Automatica 47 (2011) 2245–2250

Contents lists available at SciVerse ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

Brief paper

Cubature Kalman smoothers[☆]

Ienkaran Arasaratnam [1], Simon Haykin

Cognitive Systems Laboratory, McMaster University, Hamilton, ON L8S 4K1, Canada

ARTICLE INFO

Article history:
Received 25 May 2010
Received in revised form
22 January 2011
Accepted 9 April 2011
Available online 31 August 2011

Keywords:
Cubature Kalman filter
Fixed-interval smoothing
Rauch–Tung–Striebel Smoothing
Square-root filtering

ABSTRACT

The cubature Kalman filter (CKF) is a relatively new addition to derivative-free approximate Bayesian filters built under the Gaussian assumption. This paper extends the CKF theory to address nonlinear smoothing problems; the resulting state estimator is named the fixed-interval cubature Kalman smoother (FI-CKS). Moreover, the FI-CKS is reformulated to propagate the square-root error covariances. Although algebraically equivalent to the FI-CKS, the square-root variant ensures reliable implementation when committed to embedded systems with fixed precision or when the inference problem itself is ill-conditioned. Finally, to validate the formulation, the square-root FI-CKS is applied to track a ballistic target on reentry.

1. Introduction

In Arasaratnam and Haykin (2009), Arasaratnam and Haykin described a new nonlinear filter named the *Cubature Kalman Filter* (CKF), for hidden state estimation based on nonlinear *discrete-time* state-space models. Like the celebrated Kalman filter for linear Gaussian models, an important virtue of the CKF is its mathematical rigor. This rigor is rooted in the *third-degree spherical-radial cubature rule* for numerically computing Gaussian-weighted integrals. A unique characteristic of the CKF is the fact that the spherical-radial cubature rule leads to an even number of equally-weighted cubature points ($2n$ points, where n is the dimensionality of the state vector). These cubature points are distributed uniformly on a sphere centered at the origin.

In a related context, the *unscented Kalman filter* (UKF) due to Julier et al. has an odd number of sigma points ($(2n + 1)$ points). These sigma points are distributed on an ellipsoid with a non-zero center point (Julier, Ulhmann, & Durrant-Whyte, 2000). Whereas the cubature points of the CKF follow rigorously from the spherical-radial cubature rule, the sigma points of the UKF are the result of the so-called *unscented transformation* applied to inputs. Theoretically, there is a fundamental difference between the CKF and the UKF. The CKF follows directly from the cubature rule whose important property is that it does not entail any free parameter. In contrast, the UKF purposely introduces a nonzero scaling parameter, commonly denoted by κ. Due to κ, a nonzero center point is often associated with a weight higher than that of the remaining set of sigma points (see Section VII of Arasaratnam and Haykin (2009) and Section III of Arasaratnam, Haykin, and Hurd (2010) for more details). Although the inclusion of the free parameter κ gives freedom to the UKF when it is non-zero, it destroys many desired numerical and theoretical properties of the UKF.

The parameter $\kappa = (3 - n)$ of the 'plain' UKF is zero only when the state dimensionality is three (by 'plain' we mean the UKF without using a scaled unscented transformation). For this special case, what is truly interesting is that the sigma point set boils down to the cubature point set and the algorithmic steps of the plain UKF become identical to that of the CKF. As such, the CKF may be considered as a special case of the UKF in an algebraic sense. However, it is ironic that the observation for setting κ equal to zero has been largely overlooked in the literature on nonlinear filtering for the past many years. Note that the authors of Wu, Hu, Wu, and Hu (2006) attempted to rederive the UKF from the integration perspective using monomial rules. However, a direct use of monomial rules leads to a free parameter similarly to the original derivation proposed by Julier et al. (2000). It is unfortunate again that the authors have completely ignored the possibility of setting the free parameter u_1 in (13) of Wu et al. (2006) to be $u_1 = \sqrt{d}$. This choice leads to the CKF equations, which in turn solves the inherent stability issue of the UKF.

It is well-known that the state estimate of a smoother algorithm is more accurate than that of the corresponding filter counterpart (Meditch, 1969). The motivation of this paper is to derive a

[☆] The material in this paper was not presented at any conference. This paper was recommended for publication in revised form by Associate Editor Giancarlo Ferrari-Trecate under the direction of Editor Ian R. Petersen.

E-mail addresses: haran@ieee.org (I. Arasaratnam), haykin@mcmaster.ca (S. Haykin).

[1] Tel.: +1 519 567 5738; fax: +1 519 944 8750.

0005-1098/$ – see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.automatica.2011.08.005

CKF-based smoothing algorithm. The novel contributions of this paper are as follows. (i) Application of cubature integration to existing integration-based smoothing theory. The resulting algorithm is named the *fixed-interval cubature Kalman smoother* (FI-CKS). (ii) For improved numerical stability in systems with limited precision, we go on to develop a square-root version of the FI-CKS. The square-root FI-CKS propagates the square-roots of the error covariances. (iii) Application of the square-root FI-CKS to target tracking. The paper is organized as follows. Section 2 reviews the optimal yet conceptual Bayesian inference solution. Section 3 reviews the CKF briefly. Section 4 derives a suboptimal fixed-interval smoother, which we have named the Fixed-Interval Cubature Kalman Smoother (FI-CKS). We go on to modify the FI-CKS in a way that it propagates the square-roots of covariances for improved numerical stability in Section 5. Section 6 validates the square-root FI-CKS formulation by applying it to a target tracking problem. Section 7 concludes the paper with final remarks.

2. Optimal Bayesian smoother

Consider the following discrete-time nonlinear state-space model as shown by

$$\text{Process equation: } \mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}) + \mathbf{v}_{k-1} \quad (1)$$

$$\text{Measurement equation: } \mathbf{z}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{w}_k, \quad (2)$$

where $\mathbf{x}_k \in \mathbb{R}^n$ is a hidden or latent variable, called the 'state' of the system at time k; $\mathbf{z}_k \in \mathbb{R}^m$ is the measurement at time k; $\mathbf{f}(.)$ and $\mathbf{h}(.)$ are some known nonlinear functions, and $\mathbf{v}_{k-1}$ and $\mathbf{w}_k$ are noise samples from two independent zero-mean Gaussian processes with covariances $\mathbf{Q}_{k-1}$ and $\mathbf{R}_k$, respectively. They account for process model uncertainty and the inaccuracy of a measuring device. Based on the state-space model, this section reviews how fixed-interval smoothing is performed.

The optimal solution of fixed-interval smoothing can be obtained in two different ways—two-filter smoothing (Fraser & Potter, 1969) and forward–backward smoothing (Rauch, Tung, & Striebel, 1965). For computational reasons, we will focus only on forward–backward smoothing. Given measurements up to time $N(> k), D_N = \{\mathbf{z}_1, \mathbf{z}_2, \ldots \mathbf{z}_N\}$, in forward–backward smoothing, using Bayes' rule the smoothed density $p(\mathbf{x}_k|D_N)$ is factored as follows:

$$\begin{aligned} p(\mathbf{x}_k|D_N) &= \int_{\mathbb{R}^n} p(\mathbf{x}_k, \mathbf{x}_{k+1}|D_N)\mathrm{d}\mathbf{x}_{k+1} \\ &= \int_{\mathbb{R}^n} p(\mathbf{x}_{k+1}|D_N)p(\mathbf{x}_k|\mathbf{x}_{k+1}, D_N)\mathrm{d}\mathbf{x}_{k+1}. \end{aligned} \quad (3)$$

The Markovian nature of the state-space model implies that given knowledge of D_k and $\mathbf{x}_{k+1}$, the state $\mathbf{x}_k$ is independent of future measurements $\{\mathbf{z}_{k+1}, \ldots \mathbf{z}_N\}$. That is, we may write

$$p(\mathbf{x}_k|\mathbf{x}_{k+1}, D_N) = p(\mathbf{x}_k|\mathbf{x}_{k+1}, D_k). \quad (4)$$

Substituting (4) into (3), we get the smoothed density

$$\begin{aligned} p(\mathbf{x}_k|D_N) &= \int_{\mathbb{R}^n} p(\mathbf{x}_{k+1}|D_N)p(\mathbf{x}_k|\mathbf{x}_{k+1}, D_k)\mathrm{d}\mathbf{x}_{k+1} \\ &= p(\mathbf{x}_k|D_k)\int_{\mathbb{R}^n} \frac{p(\mathbf{x}_{k+1}|D_N)p(\mathbf{x}_{k+1}|\mathbf{x}_k)}{p(\mathbf{x}_{k+1}|D_k)}\mathrm{d}\mathbf{x}_{k+1}. \end{aligned} \quad (5)$$

It is understood from (5) that the smoother has to perform two different passes. In the *forward filtering pass*, it computes the posterior density $p(\mathbf{x}_k|D_k)$ and the predictive density $p(\mathbf{x}_{k+1}|D_k)$ until the final time step; in the *backward smoothing pass*, it recursively computes the smoothed density backward in time starting from $k = N$.

For linear Gaussian systems in both the discrete and continuous time domains, the solution to forward–backward smoothing can be exactly found and is given by the *Rauch–Tung–Striebel* (RTS) smoother (Rauch et al., 1965). For nonlinear systems, however, the optimal smoothing solution is intractable for two reasons. (i) For a multi-dimensional system, we must compute the multi-dimensional integral (5). (ii) Even after this integral is computed, it may be difficult to propagate the smoothed density through subsequent time steps because the new smoothed density is not guaranteed to remain closed with a finite summary statistic. For these reasons, we resort to approximations to obtain a suboptimal smoother.

In the past, researchers have resorted to numerical methods to obtain approximate smoothing solutions. One of the well known approximate smoothers is the extended Kalman smoother, the basic idea of which is to apply the Kalman (or RTS) smoother theory by linearizing the nonlinear process and measurement functions using the first-order Taylor series expansion evaluated at the current estimate of the state (Bar Shalom, Li, & Kirubarajan, 2001). The RTS smoother theory can be well adopted to nonlinear Gaussian filters. In Särkkä and Hartikainen (2010) and Šimandl and Duník (2009), derivative-free RTS smoothers based on the unscented transformation, central differences (or the second order Stirling's interpolation) and Gauss–Hermite quadrature are presented in a unified framework. The CKF, a relatively new filter, yields reasonably accurate and numerically stable state estimates at a minimal cost (Arasaratnam & Haykin, 2009). In the subsequent sections, we derive the CKF-based square-root fixed-interval smoother.

3. Cubature Kalman filtering

In this section, before proceeding to the development of the Fixed-Interval Cubature Kalman Smoother (FI-CKS), we briefly review the CKF first (Arasaratnam & Haykin, 2009). The CKF is derived under the assumption that the predictive density of the joint state-measurement random variable is Gaussian (Arasaratnam & Haykin, 2009). This assumption naturally leads to a Gaussian predictive and filtering density of the state. Under this assumption, the Bayesian filter reduces to the problem of how to compute integrals whose integrands are all of the form *nonlinear function* $\times$ *Gaussian*. The CKF uses a third-degree cubature rule to numerically compute the above Gaussian-weighted integrals. For example, the cubature rule approximates an n-dimensional Gaussian weighted integral as follows:

$$\int_{\mathbb{R}^n} \mathbf{f}(\mathbf{x})\mathcal{N}(\mathbf{x}; \mu, \Sigma)\mathrm{d}\mathbf{x} \approx \frac{1}{2n}\sum_{i=1}^{2n} \mathbf{f}(\mu + \sqrt{\Sigma}\xi_i)$$

where a square-root factor of the covariance Σ satisfies the relationship $\Sigma = \sqrt{\Sigma}\sqrt{\Sigma}^T$ and the set of $2n$ cubature points are given by

$$\xi_i = \begin{cases} \sqrt{n}\mathbf{e}_i, & i = 1, 2 \ldots n \\ -\sqrt{n}\mathbf{e}_{i-n}, & i = n+1, n+2 \ldots 2n. \end{cases}$$

with $\mathbf{e}_i \in \mathbb{R}^n$ denoting the i-th elementary column vector. That is, the i-th entry of $\mathbf{e}_i$ is unity and all other entries are zero. The third-degree cubature rule is exact for Gaussian-weighted integrals whose integrands are written in the form of a linear combination of monomials up to the third degree or any odd-degree (Arasaratnam & Haykin, 2009). Assuming at time k that the posterior density $p(\mathbf{x}_k|D_k) = \mathcal{N}(\hat{\mathbf{x}}_{k|k}, \mathbf{P}_{k|k})$ is known, we summarize the steps involved in the time-update and the measurement-update of the CKF as shown in Table 1 that was derived in Arasaratnam and Haykin (2009).

Table 1

CKF: Time-update	
1. Factorize	
$\mathbf{P}_{k\|k} = \mathbf{S}_{k\|k}\mathbf{S}_{k\|k}^T$.	(6)
2. Generate the cubature points ($i = 1, 2, \ldots, 2n$)	
$X_{i,k\|k} = \mathbf{S}_{k\|k}\xi_i + \hat{\mathbf{x}}_{k\|k}$.	(7)
3. Compute the propagated cubature points ($i = 1, 2, \ldots, 2n$)	
$X^*_{i,k+1\|k} = \mathbf{f}(X_{i,k\|k})$.	(8)
4. Compute the predicted state	
$\hat{\mathbf{x}}_{k+1\|k} = \frac{1}{2n}\sum_{i=1}^{2n} X^*_{i,k+1\|k}$.	(9)
5. Compute the predicted state error covariance	
$\mathbf{P}_{k+1\|k} = \frac{1}{2n}\sum_{i=1}^{2n} X^*_{i,k+1\|k}X^{*T}_{i,k+1\|k} - \hat{\mathbf{x}}_{k+1\|k}\hat{\mathbf{x}}^T_{k+1\|k} + Q_k$.	(10)
6. Compute the cross-covariance (this step is needed only for smoothing not for filtering)	
$\mathbf{P}_{k,k+1\|k} = \frac{1}{2n}\sum_{i=1}^{2n} X_{i,k\|k}X^{*T}_{i,k+1\|k} - \hat{\mathbf{x}}_{k\|k}\hat{\mathbf{x}}^T_{k+1\|k}$.	(11)
CKF: Measurement-update	
1. Factorize	
$\mathbf{P}_{k+1\|k} = \mathbf{S}_{k+1\|k}\mathbf{S}^T_{k+1\|k}$.	(12)
2. Generate the cubature points ($i = 1, 2, \ldots, 2n$)	
$X_{i,k+1\|k} = \mathbf{S}_{k+1\|k}\xi_i + \hat{\mathbf{x}}_{k+1\|k}$.	(13)
3. Compute the propagated cubature points ($i = 1, 2, \ldots, 2n$)	
$Z_{i,k+1\|k} = \mathbf{h}(X_{i,k+1\|k})$.	(14)
4. Compute the predicted measurement	
$\hat{\mathbf{z}}_{k+1\|k} = \frac{1}{2n}\sum_{i=1}^{2n} Z_{i,k+1\|k}$.	(15)
5. Compute the innovation covariance	
$\mathbf{P}_{zz,k+1\|k} = \frac{1}{2n}\sum_{i=1}^{2n} Z_{i,k+1\|k}Z^T_{i,k+1\|k} - \hat{\mathbf{z}}_{k+1\|k}\hat{\mathbf{z}}^T_{k+1\|k} + \mathbf{R}_{k+1}$.	(16)
6. Compute the cross-covariance	
$\mathbf{P}_{xz,k+1\|k} = \frac{1}{2n}\sum_{i=1}^{2n} X_{i,k+1\|k}Z^T_{i,k+1\|k} - \hat{\mathbf{x}}_{k+1\|k}\hat{\mathbf{z}}^T_{k+1\|k}$.	(17)
7. Compute the filter gain	
$\mathbf{W}_{k+1} = \mathbf{P}_{xz,k+1\|k}\mathbf{P}^{-1}_{zz,k+1\|k}$.	(18)
8. Compute the filtered state	
$\hat{\mathbf{x}}_{k+1\|k+1} = \hat{\mathbf{x}}_{k+1\|k} + \mathbf{W}_{k+1}(\mathbf{z}_{k+1} - \hat{\mathbf{z}}_{k+1\|k})$.	(19)
9. Compute the filtered state error covariance	
$\mathbf{P}_{k+1\|k+1} = \mathbf{P}_{k+1\|k} - \mathbf{W}_{k+1}\mathbf{P}_{zz,k+1\|k}\mathbf{W}^T_{k+1}$.	(20)

4. Cubature Kalman smoothing

This section derives the FI-CKS from (5) by approximating all conditional densities in question to be Gaussian distributed. It may be tempting to obtain a Gaussian smoothed density by directly substituting Gaussian expressions for the conditional densities present in (5). However, this straightforward substitution approach involves multiplications, divisions and integration of Gaussians in a multi-dimensional state-space and is therefore tedious. To make matters simpler, an indirect approach using a few readily available densities and Gaussian lemmas is taken in Section II of Särkkä (2008) and Särkkä and Hartikainen (2010). The derivation of the FI-CKS closely follows this indirect approach: the forward pass of the FI-CKS uses the CKF to compute the posterior densities up to $p(\mathbf{x}_N|D_N)$, whereas the backward pass implements a five-step procedure for rolling backwards from $p(\mathbf{x}_N|D_N)$ to $p(\mathbf{x}_1|D_N)$ as shown in Table 2.

5. Square-root cubature Kalman smoothing

In each recursion cycle, it is important that we preserve the two properties of a covariance matrix, namely, its positive definitiveness and symmetry. Unfortunately, when the FI-CKS is committed to an embedded system with limited word-length, numerical errors may lead to a loss of these properties. The accumulation of numerical errors may cause the smoother to diverge or otherwise crash. The FI-CKS involves numerically sensitive operations such as matrix square-rooting, matrix inversion and subtraction of two positive-definite matrices, which may combine to destroy the fundamental properties of a covariance matrix (see, for example, (21) and (22)). The logical procedure to preserve both properties of the covariance and to improve the numerical stability is to design a square-root version of the FI-CKS. The square-root FI-CKS propagates the square-roots of the covariance matrices and is algebraically equivalent to the FI-CKS.

In this section, to develop the square-root FI-CKS, we use the following two powerful tools:

- *The least-squares method* to compute the smoother gain.
- *The matrix triangular factorizations* or triangularizations (e.g., QR decomposition) for the covariance updates.

The least-squares method avoids computing the matrix inversion explicitly, whereas the triangularization essentially computes a triangular square-root factor of the covariance without square-rooting a squared-matrix form of the covariance.

Before writing all the required steps of the square-root FI-CKS, for convenience, we first introduce the following notations. (i) Using the symbol '/' to represent the matrix right-divide operator, we write $\mathbf{BA}^{-1}$ to be $\mathbf{B}/\mathbf{A}$; this operator applies the *forward substitution* algorithm to compute filter/smoother gains (ii) We denote a general triangularization algorithm (e.g., QR decomposition) as $\mathbf{S} = Tria(\mathbf{A})$, where $\mathbf{S}$ is a lower triangular matrix. The matrices $\mathbf{A}$ and $\mathbf{S}$ are related to each other as follows: let $\mathbf{R}$ be an upper triangular matrix obtained from the QR decomposition on $\mathbf{A}^T$; then $\mathbf{S} = \mathbf{R}^T$ (iii) We use $\mathbf{S}_{Q,k}$ and $\mathbf{S}_{R,k}$ to denote the square-roots of $\mathbf{Q}_k$ and $\mathbf{R}_k$, respectively. Table 3 presents the steps involved in the square-root FI-CKS algorithm assuming at time k that the posterior density $p(\mathbf{x}_k|D_k) = \mathcal{N}(\hat{\mathbf{x}}_{k|k}, \mathbf{P}_{k|k} = \mathbf{S}_{k|k}\mathbf{S}^T_{k|k})$ is known.

Unlike Section 3, the forward pass of the square-root FI-CKS presented in this section implements the square-root CKF. A detailed derivation of the square-root CKF can be found in Section VI of Arasaratnam et al. (2010). In the backward pass, all the steps are straightforward except the square-root smoothed state error covariance, which is derived as follows: since we may write $\mathbf{P}_{k|k} = \mathscr{X}_{k|k}\mathscr{X}^T_{k|k}$, $\mathbf{P}_{k+1|k} = (\mathscr{X}^*_{k+1|k}\mathscr{X}^{*T}_{k+1|k} + \mathbf{S}_{Q,k}\mathbf{S}^T_{Q,k})$ and $\mathbf{P}_{k,k+1|k} = \mathscr{X}_{k|k}\mathscr{X}^{*T}_{k+1|k}$, it holds that

$$\begin{pmatrix} \mathbf{P}_{k+1|k} & \mathbf{P}_{k+1,k|k} \\ \mathbf{P}_{k,k+1|k} & \mathbf{P}_{k|k} \end{pmatrix} = \begin{pmatrix} \mathscr{X}^*_{k+1|k} & \mathbf{S}_{Q,k+1} \\ \mathscr{X}_{k|k} & \mathbf{O} \end{pmatrix} \begin{pmatrix} \mathscr{X}^*_{k+1|k} & \mathbf{S}_{Q,k+1} \\ \mathscr{X}_{k|k} & \mathbf{O} \end{pmatrix}^T, \quad (36)$$

Table 2

FI-CKS: Backward pass

1. Given the posterior density $p(\mathbf{x}_k|D_k)$ at time k, compute the predictive density
$p(\mathbf{x}_{k+1}|D_k) = \mathcal{N}(\hat{\mathbf{x}}_{k+1|k}, \mathbf{P}_{k+1|k})$
2. Construct the joint conditional density of $(\mathbf{x}_k, \mathbf{x}_{k+1})$
$$p(\mathbf{x}_k, \mathbf{x}_{k+1}|D_k) = \mathcal{N}\left(\begin{pmatrix}\hat{\mathbf{x}}_{k|k}\\ \hat{\mathbf{x}}_{k+1|k}\end{pmatrix}, \begin{pmatrix}\mathbf{P}_{k|k} & \mathbf{P}_{k,k+1|k}\\ \mathbf{P}_{k+1,k|k} & \mathbf{P}_{k+1|k}\end{pmatrix}\right)$$
3. Compute the conditional density
$p(\mathbf{x}_k|\mathbf{x}_{k+1}, D_k) = \mathcal{N}(\hat{\mathbf{x}}_{k|k} + \mathbf{G}_k(\mathbf{x}_{k+1} - \hat{\mathbf{x}}_{k+1|k}), \mathbf{P}_{k|k} - \mathbf{G}_k\mathbf{P}_{k+1|k}\mathbf{G}_k^T)$
where
$$\mathbf{G}_k = \mathbf{P}_{k,k+1|k}\mathbf{P}_{k+1|k}^{-1} \tag{21}$$
4. Given the smoothed density $p(\mathbf{x}_{k+1}|D_N)$, using Bayes' rule construct the joint conditional density
$$p(\mathbf{x}_k, \mathbf{x}_{k+1}|D_N) = \mathcal{N}\left(\begin{pmatrix}\mathbf{x}_{k|k} + \mathbf{G}_k(\mathbf{x}_{k+1} - \hat{\mathbf{x}}_{k+1|k})\\ \hat{\mathbf{x}}_{k+1|N}\end{pmatrix}, \begin{pmatrix}\boldsymbol{\Sigma}_k & \mathbf{G}_k\mathbf{P}_{k+1|N}\\ \mathbf{P}_{k+1|N}\mathbf{G}_k^T & \mathbf{P}_{k+1|N}\end{pmatrix}\right)$$
where
$$\boldsymbol{\Sigma}_k = \mathbf{P}_{k|k} - \mathbf{G}_k(\mathbf{P}_{k+1|k} - \mathbf{P}_{k+1|N})\mathbf{G}_k^T \tag{22}$$
5. From $p(\mathbf{x}_k, \mathbf{x}_{k+1}|D_N)$, compute the smoothed density
$p(\mathbf{x}_k|D_N) = \mathcal{N}(\hat{\mathbf{x}}_{k|N} = \hat{\mathbf{x}}_{k|k} + \mathbf{G}_k(\hat{\mathbf{x}}_{k+1|N} - \hat{\mathbf{x}}_{k+1|k}), \mathbf{P}_{k|N} = \mathbf{P}_{k|k} - \mathbf{G}_k(\mathbf{P}_{k+1|k} - \mathbf{P}_{k+1|N})\mathbf{G}_k^T)$

Table 3

Square-root FI-CKS—forward pass $k = 0, 1 \dots (N-1)$

1. Since the square-root of the error covariance, $\mathbf{S}_{k|k}$, is available, skip the factorization (6). Compute from (7)–(9).
2. Compute the square-root of the predicted state error covariance
$$\mathbf{S}_{k+1|k} = Tria([\mathcal{X}^*_{k+1|k}\mathbf{S}_{Q,k}]). \tag{23}$$
where the weighted centered (predicted mean estimate is subtracted off) matrix
$$\mathcal{X}^*_{k+1|k} = \frac{1}{\sqrt{2n}}[X^*_{1,k+1|k} - \hat{\mathbf{x}}_{k+1|k} \dots X^*_{2n,k+1|k} - \hat{\mathbf{x}}_{k+1|k}]. \tag{24}$$
3. Compute (13)–(15).
4. Construct the following two weighted-centered matrices:
$$\mathcal{X}_{k+1|k} = \frac{1}{\sqrt{2n}}[X_{1,k+1|k} - \hat{\mathbf{x}}_{k+1|k} \dots X_{2n,k+1|k} - \hat{\mathbf{x}}_{k+1|k}]. \tag{25}$$
$$\mathcal{Z}_{k+1|k} = \frac{1}{\sqrt{2n}}[Z_{1,k+1|k} - \hat{\mathbf{z}}_{k+1|k} \dots Z_{2n,k+1|k} - \hat{\mathbf{z}}_{k+1|k}]. \tag{26}$$
5. Compute the matrices $\mathbf{T}_{11}$, $\mathbf{T}_{21}$ and $\mathbf{T}_{22}$ using the triangularization algorithm:
$$\begin{pmatrix}\mathbf{T}_{11} & \mathbf{O}\\ \mathbf{T}_{21} & \mathbf{T}_{22}\end{pmatrix} = Tria\begin{pmatrix}\mathcal{Z}_{k+1|k} & \mathbf{S}_{R,k+1}\\ \mathcal{X}_{k+1|k} & \mathbf{O}\end{pmatrix}. \tag{27}$$
6. Compute the filter gain
$$\mathbf{W}_{k+1} = \mathbf{T}_{21}/\mathbf{T}_{11}. \tag{28}$$
7. Compute the filtered state
$$\hat{\mathbf{x}}_{k+1|k+1} = \hat{\mathbf{x}}_{k+1|k} + \mathbf{W}_{k+1}(\mathbf{z}_{k+1} - \hat{\mathbf{z}}_{k+1|k}). \tag{29}$$
8. The square-root of the filtered state-error covariance
$$\mathbf{S}_{k+1|k+1} = \mathbf{T}_{22}. \tag{30}$$

Square-root FI-CKS—backward pass: $k = (N-1) \dots 1$

1. Compute the matrices $\mathbf{U}_{11}$, $\mathbf{U}_{21}$ and $\mathbf{U}_{22}$ using the triangularization algorithm:
$$\begin{pmatrix}\mathbf{U}_{11} & \mathbf{O}\\ \mathbf{U}_{21} & \mathbf{U}_{22}\end{pmatrix} = Tria\begin{pmatrix}\mathcal{X}^*_{k+1|k} & \mathbf{S}_{Q,k+1}\\ \mathcal{X}_{k|k} & \mathbf{O}\end{pmatrix}. \tag{31}$$
where the weighted centered matrix
$$\mathcal{X}_{k|k} = \frac{1}{\sqrt{2n}}[X_{1,k|k} - \hat{\mathbf{x}}_{k|k} \dots X_{m,k|k} - \hat{\mathbf{x}}_{k|k}]. \tag{32}$$
2. Compute the smoother gain
$$\mathbf{G}_k = \mathbf{U}_{21}/\mathbf{U}_{11}. \tag{33}$$
3. Compute the smoothed state
$$\hat{\mathbf{x}}_{k|N} = \hat{\mathbf{x}}_{k|k} + \mathbf{G}_k(\hat{\mathbf{x}}_{k+1|N} - \hat{\mathbf{x}}_{k+1|k}). \tag{34}$$
4. Compute the square-root of the smoothed state error covariance
$$\mathbf{S}_{k|N} = Tria\Big([\mathbf{U}_{22} \quad \mathbf{G}_k\mathbf{S}_{k+1|N}]\Big). \tag{35}$$

where $\mathbf{O} \in \mathbb{R}^{n\times n}$ is the zero matrix. Applying the triangularization procedure to the square-root factor present on the right-hand side of (36) yields

$$Tria\begin{pmatrix}\mathcal{X}^*_{k+1|k} & \mathbf{S}_{Q,k+1}\\ \mathcal{X}_{k|k} & \mathbf{O}\end{pmatrix} = \begin{pmatrix}\mathbf{U}_{11} & \mathbf{O}\\ \mathbf{U}_{21} & \mathbf{U}_{22}\end{pmatrix}, \tag{37}$$

where $\mathbf{U}_{11} \in \mathbb{R}^{n\times n}$, and $\mathbf{U}_{22} \in \mathbb{R}^{n\times n}$ are lower-triangular matrices, and $\mathbf{U}_{21} \in \mathbb{R}^{n\times n}$. Hence, we may rewrite (36) in a 'new' squared-matrix form as follows:

$$\begin{pmatrix}\mathbf{P}_{k+1|k} & \mathbf{P}_{k+1,k|k}\\ \mathbf{P}_{k,k+1|k} & \mathbf{P}_{k|k}\end{pmatrix} = \begin{pmatrix}\mathbf{U}_{11}\mathbf{U}_{11}^T & \mathbf{U}_{11}\mathbf{U}_{21}^T\\ \mathbf{U}_{21}\mathbf{U}_{11}^T & \mathbf{U}_{21}\mathbf{U}_{21}^T + \mathbf{U}_{22}\mathbf{U}_{22}^T\end{pmatrix}. \tag{38}$$

Substituting the above results into the smoother gain (21) yields

$$\mathbf{G}_k = \mathbf{U}_{21}\mathbf{U}_{11}^T(\mathbf{U}_{11}\mathbf{U}_{11}^T)^{-1} = \mathbf{U}_{21}\mathbf{U}_{11}^{-1} = \mathbf{U}_{21}/\mathbf{U}_{11}. \tag{39}$$

Let us now see how $\mathbf{P}_{k|N}$ can be written in matrix-squared form. We expand the smoother covariance, as shown by

$$\mathbf{P}_{k|N} = \mathbf{P}_{k|k} - \mathbf{G}_k\mathbf{P}_{k+1|k}\mathbf{G}_k^T + \mathbf{G}_k\mathbf{P}_{k+1|N}\mathbf{G}_k^T. \tag{40}$$

Substituting the results obtained in (38) and (39) into (40) yields

$$\begin{aligned}\mathbf{P}_{k|N} &= (\mathbf{U}_{21}\mathbf{U}_{21}^T + \mathbf{U}_{22}\mathbf{U}_{22}^T) - \mathbf{U}_{21}\mathbf{U}_{11}^{-1}(\mathbf{U}_{11}\mathbf{U}_{11}^T)\\ &\quad \times (\mathbf{U}_{21}\mathbf{U}_{11}^{-1})^T + \mathbf{G}_k\mathbf{S}_{k+1|N}(\mathbf{G}_k\mathbf{S}_{k+1|N})^T\\ &= [\mathbf{U}_{22} \quad \mathbf{G}_k\mathbf{S}_{k+1|N}][\mathbf{U}_{22} \quad \mathbf{G}_k\mathbf{S}_{k+1|N}]^T.\end{aligned} \tag{41}$$

Hence, the matrix triangularization of (41) leads to (35).

6. Application to tracking a ballistic target on reentry

Reentry Scenario. Tracking ballistic targets is useful for intercepting and destroying them before they hit the ground. To illustrate the computing power of the SR-FI-CKS, we consider a problem of tracking a ballistic target on reentry (Ristic, Arulampalam, & Gordon, 2004). Under the influence of drag and gravity acting on the target, the following differential equation governs its motion (Ristic et al., 2004):

$$\dot{\mathbf{x}}_t = [-\mathbf{x}_t[2] \underbrace{-\exp(-\gamma \mathbf{x}_t[1])\mathbf{x}_t^2[2]\mathbf{x}_t[3]}_{\text{drag}} + g \;\; 0]^T$$

$$= \mathbf{g}(\mathbf{x}_t), \tag{42}$$

where x_1 and x_2 are altitude and velocity, respectively; x_3 is a constant ballistic coefficient that depends on the target's mass, shape, cross-sectional area and air density; the state vector $\mathbf{x} = [x_1 x_2 x_3]^T$; g is gravity ($g = 9.81$ ms^{-2}) and the constant $\gamma = 1.49 \times 10^{-4}$. Applying the Euler approximation to (42) with a small integration step $\delta = (t_{k+1} - t_k)$, we get

$$\mathbf{x}_{k+1} = [\mathbf{x}_k + \delta \mathbf{g}(\mathbf{x}_k)]$$

$$= \mathbf{f}(\mathbf{x}_k), \tag{43}$$

where $\mathbf{f}(\mathbf{x}_k) = (\mathbf{x}_k[1] - \delta \mathbf{x}_k[2], \mathbf{x}_k[2] + \delta(-\exp(-\gamma \mathbf{x}_k[1])\mathbf{x}_k^2[2]\mathbf{x}_k[3] + g), \mathbf{x}_k[3])^T$.

For the model problem at hand, a radar was located at $(0, H)$ and equipped to measure the range r at a sampling time interval of T. Hence, the measurement equation is given by

$$r_k = \sqrt{M^2 + (\mathbf{x}_k[1] - H)^2} + v_k \tag{44}$$

where the measurement noise $v_k \sim \mathcal{N}(0, R_k)$, and M is the horizontal distance. To track the ballistic target, the following Bayesian smoothers were employed. (i) Fixed-interval extended Kalman smoother (FI-EKS) (ii) Fixed-interval central difference Kalman smoother (FI-CDKS) (iii) Fixed-interval 3-point quadrature Kalman smoother (FI-QKS) (iv) Fixed-interval cubature Kalman smoother (FI-CKS). Note that we do not consider the FI-UKS because it reduces to the FI-CKS when the state dimension equals three as described in Section 1. The smoothers were then compared against the CKF. We did not consider the EKF because it is well known that the CKF always outperforms the EKF (Arasaratnam & Haykin, 2009).

Experimental Data. $H = 1$ km, $M = 10$ km, $R = (30\text{ m})^2$, $T = \delta = 0.5$ s, and the true initial state was assumed to be $\mathbf{x}_0 = [61\text{ km } 3048\text{ m/s } 4.49 \times 10^{-4}]^T$. To initialize the estimators, the initial state density was assumed to be Gaussian with the mean $\hat{\mathbf{x}}_{0|0} = [62\text{ km } 3400\text{ m/s } 10^{-5}]^T$ and covariance $\mathbf{P}_{0|0} = \text{diag}([10^6 10^4 10^{-4}])$.

Performance Metric. We computed the RMSE of the i-th state component at time k, as shown by

$$\text{RMSE}_k[i] = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(\mathbf{x}_k^{(n)}[i] - \hat{\mathbf{x}}_{k|k}^{(n)}[i])^2},$$

where N is the total number of Monte Carlo simulation runs. Each trajectory was simulated for 30s and a total of $N = 1000$ independent Monte Carlo runs were made. We also computed the accumulative RMSE of the i-th state component, as shown by

$$\text{ARMSE}_k[i] = \sqrt{\frac{1}{NK}\sum_{k=1}^{K}\sum_{n=1}^{N}(\mathbf{x}_k^{(n)}[i] - \hat{\mathbf{x}}_{k|k}^{(n)}[i])^2}.$$

Observations. The RMSEs in altitude, velocity and ballistic coefficient are shown in Figs. 1–3. As expected, the FI-CKS is superior to the FI-EKS and markedly outperforms the CKF. Fig. 3 shows the performance of the Bayesian estimators in estimating the ballistic parameter. As can be seen from Fig. 3, the backward smoothing pass of the FI-CKS does not improve the parameter

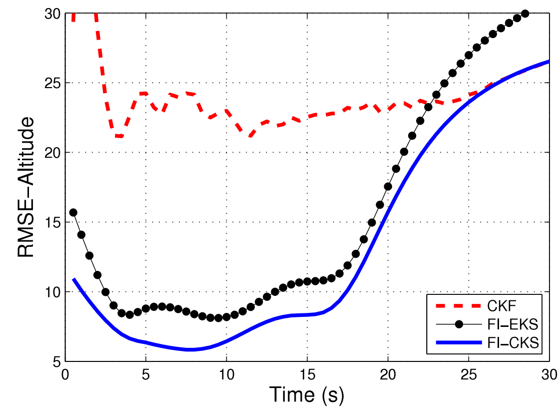

Fig. 1. RMSE in position.

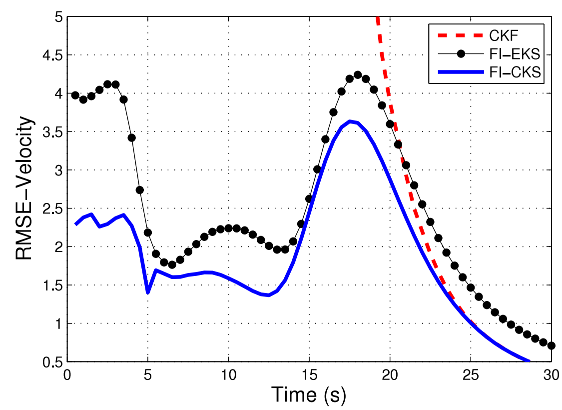

Fig. 2. RMSE in velocity.

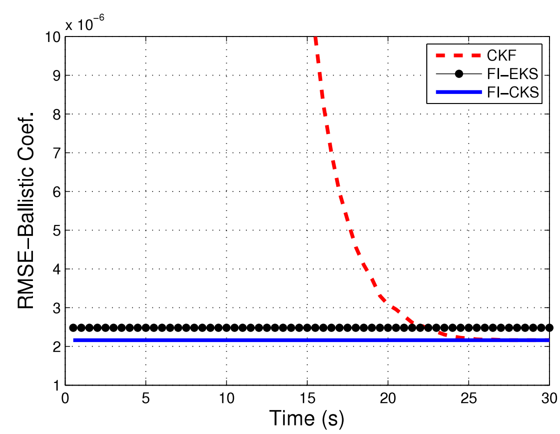

Fig. 3. RMSE in ballistic coefficient.

estimate of the CKF made at time $k = 30$. This substantiates the idea that parameters (or constant states) are not smoothable. We did not present the results of the FI-CDKS and the FI-QKS in Figs. 1–3, because the results of these smoothers and the FI-CKS were found to be nearly identical and overlapping (see, for example, Table 4).

Remark: The derivative-free estimators can be unified under a broad perspective of numerical integration (Wu et al., 2006). As such, the computational costs of all derivative-free smoothers are dominated by function evaluations, which in turn depend on the number of points used by the smoothers. For example, the FI-CDKS uses the central-difference scheme of points $(2n + 1)$, whereas the FI-CKS uses a $2n$ equally-weighted cubature points. Hence, the FI-CKS is expected to run 'slightly' faster than the FI-CDKS.

Table 4
ARMSE.

Algorithm	Position (m)	Velocity (m/s)	Ball. coef.
CKF	23.15	54.58	2.1×10^{-3}
FI-EKS	18.03	2.97	2.48×10^{-6}
FI-CDKS	15.83	2.04	2.16×10^{-6}
FI-QKS	15.81	2.02	2.16×10^{-6}
FI-CKS	15.83	2.01	2.16×10^{-6}

Although all derivative-free smoothers have performed equally in this considered problem, for high-dimensional problems, the FI-CKS may be preferred among them. The FI-CKS is the only estimator that could provide numerically stable results (due to its square-root form) at the least computational cost (due to the use of the least number of points that are all equally weighted). However, note that unlike the derivative-free smoothers, the FI-EKS uses Jacobian and can be viewed as a smoother using a single point. Therefore, from a computational perspective, the FI-EKS is the cheapest of all. The reader may refer to Table I of Wu et al. (2006) under efficiency for a similar discussion on the computational comparison of derivative-free filters.

7. Concluding remarks

In this paper, we have derived the fixed-interval cubature Kalman smoother (FI-CKS) for nonlinear and Gaussian systems. To ensure reliable implementation in hardware with limited precision, the square-root FI-CKS, which propagates the square-root error covariances, has been derived at the expense of increased computational complexity. Unlike filtering algorithms, the smoothed state-error covariance is not an integral part of smoothing algorithms. This is an important observation for two reasons. (i) Because the loss of the properties of a smoothed state-error covariance does not affect the accuracy of the smoothed estimate, a square-root reformulation of the backward smoothing pass may not be necessary. (ii) We may avoid computing the smoothed state error covariance to substantially reduce the computational cost; given that the smoother gain is computed in the forward pass, it is sufficient to propagate the first moment in the backward pass. Because the fixed-lag and fixed-point smoothers are built around the fixed-interval smoother theory, by closely following (Meditch, 1969; Särkkä & Hartikainen, 2010), we may go on to derive the square-root cubature-based fixed-lag and fixed-point smoothers. In a nutshell, the cubature Kalman smoothers are a new algorithmic addition to derivative-free estimation tool kits.

References

Arasaratnam, I., & Haykin, S. (2009). Cubature kalman filters. *IEEE Transactions on Automatic Control*, *54*(6), 1254–1269.
Arasaratnam, I., Haykin, S., & Hurd, T. R. (2010). Cubature kalman filtering for continuous-discrete systems: theory and simulations. *IEEE Transactions on Signal Processing*, *58*(10), 4977–4993.
Bar Shalom, Y., Li, X. R., & Kirubarajan, T. (2001). *Estimation with applications to tracking and navigation*. NY: Wiley & Sons.
Fraser, D., & Potter, J. (1969). The optimum linear smoother as a combination of two optimum linear filters. *IEEE Transactions on Automatic Control*, *14*(4).
Julier, S. J., Ulhmann, J. K., & Durrant-Whyte, H. F. (2000). A new method for nonlinear transformation of means and covariances in filters and estimators. *IEEE Transactions on Automatic Control*, *45*(3), 472–482.
Meditch, J. (1969). *Stochastic optimal linear estimation and control*. McGraw-Hill.
Rauch, H. E., Tung, F., & Striebel, C. T. (1965). Maximum likelihood estimates of linear dynamic systems. *AIAA Journal*, *8*(3), 1445–1450.
Ristic, B., Arulampalam, S., & Gordon, N. (2004). *Beyond the kalman filter*. MA: Artech House.
Särkkä, S. (2008). Unscented Rauch–Tung–Striebel smoother. *IEEE Transactions on Automatic Control*, *53*(3), 845–849.
Särkkä, S., & Hartikainen, J. (2010). On Gaussian optimal smoothing of non-linear state space models. *IEEE Transactions on Automatic Control*, *55*(8), 1938–1941.
Šimandl, M., & Duník, J. (2009). Derivative-free estimation methods: new results and performance analysis. *Automatica*, *45*(7), 1749–1757.
Wu, Y., Hu, D., Wu, M., & Hu, X. (2006). Numerical integration perspective on Gaussian filters. *IEEE Transactions on Signal Processing*, *54*(8), 2910–2921.

Ienkaran Arasaratnam received the B.Sc. degree (first-class honors) in the Department of Electronics and Telecommunication Engineering from the University of Moratuwa, Sri Lanka in 2003 and the M.A.Sc. and Ph.D. degrees in the Department of Electrical and Computer Engineering from McMaster University, Hamilton, ON, Canada in 2006 and 2009, respectively. For his Ph.D. dissertation titled 'Cubature Kalman Filtering: Theory and Applications', he formulated a new approximate nonlinear Bayesian filter called the 'Cubature Kalman Filter'. He is currently working as a R & D engineer at Ford motor company of Canada Ltd. His main research interests include signal processing, control and machine learning with applications to target tracking and fault diagnosis in automotive engines. Many of his research work can be found at https://sites.google.com/site/haranarasaratnam/.

Dr. Arasaratnam received the Mahapola Merit Scholarship during his undergraduate studies. He is the recipient of the Outstanding Thesis Research Award at M.A.Sc. Level in 2006, the Ontario Graduate Scholarship in Science and Technology in 2008 and NSERC's Industrial Research & Development Fellowship in 2010.

Simon Haykin (Fellow, IEEE) received the B.Sc. (first-class honors), Ph.D., and D.Sc. degrees, all in Electrical Engineering from the University of Birmingham, England. Currently, he is the University Professor at McMaster University, Hamilton, ON, Canada. He is a pioneer in adaptive signal-processing with emphasis on applications in radar and communications, an area of research which has occupied much of his professional life. In the mid 1980s, he shifted the thrust of his research effort in the direction of Neural Computation, which was re-emerging at that time. All along, he had the vision of revisiting the fields of radar and communications from a brand new perspective. That vision became a reality in the early years of this century with the publication of two seminal journal papers: (i) Cognitive Radio: Brain-empowered Wireless communications, IEEE J. Selected Areas in Communications, Feb. 2005 (ii) Cognitive Radar: A Way of the Future, IEEE J. Signal Processing, Feb. 2006. Cognitive Radio and Cognitive Radar are two important parts of a much wider and integrative field: Cognitive Dynamic Systems, research into which has become his passion.

Prof. Haykin is a Fellow of the Royal Society of Canada. He is the recipient of the Henry Booker Medal of 2002, the Honorary Degree of Doctor of Technical Sciences from ETH Zentrum, Zurich, Switzerland, in 1999, and many other medals and prizes.