14th International Conference on Information Fusion
Chicago, Illinois, USA, July 5-8, 2011

An efficient implementation of the second order extended Kalman filter

Michael Roth
Dept. of Electrical Engineering
Linköping University
Linköping, Sweden
Email: roth@isy.liu.se

Fredrik Gustafsson
Dept. of Electrical Engineering
Linköping University
Linköping, Sweden
Email: fredrik.gustafsson@liu.se

***Abstract*—The second order extended Kalman filter (EKF2) is based on a second order Taylor expansion of a nonlinear system, in contrast to the more common (first order) extended Kalman filter (EKF1). Despite a solid theoretical ground for its approximation, it is seldom used in applications, where the EKF and the unscented Kalman filter (UKF) are the standard algorithms. One reason for this might be the requirement for analytical Jacobian and Hessian of the system equations, and the high complexity that scales with the state order n_x as n_x^5. We propose a numerical algorithm which is based on an extended set of sigma points (compared to the UKF) that needs neither Jacobian nor Hessian (or numerical approximations of these). Further, it scales as n_x^4, which is an order of magnitude better than the EKF2 algorithm presented in literature.**

I. Introduction

The functional form of the optimal nonlinear filter has been known for a long time [1], and common approaches to implementable algorithms can be divided into two broad categories:

- Algorithms that approximate the nonlinear dynamical model. The most common examples are the extended Kalman filter (EKF) [2], the unscented Kalman filter (UKF) [3], [4] and related sigma point approaches [5] such as the recent quadrature and cubature Kalman filters [6] (QKF and CKF, respectively). They all represent the posterior filtering distribution with a state estimate and associated covariance matrix.
- Algorithms that approximate the posterior distribution numerically, such as the point mass filter and particle filter [7].

The standard EKF is commonly derived from a first order Taylor expansion of the state dynamics and measurement model. There is a version of the EKF that extends this concept to include also an approximation of the second order Taylor term [1], [2], [8], [9], which has been known for several decades. We will denote the resulting algorithm EKF2, to distinguish it from the standard algorithm which we will denote EKF1. Although the EKF2 is seldom used in applications, its solid basis provided by Taylor expansions motivates a reconsideration.

EKF2 resembles UKF in that they both intend to get correct first and second order moments. However, one commonly claimed advantage that UKF indeed succeeds in this fails to a simple counter example demonstrated in Table I, taken from [10].

Table I
NONLINEAR APPROXIMATIONS OF $x^T x$ FOR $x \sim \mathcal{N}(0, I_n)$. THEORETICAL DISTRIBUTION IS $\chi^2(n)$ WITH MEAN n AND VARIANCE $2n$.

EKF1	EKF2	UT1 [3]	UT2 [4]	CKF [6], SPKF [5]	Theory
$\mathcal{N}(0,0)$	$\mathcal{N}(n,2n)$	$\mathcal{N}(n,(3-n)n)$	$\mathcal{N}(n,2n^2)$	$\mathcal{N}(n,0)$	$\mathcal{N}(n,2n)$

The example shows one nonlinear function that in theory can occur in either the dynamic model or the measurement model. Consequently, there is at least one nonlinearity where the EKF2 can be expected to outperform its competitors. However, as remarked in [10], the UKF variants perform very well for a wide range of standard sensor models. We will here as an example study a coordinated turn model, which similarly to the example $x^T x$ is (close to) bilinear.

This article is meant to provide a more detailed study of the EKF2, with three contributions. First, its derivation is in many standard references shown for additive noise only [11]. A complete derivation can be found in [12] and [13]. However, we included a brief general derivation for the sake of clarity. Second, the requirement of analytical Jacobian and Hessian may seem prohibitive in some applications, and one often claimed advantage of the UKF is that it does not require any analytical derivatives, only function evaluations are needed. We show that the EFK2 can be implemented with neither analytical nor numerical approximations of the Jacobian and Hessian, which is a novel result. Third, the computational complexity (state dimension to the power of five) is perhaps one good reason for the limited use of EKF2 in practice. The third contribution addresses this issue and provides the result that the presented tools facilitate an EKF2 with lower complexity (state dimension to the power of four), compared to the straightforward implementation using explicit Jacobian and Hessian.

The outline is as follows. Section II explains the approach, introduces the notation and derives the EKF2. Section III presents the tools that facilitate a novel derivative free EKF2 algorithm, which can be seen as a sigma point approach with an extended set of sigma points. Section IV compares the computational complexity of the different variants of EKF2.

Simulation results are given in Section V, and followed by some concluding remarks in Section VI.

II. Second Order Extended Kalman Filter

A. Nonlinear State Space Model

We will consider filtering algorithms for nonlinear state space models of the form

$$x_{k+1} = f(x_k, v_k), \tag{1a}$$
$$y_k = h(x_k, e_k). \tag{1b}$$

Here, x_k is the n_x-dimensional state vector, v_k and e_k are the process and measurement noise vectors (dimensions $n_v \times 1$ and $n_e \times 1$, respectively), and y_k is the n_y-dimensional measured output vector. State transition f and measurement h are possibly nonlinear functions. The process and measurement noise signals v_k and e_k are assumed to be zero-mean Gaussian with known covariance matrices Q_k and R_k, respectively.

B. System Approximation by Taylor Expansion

Both the nonlinear state and the measurement equation in (1) can be approximated by a Taylor expansion around a given state and a nominal (zero) noise value. However, even if derivative terms higher than order two are neglected in the expansion, the expressions are quite complex. For the sake of clarity, new variables that comprise state and noise are introduced.

First, using $\gamma_k = (x_k^T, v_k^T)^T$, (1a) can be rewritten as $x_{k+1} = f(\gamma_k)$. A Taylor expansion for each scalar element x_{k+1}^l of the state vector x_{k+1} around a known $\hat{\gamma}_k$ is then given by

$$\begin{aligned} x_{k+1}^l \approx\ & f_l(\hat{\gamma}_k) + (\nabla_\gamma f_l(\hat{\gamma}_k))^T(\gamma_k - \hat{\gamma}_k) \\ & + \frac{1}{2}(\gamma_k - \hat{\gamma}_k)^T \nabla_\gamma^2 f_l(\hat{\gamma}_k)(\gamma_k - \hat{\gamma}_k), \\ & l \in \{1, \dots, n_x\}, \end{aligned} \tag{2a}$$

where $(\nabla_\gamma f_l(\hat{\gamma}_k))^T$ is a row vector containing the partial derivatives of $f_l(\gamma_k)$ with respect to all elements of γ_k, evaluated at $\hat{\gamma}_k$. Accordingly, $\nabla_\gamma^2 f_l(\hat{\gamma}_k)$ is a Hessian matrix containing second order partial derivatives.

In a similar way, (1b) can be rewritten using $\delta_k = (x_k^T, e_k^T)^T$ and approximated about some given $\hat{\delta}_k$. One component of y_k is given by

$$\begin{aligned} y_k^m \approx\ & h_m(\hat{\delta}_k) + (\nabla_\delta h_m(\hat{\delta}_k))^T(\delta_k - \hat{\delta}_k) \\ & + \frac{1}{2}(\delta_k - \hat{\delta}_k)^T \nabla_\delta^2 h_m(\hat{\delta}_k)(\delta_k - \hat{\delta}_k), \\ & m \in \{1, \dots, n_y\}. \end{aligned} \tag{2b}$$

The presented first and second order derivatives, for instance $(\nabla_\gamma f_l(\hat{\gamma}_k))^T$ or $\nabla_\delta^2 h_m(\hat{\delta}_k)$, are concatenated vectors and block matrices of derivatives w.r.t. x, and v or e. Also, mixed second order derivatives occur in the Hessians. Stacking for instance all n_y rows $(\nabla_x h_m(\hat{\delta}_k))^T$ yields the Jacobian matrix $(\nabla_x h(\hat{\delta}_k))^T$ which is used in the Kalman gain of the following subsection.

A large subclass of nonlinear state and measurement equations, namely bilinear nonlinearities, can be fully represented by a second order Taylor expansion. Equation (2a), for instance, is exact for a bilinear f.

C. EKF2 Recursions

Given an initial state x_0 and its covariance matrix P_0, the filtering algorithm consists of recursively calling time and measurement updates.

1) Time Update: A predicted estimate $\hat{x}_{k+1|k}^l$ of x_{k+1}^l can be governed by taking the expectation of (2a) given y_k:

$$\begin{aligned} \hat{x}_{k+1|k}^l = f_l(\hat{\gamma}_{k|k}) & + \frac{1}{2}\operatorname{tr}\Big(\nabla_x^2 f_l(\hat{\gamma}_{k|k}) P_{k|k}\Big) \\ & + \frac{1}{2}\operatorname{tr}\Big(\nabla_v^2 f_l(\hat{\gamma}_{k|k}) Q_k\Big). \end{aligned} \tag{3}$$

It is assumed that $\hat{\gamma}_{k|k} = (\hat{x}_{k|k}^T, 0^T)^T = \mathrm{E}((x_k^T, v_k^T)^T | y_k)$. Consequently, odd terms in γ_k vanish when performing the expectation. Furthermore, we do not consider correlation between v_k and x_k which can be expressed by $\operatorname{cov}(\gamma_k | y_k) = \operatorname{blkdiag}(P_{k|k}, Q_k)$, so that no mixed second order derivatives occur in (3). See [12] and [13] for a treatment of such cases. The predicted covariance matrix $P_{k+1|k} = \operatorname{cov}(x_{k+1}|y_k)$ is given by its elements in row l and column m

$$\begin{aligned} P_{k+1|k}^{lm} = & (\nabla_x f_l(\hat{\gamma}_{k|k}))^T P_{k|k} \nabla_x f_m(\hat{\gamma}_{k|k}) \\ & + (\nabla_v f_l(\hat{\gamma}_{k|k}))^T Q_k \nabla_v f_m(\hat{\gamma}_{k|k}) \\ & + \frac{1}{2}\operatorname{tr}\Big(\nabla_x^2 f_l(\hat{\gamma}_{k|k}) P_{k|k} \nabla_x^2 f_m(\hat{\gamma}_{k|k}) P_{k|k}\Big) \\ & + \frac{1}{2}\operatorname{tr}\Big(\nabla_v^2 f_l(\hat{\gamma}_{k|k}) Q_k \nabla_v^2 f_m(\hat{\gamma}_{k|k}) Q_k\Big) \end{aligned} \tag{4}$$

where $l, m \in \{1, \dots, n_x\}$. The covariance computation is straightforward but requires a Gaussian assumption on x_k.

2) Measurement Update: Performing the expectation of (2b) about a state prediction yields an output estimate (or its components)

$$\begin{aligned} \hat{y}_{k|k-1}^m = h_m(\hat{\delta}_{k|k-1}) & + \frac{1}{2}\operatorname{tr}\Big(\nabla_x^2 h_m(\hat{\delta}_{k|k-1}) P_{k|k-1}\Big) \\ & + \frac{1}{2}\operatorname{tr}\Big(\nabla_e^2 h_m(\hat{\delta}_{k|k-1}) R_k\Big). \end{aligned} \tag{5}$$

The output covariance matrix $S_k = \operatorname{cov}(y_k | y_{k-1})$ is given by its elements ($l, m \in \{1, \dots, n_y\}$):

$$\begin{aligned} S_k^{lm} = & (\nabla_x h_l(\hat{\delta}_{k|k-1}))^T P_{k|k-1} \nabla_x h_m(\hat{\delta}_{k|k-1}) \\ & + (\nabla_e h_l(\hat{\delta}_{k|k-1}))^T R_k \nabla_e h_m(\hat{\delta}_{k|k-1}) \\ & + \frac{1}{2}\operatorname{tr}\Big(\nabla_x^2 h_l(\hat{\delta}_{k|k-1}) P_{k|k-1} \nabla_x^2 h_m(\hat{\delta}_{k|k-1}) P_{k|k-1}\Big) \\ & + \frac{1}{2}\operatorname{tr}\Big(\nabla_e^2 h_l(\hat{\delta}_{k|k-1}) R_k \nabla_e^2 h_m(\hat{\delta}_{k|k-1}) R_k\Big). \end{aligned} \tag{6}$$

Again we have assumed uncorrelated noise: $\mathrm{cov}(\delta_k|y_{k-1}) = \mathrm{blkdiag}(P_{k|k-1}, R_k)$. Next, the Kalman gain can be obtained under the assumption that x_k and y_k are jointly Gaussian distributed according to

$$\begin{pmatrix} x_k \\ y_k \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \hat{x}_{k|k-1} \\ \hat{y}_{k|k-1} \end{pmatrix}, \begin{pmatrix} P_{k|k-1} & M_k \\ M_k^T & S_k \end{pmatrix} \right) \tag{7}$$

with cross terms given by $M_k = \mathrm{cov}(x_k, y_k|y_{k-1}) = P_{k|k-1}\nabla_x h(\hat{\delta}_{k|k-1})$. Using a well known Lemma from optimal estimation theory, described in e.g. [9], the posterior estimate can be derived as conditional expectation of x_k given y_k:

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + M_k S_k^{-1}(y_k - \hat{y}_{k|k-1}). \tag{8}$$

The iteration is concluded by updating the covariance matrix

$$P_{k|k} = P_{k|k-1} - M_k S_k^{-1} M_k^T. \tag{9}$$

III. Sigma Point Approximation of Derivative Terms in the EKF2

In order lessen the computational burden of the EKF2, we will now introduce methods for approximating derivative terms in the EKF2 recursion.

We consider an arbitrary nonlinear vector valued function $g(x)$ with n_z scalar components $g_l(x)$. For notational simplicity let $\mathcal{J}_l = (\nabla_x g_l(x))^T$ be a row vector containing the partial derivatives of $g_l(x)$ with respect to all n_x components of x, and $\mathcal{H}_l = \nabla_x^2 g_l(x)$ the $n_x \times n_x$ Hessian of $g_l(x)$. Stacking all $\mathcal{J}_l$ yields the Jacobian matrix $\mathcal{J}$.

Let $\hat{x} = \mathrm{E}(x)$ and $\mathcal{P} = \mathrm{cov}(x)$. Using the singular value decomposition, the symmetric $n_x \times n_x$ covariance matrix can be written as

$$\mathcal{P} = \mathcal{U}\mathcal{S}\mathcal{U}^T = \sum_{i=1}^{n_x} s_i u_i u_i^T. \tag{10}$$

Here, s_i is the i-th singular value of $\mathcal{P}$ and u_i the i-th column of the unitary matrix $\mathcal{U}$.

Next, a set of sigma points around $\hat{x}$, as used in the unscented transformation [3], can be chosen systematically. With hindsight that the conventional set comprising $2n_x+1$ vectors in x will not suffice for our purpose, the set is extended by adding $n_x^2 - n_x$ distinct sigma points. The set members are constructed according to:

$$x^{(0)} = \hat{x}, \tag{11a}$$

$$x^{(\pm i)} = \hat{x} \pm \alpha\sqrt{n_x s_i} u_i, \quad i = 1, \ldots, n_x, \tag{11b}$$

$$x^{(\pm ij)} = \hat{x} \pm \alpha\sqrt{n_x}(\sqrt{s_i}u_i + \sqrt{s_j}u_j), \quad i, j = 1, \ldots, n_x, \quad i \neq j. \tag{11c}$$

The scalar parameter α determines the spread of sigma points. Evaluating the nonlinear function $g(x)$ for each member of $\{x^{(0)}, x^{(\pm i)}, x^{(\pm ij)}\}$ yields a set of transformed sigma points denoted $\{z^{(0)}, z^{(\pm i)}, z^{(\pm ij)}\}$.

Theorem 1. *Given a function $g(x)$, the matrix $\mathcal{P}$ of (10), and evaluates of $g(x)$ at the sigma points of (11), the following expressions hold:*

$$\mathcal{J}\mathcal{P} = \lim_{\alpha\to 0} \frac{1}{2\alpha\sqrt{n}_x} \sum_{i=1}^{n_x} (z^{(i)} - z^{(-i)})\sqrt{s}_i u_i^T, \tag{12a}$$

$$\mathcal{J}\mathcal{P}\mathcal{J}^T = \lim_{\alpha\to 0} \frac{1}{4\alpha^2 n_x} \sum_{i=1}^{n_x} (z^{(i)} - z^{(-i)})(z^{(i)} - z^{(-i)}), \tag{12b}$$

$$\mathrm{tr}(\mathcal{H}_l\mathcal{P}) = \lim_{\alpha\to 0} \frac{1}{\alpha^2 n_x} \sum_{i=1}^{n_x} \left(z_l^{(i)} + z_l^{(-i)} - 2z_l^{(0)} \right), \tag{12c}$$

$$\mathrm{tr}(\mathcal{H}_l\mathcal{P}\mathcal{H}_m\mathcal{P}) = \lim_{\alpha\to 0} \frac{1}{\alpha^4 n_x^2} \left(\sum_{i=1}^{n_x} \left(z_l^{(i)} + z_l^{(-i)} - 2z_l^{(0)} \right) \left(z_m^{(i)} + z_m^{(-i)} - 2z_m^{(0)} \right) + \frac{1}{4} \sum_{i=1}^{n_x} \sum_{j=1, j\neq i}^{n_x} \left(z_l^{(ij)} + z_l^{(-ij)} - z_l^{(i)} - z_l^{(-i)} - z_l^{(j)} - z_l^{(-j)} + 2z_l^{(0)} \right) \left(z_m^{(ij)} + z_m^{(-ij)} - z_m^{(i)} - z_m^{(-i)} - z_m^{(j)} - z_m^{(-j)} + 2z_m^{(0)} \right) \right). \tag{12d}$$

Proof: In the following part we will omit the limit expression for α. However, it is silently assumed that α is chosen in a way such that the following Taylor expansion about the central sigma point $\hat{x}$ holds:

$$z_l^{(\pm i)} \approx z_l^{(0)} \pm \mathcal{J}_l \alpha\sqrt{n_x s_i} u_i + \frac{\alpha^2 n_x}{2} s_i u_i^T \mathcal{H}_l u_i. \tag{13}$$

a) $\mathcal{J}\mathcal{P}$: In order to arrive at the expression of (12a), subtract two transformed sigma points from (11b). Using (13) such a difference is given by

$$z_l^{(i)} - z_l^{(-i)} = 2\mathcal{J}_l \alpha\sqrt{n_x s_i} u_i. \tag{14}$$

Multiplication by $\sqrt{s_i}u_i^T$ and summation over all i differences yields

$$\sum_{i=1}^{n_x} (z_l^{(i)} - z_l^{(-i)})\sqrt{s}_i u_i^T = 2\alpha\sqrt{n}_x \mathcal{J}_l \sum_{i=1}^{n_x} s_i u_i u_i^T \tag{15}$$

which can be rearranged as

$$\mathcal{J}_l\mathcal{P} = \frac{1}{2\alpha\sqrt{n}_x} \sum_{i=1}^{n_x} (z_l^{(i)} - z_l^{(-i)})\sqrt{s}_i u_i^T. \tag{16}$$

Combining all n_z such rows yields the matrix product of (12a).

b) $\mathcal{J}\mathcal{P}\mathcal{J}^T$: Starting from the result (14), we obtain the following for the product of two sigma point differences:

$$(z_l^{(i)} - z_l^{(-i)})(z_m^{(i)} - z_m^{(-i)}) = 4\alpha^2 n_x \mathcal{J}_l s_i u_i u_i^T \mathcal{J}_m^T. \tag{17}$$

Again, summation over all i yields

$$\sum_{i=1}^{n_x}(z_l^{(i)} - z_l^{(-i)})(z_m^{(i)} - z_m^{(-i)}) = 4\alpha^2 n_x \mathcal{J}_l \mathcal{P} \mathcal{J}_m^T \tag{18}$$

which can be rearranged and evaluated for all l and m to give the result of (12b).

c) $\operatorname{tr}(\mathcal{H}_l\mathcal{P})$*:* Adding a pair $z_l^{(\pm i)}$ and subtracting the transformed central point yields just the Hessian term of the Taylor expansion (13):

$$\begin{aligned} z_l^{(i)} + z_l^{(-i)} - 2z_l^{(0)} &= \alpha^2 n_x s_i u_i^T \mathcal{H}_l u_i \\ &= \alpha^2 n_x \operatorname{tr}(\mathcal{H}_l s_i u_i u_i^T). \end{aligned} \tag{19}$$

Summation over all i and rearranging the equation gives (12c).

d) $\operatorname{tr}(\mathcal{H}_l\mathcal{P}\mathcal{H}_m\mathcal{P})$*:* As the result (12d) takes a bit more effort to compute, we will start by inspecting the quantity we wish to approximate first. Using the singular value decomposition the trace term of interest can be written as

$$\operatorname{tr}(\mathcal{H}_l\mathcal{P}\mathcal{H}_m\mathcal{P}) = \operatorname{tr}\Big(\mathcal{H}_l\Big(\sum_{i=1}^{n_x} s_i u_i u_i^T\Big)\mathcal{H}_m\Big(\sum_{j=1}^{n_x} s_j u_j u_j^T\Big)\Big) \tag{20}$$

which is a sum of n_x^2 terms. Having a closer look at one of the summands

$$\operatorname{tr}\Big(\mathcal{H}_l s_i u_i u_i^T \mathcal{H}_m s_j u_j u_j^T\Big) \tag{21}$$

reveals that the conventional sigma points from (11a-11b), as used in the unscented transformation, do not suffice to approximate all n_x^2 summands. However, n_x terms for $i = j$ can be represented by the following product:

$$\begin{aligned} &\operatorname{tr}\Big(\mathcal{H}_l s_i u_i u_i^T \mathcal{H}_m s_i u_i u_i^T\Big) \\ &\quad = \frac{1}{\alpha^4 n_x^2}\Big(z_l^{(i)} + z_l^{(-i)} - 2z_l^{(0)}\Big)\Big(z_m^{(i)} + z_m^{(-i)} - 2z_m^{(0)}\Big). \end{aligned} \tag{22}$$

The remaining terms of (20) cannot be approximated by direct multiplication of transformed sigma points that have been chosen in a clever way. Instead, we make use of a larger number of sigma points and introduce an intermediate step.

In order to keep the methodology used for deriving (12a-12c), consider the second order Taylor expansion of the transformed extended sigma points of (11c):

$$\begin{aligned} z_l^{(\pm ij)} &= z_l^{(0)} \pm \alpha\sqrt{n_x}\mathcal{J}_l(\sqrt{s_i}u_i + \sqrt{s_j}u_j) \\ &\quad + \frac{\alpha^2 n_x}{2}(\sqrt{s_i}u_i + \sqrt{s_j}u_j)^T \mathcal{H}_l(\sqrt{s_i}u_i + \sqrt{s_j}u_j). \end{aligned} \tag{23}$$

Again, adding a pair of $z_l^{(\pm ij)}$ yields just the Hessian term that takes the form

$$\begin{aligned} &z_l^{(ij)} + z_l^{(-ij)} - 2z_l^{(0)} \\ &\quad = \alpha^2 n_x(\sqrt{s_i}u_i + \sqrt{s_j}u_j)^T \mathcal{H}_l(\sqrt{s_i}u_i + \sqrt{s_j}u_j) \\ &\quad = \alpha^2 n_x(s_i u_i^T \mathcal{H}_l u_i + s_j u_j^T \mathcal{H}_l u_j + \underline{2\sqrt{s_i s_j} u_i^T \mathcal{H}_l u_j}). \end{aligned} \tag{24}$$

The underlined term is a desired intermediate result that is needed for further computations whereas the first two terms have been computed in (19). Thus, subtracting the known parts from (24), we can isolate the desired summand:

$$\begin{aligned} &2\alpha^2 n_x \sqrt{s_i s_j} u_i^T \mathcal{H}_l u_j \\ &\quad = z_l^{(ij)} + z_l^{(-ij)} - 2z_l^{(0)} \\ &\qquad - (z_l^{(i)} + z_l^{(-i)} - 2z_l^{(0)}) - (z_l^{(j)} + z_l^{(-j)} - 2z_l^{(0)}) \\ &\quad = z_l^{(ij)} + z_l^{(-ij)} - z_l^{(i)} - z_l^{(-i)} - z_l^{(j)} - z_l^{(-j)} + 2z_l^{(0)}. \end{aligned} \tag{25}$$

Next, multiplication by a similar intermediate result of $g_m(x)$ and taking the trace of the scalar result provides the unknown summands (for $i \neq j$) of (20):

$$\begin{aligned} &4\alpha^4 n_x^2 \operatorname{tr}\Big(\mathcal{H}_l s_i u_i u_i^T \mathcal{H}_m s_j u_j u_j^T\Big) \\ &= \Big(z_l^{(ij)} + z_l^{(-ij)} - z_l^{(i)} - z_l^{(-i)} - z_l^{(j)} - z_l^{(-j)} + 2z_l^{(0)}\Big) \\ &\quad \Big(z_m^{(ij)} + z_m^{(-ij)} - z_m^{(i)} - z_m^{(-i)} - z_m^{(j)} - z_m^{(-j)} + 2z_m^{(0)}\Big). \end{aligned} \tag{26}$$

This concludes the proof. ∎

Using Theorem 1, several terms in the EKF2 recursions can be replaced. More precisely, all products of Jacobian/Hessian and covariance matrices can be replaced by sums of transformed sigma points. The presented method thus provides a novel derivative free EKF2 algorithm.

IV. Computational Complexity

Theorem 1 facilitates coding derivative free filters and reduces the number of floating point operations by replacing costly matrix multiplications. The following section describes the inherent gain in computational complexity and compares it to an EKF2 implementation that is based on numerical derivatives.

Most critical, with regard to computation, are terms of the form $\operatorname{tr}(\mathcal{H}_l\mathcal{P}\mathcal{H}_m\mathcal{P})$ which occur in the covariance updates (4) and (6). For explanatory reasons we will focus on the state covariance prediction $P_{k+1|k}$ and, without loss of generality, assume that the state dimension is larger than the process noise dimension: $n_x > n_v$.

A. Numerical Derivative EKF2

For each scalar component of $f(x_k, v_k)$, a Hessian w.r.t. x must be computed. In total, n_x matrices with n_x^2 entries must be stored and hence the memory requirements are of

Table II
SIMULATION PARAMETERS.

$\hat{x} \sim \mathcal{N}((0,0,0,0,0)^T, \mathrm{diag}(5000^2, 5000^2, 0.01, 0.01, 0.01))$
$p \sim \mathcal{N}((0,0,0,0,0)^T, I_5)$

$\mathcal{O}(n_x^3)$. Furthermore, each Hessian requires the evaluation of $4n_x^2$ points when using a central difference scheme. The subsequent matrix multiplications ($\mathcal{H}_l\mathcal{P}\mathcal{H}_m\mathcal{P}$) require $\mathcal{O}(n_x^3)$ floating point operations each that have to be carried out for $(n_x^2 - n_x)/2$ entries of $P_{k|k-1}$ (exploiting symmetry) which leads to a flop count of $\mathcal{O}(n_x^5)$ in total.

B. Sigma Point EKF2

An extended set of sigma points needs to be evaluated and stored for each component of f. This implies $\mathcal{O}(n_x^2)$ function calls of f and storage requirements of $\mathcal{O}(n_x^3)$. The following summations are of $\mathcal{O}(n_x^2)$ and must be carried out for each matrix element of $P_{k|k-1}$. Thus, the overall flop count is of $\mathcal{O}(n_x^4)$ — one order of magnitude better than its numerical derivative counterpart.

V. SIMULATION STUDY

The following examples primarily aim at showing the validity of the approximations of Theorem 1. For a practical application of EKF2 the reader is referred to [14] where its use proves benefitial in IMU strapdown integration: an arising nonlinear acceleration uncertainty is a result of an uncertain rotation from sensor mounted to global coordinate frame. Here EKF2 provides accurate mean and covariance estimates of the uncertain acceleration whereas EKF1 severely underestimates the covariance. Related work shows that the UKF instead overestimates the covariance.

A. Example 1

We consider a motion model that describes a coordinated turn in two dimensions, found in e.g. [15]. The first two states contain the position in Cartesian coordinates, the third and fourth state are Cartesian velocities, and the fifth state is the angular velocity of the object of interest on its circular path. The continuous time equations

$$\dot{x}_1(t) = x_3(t) \tag{27a}$$
$$\dot{x}_2(t) = x_4(t) \tag{27b}$$
$$\dot{x}_3(t) = -x_5(t)x_3(t) \tag{27c}$$
$$\dot{x}_4(t) = x_5(t)x_4(t) \tag{27d}$$
$$\dot{x}_5(t) = 0 \tag{27e}$$

reveal that the state transition $\dot{x}(t) = f(x(t))$ is bilinear, i.e. its first order partial derivatives w.r.t. the components of $x(t)$ are linear functions of the state $x(t)$.

However, the system must be discretized in order to use it in the discrete time filtering framework of Section II. An exact zero order hold sampled model ($x_k = x(kT)$) of (27) is given by

$$x_{k+1}^1 = x_k^1 + \frac{x_k^3}{x_k^5}\sin(x_k^5 T) - \frac{x_k^4}{x_k^5}(1 - \cos(x_k^5 T)) \tag{28a}$$
$$x_{k+1}^2 = x_k^2 + \frac{x_k^3}{x_k^5}(1 - \cos(x_k^5 T)) + \frac{x_k^4}{x_k^5}\sin(x_k^5 T) \tag{28b}$$
$$x_{k+1}^3 = x_k^3\cos(x_k^5 T) - x_k^4\sin(x_k^5 T) \tag{28c}$$
$$x_{k+1}^4 = x_k^3\sin(x_k^5 T) + x_k^4\cos(x_k^5 T) \tag{28d}$$
$$x_{k+1}^5 = x_k^5 \tag{28e}$$

and can be written compactly in the form of (1a) as $x_{k+1} = f(x_k)$. It is assumed that the sampling interval T is chosen small so that the bilinear characteristics are preserved during discretization. Using symbolic software tools, the Jacobian $\mathcal{J}$ and all Hessians $\mathcal{H}_l$ of (28) are available for evaluation purposes. Please note that a process noise input in (27) and (28) has been omitted for this demonstration.

Next, we will show the approximation quality of the terms in Theorem 1 for the function (28), random points $\hat{x}$, and random covariance matrices $\mathcal{P}$ with selected matrix norm. Monte Carlo simulations are run for the settings that are displayed in Table II. Here, the random vector p can be used to construct a positive definite $\mathcal{P}$ with a chosen matrix norm according to

$$\mathcal{P} = \frac{\|\mathcal{P}\|_2}{\|pp^T\|_2} pp^T. \tag{29}$$

The analytical expressions for the Jacobian products $\mathcal{J}\mathcal{P}$, $\mathcal{J}\mathcal{P}\mathcal{J}^T$, and Hessian products $\mathrm{tr}(\mathcal{H}_l\mathcal{P})$ and $\mathrm{tr}(\mathcal{H}_l\mathcal{P}\mathcal{H}_m\mathcal{P})$ are computed for all $l, m \in \{1, \ldots, 5\}$ and combined into vectors and matrices, similar to their use in the EKF2. Likewise, the approximated expressions are calculated using Theorem 1. Subsequently calculated error differences between the analytical products and their approximations can be assessed by interpreting their 2-norms.

Resulting error norms as a function of $\|\mathcal{P}\|_2$, averaged over 100 Monte Carlo, are shown in Figure 1. The simulation outcome demonstrates the dependency on the size of $\mathcal{P}$. Although the error norms increase for larger gains of $\mathcal{P}$, the error stays within viable bounds for a prospective filtering application.

B. Example 2

An extension to Example 1 demonstrates the performance of the approximations in an actual filtering context. We now consider a motion model of the form (28), but with a white noise input added to equation (28e) such that

$$x_{k+1}^5 = x_k^5 + v_k. \tag{30}$$

The aim is to assess the quality of a time update using the sigma point approximations of Section III in the EKF2 algorithm of Section II. Therefore, randomly created states x_k

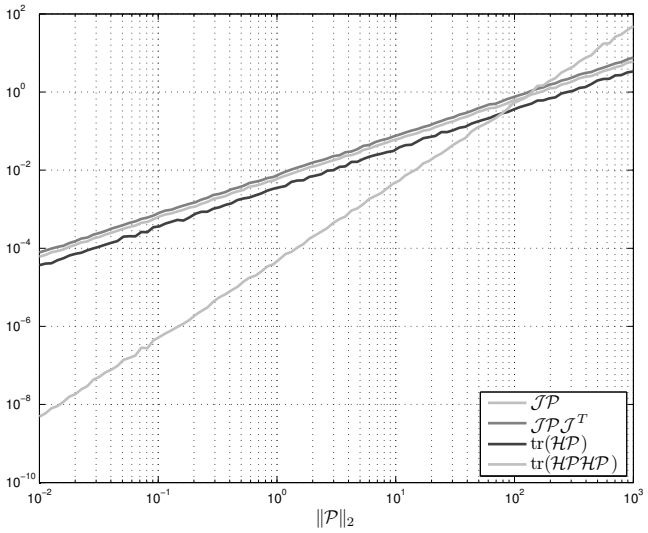

Figure 1. Error norms of the differences between analytical expressions and sigma point approximations of derivative terms for a coordinated turn state transition in Example 1.

Table III
SIMULATION RESULTS, EXAMPLE 2.

$\|x_{k+1}^{MC} - \hat{x}_{k+1\|k}\|_2 / \|x_{k+1}^{MC}\|_2$	$2.9 \cdot 10^{-5}$
$\|P_{k+1}^{MC} - P_{k+1\|k}\|_2 / \|P_{k+1}^{MC}\|_2$	$1.7 \cdot 10^{-2}$

and random unit gain covariance matrices P_k are propagated through the system (28a-28d, 30).

"True" propagated values x_{k+1}^{MC} and P_{k+1}^{MC} are obtained using a simulation based Monte Carlo transformation [10] that involves 10000 calls of the system (28a-28d, 30) with different state $x_k^{MC} \sim \mathcal{N}(x_k, P_k)$ and process noise realizations $v_k^{MC} \sim \mathcal{N}(0, Q_k)$ per state transition.

A predicted estimate $\hat{x}_{k+1|k}$ and its covariance $P_{k+1|k}$ are calculated using the time update equations of Section II with the approximations of Section III.

Subsequently, one step prediction errors can be computed by subtracting "true" and predicted state and covariance, respectively. Here, we choose to display relative errors that are scaled by $\|x_{k+1}^{MC}\|_2$ and $\|P_{k+1}^{MC}\|_2$. Table III shows the definitions and experimental results for the error terms, averaged over 10000 different experiments with random x_k, P_k, and a fixed $Q_k = 0.02$. The order of magnitude of the shown errors encourage further applications of the novel EKF2.

VI. CONCLUSION

We consider the second order extended Kalman filter (EKF2) as a theoretically sound alternative to other similar algorithms such as the unscented Kalman filter (UKF) and its variants based on sigma points. We have been able to formulate EKF2 in a sigma point framework, where the main advantages are first that no analytical or numerical explicit expressions of the Jacobian and Hessian are required. Only function evaluations are needed. Here, the standard set of sigma points used in all sigma point filters has to be extended to get correct first and second order moments in the case of transforming a Gaussian vector. We have shown that the computational complexity decreases from n_x^5 for the straightforward algorithm known in literature to n_x^4. It is still an order of magnitude larger than UKF. Its main potential should be for the class of nonlinear dynamic models *or* nonlinear measurement models that are (close to) bilinear, compare to the example in Table I.

ACKNOWLEDGMENT

This work has been supported by the MC Impulse (Monte Carlo based Innovative Management and Processing for an Unrivalled Leap in Sensor Exploitation) project, a European Commission FP7 Marie Curie Initial Training Network.

REFERENCES

[1] A. H. Jazwinski, *Stochastic Processes and Filtering Theory*. Academic Press, 1970.

[2] M. Athans, R. Wishner, and A. Bertolini, "Suboptimal state estimation for continuous-time nonlinear systems from discrete noisy measurements," *Automatic Control, IEEE Transactions on*, vol. 13, no. 5, pp. 504–514, 1968.

[3] S. Julier, J. Uhlmann, and H. Durrant-Whyte, "A new approach for filtering nonlinear systems," in *American Control Conference, 1995. Proceedings of the*, vol. 3, 1995, pp. 1628–1632 vol.3.

[4] E. Wan and R. van der Merwe, "The unscented Kalman filter for nonlinear estimation," in *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000*, 2000, pp. 153–158.

[5] B. M. Quine, "A derivative-free implementation of the extended Kalman filter," *Automatica*, vol. 42, no. 11, pp. 1927–1934, 2006.

[6] I. Arasaratnam and S. Haykin, "Cubature Kalman filters," *Automatic Control, IEEE Transactions on*, vol. 54, no. 6, pp. 1254–1269, 2009.

[7] F. Gustafsson, "Particle filter theory and practice with positioning applications," *Aerospace and Electronic Systems Magazine, IEEE*, vol. 25, no. 7, pp. 53–82, 2010.

[8] A. Gelb, *Applied Optimal Estimation*. The MIT Press, 1974.

[9] B. D. O. Anderson and J. B. Moore, *Optimal Filtering*. Prentice Hall, 1979.

[10] F. Gustafsson and G. Hendeby, "Some relations between extended and unscented Kalman filters," *Signal Processing, IEEE Transactions on*, 2011, submitted.

[11] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*, 1st ed. Wiley-Interscience, 2001.

[12] H. Tanizaki, *Nonlinear Filters: Estimation and Applications*, 2nd ed. Springer, 1996.

[13] G. Hendeby, "Performance and implementaion aspects of nonlinear filtering," Linköping Studies in Science and Technology. Dissertations. No. 1161, 2008.

[14] M. Zhang, J. D. Hol, H. Luinge, and L. Slot, "Second order nonlinear uncertainty modeling in strapdown integration using MEMS IMU," in *14th International Conference on Information Fusion (Fusion 2011)*, Chicago, USA, 2011.

[15] F. Gustafsson, *Statistical Sensor Fusion*. Studentlitteratur AB, Mar. 2010.