{d}\boldsymbol{x}_{k+1}
\end{aligned}
\tag{11}
$$

B. Derivation of Sigma-Points Smoother for a Single Stage

The sigma-points smoother can be derived from the second line of Eq. (11) by making the usual sigma-points approximation that the distributions in question are Gaussian. Let the notation $\mathcal{N}(\boldsymbol{x};\boldsymbol{\mu},P)$ denote the Gaussian (i.e., normal) probability distribution for the random vector $\boldsymbol{x}$ that has a mean equal to $\boldsymbol{\mu}$ and a covariance matrix equal to P. Then the Gaussian approximations of the two probability distribution functions given on the second line of Eq. (11) take the forms

$$
\begin{aligned}
&p(\boldsymbol{x}_k,\boldsymbol{w}_k|\boldsymbol{x}_{k+1},\boldsymbol{y}_1,\ldots,\boldsymbol{y}_k)\\
&\quad=\mathcal{N}\left(\begin{bmatrix}\boldsymbol{x}_k\\ \boldsymbol{w}_k\end{bmatrix};\begin{bmatrix}\hat{\boldsymbol{x}}_k+J_{xk}(\boldsymbol{x}_{k+1}-\bar{\boldsymbol{x}}_{k+1})\\ J_{wk}(\boldsymbol{x}_{k+1}-\bar{\boldsymbol{x}}_{k+1})\end{bmatrix},\begin{bmatrix}\hat{P}_k & 0\\ 0 & Q_k\end{bmatrix}\right.\\
&\qquad\left.-\begin{bmatrix}J_{xk}\\ J_{wk}\end{bmatrix}\bar{P}_{k+1}\begin{bmatrix}J_{xk}^T & J_{wk}^T\end{bmatrix}\right)
\end{aligned}
\tag{12a}
$$

$$
p(\boldsymbol{x}_{k+1}|\boldsymbol{y}_1,\ldots\boldsymbol{y}_N)=\mathcal{N}\left(\boldsymbol{x}_{k+1};\boldsymbol{x}_{k+1}^*,P_{k+1}^*\right)
\tag{12b}
$$

where the matrices J_{xk} and J_{wk} are computed using sigma-points as follows:

$$
\begin{aligned}
J_{xk}&=E\{(\boldsymbol{x}_k-\hat{\boldsymbol{x}}_k)(\boldsymbol{x}_{k+1}-\bar{\boldsymbol{x}}_{k+1})^T\}\bar{P}_{k+1}^{-1}\\
&\cong\left[\sum_{i=0}^{2(n+m+p)}W_c^i\left(\mathcal{X}_k^i-\hat{\boldsymbol{x}}_k\right)\left(\tilde{\mathcal{X}}_{k+1}^i-\bar{\boldsymbol{x}}_{k+1}\right)^T\right]\bar{P}_{k+1}^{-1}
\end{aligned}
\tag{13a}
$$

$$
\begin{aligned}
J_{wk}&=E\{\boldsymbol{w}_k(\boldsymbol{x}_{k+1}-\bar{\boldsymbol{x}}_{k+1})^T\}\bar{P}_{k+1}^{-1}\\
&\cong\left[\sum_{i=0}^{2(n+m+p)}W_c^i\mathcal{W}_k^i\left(\tilde{\mathcal{X}}_{k+1}^i-\bar{\boldsymbol{x}}_{k+1}\right)^T\right]\bar{P}_{k+1}^{-1}
\end{aligned}
\tag{13b}
$$

The probability density in Eq. (12b) is obvious from the definitions of $\boldsymbol{x}_k^*$ and P_{k+1}^*. The derivation of Eq. (12a) is a straightforward application of results for Gaussian distribution conditioning applied to derive the Gaussian approximation of $p(\boldsymbol{x}_k,\boldsymbol{w}_k|\boldsymbol{x}_{k+1},\boldsymbol{y}_1,\ldots,\boldsymbol{y}_k)=p(\boldsymbol{x}_k,\boldsymbol{w}_k,\boldsymbol{x}_{k+1}|\boldsymbol{y}_1,\ldots,\boldsymbol{y}_k)/p(\boldsymbol{x}_{k+1}|\boldsymbol{y}_1,\ldots,\boldsymbol{y}_k)$. As an interim step in this derivation, the Gaussian approximation parameters of $p(\boldsymbol{x}_k,\boldsymbol{w}_k,\boldsymbol{x}_{k+1}|\boldsymbol{y}_1,\ldots,\boldsymbol{y}_k)$ are computed using the sigma-points results in Eqs. (13a) and (13b):

$$
\begin{aligned}
&p(\boldsymbol{x}_k,\boldsymbol{w}_k,\boldsymbol{x}_{k+1}|\boldsymbol{y}_1,\ldots,\boldsymbol{y}_k)\\
&\quad=\mathcal{N}\left(\begin{bmatrix}\boldsymbol{x}_k\\ \boldsymbol{w}_k\\ \boldsymbol{x}_{k+1}\end{bmatrix};\begin{bmatrix}\hat{\boldsymbol{x}}_k\\ 0\\ \bar{\boldsymbol{x}}_{k+1}\end{bmatrix},\begin{bmatrix}\hat{P}_k & 0 & J_{xk}\bar{P}_{k+1}\\ 0 & Q_k & J_{wk}\bar{P}_{k+1}\\ \bar{P}_{k+1}J_{xk}^T & \bar{P}_{k+1}J_{wk}^T & \bar{P}_{k+1}\end{bmatrix}\right)
\end{aligned}
\tag{14}
$$

The covariance matrix of the Gaussian distribution in Eq. (12a) is singular for a linear problem, but it is normally nonsingular for a nonlinear problem due to added variations that are caused by the nonlinearities. In the singular case, the distribution is understood to collapse to a Dirac delta function in each singular direction.

An algebraic derivation can be used to show that the product of the two normal distributions in Eqs. (12a) and (12b) can be rewritten as the product of the following two normal distributions:

$$
\begin{aligned}
&\mathcal{N}\left(\begin{bmatrix}\boldsymbol{x}_k\\ \boldsymbol{w}_k\end{bmatrix};\begin{bmatrix}\hat{\boldsymbol{x}}_k+J_{xk}(\boldsymbol{x}_{k+1}^*-\bar{\boldsymbol{x}}_{k+1})\\ J_{wk}(\boldsymbol{x}_{k+1}^*-\bar{\boldsymbol{x}}_{k+1})\end{bmatrix},\begin{bmatrix}\hat{P}_k & 0\\ 0 & Q_k\end{bmatrix}\right.\\
&\qquad\left.+\begin{bmatrix}J_{xk}\\ J_{wk}\end{bmatrix}(P_{k+1}^*-\bar{P}_{k+1})\begin{bmatrix}J_{xk}^T & J_{wk}^T\end{bmatrix}\right)
\end{aligned}
\tag{15a}
$$

$$
\mathcal{N}\,(\boldsymbol{x}_{k+1};\breve{\boldsymbol{x}}_{k+1}[\boldsymbol{x}_k,\boldsymbol{w}_k],\breve{P}_{k+1})
\tag{15b}
$$

where

$$
\begin{aligned}
\breve{\boldsymbol{x}}_{k+1}[\boldsymbol{x}_k,\boldsymbol{w}_k]&=\boldsymbol{x}_{k+1}^*+P_{k+1}^*[J_{xk}^T\quad J_{wk}^T]\left\{\begin{bmatrix}\hat{P}_k & 0\\ 0 & Q_k\end{bmatrix}\right.\\
&\quad\left.+\begin{bmatrix}J_{xk}\\ J_{wk}\end{bmatrix}(P_{k+1}^*-\bar{P}_{k+1})[J_{xk}^T\quad J_{wk}^T]\right\}^{-1}\\
&\quad\times\begin{bmatrix}\boldsymbol{x}_k-\hat{\boldsymbol{x}}_k-J_{xk}(\boldsymbol{x}_{k+1}^*-\bar{\boldsymbol{x}}_{k+1})\\ \boldsymbol{w}_k-J_{wk}(\boldsymbol{x}_{\mathrm{k}+1}^*-\bar{\boldsymbol{x}}_{k+1})\end{bmatrix}
\end{aligned}
\tag{16a}
$$

$$
\begin{aligned}
\breve{P}_{k+1}&=P_{k+1}^*-P_{k+1}^*[J_{xk}^T\quad J_{wk}^T]\left\{\begin{bmatrix}\hat{P}_k & 0\\ 0 & Q_k\end{bmatrix}+\begin{bmatrix}J_{xk}\\ J_{wk}\end{bmatrix}\left(P_{k+1}^*\right.\right.\\
&\quad\left.\left.-\bar{P}_{k+1}\right)[J_{xk}^T\quad J_{wk}^T]\right\}^{-1}\begin{bmatrix}J_{xk}\\ J_{wk}\end{bmatrix}P_{k+1}^*
\end{aligned}
\tag{16b}
$$

The derivation starts with the exponential formulas for the normal distributions in Eqs. (12a), (12b), (15a), and (15b), and it uses these formulas to show that the two distribution products are equal. This straightforward, though lengthy, derivation has been omitted for the sake of brevity.

The integral on the second line of Eq. (11) can be carried out if one replaces its integrand with the approximation that equals the product of the two normal distributions in Eqs. (15a) and (15b). The integral is carried out over $\boldsymbol{x}_{k+1}$, and $\boldsymbol{x}_{k+1}$ appears in this product only as the argument of the normal distribution in Eq. (15b). Therefore, the unit normalization property of this distribution implies that the integral evaluates to

$$
\begin{aligned}
&p(\boldsymbol{x}_k,\boldsymbol{w}_k|\boldsymbol{y}_1,\ldots,\boldsymbol{y}_N)\\
&\quad=\mathcal{N}\left(\begin{bmatrix}\boldsymbol{x}_k\\ \boldsymbol{w}_k\end{bmatrix};\begin{bmatrix}\hat{\boldsymbol{x}}_k+J_{xk}\left(\boldsymbol{x}_{k+1}^*-\bar{\boldsymbol{x}}_{k+1}\right)\\ J_{wk}\left(\boldsymbol{x}_{k+1}^*-\bar{\boldsymbol{x}}_{k+1}\right)\end{bmatrix},\begin{bmatrix}\hat{P}_k & 0\\ 0 & Q_k\end{bmatrix}\right.\\
&\qquad\left.+\begin{bmatrix}J_{xk}\\ J_{wk}\end{bmatrix}\left(P_{k+1}^*-\bar{P}_{k+1}\right)\begin{bmatrix}J_{xk}^T & J_{wk}^T\end{bmatrix}\right)
\end{aligned}
\tag{17}
$$

i.e., the integral equals the normal distribution in Eq. (15a).

Equation (17) implies that the smoothed state and process noise at stage k are

$$
\boldsymbol{x}_k^*=\hat{\boldsymbol{x}}_k+J_{xk}\left(\boldsymbol{x}_{k+1}^*-\bar{\boldsymbol{x}}_{k+1}\right)\quad\text{and}\quad\boldsymbol{w}_k^*=J_{wk}\left(\boldsymbol{x}_{k+1}^*-\bar{\boldsymbol{x}}_{k+1}\right)
\tag{18}
$$

and that their estimation error covariances are

$$
\begin{aligned}
P_k^*&=\hat{P}_k+J_{xk}\left(P_{k+1}^*-\bar{P}_{k+1}\right)J_{xk}^T\quad\text{and}\\
Q_k^*&=Q_k+J_{wk}\left(P_{k+1}^*-\bar{P}_{k+1}\right)J_{wk}^T
\end{aligned}
\tag{19}
$$

where $Q_k^*=E\{(\boldsymbol{w}_k-\boldsymbol{w}_k^*)(\boldsymbol{w}_k-\boldsymbol{w}_k^*)^T\}$.

The smoothing calculations in Eqs. (18) and (19) are equivalent to the RTS smoother of [15] if one recognizes that J_{xk} serves as the smoother gain matrix. The difference between the present algorithm and the RTS derivation is that J_{xk} is computed using a sigma-points computation instead of a linear statistical analysis. It is straightforward to demonstrate that the sigma-points formula for J_{xk} in Eq. (13a) reduces to the gain formula given in [15] in the case of a linear system.

C. Entire Sigma-Points Fixed-Interval Smoothing Algorithm

Using the preceding equations, the sigma-points solution to the fixed-interval smoothing problem is implemented by the following steps:

1) Perform forward filtering pass using sigma-points filter (UKF). During this process compute and store $\hat{\boldsymbol{x}}_0, \ldots, \hat{\boldsymbol{x}}_N$, $\hat{P}_0, \ldots, \hat{P}_N$, $\bar{\boldsymbol{x}}_1, \ldots, \bar{\boldsymbol{x}}_N$, $\bar{P}_1, \ldots, \bar{P}_N$, $J_{x0}, \ldots, J_{x(N-1)}$, and $J_{w0}, \ldots, J_{w(N-1)}$, where $J_{x0}, \ldots, J_{x(N-1)}$ and $J_{w0}, \ldots, J_{w(N-1)}$ are computed using, respectively, Eqs. (13a) and (13b).

2) Set $\boldsymbol{x}_N^* = \hat{\boldsymbol{x}}_N$, $P_N^* = \hat{P}_N$, and $k = N$.

3) Decrement k by 1 and stop if $k < 0$; otherwise, go to Step 4.

4) Compute $\boldsymbol{x}_k^*$ and $\boldsymbol{w}_k^*$ using Eq. (18) and compute P_k^* and Q_k^* using Eq. (19).

5) Go to Step 3.

Sigma-points smoothing may be done by computing only the $\boldsymbol{x}_k$ estimates. This formulation is a straightforward simplification of the general equations and may be useful for some systems. It gives the parameters of the Gaussian approximation of the marginal density $p(\boldsymbol{x}_k|\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N)$.

Note that this algorithm does not compute new sigma points during its backward pass. It may be possible to improve the algorithm by adding such a feature, but a good method for computing new points has yet to be developed. An additional level of sophistication might be to embed sigma-points recomputation within an outer iteration that refilters and resmoothes after the selection of new points. All such developments are left as possible topics for future research.

IV. Alternative Sigma-Points Smoothing Algorithm Derivation

An equivalent alternate form of the sigma-points smoothing algorithm can be derived based on the idea of a pseudomeasurement. Although somewhat more complicated, this alternate derivation yields insights into the nature of sigma-points smoothing.

A. Bayesian Derivation of a Pseudomeasurement Probability Density Function

The Bayesian analysis of Eq. (11) can be extended to derive a smoothing recursion that can be cast in the form of a measurement update. The analysis proceeds as follows:

$$
\begin{aligned}
&p(\boldsymbol{x}_k, \boldsymbol{w}_k|\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N) \\
&\quad = \int p(\boldsymbol{x}_k, \boldsymbol{w}_k|\boldsymbol{x}_{k+1}, \boldsymbol{y}_1, \ldots, \boldsymbol{y}_k) p(\boldsymbol{x}_{k+1}|\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N)\,\mathrm{d}\boldsymbol{x}_{k+1} \\
&\quad = \int \left[\frac{p(\boldsymbol{x}_{k+1}|\boldsymbol{x}_k, \boldsymbol{w}_k, \boldsymbol{y}_1, \ldots, \boldsymbol{y}_k) p(\boldsymbol{x}_k, \boldsymbol{w}_k|\boldsymbol{y}_1, \ldots, \boldsymbol{y}_k)}{p(\boldsymbol{x}_{k+1}|\boldsymbol{y}_1, \ldots, \boldsymbol{y}_k)}\right] \\
&\qquad \times p(\boldsymbol{x}_{k+1}|\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N)\,\mathrm{d}\boldsymbol{x}_{k+1} \\
&\quad = \int \left[\frac{p(\boldsymbol{x}_{k+1}|\boldsymbol{x}_k, \boldsymbol{w}_k) p(\boldsymbol{x}_k, \boldsymbol{w}_k|\boldsymbol{y}_1, \ldots, \boldsymbol{y}_k)}{p(\boldsymbol{x}_{k+1}|\boldsymbol{y}_1, \ldots, \boldsymbol{y}_k)}\right] \\
&\qquad \times p(\boldsymbol{x}_{k+1}|\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N)\,\mathrm{d}\boldsymbol{x}_{k+1} \\
&\quad = \int \left[\frac{\delta\{\boldsymbol{x}_{k+1} - \boldsymbol{f}_k(\boldsymbol{x}_k, \boldsymbol{w}_k)\} p(\boldsymbol{x}_k, \boldsymbol{w}_k|\boldsymbol{y}_1, \ldots, \boldsymbol{y}_k)}{p(\boldsymbol{x}_{k+1}|\boldsymbol{y}_1, \ldots, \boldsymbol{y}_k)}\right] \\
&\qquad \times p(\boldsymbol{x}_{k+1}|\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N)\,\mathrm{d}\boldsymbol{x}_{k+1} \\
&\quad = \frac{p(\boldsymbol{x}_{k+1} = \boldsymbol{f}_k(\boldsymbol{x}_k, \boldsymbol{w}_k)|\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N) p(\boldsymbol{x}_k, \boldsymbol{w}_k|\boldsymbol{y}_1, \ldots, \boldsymbol{y}_k)}{p(\boldsymbol{x}_{k+1} = \boldsymbol{f}_k(\boldsymbol{x}_k, \boldsymbol{w}_k)|\boldsymbol{y}_1, \ldots, \boldsymbol{y}_k)}
\end{aligned}
\tag{20}
$$

where the function $\delta\{\}$ in the second-to-last equation is the multidimensional Dirac delta function.

The result in Eq. (20) can be used to develop a smoothing update as though it were a standard measurement update. Consider the following probability distribution:

$$
p(\boldsymbol{\psi}_k|\boldsymbol{x}_k, \boldsymbol{w}_k) = c_1 \frac{p(\boldsymbol{x}_{k+1} = \boldsymbol{f}_k(\boldsymbol{x}_k, \boldsymbol{w}_k)|\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N)}{p(\boldsymbol{x}_{k+1} = \boldsymbol{f}_k(\boldsymbol{x}_k, \boldsymbol{w}_k)|\boldsymbol{y}_1, \ldots, \boldsymbol{y}_k)} \tag{21}
$$

for some as-yet-to-be-determined pseudomeasurement $\boldsymbol{\psi}_k$ that, in effect, contains all of the information about $\boldsymbol{x}_k$ and $\boldsymbol{w}_k$ that is available from the data $\boldsymbol{y}_{k+1}, \ldots, \boldsymbol{y}_N$. The constant c_1 in Eq. (21) is a normalizing constant.[†]

If one substitutes the conditional probability density function of Eq. (21) into the last line of Eq. (20), then the resulting formula for $p(\boldsymbol{x}_k, \boldsymbol{w}_k|\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N)$ is equivalent to the standard Bayesian formula that would be used to develop a Kalman filter update of $p(\boldsymbol{x}_k, \boldsymbol{w}_k|\boldsymbol{y}_1, \ldots, \boldsymbol{y}_k)$ based on the pseudomeasurement $\boldsymbol{\psi}_k$. This equivalence allows one to develop the smoother update of $[\boldsymbol{x}_k; \boldsymbol{w}_k]$ as though it were a standard Kalman filter update. It is straightforward to develop this update using sigma-points methods if one can derive an appropriate pseudomeasurement along with its corresponding nonlinear model.

B. Derivation of Pseudomeasurement Equation

This subsection derives a pseudomeasurement equation for $\boldsymbol{\psi}_k$ that yields the conditional measurement probability distribution given in Eq. (21). In keeping with the sigma-points approach, this equation will be derived using the approximation that all probability distributions in question are Gaussian. Therefore, the distribution $p(\boldsymbol{x}_{k+1}|\boldsymbol{y}_1, \ldots, \boldsymbol{y}_k)$ is completely defined by its mean $\bar{\boldsymbol{x}}_{k+1}$ and its covariance $\bar{P}_{k+1}$. Similarly, the distribution $p(\boldsymbol{x}_{k+1}|\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N)$ is completely defined by $\boldsymbol{x}_{k+1}^*$ and P_{k+1}^*. The conditional pseudomeasurement probability distribution takes the form

$$
\begin{aligned}
p(\boldsymbol{\psi}_k|\boldsymbol{x}_k, \boldsymbol{w}_k) = c_2 \exp\biggl\{&-\frac{1}{2}\Bigl[\boldsymbol{f}_k(\boldsymbol{x}_k, \boldsymbol{w}_k) \\
&- \boldsymbol{x}_{k+1}^*\Bigr]^T \Bigl[P_{k+1}^*\Bigr]^{-1} \Bigl[\boldsymbol{f}_k(\boldsymbol{x}_k, \boldsymbol{w}_k) - \boldsymbol{x}_{k+1}^*\Bigr] + \frac{1}{2}\Bigl[\boldsymbol{f}_k(\boldsymbol{x}_k, \boldsymbol{w}_k) \\
&- \bar{\boldsymbol{x}}_{k+1}\Bigr]^T \bar{P}_{k+1}^{-1} \Bigl[\boldsymbol{f}_k(\boldsymbol{x}_k, \boldsymbol{w}_k) - \bar{\boldsymbol{x}}_{k+1}\Bigr]\biggr\}
\end{aligned}
\tag{22}
$$

where c_2 is another normalizing constant.

As preparation for deriving the pseudomeasurement equation, it is helpful to define a smoother square root covariance update matrix $S_{s(k+1)}$ and a normalized smoother innovation vector $\boldsymbol{\nu}_{s(k+1)}$. These quantities are defined through the following relationships between the smoothed and a priori state estimates and error covariances at sample $k+1$:

$$
P_{k+1}^* = \bar{P}_{k+1} - S_{s(k+1)}^T S_{s(k+1)} \quad \text{and} \quad \boldsymbol{x}_{k+1}^* = \bar{\boldsymbol{x}}_{k+1} + S_{s(k+1)}^T \boldsymbol{\nu}_{s(k+1)} \tag{23}
$$

At the terminal stage, $S_{s(N)} = \mathcal{K}_N^T$ and $\boldsymbol{\nu}_{s(N)} = \boldsymbol{\nu}_{y(N)}$ from the UKF's last measurement update in Eq. (10). At preceding stages, $S_{s(k+1)}$ and $\boldsymbol{\nu}_{s(k+1)}$ are built up from $\mathcal{K}_{k+1}^T$, $\boldsymbol{\nu}_{y(k+1)}$, and related quantities that are derived during the smoothing update for that stage, as will be demonstrated in Eqs. (30) and (31). Note that it is not necessary to attach any "physical" meaning to the matrix $S_{s(k+1)}$ or to the vector $\boldsymbol{\nu}_{s(k+1)}$. They are only intermediate mathematical quantities that are useful for reducing the complexity of the algorithm to manageable levels. The dimension of $\boldsymbol{\nu}_{s(k+1)}$ equals the number of rows in $S_{s(k+1)}$, and this dimension is never greater than n.

Given these quantities, the pseudomeasurement equation takes the form

[†] A ratio of probability density functions may not have a finite integral, but the ratio in Eq. (21) will have a finite integral over the pseudomeasurement $\boldsymbol{\psi}_k$ if $\boldsymbol{\psi}_k$ is properly defined, which implies that the normalizing constant c_1 exists.

$$\boldsymbol{\psi}_k = \boldsymbol{h}_{\psi k}(\boldsymbol{x}_k, \boldsymbol{w}_k) + \boldsymbol{v}_{\psi k} \tag{24}$$

where the "measured" value of the pseudomeasurement is

$$\boldsymbol{\psi}_k = \boldsymbol{\nu}_{s(k+1)} + S_{s(k+1)}\left[P^*_{k+1}\right]^{-1}\boldsymbol{x}^*_{k+1} \tag{25}$$

the nonlinear function that models the pseudomeasurement is

$$\boldsymbol{h}_{\psi k}(\boldsymbol{x}_k, \boldsymbol{w}_k) = S_{s(k+1)}\left[P^*_{k+1}\right]^{-1}\boldsymbol{f}_k(\boldsymbol{x}_k, \boldsymbol{w}_k) \tag{26}$$

and $\boldsymbol{v}_{\psi k}$ is a zero-mean Gaussian white-noise vector with covariance

$$R_{\psi k} = E\left\{\boldsymbol{v}_{\psi k}\boldsymbol{v}^T_{\psi k}\right\} = I + S_{s(k+1)}\left[P^*_{k+1}\right]^{-1}S^T_{s(k+1)} \tag{27}$$

The consistency of this pseudomeasurement formulation with Eq. (22) can be proved by forming the conditional pseudomeasurement probability distribution

$$p(\boldsymbol{\psi}_k|\boldsymbol{x}_k, \boldsymbol{w}_k) = c_3 \exp\left\{-\frac{1}{2}[\boldsymbol{\psi}_k - \boldsymbol{h}_{\psi k}(\boldsymbol{x}_k, \boldsymbol{w}_k)]^T R^{-1}_{\psi k}[\boldsymbol{\psi}_k - \boldsymbol{h}_{\psi k}(\boldsymbol{x}_k, \boldsymbol{w}_k)]\right\} \tag{28}$$

One can demonstrate that this expression is equivalent to the expression in Eq. (22) for a suitably chosen normalizing constant c_3. The key to proving this equivalence is to use the definitions of $\boldsymbol{\psi}_k$, $\boldsymbol{h}_{\psi k}(\boldsymbol{x}_k, \boldsymbol{w}_k)$, and $R_{\psi k}$ in Eqs. (25–27). The actual proof involves a lengthy set of algebraic manipulations that have been omitted for the sake of brevity.

The pseudomeasurement model in Eqs. (24–27) amounts to a transformation of the relationship $\boldsymbol{x}^*_{k+1} = \boldsymbol{f}_k(\boldsymbol{x}_k, \boldsymbol{w}_k) +$ noise. One effect of this transformation is to delete from $\boldsymbol{x}^*_{k+1}$ and P^*_{k+1} all information that is based on the data $\{\boldsymbol{y}_1, \ldots \boldsymbol{y}_k\}$ because that information has already been incorporated into $\hat{\boldsymbol{x}}_k$ and $\hat{P}_k$. The transformation also results in projection onto a subspace if $S_{s(k+1)}$ has fewer than n rows. Projection occurs near the end of the smoothing interval if there are less than n independent scalar measurements in the data set $\{\boldsymbol{y}_{k+1}, \ldots \boldsymbol{y}_N\}$. As an example of projection, suppose that $k = N - 1$, that there are $p = 2$ measurements, and that there are $n = 5$ states. Then there are only two new pieces of scalar information in the five equations $\boldsymbol{x}^*_N = \boldsymbol{f}_{N-1}(\boldsymbol{x}_{N-1}, \boldsymbol{w}_{N-1}) +$ noise. Therefore, $S_{s(N)}$ will be a 2 × 5 matrix, and the pseudomeasurement $\boldsymbol{\psi}_{N-1}$ will have only two elements.

Note that the pseudomeasurement model in Eqs. (25–27) is not unique. Any consistent rescaling of the measurements, the nonlinear model function, and the error covariance will yield equivalence between Eqs. (22) and (28). Any such rescaling, however, will not alter the smoothing updates that are derived in the next subsection.

C. Sigma-Points Pseudomeasurement Update to Accomplish Smoothing

Equations (20) and (21) imply that the smoothed estimates of $\boldsymbol{x}_k$ and $\boldsymbol{w}_k$ can be computed by using the pseudomeasurement in Eq. (24) to update the a posteriori estimates $\hat{\boldsymbol{x}}_k$ and $\hat{\boldsymbol{w}}_k$ (=0). The pseudomeasurement equation is nonlinear. Therefore, a sigma-points approach is used to perform the update. This approach reuses the sigma points that have been generated in Eqs. (3a–3g) during the forward filtering pass, and it performs operations that are similar to those given in Eqs. (4), (5a), (5c), (5d), and (7–10) for the filter's measurement update. The pseudoupdate computations are

$$\boldsymbol{\psi}^i_k = \boldsymbol{h}_{\psi k}\left(\mathcal{X}^i_k, \mathcal{W}^i_k\right) \quad \text{for } i = 0, \ldots, 2(n+m+p) \tag{29a}$$

$$\hat{\boldsymbol{\psi}}_k = \sum_{i=0}^{2(n+m+p)} W^i_a \boldsymbol{\psi}^i_k \tag{29b}$$

$$P_{x\psi k} = \sum_{i=0}^{2(n+m+p)} W^i_c\left(\mathcal{X}^i_k - \hat{\boldsymbol{x}}_k\right)\left(\boldsymbol{\psi}^i_k - \hat{\boldsymbol{\psi}}_k\right)^T \tag{29c}$$

$$P_{w\psi k} = \sum_{i=0}^{2(n+m+p)} W^i_c\mathcal{W}^i_k\left(\boldsymbol{\psi}^i_k - \hat{\boldsymbol{\psi}}_k\right)^T \tag{29d}$$

$$P_{\psi\psi k} = \sum_{i=0}^{2(n+m+p)} W^i_c\left(\boldsymbol{\psi}^i_k - \hat{\boldsymbol{\psi}}_k\right)\left(\boldsymbol{\psi}^i_k - \hat{\boldsymbol{\psi}}_k\right)^T + R_{\psi k} \tag{29e}$$

$$S^T_{\psi k}S_{\psi k} = P_{\psi\psi k} \tag{29f}$$

$$\boldsymbol{\nu}_{\psi(k)} = S^{-T}_{\psi k}\left[\boldsymbol{\psi}_k - \hat{\boldsymbol{\psi}}_k\right] \tag{29g}$$

$$\mathcal{K}_{x\psi(k)} = P_{x\psi k}S^{-1}_{\psi k} \quad \text{and} \quad \mathcal{K}_{w\psi(k)} = P_{w\psi k}S^{-1}_{\psi k} \tag{29h}$$

$$\boldsymbol{x}^*_k = \hat{\boldsymbol{x}}_k + \mathcal{K}_{x\psi(k)}\boldsymbol{\nu}_{\psi(k)} \quad \text{and} \quad \boldsymbol{w}^*_k = \mathcal{K}_{w\psi(k)}\boldsymbol{\nu}_{\psi(k)} \tag{29i}$$

$$\begin{gathered} P^*_k = \hat{P}_k - \mathcal{K}_{x\psi(k)}\mathcal{K}^T_{x\psi(k)} \\ Q^*_k = Q_k - \mathcal{K}_{w\psi(k)}\mathcal{K}^T_{w\psi(k)}, \quad \text{and} \quad P^*_{xwk} = -\mathcal{K}_{x\psi(k)}\mathcal{K}^T_{w\psi(k)} \end{gathered} \tag{29j}$$

The final two equations in this sequence yield the sigma-points approximations to the smoothed state and process noise at sample k along with their corresponding error covariance matrices.

D. Recursive Backward Smoothing Pass

The backward smoothing pass recursively executes the operations in Eqs. (25), (27), and (29a–29j) backward in time starting at sample $k = N - 1$ and terminating at sample $k = 0$. The following quantities must be passed from the recursion at sample $k + 1$ to the recursion at sample k: $\boldsymbol{x}^*_{k+1}$, P^*_{k+1}, $\boldsymbol{\nu}_{s(k+1)}$, and $S_{s(k+1)}$. For sample $k = N - 1$, these quantities are available from the terminal state of the forward filtering pass: $\boldsymbol{x}^*_N = \hat{\boldsymbol{x}}_N$, $P^*_N = P_N$, $\boldsymbol{\nu}_{s(N)} = \boldsymbol{\nu}_{y(N)}$, and $S_{s(N)} = \mathcal{K}^T_N$. For other stages, $\boldsymbol{x}^*_{k+1}$ and P^*_{k+1} will be available from the computations in Eqs. (29i) and (29j) at sample $k + 1$, but the method for determining $S_{s(k+1)}$ and $\boldsymbol{\nu}_{s(k+1)}$ remains to be defined.

The matrix $S_{s(k+1)}$ and the vector $\boldsymbol{\nu}_{s(k+1)}$ are formed by combining the filter's measurement update in Eq. (10) and the smoother's pseudomeasurement update in Eqs. (29i) and (29j). A comparison of these equations with Eq. (23) demonstrates that the following formulas yield $S_{s(k+1)}$ and $\boldsymbol{\nu}_{s(k+1)}$ values which satisfy the relationships in Eq. (23):

$$S_{s(k+1)} = \begin{bmatrix} \mathcal{K}^T_{k+1} \\ \mathcal{K}^T_{x\psi(k+1)} \end{bmatrix} \quad \text{and} \quad \boldsymbol{\nu}_{s(k+1)} = \begin{bmatrix} \boldsymbol{\nu}_{y(k+1)} \\ \boldsymbol{\nu}_{\psi(k+1)} \end{bmatrix} \tag{30}$$

There is one drawback to the use of Eq. (30) to generate $S_{s(k+1)}$ and $\boldsymbol{\nu}_{s(k+1)}$: The number of rows in $S_{s(k+1)}$, which equals the number of elements in $\boldsymbol{\nu}_{s(k+1)}$, will increase with each successive iteration of Eq. (30) for each new value of the stage index k during the smoother's backward recursion. A bound on the number of rows in $S_{s(k+1)}$ can be enforced by modifying Eq. (30) through the use of orthogonal/upper-triangular (QR) factorization [16] if that number would otherwise exceed the state dimension n. Recall that the matrix $S_{s(k+1)}$ has n columns. If Eq. (30) would give it more than n rows, then Eq. (30) gets replaced by the following calculations:

$$Q_{s(k+1)}\begin{bmatrix} S_{s(k+1)} \\ 0 \end{bmatrix} = \begin{bmatrix} \mathcal{K}_{k+1}^T \\ \mathcal{K}_{x\psi(k+1)}^T \end{bmatrix} \quad \text{and}$$
$$\boldsymbol{\nu}_{s(k+1)} = \begin{bmatrix} I & 0 \end{bmatrix} Q_{s(k+1)}^T \begin{bmatrix} \boldsymbol{\nu}_{y(k+1)} \\ \boldsymbol{\nu}_{\psi(k+1)} \end{bmatrix} \tag{31}$$

where $Q_{s(k+1)}$ is a square, orthonormal matrix (i.e., $Q_{s(k+1)}Q_{s(k+1)}^T = Q_{s(k+1)}^T Q_{s(k+1)} = I$), $S_{s(k+1)}$ is an $n \times n$ upper-triangular matrix, and $\boldsymbol{\nu}_{s(k+1)}$ is an n-dimensional column vector. The matrices $Q_{s(k+1)}$ and $S_{s(k+1)}$ can be computed from the block matrix on the right-hand side of the first equation via standard QR-factorization techniques [16]. It is straightforward to show that the transformation in Eq. (31) produces a matrix $S_{s(k+1)}$ and a vector $\boldsymbol{\nu}_{s(k+1)}$ that satisfy the relationships in Eq. (23).

E. Equivalence to Algorithm of Section III

The sigma-points smoothing algorithm of Section III is exactly equivalent to the algorithm derived in the present section. One way to show this is to perform algebraic manipulations of the two algorithms' smoothing equations to demonstrate that they yield identical updates. Such an analysis has been carried out, but it is very lengthy, and it has been omitted for the sake of brevity.

An alternative demonstration of equivalence starts from the first line of Eq. (20). One substitutes into the right-hand side of this equation the approximate normal distributions given in Eqs. (12a) and (12b). Next, one proceeds with the remainder of the analysis of Eq. (20) using related normal distributions. This modified form of Eq. (20) gives rise to a ratio of normal probability distributions in Eq. (21) that is exactly equal to the distribution given in Eq. (22). This equality guarantees that the subsequent algorithm steps will exactly reproduce the results of Section III.

The backward smoothing update equations of Section III are much less complicated than those of the present section. Thus, most practitioners will prefer to use the former sigma-points smoother.

The present algorithm's chief benefit is its notion of a pseudomeasurement update. This notion applies regardless of whether one makes the Gaussian assumption needed to derive a sigma-points smoother. Therefore, the present algorithm may serve as a starting point for the development of other types of nonlinear smoothers. In addition, the present algorithm's form may make it more conducive to the development of a square root implementation of a sigma-points smoother.

V. Smoother Evaluation Using a Truth-Model Simulation

A. Test Cases that Use a Truth-Model Simulation of an Example Problem

A truth-model simulation has been used to generate data for purposes of testing this paper's sigma-points smoothing algorithm. The simulation models a challenging nonlinear estimation problem that is borrowed from [3]. The goal of the estimator is to determine the attitude, attitude-rate, and moment-of-inertia matrix of a microsatellite that starts out tumbling at a rate of 2 deg/s while flying in a circular orbit at an altitude of 823 km and an inclination of 82 deg. The only measurements come from a three-axis magnetometer, which makes the problem undersensed. The system is observable because the magnetic field vector undergoes large orientation changes with respect to inertial space as the satellite moves along its orbit. Small attitude estimation errors can be achieved only if the filter/smoother can accurately estimate the moment-of-inertia matrix because this matrix is needed to use the Euler attitude dynamics equation to propagate attitude-rate and attitude estimates along significant portions of the orbit. The principal moments of inertia are nearly equal. This fact makes the estimator's tumbling model highly sensitive to small errors in the moment-of-inertia matrix because the directions of its principal axes, which define the nutation axes, are highly sensitive to such errors.

The truth-model attitude dynamics simulation employs realistic models of errors that are not truly random processes. These include torques due to atmospheric drag, due to radiation pressure from the sun and the Earth's albedo, and due to the gravity-gradient effect. Measurement errors include thermal noise and residual errors in the magnetometer scale factors, relative alignments, and biases that might remain after an attitude-independent on-orbit calibration.

This attitude estimation problem represents a good test case for the smoothing algorithm. It uses a realistic process noise model that is based on physics; it is not simply a realization of the stochastic model used by the smoother. The true accuracy of the sigma-points smoother can be evaluated because the "truth" states are available from the simulation. The use of real flight data for such an evaluation, although more impressive at first glance, can lead to difficulties because the true states may not be known.

A mathematical description of the nonlinear attitude estimation problem model is given in [3]. That description includes detailed definitions of the state and process noise vectors, the dynamics model, and the measurement model. Only a brief description of that model is given here. The dynamics model includes an attitude kinematics equation and Euler's equation for the dynamics of the attitude-rate vector. The state has $n = 12$ elements that consist of the three elements of the body-referenced attitude-rate vector, a three-parameter attitude representation, and a six-parameter representation of the moment-of-inertia matrix. The process noise vector has $m = 3$ elements that represent disturbance torques. The three-element measurement from the vector magnetometer is rescaled to become a unit direction vector because its length does not contain attitude information. Therefore, the estimator only sees $p = 2$ measurements.

The UKS has been tested on the same three cases of the attitude estimation problem that are considered in [3]. Test case 1 begins with small initial estimation errors in the attitude (0.42 deg) and the attitude-rate (0.18 deg/s) and with inertia matrix errors that are no larger than 4.23% of the maximum principal inertia. Test case 2 starts with a 180 deg attitude error in rotation about the initial magnetic field, which is the worst-case direction because it is the direction about which attitude is initially unsensed. Its initial rate error is 1.54 deg/s, and its initial moment-of-inertia error is the same as in case 1. Test case 3 has an initial attitude error of 32.75 deg, an initial attitude-rate error of 18 deg/s, and the same initial error in its moment-of-inertia estimate as for cases 1 and 2. The 18 deg/s initial attitude-rate error is directed about the magnetic field vector, making it initially unsensed. Given the 0.1 Hz magnetometer sampling rate, this magnitude of the rate error equals the aliasing frequency of one half cycle per sample, which makes case 3 particularly difficult.

The UKS has used the sigma-points scaling parameters $\alpha = 1$, $\beta = 2$, and $\kappa = -12$ for all three attitude determination cases. These values correspond to UKF scaling A of [3], which was one of two reasonable parameter sets that were found for this estimation problem. The UKF also uses an ad hoc fix that is described in [3]. This fix counteracts the troublesome effects of a discontinuity in the dynamics propagation model that is caused by the use of a two-argument arctangent function to determine an attitude angle that has a 2π ambiguity. The fix unwinds this ambiguity in any given set of sigma points so that the attitude angles in question all lie within $\pm\pi$ radians of each other.

B. Comparison Smoothing Algorithms

The sigma-points smoother has been compared to two competing smoothers. For the remainder of this paper, the sigma-points smoother is called the unscented Kalman smoother (UKS).

The first competing smoother is an extended smoother that performs an RTS backward smoothing pass [15] using data from a forward filtering pass that has been performed by an EKF. This smoother works with the same linearized dynamics and measurement models that the EKF uses. These models take the form

$$\boldsymbol{x}_{k+1} = \bar{\boldsymbol{x}}_{k+1} + \left(\left.\frac{\partial \boldsymbol{f}_k}{\partial \boldsymbol{x}_k}\right|_{(\hat{\boldsymbol{x}}_k,0)}\right)(\boldsymbol{x}_k - \hat{\boldsymbol{x}}_k) + \left(\left.\frac{\partial \boldsymbol{f}_k}{\partial \boldsymbol{w}_k}\right|_{(\hat{\boldsymbol{x}}_k,0)}\right)\boldsymbol{w}_k \tag{32a}$$

$$\boldsymbol{y}_{k+1} = \boldsymbol{h}_{k+1}(\bar{\boldsymbol{x}}_{k+1}) + \left(\frac{\partial \boldsymbol{h}_{k+1}}{\partial \boldsymbol{x}_{k+1}}\bigg|_{\bar{\boldsymbol{x}}_{k+1}}\right)(\boldsymbol{x}_{k+1} - \bar{\boldsymbol{x}}_{k+1}) + \boldsymbol{v}_{k+1} \quad (32b)$$

with $\bar{\boldsymbol{x}}_{k+1} = \boldsymbol{f}_k(\hat{\boldsymbol{x}}_k, 0)$. This smoother is called the extended Kalman smoother (EKS) in the remainder of this paper.

The second competing smoother algorithm is the nonlinear batch smoother that is defined in Secs. III.B and III.C of [3]. This algorithm is an implementation of the iterative Gauss–Newton method [16]. It solves for the values of $\boldsymbol{x}_0^*$ and $\boldsymbol{w}_0^*$ through $\boldsymbol{w}_{N-1}^*$ that minimize a negative-log-probability cost function which equals the sum of the squares of the weighted magnetometer measurement errors, the weighted norms of $\boldsymbol{w}_0^*$ through $\boldsymbol{w}_{N-1}^*$, and the weighted norm of the difference between $\boldsymbol{x}_0^*$ and $\hat{\boldsymbol{x}}_0$. The residual magnetometer errors at samples 1 through N are determined by using the dynamics model in Eq. (1a) and the measurement model in Eq. (1b) to determine the smoothed quantities $\boldsymbol{x}_1^*$ through $\boldsymbol{x}_N^*$ and $\boldsymbol{y}_1^*$ through $\boldsymbol{y}_N^*$. The output of this smoother constitutes the exact solution to the nonlinear maximum a posteriori smoothing problem. This smoother is called the Gauss–Newton smoother (GNS) for the remainder of this paper.

C. Truth-Model Simulation Test Results

The UKS and the two competing smoothers have been tested on the three cases of the attitude estimation problem. The results for these cases are illustrated by the attitude estimation error time histories that are plotted in Figs. 1–3. Each of these figures plots five time histories of the total attitude estimation error. This error is the Euler rotation angle needed to transform from the true attitude to the estimated attitude. Each figure includes two filtered estimation error time histories and three smoothed error time histories.

The two filtered time histories are those of the EKF (dark gray dotted curves) and the UKF (black dashed curves). They are included to show the accuracy of the forward filtering passes that have generated data for use by the EKS and UKS backward smoothing passes. These error time histories also provide a contrast for the smoothed results because filtered results usually display an initial error decay transient that should be largely absent from smoothed results.

The three smoothed time histories are those of the EKS (light gray dash-dotted curves), the UKS (dark gray solid curves) and the GNS (black dotted curves). The EKS is included to determine whether the UKS offers advantages over this existing “standard” method. The GNS is plotted as a sort of “gold standard.” Its accuracy is likely to be the best that is achievable, or nearly so.

The vertical scale on Fig. 1 differs from that of Figs. 2 and 3. Figure 1’s vertical scale is linear, but Figs. 2 and 3 have logarithmic vertical scales. A linear scale is reasonable for Fig. 1 because all of its errors are relatively small, less than 20 deg even during the filters’ initial transients. Logarithmic scales are needed for the other two plots because they allow one to see the large 180 deg errors caused by the large initial uncertainties while simultaneously allowing one to distinguish the subdegree accuracy of the very best smoothers.

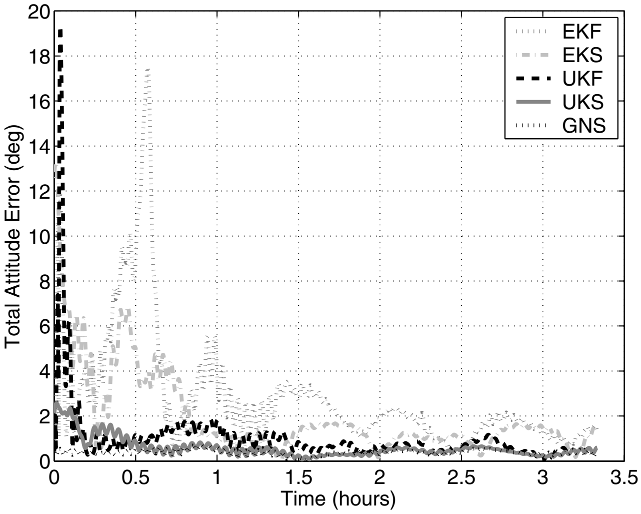

Fig. 1 Attitude estimation error time histories of the EKF, the EKS, the UKF, the UKS, and the GNS on a problem with small initial attitude and rate errors (test case 1).

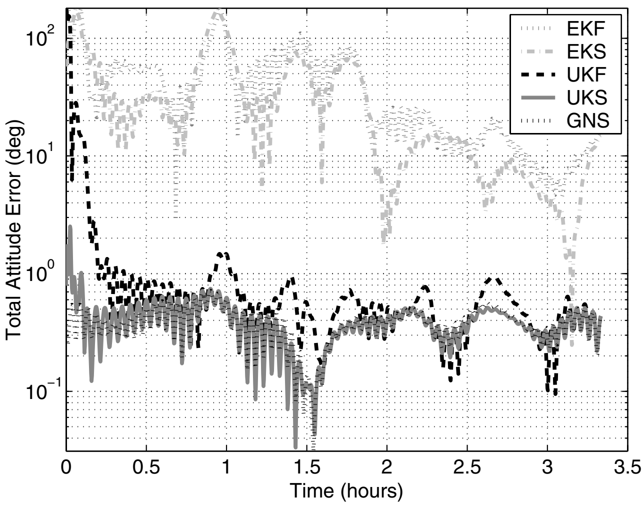

Fig. 2 Time histories of attitude errors for two filters and three smoothers for a problem with a 180 deg initial attitude error (test case 2).

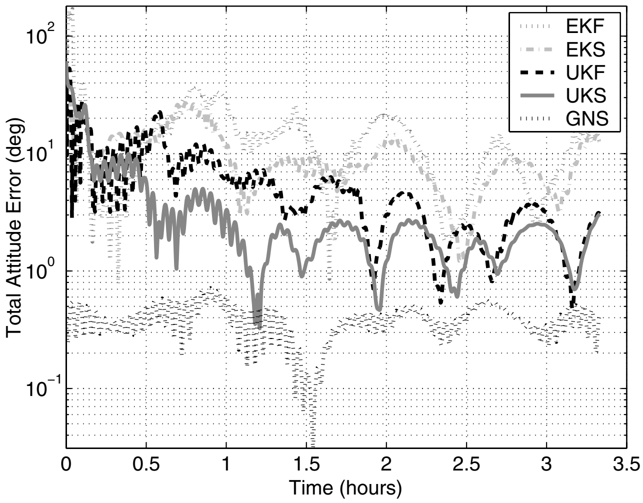

Fig. 3 Comparison of attitude error time histories for a case with a large initial rate error (test case 3).

The accuracy ranking of the five estimators is the same on all three test cases. Their ranking in order of improving accuracy is the EKF, the EKS, the UKF, the UKS, and the GNS. Each smoother performs better, on average, than its corresponding filter. The sigma-points methods perform better than the extended Kalman methods, but the Gauss–Newton smoother performs the best of all. These results are as expected.

The UKS works very well in cases 1 and 2. It achieves an accuracy that is comparable to that of the GNS for most of the smoothing interval, an accuracy that is on the order of 0.7 deg or better. Its accuracy degrades in comparison to the GNS only near the beginning of the interval, but its initial accuracy is much better than that of its corresponding filter, the UKF. In Fig. 1, the UKF has an initial transient spike to a peak error of 19.3 deg, but the peak error of the UKS during this transient period is only 2.6 deg. Although this latter error is more than three times larger than the peak error of the GNS, the UKS still demonstrates an ability to significantly improve accuracy in comparison to a filtered estimate during the initial transient period. Surprisingly, the performance of the UKS on test case 2 is almost identical to its performance on test case 1 despite the

180 deg initial attitude error for case 2: compare the solid gray curves in Figs. 1 and 2. This represents a significant ability of the UKS to take advantage of its noncausal data processing even though its initial sigma-points from its filtered run are distributed about a wildly wrong initial estimate. The EKS, in contrast, is not able to eliminate the initial transient errors of the EKF for case 2; its peak error in Fig. 2 is 179.3 deg.

Test case 3 in Fig. 3 serves as a reminder that sigma-points methods may not perform well in all situations. Although still better than the extended methods, the UKF and the UKS are both significantly less accurate than the GNS, and they both experience initial transients. The UKS has better accuracy than the UKF, on average, but its total attitude error regularly wanders above 2 deg during the second half of the smoothing interval, even though the error transients have died out. The GNS is more than four times as accurate as the UKS during this same interval. If it were possible to develop a UKS that selected new sigma points during its backward pass, possibly within an iterative outer loop, then the accuracy of such a UKS on this difficult problem might approach that of the GNS.

In summary, the truth-model simulation results indicate that the sigma-points backward-pass smoother can provide the usual accuracy improvements of a smoother vs a filter when run using data from a UKF forward filtering pass. The sigma-points smoother is more accurate than an extended Kalman smoother for the nonlinear cases considered, and its accuracy can approach that of a Gauss–Newton iterative nonlinear smoother on some example problems. Being a local method, however, a sigma-points smoother can still experience degraded accuracy if its corresponding UKF fails to achieve sufficient accuracy so that its sigma points are not representative of the true error statistics of the underlying estimation problem.

A UKS can be executed in a small fraction of the time required to execute a GNS, and the encoding of a UKS algorithm is much simpler than the encoding of a GNS algorithm. Therefore, a UKS is preferred in situations where its sigma-points approximation yields sufficient accuracy.

VI. Conclusions

The sigma-points approximation technique has been applied to the problem of fixed-interval smoothing for discrete-time nonlinear systems. The sigma-points smoothing algorithm starts by executing a forward sigma-points filtering pass through the data batch, and it finishes with a backward smoothing pass. The algorithm does not require the inversion of the dynamics function.

Two equivalent versions of the algorithm have been developed from the principles of Bayesian analysis. In one form of the algorithm, the backward smoothing pass is almost identical to that of the Rauch–Tung–Striebel smoother, except that its smoothing gains are computed using sigma points. The other form of the algorithm reveals that smoothing can be accomplished by updating the a posteriori filtered state using a pseudomeasurement. The pseudomeasurement relates the nonlinear dynamics state transition function to the smoothed and a priori state estimates at the subsequent sample. The pseudomeasurement model is nonlinear, and the corresponding update is accomplished by using standard sigma-points methods.

The sigma-points smoothing algorithm has been tested using a truth-model simulation of a difficult attitude determine problem. The sigma-points smoother achieves significant accuracy improvements over a sigma-points filter because its noncausal computations enable it to exploit the availability of additional data that come after the sample time to which a given state estimate applies. Such improvements are most notable near the initial portion of a data batch, where a sigma-points filter has significant error transients that are largely eliminated by the sigma-points smoother. The sigma-points smoother demonstrates superior performance to that of an extended Kalman smoother, and on some problems, it achieves accuracies that approach those of an iterative nonlinear batch filter. On one test problem, however, its accuracy is seriously degraded in comparison to a nonlinear batch filter because its sigma-points samples are rendered statistically inaccurate by degraded performance of its forward-pass sigma-points filter.

References

[1] Bar-Shalom, Y., Li, X.-R., and Kirubarajan, T., *Estimation with Applications to Tracking and Navigation*, J. Wiley & Sons, New York, 2001, pp. 200–217, 333–338, 381–395.

[2] Crassidis, J. L., and Junkins, J. L., *Optimal Estimation of Dynamic Systems*, Chapman & Hall/CRC, New York, 2004, pp. 285–292, 343–385.

[3] Psiaki, M. L., "Backward-Smoothing Extended Kalman Filter," *Journal of Guidance, Control, and Dynamics*, Vol. 28, No. 5, Sept.–Oct. 2005, pp. 885–894.

[4] 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.

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

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

[7] Ristic, B., Arulampalam, S., and Gordon, N., *Beyond the Kalman Filter*, Artech House, Boston, 2004, pp. 35–65.

[8] 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.

[9] Psiaki, M. L., "Estimation Using Quaternion Probability Densities on the Unit Hypersphere," AAS, Paper 05-461, June 2005.

[10] Psiaki, M. L., Theiler, J., Bloch, J., Ryan, S., Dill, R. W., and Warner, R. E., "ALEXIS Spacecraft Attitude Reconstruction with Thermal/Flexible Motions Due to Launch Damage," *Journal of Guidance, Control, and Dynamics*, Vol. 20, No. 5, Sept.–Oct. 1997, pp. 1033–1041.

[11] Humphreys, T. E., Psiaki, M. L., Klatt, E. M., Powell, S. P., and Kintner, P. M., Jr., "Magnetometer-Based Attitude and Rate Estimation for a Spacecraft with Wire Booms," *Journal of Guidance, Control, and Dynamics*, Vol. 28, No. 4, July–Aug. 2005, pp. 584–593.

[12] Venkatasubramanian, V., and Leung, H., "Chaos Based Semi-Blind System Identification Using an EM-UKS Estimator," *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, IEEE, Los Alamitos, CA, Oct. 2005, pp. 2873–2878.

[13] Ypma, A., and Heskes, T., "Novel Approximations for Inference in Nonlinear Dynamical Systems Using Expectation Propagation," *Neurocomputing; Variable Star Bulletin*, Vol. 69, Nos. 1–3, Dec. 2005, pp. 85–99.

[14] Mao, X., Wada, M., and Hashimoto, H., "Nonlinear Iterative Algorithm for GPS Positioning with Bias Model," *Proceedings of the IEEE Intelligent Transportation Systems Conference*, IEEE, Los Alamitos, CA, Oct. 2004, pp. 684–689.

[15] Rauch, H. E., Tung, F., and Striebel, C. T., "Maximum Likelihood Estimation for Linear Dynamic Systems," *AIAA Journal*, Vol. 3, No. 8, Aug. 1965, pp. 1445–1450.

[16] Gill, P. E., Murray, W., and Wright, M. H., *Practical Optimization*, Academic Press, New York, 1981, pp. 36–40, 134–136.