

 Open access • Journal Article • DOI:10.1109/TSP.2011.2172431

Some Relations Between Extended and Unscented Kalman Filters — [Source link](#)

Fredrik Gustafsson, Gustaf Hendeby





Institutions: Linköping University

Published on: 01 Feb 2012 - IEEE Transactions on Signal Processing (IEEE Signal Processing Society)

Topics: Unscented transform, Extended Kalman filter, Invariant extended Kalman filter, Kalman filter and Riccati equation

Related papers:

- [Unscented filtering and nonlinear estimation](#)
- [Cubature Kalman Filters](#)
- [A new method for the nonlinear transformation of means and covariances in filters and estimators](#)
- [A New Approach to Linear Filtering and Prediction Problems](#)
- [The unscented Kalman filter for nonlinear estimation](#)

Share this paper:    

View more about this paper here: <https://typeset.io/papers/some-relations-between-extended-and-unscented-kalman-filters-4h4nmtr34y>

Some Relations Between Extended and Unscented Kalman Filters

Fredrik Gustafsson and Gustaf Hendeby

Linköping University Post Print

N.B.: When citing this work, cite the original article.

©2012 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

Fredrik Gustafsson and Gustaf Hendeby, Some Relations Between Extended and Unscented Kalman Filters, 2012, IEEE Transactions on Signal Processing, (60), 2, 545-555.

<http://dx.doi.org/10.1109/TSP.2011.2172431>

Postprint available at: Linköping University Electronic Press

<http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-75272>

Some relations between extended and unscented Kalman filters

Fredrik Gustafsson, *Senior Member IEEE* and Gustaf Hendeby, *Member IEEE*

Abstract—The unscented Kalman filter (UKF) has become a popular alternative to the extended Kalman filter (EKF) during the last decade. UKF propagates the so called sigma points by function evaluations using the unscented transformation (UT), and this is at first glance very different from the standard EKF algorithm which is based on a linearized model. The claimed advantages with UKF are that it propagates the first two moments of the posterior distribution and that it does not require gradients of the system model. We point out several less known links between EKF and UKF in terms of two conceptually different implementations of the Kalman filter: the standard one based on the discrete Riccati equation, and one based on a formula on conditional expectations that does not involve an explicit Riccati equation. First, it is shown that the sigma point function evaluations can be used in the classical EKF rather than an explicitly linearized model. Second, a less cited version of the EKF based on a second order Taylor expansion is shown to be quite closely related to UKF. The different algorithms and results are illustrated with examples inspired by core observation models in target tracking and sensor network applications.

Index Terms—extended Kalman filter, unscented Kalman filter, transformations

I. INTRODUCTION

This contribution compares various approaches for how to propagate a Gaussian approximate state distribution for a nonlinear system

$$x_{k+1} = f(x_k, u_k) + v_k, \quad (1a)$$

$$y_k = h(x_k, u_k) + e_k. \quad (1b)$$

The nonlinear filters in this study are in one way or another related to the Taylor expansion of a nonlinear function $z = g(x)$ around an estimate \hat{x} ,

$$z = g(x) = g(\hat{x}) + g'(\hat{x})(x - \hat{x}) + \underbrace{\frac{1}{2}(x - \hat{x})^T g''(\xi(x))(x - \hat{x})}_{r(x; \hat{x}, g''(\xi(x))),} \quad (2)$$

where $x \in \mathbb{R}^{n_x}$ and (initially for notational convenience) $z \in \mathbb{R}^1$. Here, g' denotes the Jacobian and g'' the Hessian of the function $g(x)$, defined in the appendix, and $\xi(x)$ is a point in the neighborhood of \hat{x} . The equality holds for a $\xi(x)$ in a

Copyright (c) 2011 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org.

F. Gustafsson is with the Division of Automatic Control, Dept. of Electrical Engineering, Linköping University, SE-58183 Linköping, Sweden (e-mail: fredrik@isy.liu.se). G. Hendeby is with the Competence Unit Informatics, Division of Information Systems at the Swedish Defense Research Agency (FOI) in Linköping, Sweden (e-mail: gustaf.hendeby@foi.se).

This work was supported by the Swedish research council (VR).

neighborhood of \hat{x} if a convergent Taylor series exists for g in the region, and is otherwise just an approximation. Basically, as an overview, the following algorithms apply:

- The *extended Kalman filter* (EKF) [1], [2] is based on the first two terms in (2). This works fine as long as the rest term is small. Small here relates both to the state estimation error and the degree of nonlinearity of g . Basically, as a rule of thumb, the rest term is negligible if either the model is almost affine, or the SNR is high, in which case the estimation error can be considered sufficiently small.
- The second order compensated EKF [3], [4], [5] approximates the rest term $r(x; \hat{x}, g''(\xi))$ with $r(x; \hat{x}, g''(\hat{x}))$, and compensates for the mean and variance of this term.
- The *unscented Kalman filter* (UKF) [6], [7] can be interpreted, as will be demonstrated, as implicitly estimating the first terms (but not the Jacobian and Hessian themselves) in the nonlinear transformation in (2).

The standard forms of the KF and EKF include a discrete-time algebraic Riccati equation (DARE) for propagating the state covariance, while the UKF in its proposed form is based on a different principle in linear estimation and has no explicit DARE. Further, UKF is based on function evaluations of $g(x)$ only, so neither the Jacobian nor the Hessian are needed. This is a first claimed advantage of the UKF:

[7]: “...UT is not the same as using a central difference scheme to calculate the Jacobian.”

This is indeed true, but, as we will point out, there is a duality in the implementation. UKF can be implemented with Riccati equations, and the EKF (and even the linear KF) can be implemented without Riccati equations using only function evaluations.

The core tool in the analytic results is the underlying transform approximations of a nonlinear mapping $z = g(x)$, providing a Gaussian approximation $\mathcal{N}(\mu_z, P_z)$ of the stochastic variable z . It is often stated that the unscented transform (UT) gives the correct first and second order moments ($\mu_z = \mathbb{E}(z)$ and $P_z = \text{COV}(z)$):

[7]: “Any set of sigma points that encodes the mean and covariance correctly, . . . , calculates the projected mean and covariance correctly to the second order”

However, we show with a simple counter-example that this is not the case, even for a quadratic function of Gaussian variables. We also show analytically that the UT generally does not give the same elements in the covariance as the second order Taylor expansion, which should at least be exact for quadratic functions of Gaussian variables. On the

other hand, we show that the UT gives a good approximation of many common sensor models in tracking and navigation applications.

The two quotes from [7] are actually cited almost literally in a large number of papers on UKF, so this gives a strong motivation for revisiting and clarifying the various links described in this contribution. The outline is as follows. The transformations, the basic relations and some numerical illustrations are given in Section II. Section III discusses the classical implementation of EKF, and shows how the sigma points of the UT can be used to estimate derivatives such that the need for Jacobians and Hessians is eliminated. Section IV gives a general version of the Riccati-free nonlinear filter, where the transform approximation can be chosen individually for the time and measurement update, respectively. Section V concludes the paper.

II. NONLINEAR TRANSFORMATIONS

This section summarizes different methods to approximate the distribution of a nonlinear mapping $z = g(x)$ of a Gaussian variable x with a Gaussian distribution,

$$x \sim \mathcal{N}(\mu_x, P_x) \rightarrow z \overset{\text{approx}}{\sim} \mathcal{N}(\mu_z, P_z). \quad (3)$$

The following subsections describe different approaches to approximate μ_z and P_z . The symbol $\overset{\text{approx}}{\sim}$ is here used to indicate a distribution approximation. The underlying idea is that it is often easier to approximate a distribution than a general nonlinear function, so the analytic distribution of $g(x)$ will not be considered here.

A. Taylor Transformations

Consider a general nonlinear transformation and its second order Taylor expansion

$$z = g(x) = g(\mu_x) + g'(\mu_x)(x - \mu_x) + \underbrace{\left[\frac{1}{2}(x - \mu_x)^T g''(\xi)(x - \mu_x) \right]}_{r(x; \mu_x, g''(\xi))}_i, \quad (4)$$

where n_x is the dimension of the vector $x \in \mathbb{R}^{n_x}$, and $z \in \mathbb{R}^{n_z}$. The Hessian of the i^{th} component of g is denoted g''_i , with $i = 1, \dots, n_z$. The notation $[v_i]_i$ is used to denote a vector in which element i is v_i . Analogously, the notation $[m_{ij}]_{ij}$ will be used to denote the matrix where the (i, j) element is m_{ij} . Theorem 1 gives the theoretical mean and covariance of (4) when ξ is substituted with μ_x . The equality holds for a $\xi(x)$ in a neighborhood of \hat{x} if a convergent Taylor series exists for g in the region, and is otherwise just an approximation.

Theorem 1 (First moments of the Taylor transformation)
Consider the mapping

$$z = g(\mu_x) + g'(\mu_x)(x - \mu_x) + \left[\frac{1}{2}(x - \mu_x)^T g''_i(\mu_x)(x - \mu_x) \right]_i, \quad (5a)$$

from \mathbb{R}^{n_x} to \mathbb{R}^{n_z} . Let $E(x) = \mu_x$ and $\text{COV}(x) = P$, then the first moment of z is given by

$$\mu_z = g(\mu_x) + \frac{1}{2}[\text{tr}(g''_i(\mu_x)P)]_i. \quad (5b)$$

Further, let $x \sim \mathcal{N}(\mu_x, P)$, then the second moment of z is given by

$$P_z = g'(\mu_x)P(g'(\mu_x))^T + \frac{1}{2} \left[\text{tr}(g''_i(\mu_x)P g''_j(\mu_x)P) \right]_{ij}, \quad (5c)$$

with $i, j = 1, \dots, n_z$.

The result is in for instance [5] given without proof, so we include it here for completeness and for preparing for Theorem 2.

Proof: Suppose without loss of generality that $\mu_x = E[x] = 0$ and $\text{COV}(x) = P$. Further, to simplify notation, let $G = g''(\mu_x)$. Then, one direct way to express the expected value of the rest term is to use the trace linearity property and $\text{tr}(AB) = \text{tr}(BA)$,

$$E[x^T G x] = E[\text{tr}(G x x^T)] = \text{tr}(G E[x x^T]) = \text{tr}(G P). \quad (6)$$

The variance is more complicated to compute, and the Gaussian assumption is needed. Below, a derivation of both mean and variance is provided.

First, let $q = G^{1/2}x$, where $G = G^{T/2}G^{1/2}$, so that $\text{COV}(q) = G^{1/2}P G^{T/2}$. Then,

$$x^T G x = q^T q. \quad (7)$$

The *singular value decomposition* (SVD) $G^{1/2}P G^{T/2} = U \Sigma U^T$, where the diagonal elements of Σ are $\sigma_1^2, \dots, \sigma_{n_x}^2$ (the variance of the noise in the direction of the respective eigen vectors), gives a second transformation $v = U^T q$ which does not change the eigenvalues of the covariance, and in particular its trace is the same, since $\text{tr}(U \Sigma U^T) = \text{tr}(\Sigma U^T U) = \text{tr}(\Sigma)$. Thus (using $E(x^4) = 3(E(x^2))^2 = 3\sigma^4$ for scalar zero mean σ^2 variance Gaussian variables),

$$E[v^T v] = \sum_{i=1}^{n_x} \sigma_i^2, \quad (8a)$$

$$E[(v^T v)^2] = \sum_{i=1}^{n_x} 3\sigma_i^4 + \sum_{i \neq j} 2\sigma_i^2 \sigma_j^2, \quad (8b)$$

$$\text{Var}[v^T v] = E[(v^T v)^2] - (E[v^T v])^2 = 2 \sum_{i=1}^{n_x} \sigma_i^4. \quad (8c)$$

Now, the sum of the diagonal elements of a matrix Σ^2 can be expressed as the trace of the square of the matrix in the SVD, so

$$\text{Var}[v^T v] = 2 \text{tr}(G^{1/2} P G^{T/2} G^{1/2} P G^{T/2}) = 2 \text{tr}(G P G P). \quad (9)$$

Further, if the function g is vector-valued, the covariance between different rows can be derived in a similar way. Let $G_i = g''_i(\mu_x)$ be the Hessian of the i^{th} row of $g(x)$. Then the result is

$$E[x^T G_i x x^T G_j x] - E[x^T G_i x] E[x^T G_j x] = 2 \text{tr}(G_i P G_j P). \quad (10)$$

In summary, the rest term for a vector valued function $g(z)$ has mean and covariance given by

$$E((x - \mu_x)^T g''(\mu_x)(x - \mu_x)) = [\text{tr}(g_i''(\mu_x)P)]_i, \quad (11a)$$

$$\text{COV}((x - \mu_x)^T g''(\mu_x)(x - \mu_x)) = [2 \text{tr}(g_i''(\mu_x)P g_j''(\mu_x)P)]_{ij}. \quad (11b)$$

This concludes the proof. \blacksquare

The following remarks are important:

- For quadratic functions $g_i(x) = a_i + B_i x + \frac{1}{2} x^T C_i x$, the Hessian $g_i''(x) = C_i$ is independent of x . That is, Theorem 1 gives the correct first and second order moments.
- For polynomial functions $g(x)$, the principle of moment matching can be applied to compute μ_z and P_z analytically, see [8]. For a Gaussian x , all moments of z can be expressed as polynomial functions of μ_x and P_x .
- The mean and covariance can also be derived from a linear regression formulation, where the sigma points become the regressors [9].
- The moment integrals

$$\mu_z = \int g(x)p(x) dx, \quad (12a)$$

$$P_z = \int g(x)g^T(x)p(x) dx, \quad (12b)$$

where $p(x)$ is the probability density function of x , can be approximated with numerical integration techniques. The Gauss-Hermite quadrature rule is examined in [10], and the cubature rule is investigated in [11]. The latter reference gives a nice link to the unscented transform which we will come back to.

To summarize the theorem, the first order Taylor approximation (TT1) can be used to form an approximate Gaussian distribution for $z = g(x)$ as

$$\text{TT1: } x \sim \mathcal{N}(\mu_x, P) \rightarrow z \overset{\text{approx}}{\sim} \mathcal{N}(g(\mu_x), g'(\mu_x)P(g'(\mu_x))^T). \quad (13)$$

Further, the second order Taylor approximation (TT2) leads to a Gaussian approximation with mean and covariance provided by the theorem as

$$\text{TT2: } x \sim \mathcal{N}(\mu_x, P) \rightarrow z \overset{\text{approx}}{\sim} \mathcal{N}\left(g(\mu_x) + \frac{1}{2}[\text{tr}(g_i''(\mu_x)P)]_i, g'(\mu_x)P(g'(\mu_x))^T + \frac{1}{2}[\text{tr}(P g_i''(\mu_x)P g_j''(\mu_x))]_{ij}\right). \quad (14)$$

It is a trivial fact that the gradient and Hessian in TT1 and TT2, respectively, can both be computed using numerical methods. It is worth stressing that both $g'_i(x)$ and $g''_i(x)$ are in all illustrations computed using numerical methods. That is, only function evaluations of the nonlinear function $g(x)$ are assumed to be available. However, as we will demonstrate in Theorem 3 and the following discussion, there is a numerical method to approximate the terms actually needed in TT2, which is one order of magnitude more efficient than approximating the Jacobian and Hessian explicitly.

B. Monte Carlo Transformation

The *Monte Carlo Transformation* (MCT) provides a general framework to compute an accurate approximation, which asymptotically should be the best possible one. The method is straightforward. First, generate a number N of random points $x^{(i)}$, let these pass the nonlinear function, and then estimate the mean and covariance as follows:

$$x^{(i)} \sim \mathcal{N}(\mu_x, P), \quad i = 1, \dots, N,$$

$$z^{(i)} = g(x^{(i)}),$$

$$\mu_z = \frac{1}{N} \sum_{i=1}^N z^{(i)},$$

$$P_z = \frac{1}{N-1} \sum_{i=1}^N (z^{(i)} - \mu_z)(z^{(i)} - \mu_z)^T.$$

The law of large numbers assures that these estimates converge to the true values, which makes the MCT well suited for validation purposes.

C. Unscented Transformation

The *unscented transform* (UT) is in a sense similar to the MCT approach in that it selects a number of points $x^{(i)}$, maps these to $z^{(i)} = g(x^{(i)})$, and then estimates the mean and covariance in the standard way. The difference lies in how the points $x^{(i)}$ are selected.

First define, u_i and σ_i from the SVD of the covariance matrix P ,

$$P = UDU^T = \sum_{i=1}^{n_x} \sigma_i^2 u_i u_i^T,$$

where $u_i = U_{:,i}$ is the i 'th column of U and $\sigma_i^2 = \Sigma_{i,i}$ is the i 'th diagonal element of Σ . Then, let

$$x^{(0)} = \mu_x, \quad x^{(\pm i)} = \mu_x \pm \sqrt{n_x + \lambda} \sigma_i u_i, \quad (15a)$$

$$\omega^{(0)} = \frac{\lambda}{n_x + \lambda}, \quad \omega^{(\pm i)} = \frac{1}{2(n_x + \lambda)}, \quad (15b)$$

where $i = 1, \dots, n_x$. Let $z^{(i)} = g(x^{(i)})$, and apply

$$\mu_z = \sum_{i=-n_x}^{n_x} \omega^{(i)} z^{(i)}, \quad (16a)$$

$$P_z = \sum_{i=-n_x}^{n_x} \omega^{(i)} (z^{(i)} - \mu_z)(z^{(i)} - \mu_z)^T + (1 - \alpha^2 + \beta)(z^{(0)} - \mu_z)(z^{(0)} - \mu_z)^T, \quad (16b)$$

where $\omega^{(0)} + (1 - \alpha^2 + \beta)$ is often denoted $\omega_c^{(0)}$ and used to make the notation more compact for the covariance matrix expression.

The design parameters of UT have here the same notation as in UKF literature (e.g., [12]):

- λ is defined by $\lambda = \alpha^2(n_x + \kappa) - n_x$.
- α controls the spread of the sigma points and is suggested to be approximately 10^{-3} .
- β compensates for the distribution, and should be chosen to $\beta = 2$ for Gaussian distributions.
- κ is usually chosen to zero.

Tab I: Different versions of the UT (counting the CT as a UT version given appropriate parameter choice) in (15) using the definition $\lambda = \alpha^2(n_x + \kappa) - n_x$.

Parameter	UT1 [6]	UT2 [12]	CT [11]	DFT [13]
α	$\sqrt{3/n_x}$	10^{-3}	1	-
β	$3/n_x - 1$	2	0	-
κ	0	0	0	-
λ	$3 - n_x$	$10^{-6}n_x - n_x$	0	0
$\sqrt{n_x + \lambda}$	$\sqrt{3}$	$10^{-3}\sqrt{n_x}$	$\sqrt{n_x}$	$\frac{1}{a}\sqrt{n_x}$
$\omega^{(0)}$	$1 - n_x/3$	-10^6	0	0

Note that $n_x + \lambda = \alpha^2 n_x$ when $\kappa = 0$, and that for $n_x + \lambda \rightarrow 0^+$ the central weight $\omega^{(0)} \rightarrow -\infty$. Furthermore, $\sum_i \omega^{(i)} = 1$. We will consider the following two versions of UT summarized in Table I, corresponding to the original one in [6] and an improved one in [12].

The *cubature transform* (CT), that is used in the *cubature Kalman filter* (CKF, [11]), is derived using different principles than the UT. However, it still fits the UT framework for a particular parameter tuning. The CT parameters are given in Table I for comparison.

The derivative-free EKF (DF-EKF) in [13] avoids the center sigma points just as the CT, but includes also an arbitrary scaling factor a to the other sigma points. For the case $a = 1$, the method coincides with CKF. Here the transformation used is denoted *derivative-free transform* (DFT).

In summary, TT1 is a computationally cheap approximation, TT2 aims at computing the correct mean and covariance by taking care of the second order term in the Taylor expansion (for functions $g(x)$ quadratic in x , TT2 is completely correct, otherwise it is often a good approximation), the MC approach is always asymptotically correct (if the moment exists), and that the UT is a fairly good compromise between TT2 and MC, that improves computational complexity to MC while being simpler to implement than TT2.

The unscented transform may have a negative weight for the center point $z^{(0)}$. This might cause problems when implementing the UKF, for instance using the square root form. On the other hand, the cubature filter described in [11] has a similar set of sigma points. The points all have positive weights, and the central point is left out.

D. Analytical Comparison of TT2 and UT

In the following theorem, the relation between TT2 and UT will be analyzed, and expressions for the resulting mean and covariance are given and interpreted in the limit as the sigma points in the UT approach the center point.

Theorem 2 (Asymptotic property of UT) Consider the mapping $z = g(x)$ from \mathbb{R}^{n_x} to \mathbb{R}^{n_z} of the stochastic variable x with mean μ_x and covariance P_x . The UT yields the following mean μ_z^{UT} and covariance P_z^{UT} asymptotically as $\sqrt{n_x + \lambda} \rightarrow 0^+$ in UT2.

$$\mu_z^{\text{UT}} = g(\mu_x) + \frac{1}{2} [\text{tr}(g_i'' P)]_i, \quad (17a)$$

$$P_z^{\text{UT}} = g'(\mu_x) P (g'(\mu_x))^T + \frac{(\beta - \alpha^2)}{4} [\text{tr}(P g_i''(\mu_x)) \text{tr}(P g_j''(\mu_x))]_{ij} \quad (17b)$$

For $n_x = 1$, equality $P_z^{\text{TT2}} = P_z^{\text{UT}}$ holds if $\beta - \alpha^2 = 2$.

Proof: Reorganizing the terms in (16) gives

$$\mu_z^{\text{UT}} = z^{(0)} + \frac{1 - \omega^{(0)}}{2n_x} \sum_{i=1}^{n_x} (z^{(i)} - 2z^{(0)} + z^{(-i)}) \quad (18a)$$

$$\begin{aligned} P_z^{\text{UT}} &= (\omega^{(0)} + (1 - \alpha^2 + \beta))(z^{(0)} - \mu_z)(\cdot)^T \\ &+ \sum_{i \neq 0} \frac{1 - \omega^{(0)}}{2n_x} (z^{(i)} - \mu_z)(\cdot)^T \\ &= (1 - \alpha^2 + \beta) \frac{(1 - \omega^{(0)})^2}{4n_x^2} \left(\sum_{i=1}^{n_x} (z^{(i)} - 2z^{(0)} + z^{(-i)}) \right) \left(\cdot \right)^T \\ &- \frac{(1 - \omega^{(0)})^2}{4n_x^2} \left(\sum_{i=1}^{n_x} (z^{(i)} - 2z^{(0)} + z^{(-i)}) \right) \left(\cdot \right)^T \\ &+ \frac{1 - \omega^{(0)}}{2n_x} \sum_{i=-n_x}^{n_x} (z^{(i)} - z^{(0)})(\cdot)^T. \end{aligned} \quad (18b)$$

With the sigma points in (15), differences can be constructed that, in the limit as $n_x + \lambda \rightarrow 0^+$ (i.e., $\alpha \rightarrow 0^+$ with $\kappa = 0$), yield the derivatives:

$$\frac{z^{(i)} - z^{(0)}}{\sigma_i \sqrt{n_x + \lambda}} \rightarrow g'(\mu_x) u_i \quad (19)$$

$$\frac{z^{(i)} - 2z^{(0)} + z^{(-i)}}{\sigma_i^2 (n_x + \lambda)} \rightarrow [u_i^T g_k''(\mu_x) u_i]_k. \quad (20)$$

Note that $n_x + \lambda = n_x / (1 - \omega^{(0)})$.

Using this, the limit case of (18) can be evaluated,

$$\mu_z^{\text{UT}} \rightarrow g(\mu_x) + \frac{1}{2} [\text{tr}(g_i''(\mu_x) P)]_i \quad (21a)$$

and

$$\begin{aligned} P_z^{\text{UT}} &\rightarrow g'(\mu_x) P (g'(\mu_x))^T \\ &+ \frac{(\beta - \alpha^2)}{4} [\text{tr}(P g_i''(\mu_x)) \text{tr}(P g_j''(\mu_x))]_{ij}. \end{aligned} \quad (21b)$$

By comparing (21) and (5) for a scalar $z = g(x)$, both TT2 and UT asymptotically gives the same result. In general, the covariances of TT2 and UT differ since

$$\begin{aligned} P_z^{\text{TT2}} - P_z^{\text{UT}} &= \frac{1}{2} [\text{tr}(P g_i''(\mu_x) P g_j''(\mu_x))]_{ij} \\ &- \frac{(\beta - \alpha^2)}{4} [\text{tr}(P g_i''(\mu_x)) \text{tr}(P g_j''(\mu_x))]_{ij} \end{aligned} \quad (22)$$

Note that $\text{tr}(AB) = \text{tr}(A)\text{tr}(B)$ if A and B are scalar, but this is in general not the case. Even with diagonal matrices, the result may differ. Consider for instance the example $\text{tr}(I_2)\text{tr}(I_2) = 4 \neq 2 = \text{tr}(I_2 I_2)$. One explanation for this discrepancy is that the UT cannot express the mixed second order derivatives needed for the TT2 compensation term without increasing the number of sigma points. The quality of this approximation depends on the transformation and must be analyzed for the case at hand.

E. Numerical Comparisons

We here provide some examples where the following methods are compared:

TT1	First order Taylor expansion leading to Gauss' approximation formula.
TT2	Second order Taylor expansion, which compensates the mean and covariance with the quadratic second order term.
UT	The unscented transformations UT1 and UT2. UT2 will be the default one in the sequel if the number is not indicated.
MCT	The Monte Carlo transformation approach, which in the limit should compute correct moments.

Tables II and III summarize the results.

Example 1 (Sum of squares) *The following mapping has a well-known distribution*

$$z = g(x) = x^T x, \quad x \sim \mathcal{N}(0, I_n) \Rightarrow z \sim \chi^2(n). \quad (23)$$

This distribution has mean n and variance $2n$. For the Taylor expansion, we have

$$g'(\mu) = 0, \quad g''(\mu) = 2I_n.$$

It follows that

$$\begin{aligned} \mu_z^{\text{TT1}} &= 0, & P_z^{\text{TT1}} &= 0, \\ \mu_z^{\text{TT2}} &= n, & P_z^{\text{TT2}} &= \frac{1}{2} \cdot 4n = 2n, \\ \mu_z^{\text{UT1}} &= n, & P_z^{\text{UT1}} &= (3 - n)n, \\ \mu_z^{\text{UT2}} &= n, & P_z^{\text{UT2}} &= \frac{1}{2} \cdot 2n \cdot 2n = 2n^2, \\ \mu_z^{\text{CT}} &= n, & P_z^{\text{CT}} &= 2n \cdot \frac{1}{2n} \cdot (1 - 1)^2 = 0. \end{aligned}$$

That is, TT1 fails completely and TT2 works perfectly. UT gives correct mean. The standard version of UT gives negative variance, while the modified one overestimates the variance, and CT gives zero variance, as seen in Tab II.

Tab II: 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 $2n$. The mean and variance are below summarized as a Gaussian distribution.

n	1	2	3	4	5	n
TT1	$\mathcal{N}(0, 0)$	$\mathcal{N}(0, 0)$	$\mathcal{N}(0, 0)$	$\mathcal{N}(0, 0)$	$\mathcal{N}(0, 0)$	$\mathcal{N}(0, 0)$
TT2	$\mathcal{N}(1, 2)$	$\mathcal{N}(2, 4)$	$\mathcal{N}(3, 6)$	$\mathcal{N}(4, 8)$	$\mathcal{N}(5, 10)$	$\mathcal{N}(n, 2n)$
UT1	$\mathcal{N}(1, 2)$	$\mathcal{N}(2, 2)$	$\mathcal{N}(3, 0)$	$\mathcal{N}(4, -4)$	$\mathcal{N}(5, -10)$	$\mathcal{N}(n, (3 - n)n)$
UT2	$\mathcal{N}(1, 2)$	$\mathcal{N}(2, 8)$	$\mathcal{N}(3, 18)$	$\mathcal{N}(4, 32)$	$\mathcal{N}(5, 50)$	$\mathcal{N}(n, 2n^2)$
CT	$\mathcal{N}(1, 0)$	$\mathcal{N}(2, 0)$	$\mathcal{N}(3, 0)$	$\mathcal{N}(4, 0)$	$\mathcal{N}(5, 0)$	$\mathcal{N}(n, 0)$

Example 2 (Radar measurements) *Consider the mapping of range and bearing to Cartesian coordinates*

$$z = g(x) = \begin{pmatrix} x_1 \cos x_2 \\ x_1 \sin x_2 \end{pmatrix}. \quad (24)$$

For the first case in Table III, $\mu_x = (3, 0)^T$, we have the Taylor expansion

$$g'(\mu) = \begin{pmatrix} 1 & 0 \\ 0 & 3 \end{pmatrix}, \quad g_1''(\mu) = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}, \quad g_2''(\mu) = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

Note that all higher order derivatives have the unit norm, $\|g^{(n)}(x)\| = 1$ for all $n \geq 2$, so the second order Taylor expansion cannot be regarded as an accurate approximation. This is particularly the case when the angular error is large, as it is designed to be here.

It follows from (14) and Theorem 2 that

$$\mu_z^{\text{UT}} = \mu_z^{\text{TT2}} = \begin{pmatrix} 0 \\ 0 \end{pmatrix},$$

and that the covariance approximations differ. The results are available in Tab III.

Tab III: Nonlinear approximations of the radar observations (range and bearing) to Cartesian position mapping $z = (x_1 \cos x_2, x_1 \sin x_2)^T$ for three different distributions of x . The mean and variance are below summarized as a Gaussian distribution. The number of Monte Carlo simulations is 10 000.

Method	x		
	$\mathcal{N}(\begin{pmatrix} 3.0 \\ 0.0 \end{pmatrix}, \begin{pmatrix} 1.0 & 0.0 \\ 0.0 & 1.0 \end{pmatrix})$	$\mathcal{N}(\begin{pmatrix} 3.0 \\ 0.5 \end{pmatrix}, \begin{pmatrix} 1.0 & 0.0 \\ 0.0 & 1.0 \end{pmatrix})$	$\mathcal{N}(\begin{pmatrix} 3.0 \\ 0.8 \end{pmatrix}, \begin{pmatrix} 1.0 & 0.0 \\ 0.0 & 1.0 \end{pmatrix})$
TT1	$\mathcal{N}(\begin{pmatrix} 3.0 \\ 0.0 \end{pmatrix}, \begin{pmatrix} 1.0 & 0.0 \\ 0.0 & 9.0 \end{pmatrix})$	$\mathcal{N}(\begin{pmatrix} 2.6 \\ 1.5 \end{pmatrix}, \begin{pmatrix} 3.0 & -3.5 \\ -3.5 & 7.0 \end{pmatrix})$	$\mathcal{N}(\begin{pmatrix} 2.1 \\ 2.1 \end{pmatrix}, \begin{pmatrix} 5.0 & -4.0 \\ -4.0 & 5.0 \end{pmatrix})$
TT2	$\mathcal{N}(\begin{pmatrix} 2.0 \\ -0.0 \end{pmatrix}, \begin{pmatrix} 3.0 & 0.0 \\ 0.0 & 10.0 \end{pmatrix})$	$\mathcal{N}(\begin{pmatrix} -1.4 \\ 0.5 \end{pmatrix}, \begin{pmatrix} 27.0 & 2.5 \\ 2.5 & 9.0 \end{pmatrix})$	$\mathcal{N}(\begin{pmatrix} 2.1 \\ 2.1 \end{pmatrix}, \begin{pmatrix} 9.0 & 0.0 \\ 0.0 & 13.0 \end{pmatrix})$
UT1	$\mathcal{N}(\begin{pmatrix} 1.8 \\ 0.0 \end{pmatrix}, \begin{pmatrix} 3.7 & 0.0 \\ 0.0 & 2.9 \end{pmatrix})$	$\mathcal{N}(\begin{pmatrix} 1.6 \\ 0.9 \end{pmatrix}, \begin{pmatrix} 3.5 & 0.3 \\ 0.3 & 3.1 \end{pmatrix})$	$\mathcal{N}(\begin{pmatrix} 1.3 \\ 1.3 \end{pmatrix}, \begin{pmatrix} 3.3 & 0.4 \\ 0.4 & 3.3 \end{pmatrix})$
UT2	$\mathcal{N}(\begin{pmatrix} 1.5 \\ 0.0 \end{pmatrix}, \begin{pmatrix} 5.5 & 0.0 \\ 0.0 & 9.0 \end{pmatrix})$	$\mathcal{N}(\begin{pmatrix} 1.3 \\ 0.8 \end{pmatrix}, \begin{pmatrix} 6.4 & -1.5 \\ -1.5 & 8.1 \end{pmatrix})$	$\mathcal{N}(\begin{pmatrix} 1.1 \\ 1.1 \end{pmatrix}, \begin{pmatrix} 7.2 & -1.7 \\ -1.7 & 7.2 \end{pmatrix})$
CT	$\mathcal{N}(\begin{pmatrix} 1.73 \\ 0.0 \end{pmatrix}, \begin{pmatrix} 2.6 & 0.0 \\ 0.0 & 4.39 \end{pmatrix})$	$\mathcal{N}(\begin{pmatrix} 1.52 \\ 0.83 \end{pmatrix}, \begin{pmatrix} 3.01 & -0.75 \\ -0.75 & 3.98 \end{pmatrix})$	$\mathcal{N}(\begin{pmatrix} 1.21 \\ 1.24 \end{pmatrix}, \begin{pmatrix} 3.52 & -0.893 \\ -0.893 & 3.47 \end{pmatrix})$
MCT	$\mathcal{N}(\begin{pmatrix} 1.8 \\ 0.0 \end{pmatrix}, \begin{pmatrix} 2.5 & 0.0 \\ 0.0 & 4.4 \end{pmatrix})$	$\mathcal{N}(\begin{pmatrix} 1.6 \\ 0.9 \end{pmatrix}, \begin{pmatrix} 2.9 & -0.8 \\ -0.8 & 3.9 \end{pmatrix})$	$\mathcal{N}(\begin{pmatrix} 1.3 \\ 1.3 \end{pmatrix}, \begin{pmatrix} 3.4 & -1.0 \\ -1.0 & 3.4 \end{pmatrix})$

Example 3 (TOA, DOA and RSS measurements) *The basic measurements in sensor networks [14] are time of arrival (TOA), direction of arrival (DOA) and received signal strength (RSS). These all relate to the position in a nonlinear way. Range measurements in two ($n = 2$) and three ($n = 3$) dimensions, respectively, are given by*

$$g_{\text{TOA}}(x) = \|x\| = \sqrt{\sum_{i=1}^n x_i^2}.$$

Received signal strength in two dimensions in dB scale (where the measurement noise can be seen as additive and Gaussian [14]) is given by

$$g_{\text{RSS}}(x) = c_0 - c_2 \cdot 10 \log_{10}(\|x\|^2).$$

Finally, direction of arrival is expressed as

$$g_{\text{DOA}}(x) = \arctan2(x_1, x_2),$$

where $\arctan2$ is the four quadrant arc-tangent function. The resulting approximation depends a lot on the assumed Gaussian distribution of position. We choose a distribution which is typical in single sensor tracking applications, where the prior distribution before the measurement update is uncertain in the direction tangential to the measurement information. The results are summarized in Table IV.

The conclusion from the example, and many similar tests with other prior distributions of the position, is that TT1 is inferior

Tab IV: Numerical comparison of approximate transformations for nonlinear measurement models in sensor network applications. The mean and covariance are in each case summarized as an approximate Gaussian distribution.

TOA 2D $g(x) = \ x\ $	$x \sim \mathcal{N}([3; 0], [1, 0; 0, 10])$
TT1	$\mathcal{N}(3, 1)$
TT2	$\mathcal{N}(4.67, 6.56)$
UT1	$\mathcal{N}(4.08, 3.34)$
UT2	$\mathcal{N}(4.67, 6.56)$
CT	$\mathcal{N}(4.19, 2, 42)$
MCT	$\mathcal{N}(4.25, 2.4)$
TOA 3D $g(x) = \ x\ $	$x \sim \mathcal{N}([3; 0; 0], [1, 0, 0; 0, 10, 0; 0, 0, 10])$
TT1	$\mathcal{N}(3, 1)$
TT2	$\mathcal{N}(6.33, 12.1)$
UT1	$\mathcal{N}(5.16, 3.34)$
UT2	$\mathcal{N}(6.33, 23.2)$
CT	$\mathcal{N}(5.16, 3.34)$
MCT	$\mathcal{N}(5.17, 2.95)$
DOA $g(x) = \arctan(x_2, x_1)$	$x \sim \mathcal{N}([3; 0], [10, 0; 0, 1])$
TT1	$\mathcal{N}(0, 0.111)$
TT2	$\mathcal{N}(0, 0.235)$
UT1	$\mathcal{N}(0.524, 1.46)$
UT2	$\mathcal{N}(0, 0.111)$
CT	$\mathcal{N}(0.785, 1.95)$
MCT	$\mathcal{N}(-0.004, 1.38)$
RSS $g(x) = 10 - 20 \log_{10}(\ x\ ^2)$	$x \sim \mathcal{N}([3; 0], [10, 0; 0, 1])$
TT1	$\mathcal{N}(-18, 12.1)$
TT2	$\mathcal{N}(-21.1, 36.5)$
UT1	$\mathcal{N}(-19.9, 25.5)$
UT2	$\mathcal{N}(-21.1, 31.6)$
CT	$\mathcal{N}(-20.1, 21.3)$
MCT	$\mathcal{N}(-20.1, 16.4)$

and that the UT, and in particular the tuning provided by the CT, is to be preferred to TT2. However, all of TT1, TT2, and UT can be arbitrarily bad compared to the MCT.

It should be remarked, though, that all the cases in Examples 2 and 3 are deliberately designed to excite higher order terms in the Taylor expansion. As the range to the target increases, the higher order terms will decrease with a rate that for the higher order terms is higher than that for the lower order terms. Based on the preceding analysis, one may expect that the TT2 will converge faster than TT1 and UT.

III. DARE-BASED EXTENDED KALMAN FILTER

Here, detailed recursions are given for the *extended Kalman filter* (EKF) without and with second order compensation, respectively. The function $f(x, u)$ is here more compactly written $f(x)$, and similarly $h(x) = h(x, u)$.

A. EKF Algorithms

Using the transformation approximation TT1 and TT2, respectively, immediately gives the two Riccati-based EKF filters in Algorithm 1.

The common EKF should work well when the bias and variance contribution of the second order Taylor term is

Algorithm 1 DARE-based EKF and EKF2

The EKF2, using the TT2 transformation, for the model (1) is given by the following recursions initialized with $\hat{x}_{1|0}$ and $P_{1|0}$:

$$S_k = R_k + h'(\hat{x}_{k|k-1})P_{k|k-1}(h'(\hat{x}_{k|k-1}))^T + \frac{1}{2} [\text{tr}(h_i''(\hat{x}_{k|k-1})P_{k|k-1}h_j''(\hat{x}_{k|k-1})P_{k|k-1})]_{ij}$$

$$K_k = P_{k|k-1}(h'(\hat{x}_{k|k-1}))^T S_k^{-1} \quad (25a)$$

$$\varepsilon_k = y_k - h(\hat{x}_{k|k-1}) - \frac{1}{2} [\text{tr}(h_i''P_{k|k-1})]_i \quad (25b)$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k \varepsilon_k \quad (25c)$$

$$P_{k|k} = P_{k|k-1} \quad (25d)$$

$$- P_{k|k-1}(h'(\hat{x}_{k|k-1}))^T S_k^{-1} h'(\hat{x}_{k|k-1})P_{k|k-1}$$

$$\hat{x}_{k+1|k} = f(\hat{x}_{k|k}) + \frac{1}{2} [\text{tr}(f_i''P_{k|k})]_i \quad (25e)$$

$$P_{k+1|k} = Q_k + f'(\hat{x}_{k|k})P_{k|k}(f'(\hat{x}_{k|k}))^T + \frac{1}{2} [\text{tr}(f_i''(\hat{x}_{k|k})P_{k|k}f_j''(\hat{x}_{k|k})P_{k|k})]_{ij}. \quad (25f)$$

The EKF, using the TT1 transformation, is obtained by letting both Hessians f'' and h'' be zero.

negligible to the noise,

$$\frac{1}{4} \left[\text{tr}(f_i''(\hat{x}_{k|k})P_{k|k})^T \text{tr}(f_j''(\hat{x}_{k|k})P_{k|k}) \right]_{ij} + \frac{1}{2} [\text{tr}(f_i''(\hat{x}_{k|k})P_{k|k}f_j''(\hat{x}_{k|k})P_{k|k})]_{ij} \ll Q_k, \quad (26a)$$

$$\frac{1}{4} \left[\text{tr}(h_i''(\hat{x}_{k|k-1})P_{k|k-1})^T \text{tr}(h_j''(\hat{x}_{k|k-1})P_{k|k-1}) \right]_{ij} + \frac{1}{2} [\text{tr}(h_i''(\hat{x}_{k|k-1})P_{k|k-1}h_j''(\hat{x}_{k|k-1})P_{k|k-1})]_{ij} \ll R_k. \quad (26b)$$

Here, $0 \ll A$ means that the eigenvalues of A are all much greater than zero. These are conditions that can be monitored on-line, but with a large computational overhead, or analyzed off-line based on only the model and typical operating points.

B. Numerical Approximations of Gradients and Taylor Terms

The standard form of the EKF involves symbolic derivatives. However, numeric derivatives may be preferred in the following cases:

- The nonlinear function is too complex to be differentiated. For instance, it may involve a computer vision algorithm or a database look-up.
- The derivatives are too complex functions, requiring too much computer code, memory or computations to be evaluated.
- A user-friendly algorithm is desired, with as few user inputs as possible.

The derivatives can then be approximated numerically, for instance by

$$\frac{\partial g(x)}{\partial x_i} \approx \frac{g(x + \Delta e_i) - g(x)}{\Delta}, \quad (27a)$$

$$\frac{\partial^2 g(x)}{\partial x_i \partial x_j} \approx \frac{1}{\Delta^2} (g(x + \Delta e_i + \Delta e_j) - g(x + \Delta e_i) - g(x + \Delta e_j) + g(x)). \quad (27b)$$

The number of function evaluations is $n_x + 1$ for the difference in (27a) ($2n_x$ for a central difference) and $n_x^2 + n_x + 1$ for difference in (27b) ($4n_x^2$ for a central difference). This should be compared to the total complexity of EKF2, which is of order n_x^3 . These numerical approximations of the Jacobian and the Hessian can be used in (25).

However, we next derive an alternative implementation using the sigma points, where these matrices never need to be formed. This algorithm is fundamentally different from other approaches in literature for derivative free (derivative free here means that neither analytical derivatives nor numerical approximations of the Jacobian or the Hessian are required) implementation of the EKF, such as DF-EKF in [13].

Theorem 3 (Sigma-point based DARE EKF) *Consider the mapping $z = g(x)$ for $x \sim \mathcal{N}(\hat{x}, P)$. Given the transformed sigma-points $z^{(i)} = g(x^{(i)})$ in (15), the terms in Algorithm 1 involving Jacobians and Hessians can be approximated arbitrarily well as $\alpha \rightarrow 0$ with*

$$\begin{aligned} \lim_{\alpha \rightarrow 0^+} \sum_{i=1}^{n_x} \sigma_i \frac{z_k^{(i)} - z_k^{(-i)}}{2\alpha\sqrt{n_x}\sigma_i} u_i^T &= g'_k(\hat{x})P, \\ \lim_{\alpha \rightarrow 0^+} \sum_{i=1}^{n_x} \sigma_i \left(\frac{z_k^{(i)} - z_k^{(-i)}}{2\alpha\sqrt{n_x}\sigma_i} \right) (\cdot)^T &= g'_k(\hat{x})P(g'_k(\hat{x}))^T, \\ \lim_{\alpha \rightarrow 0^+} \sum_{i=1}^{n_x} \sigma_i \frac{z_k^{(i)} - 2z_k^{(0)} + z_k^{(-i)}}{\alpha^2\sigma_i n_x} &= \text{tr}(g''_k(\hat{x})P). \end{aligned}$$

Further, for a scalar x ,

$$\begin{aligned} \lim_{\alpha \rightarrow 0^+} \sum_{i=1}^{n_x} \sum_{j=1}^{n_x} \frac{z_k^{(i)} - 2z_k^{(0)} + z_k^{(-i)}}{\alpha^2\sigma} \frac{z_l^{(j)} - 2z_l^{(0)} + z_l^{(-j)}}{\alpha^2\sigma} \\ = \text{tr}(g''_k(\hat{x})P g''_l(\hat{x})P). \end{aligned}$$

Proof: Using the SVD

$$P = UDU^T = \sum_{i=1}^{n_x} \sigma_i^2 u_i u_i^T, \quad (28)$$

the sigma points in (15) can be written, using $n_x + \lambda = \alpha^2 n_x$ when $\kappa = 0$,

$$x^0 = \hat{x}, \quad (29a)$$

$$x^{(\pm i)} = \hat{x} \pm \alpha\sqrt{n_x}\sigma_i u_i, \quad i = 1, 2, \dots, n_x. \quad (29b)$$

The Taylor expansion (2) for the transformed sigma points can then be written

$$\begin{aligned} z_k^{(\pm i)} &= g_k(x^{(\pm i)}) \\ &= g_k(\hat{x}) \pm \alpha\sqrt{n_x}\sigma_i g'_k(\hat{x})u_i + \frac{\alpha^2 n_x \sigma_i}{2} u_i^T g''_k(\hat{x})u_i. \end{aligned} \quad (30)$$

Note that the second order rest term is accurate only in a small neighborhood of \hat{x} , so the sigma points should be chosen close to \hat{x} , which means that α should be small.

The first and second order terms in the Taylor expansion can now be resolved using the following linear combinations,

$$\frac{z_k^{(i)} - z_k^{(-i)}}{2\alpha\sqrt{n_x}\sigma_i} \rightarrow g'_k(\hat{x})u_i, \quad (31a)$$

$$\frac{z_k^{(i)} - 2z_k^{(0)} + z_k^{(-i)}}{\alpha^2\sigma_i n_x} \rightarrow u_i^T g''_k(\hat{x})u_i, \quad (31b)$$

as $\alpha \rightarrow 0^+$. Taking the weighted sum of the term in (31), we get

$$\sum_{i=1}^{n_x} \sigma_i \frac{z_k^{(i)} - z_k^{(-i)}}{2\alpha\sqrt{n_x}\sigma_i} u_i^T \rightarrow \sum_{i=1}^{n_x} g'_k(\hat{x})\sigma_i u_i u_i^T g'_k(\hat{x})P. \quad (32a)$$

Similarly, summing quadratic forms of (31) gives

$$\begin{aligned} \sum_{i=1}^{n_x} \sigma_i \left(\frac{z_k^{(i)} - z_k^{(-i)}}{2\alpha\sqrt{n_x}\sigma_i} \right) \left(\frac{z_k^{(i)} - z_k^{(-i)}}{2\alpha\sqrt{n_x}\sigma_i} \right)^T \\ \rightarrow \sum_{i=1}^{n_x} \sigma_i g'_k(\hat{x})u_i (g'_k(\hat{x})u_i)^T = g'_k(\hat{x}) \sum_{i=1}^{n_x} \sigma_i u_i u_i^T (g'_k(\hat{x}))^T \\ = g'_k(\hat{x})P(g'_k(\hat{x}))^T \end{aligned} \quad (32b)$$

Further,

$$\begin{aligned} \sum_{i=1}^{n_x} \sigma_i \frac{z_k^{(i)} - 2z_k^{(0)} + z_k^{(-i)}}{\alpha^2\sigma_i n_x} \rightarrow \sum_{i=1}^{n_x} \sigma_i u_i^T g''_k(\hat{x})u_i, \\ = \text{tr} \left(g''_k(\hat{x}) \sum_{i=1}^{n_x} \sigma_i u_i u_i^T \right) = \text{tr}(g''_k(\hat{x})P) \end{aligned} \quad (32c)$$

For the final statement in the theorem, note that

$$\begin{aligned} \text{tr}(g''_k(\hat{x})P g''_l(\hat{x})P) &= \text{tr} \left(g''_k(\hat{x}) \sum_{i=1}^{n_x} \sigma_i u_i u_i^T g''_l(\hat{x}) \sum_{j=1}^{n_x} \sigma_j u_j u_j^T \right) \\ &= \sum_{i=1}^{n_x} \sum_{j=1}^{n_x} \sigma_i \sigma_j (u_j^T g''_k(\hat{x})u_i) (u_i^T g''_l(\hat{x})u_j). \end{aligned} \quad (32d)$$

Now, (32d) can be simplified if $n_x = 1$ so only symmetric factors $u_i^T g''_k(\hat{x})u_j = g''_l(\hat{x})$ remain. This concludes the proof. \blacksquare

That is, the standard EKF can be implemented without forming the Jacobians $f'(x)$ and $h'(x)$, neither analytically nor numerically. This holds also for the second order EKF, where neither the Jacobians nor Hessians need to be formed.

It is interesting to compare the computational complexity of the three alternatives of using analytical Jacobian (and Hessian), numerical approximation of these and finally the numerical approximation of the terms needed in the EKF. For the standard EKF, the matrix times matrix multiplication FPF^T is of complexity $\mathcal{O}(n_x^3)$, which is also the case for the first two terms in Theorem 3. This fact is well known. Next, consider the second order EKF. The complexity of computing $g''_k P$ is of complexity $\mathcal{O}(n_x^3)$.

The last statement in Theorem 3 holds only for a scalar state, which is a rather limited result. There is no apparent way to generalize this using only the sigma points in the UT. However, by also including the ‘‘corner’’ sigma points, this issue can be resolved.

To motivate this statement, define an extended set of sigma points where the points

$$x^{(ij)} = \hat{x} + \alpha\sqrt{n_x}\sigma_i u_i + \alpha\sqrt{n_x}\sigma_j u_j, \quad (33a)$$

$$x^{(-ij)} = \hat{x} - \alpha\sqrt{n_x}\sigma_i u_i - \alpha\sqrt{n_x}\sigma_j u_j, \quad (33b)$$

are added to the set in (15), and $z^{(ij)} = g(x^{(ij)})$ is defined for each new sigma point. The derivation is based on the observation that

$$z_k^{(ij)} = g_k(\hat{x}) + \alpha\sqrt{n_x}g_k'(\hat{x})(\sigma_i u_i + \sigma_j u_j) + \frac{\alpha^2 n_x}{2}(\sigma_i u_i + \sigma_j u_j)^T g_k''(\hat{x})(\sigma_i u_i + \sigma_j u_j) \quad (34a)$$

$$z_k^{(-ij)} = g_k(\hat{x}) - \alpha\sqrt{n_x}g_k'(\hat{x})(\sigma_i u_i + \sigma_j u_j) + \frac{\alpha^2 n_x}{2}(\sigma_i u_i + \sigma_j u_j)^T g_k''(\hat{x})(\sigma_i u_i + \sigma_j u_j) \quad (34b)$$

One can then show that

$$\frac{z_k^{(ij)} + z_k^{(-ij)} - 2z_k^{(0)}}{\alpha^2 n_x} = \sigma_i^2 u_i^T g_k''(\hat{x}) u_i + \sigma_j^2 u_j^T g_k''(\hat{x}) u_j + 2\sigma_i \sigma_j u_i^T g_k''(\hat{x}) u_j. \quad (35)$$

Here, the first two terms can be computed with the standard sigma points using (32c), while the last term is what is needed to evaluate (32d) in the multivariable case. The details are outside the scope of this contribution. There are actually two advantages of implementing the second order EKF in this way, even compared to the case where the analytical Hessian G is available. First, the terms $\text{tr}(G_k P)$ and $\text{tr}(G_k P G_l P)$ are here numerically approximated in a basis that automatically incorporates P , so the product $G_l P$ is not needed to form explicitly. The direct computation of $\text{tr}(G_k P)$ for all $k = 1, \dots, n_x$ is of complexity n_x^3 (only the diagonal terms need to be computed for the trace), while the numerical approximation in (32c) is only $\mathcal{O}(n_x^2)$. Further, direct evaluation of $\text{tr}(G_k P G_l P)$ for all k and l is of complexity $\mathcal{O}(n_x^5)$, while the numerical approximation in (32d) is $\mathcal{O}(n_x^4)$.

In summary, the transformed sigma points can be used to approximate the linear term and rest term in the Taylor expansion (2), without explicitly computing the Jacobian and Hessian of f and h . This is one sound motivation for propagating the sigma points through the nonlinearity. From Theorem 3 and the following discussion, we make the following remarks on the EKF, assuming for simplicity additive noise processes:

- Equations (32a,b) give the gradient needed in the standard EKF (25). That is, h'_x can be substituted with one of these approximations on all occasions.
- Equation (32c) provides the mean corrections in the second order EKF (25a,e).
- Equation (32d) provides the covariance corrections in the second order EKF (25f,h) for a scalar x , but as we have argued this can be resolved by using more sigma points (33), leading to a computationally efficient implementation.

IV. RICCATI-FREE EKF AND UKF

The Kalman filter equations are often obscured by the complexity of the Riccati equation. However, one key idea

in the UKF is based on a result from optimal filtering, where UT but also TT1, TT2 and MCT can be used.

As a brief review, the basic idea is to consider the nonlinear transformation

$$z = \begin{pmatrix} x \\ g(x, w) \end{pmatrix} \quad (36)$$

of the state x and a stochastic variable w , both assumed Gaussian distributed, using the prior

$$\bar{x} = \begin{pmatrix} x \\ w \end{pmatrix} \sim \mathcal{N} \left(\begin{pmatrix} \hat{x} \\ 0 \end{pmatrix}, \begin{pmatrix} P^x & 0 \\ 0 & P^w \end{pmatrix} \right). \quad (37)$$

The transformed variables can then be approximated with the following Gaussian distribution, using TT1, TT2, UT, or MCT,

$$z \sim \mathcal{N} \left(\begin{pmatrix} z^x \\ z^g \end{pmatrix}, \begin{pmatrix} P^{xx} & P^{xg} \\ P^{gx} & P^{gg} \end{pmatrix} \right). \quad (38)$$

The quality of the approximation depends on the nonlinearity and the method used. Assuming an observation g^{obs} of the nonlinear relation $g(x, w)$, a well-known result (see for instance Lemma 7.1 in [15]) states that

$$K = P^{xg} (P^{gg})^{-1}, \quad (39)$$

$$\hat{x} = z^x + K(g^{obs} - z^g). \quad (40)$$

The basic idea is thus to approximate the covariance matrix for $(x^T, g^T)^T$ numerically and compute the Kalman gain K from its block matrix decomposition. Algorithm 2 gives the general algorithm. Note that the process noise does not need to be additive in this approach. These transformations provide a framework for nonlinear filtering from which the following different combinations of transforms can be done:

- The EKF obtained using TT1 above is equivalent to the EKF in (25).
- The EKF version obtained using TT2 above is equivalent to the second order compensated EKF in (25).
- The Monte Carlo approach should potentially be the most accurate, given that a sufficient number of samples are used, since it asymptotically computes the correct first and second order moments.
- The UKF is obtained by using the UT (1, 2 or other variants) in both time and measurement updates above.
- One should be aware of that it is not advisable to start with a large initial covariance P_0 when using UKF, since the sigma points are then located far from the true state, in contrast to the EKF variants.
- There is a freedom to mix transform approximations in the time and measurement update.
- If the observation model is linear, the usual Kalman filter measurement update should be performed. The same holds for a linear dynamic model.

The actual performance for the 16 different combinations depends of course on the degree of nonlinearity in the system model. As a rule of thumb, the choice can be guided by studying the nonlinear mappings in the dynamic model and sensor model individually. For target tracking and navigation applications, it is often the nonlinear sensor model that gives the greatest filtering challenge as pointed out in [16].

Algorithm 2 Nonlinear Transformation-Based Filtering

The nonlinear transform-based filter for the model (1) is given by the following recursions initialized with $\hat{x}_{1|0}$ and $P_{1|0}$:

1) **Measurement update:** Let

$$\bar{x} = \begin{pmatrix} x_k \\ e_k \end{pmatrix} \sim \mathcal{N} \left(\begin{pmatrix} \hat{x}_{k|k-1} \\ 0 \end{pmatrix}, \begin{pmatrix} P_{k|k-1} & 0 \\ 0 & R_k \end{pmatrix} \right) \quad (41a)$$

$$z = \begin{pmatrix} x_k \\ y_k \end{pmatrix} = \begin{pmatrix} x_k \\ h(x_k) + e_k \end{pmatrix} \quad (41b)$$

The transformation approximation (UT, MC, TT1, TT2) gives

$$z \sim \mathcal{N} \left(\begin{pmatrix} \hat{x}_{k|k-1} \\ \hat{y}_{k|k-1} \end{pmatrix}, \begin{pmatrix} P_{k|k-1}^{xx} & P_{k|k-1}^{xy} \\ P_{k|k-1}^{yx} & P_{k|k-1}^{yy} \end{pmatrix} \right) \quad (41c)$$

The measurement update is then

$$K_k = P_{k|k-1}^{xy} (P_{k|k-1}^{yy})^{-1}, \quad (41d)$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k (y_k - \hat{y}_{k|k-1}), \quad (41e)$$

$$P_{k|k}^{xx} = P_{k|k-1}^{xx} - K_k P_{k|k-1}^{yy} K_k^T. \quad (41f)$$

2) **Time update:** Let

$$\bar{x} = \begin{pmatrix} x_k \\ v_k \end{pmatrix} \sim \mathcal{N} \left(\begin{pmatrix} \hat{x}_{k|k} \\ 0 \end{pmatrix}, \begin{pmatrix} P_{k|k} & 0 \\ 0 & Q_k \end{pmatrix} \right) \quad (41g)$$

$$z = x_{k+1} = f(x_k) + v_k. \quad (41h)$$

The transformation approximation (UT, MC, TT1, TT2) gives

$$z \sim \mathcal{N} (\hat{x}_{k+1|k}, P_{k+1|k}). \quad (41i)$$

Example 4 (Bearings-Only Tracking) The next example exemplifies the common bearings only problem depicted in Figure 1. Here a situation where a target, known to be in an approximate location quantified by $\hat{x}_{0|0}$ and $P_{0|0}$, is being triangulated using bearings-only measurements θ . This can be

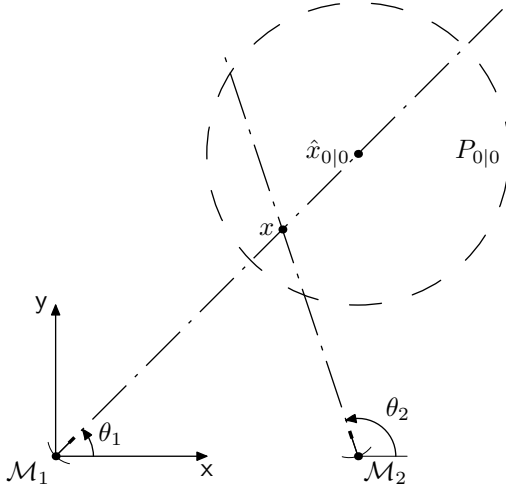


Fig 1: Bearings-only problem. Two measurements are used, one from \mathcal{M}_1 and one from \mathcal{M}_2 .

mathematically described as

$$x_{k+1} = f(x_k) + v_k = x_k + v_k$$

$$y_k = h(x_k) + e_k = \arctan2(x_k - x_k^0, y_k - y_k^0) + e_k,$$

where the state $x = (x \ y)^T$ is the Cartesian position of the target, $v_k \equiv 0$ for clarity, and $\text{COV}(e_k) = R$. For this situation, the gradients needed to perform filtering using an EKF are

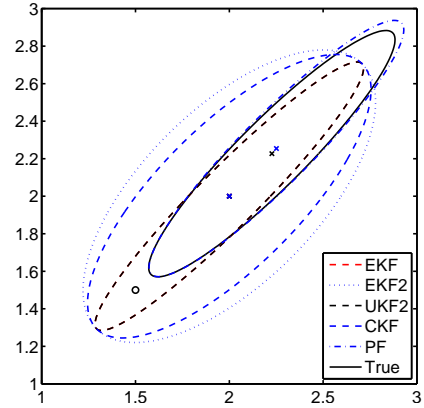
$$F = I, \quad H = \frac{1}{(x - x^0)^2 + (y - y^0)^2} \begin{pmatrix} -(y - y^0) \\ x - x^0 \end{pmatrix}.$$

Note, the first order approximation of $\arctan2$ is best for $|\frac{y}{x}| \gg 0$.

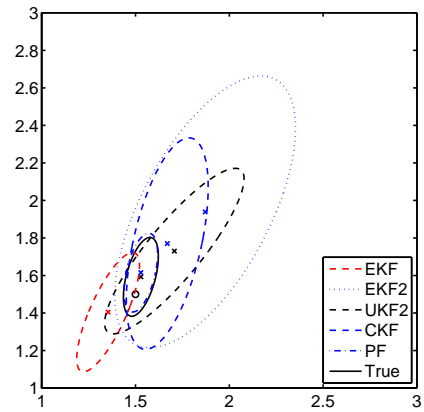
Now, assume

$$x_0 = \begin{pmatrix} 1.5 \\ 1.5 \end{pmatrix}, \quad \hat{x}_{0|0} = \begin{pmatrix} 2 \\ 2 \end{pmatrix}, \quad P_{0|0} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix},$$

and that the bearing to the target is measured first from the position $\mathcal{M}_1 = (0, 0)$ and then from $\mathcal{M}_2 = (2, 0)$, as depicted in Figure 1. Figure 2 depicts the estimates based on this new \hat{x}_0 and noise-free measurements for the different filters.



(a) Measurements from: \mathcal{M}_1 (EKF and UKF almost coincide, as do PF and true.)



(b) Measurements from: $\mathcal{M}_1 + \mathcal{M}_2$

Fig 2: Estimate with $\hat{x}_0 = (0, 0)^T$, based on one and two measurements. (Estimates are denoted with \times in the center of each covariance ellipse, and the true target position is denoted with \circ .)

Tab V: Mean square error filter performance for 1 000 Monte Carlo simulations. True posterior is computed by a point-mass filter (PMF) with a dense grid, and the particle filter (PF) performance is given for comparison.

Filter	Measurements		
	None	\mathcal{M}_1	$\mathcal{M}_1 + \mathcal{M}_2$
True	2.01	0.95	0.06
EKF	2.01	1.33	1.23
EKFII	2.01	1.38	0.79
UKF	2.01	1.33	1.00
CKF	2.01	1.45	1.12
PF	2.01	0.95	0.06

The variance of the estimation error based on Monte Carlo simulations of the problem specified above, using the described filters, yield the result in Table V. The table somewhat contradicts the previous results. One thing to observe is that the UKF outperforms the EKF. Hence, it seems that the conservative P matrix actually pays off.

Finally, note that the PF and the inferred distribution is almost identical. Worth noticing, though, is the substantially better estimates achieved with the PF compared to the other used filters. Hence, this is a situation when the PF pays off.

V. CONCLUSIONS

For nonlinear filtering problems where the nonlinearity is severe compared to the prior state information, the classical extended Kalman filter (EKF) “stinks” compared to the unscented Kalman filter (UKF), which has been concluded in a large number of applications. We have shown that the less cited EKF2 based on a second order KF is closely related to the UKF. Indeed, in a way EKF2 approximates the first two moments in a more accurate way for multi-variable transformations. The comparison is performed in terms of the corresponding transformations of a nonlinear mapping $z = g(x)$ for x being Gaussian. The unscented transform (UT) does not give the correct second order moments even for quadratic functions, an often stated property. This was demonstrated with the simple counter-example $g(x) = x^T x$ which has an analytical solution. On the other hand, for many standard sensor models, the UT performs very well.

APPENDIX

The Jacobian $g'(x)$ and Hessian $g''(x)$ for a scalar function $g(x)$ are defined as

$$g'(x) = \begin{pmatrix} \frac{\partial g(x)}{\partial x_1} & \frac{\partial g(x)}{\partial x_2} & \dots & \frac{\partial g(x)}{\partial x_n} \end{pmatrix}, \quad (43a)$$

$$g''(x) = \begin{pmatrix} \frac{\partial^2 g(x)}{\partial x_1 \partial x_1} & \frac{\partial^2 g(x)}{\partial x_1 \partial x_2} & \dots & \frac{\partial^2 g(x)}{\partial x_1 \partial x_n} \\ \frac{\partial^2 g(x)}{\partial x_2 \partial x_1} & \frac{\partial^2 g(x)}{\partial x_2 \partial x_2} & \dots & \frac{\partial^2 g(x)}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 g(x)}{\partial x_n \partial x_1} & \frac{\partial^2 g(x)}{\partial x_n \partial x_2} & \dots & \frac{\partial^2 g(x)}{\partial x_n \partial x_n} \end{pmatrix}. \quad (43b)$$

REFERENCES

[1] G. L. Smith, S. F. Schmidt, and L. A. McGee, “Application of statistical filter theory to the optimal estimation of position and velocity on board a circumlunar vehicle,” NASA, Tech. Rep. TR R-135, 1962.

[2] S. Schmidt, “Application of state-space methods to navigation problems,” *Advances in Control Systems*, pp. 293–340, 1966.

[3] M. Athans, R. Wishner, and A. Bertolini, “Suboptimal state estimation for continuous-time nonlinear systems from discrete noisy measurements,” *IEEE Transactions on Automatic Control*, 1968.

[4] P. S. Maybeck, *Stochastic Models, Estimation, and Control, volume 2*. Academic Press, 1982.

[5] Y. Bar-Shalom, X. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation: Theory, Algorithms and Software*. John Wiley & Sons, 2001.

[6] S. Julier, J. Uhlmann, and H. F. Durrant-Whyte, “A new approach for filtering nonlinear systems,” in *IEEE American Control Conference*, 1995, pp. 1628–1632.

[7] S. Julier and J. Uhlmann, “Unscented filtering and nonlinear estimation,” *Proceedings of the IEEE*, vol. 92, no. 3, pp. 401–422, 2004.

[8] S.Saha, P. Mandal, Y. Boers, H. Driessen, and A. Bagchi, “Gaussian proposal density using moment matching in SMC methods,” *Journal Statistics and Computing*, 2009.

[9] T. Lefebvre, H. Bruyninckx, and J. DeSchutter, “Comment on ”a new method for the nonlinear transformation of means and covariances in filters and estimators,” *IEEE Transactions on Automatic Control*, vol. 47, no. 8, pp. 1406–1408, 2002.

[10] I. Arasaratnam, S. Haykin, and R. Elliot, “Discrete-time nonlinear filtering algorithms using Gauss-Hermite quadrature,” *Proceedings of IEEE*, vol. 95, pp. 953–977, 2007.

[11] I. Arasaratnam and S. Haykin, “Cubature Kalman filters,” *IEEE Transactions on Automatic Control*, vol. 54, pp. 1254–1269, 2009.

[12] E. Wan and R. van der Merwe, “The unscented Kalman filter for nonlinear estimation,” in *Proc. of IEEE Symposium (AS-SPCC)*, pp. 153–158.

[13] B. Quine, “A derivative-free implementation of the extended Kalman filter,” *Automatica*, vol. 42, pp. 1927–1934, 2006.

[14] F. Gustafsson and F. Gunnarsson, “Mobile positioning using wireless networks: possibilities and fundamental limitations based on available wireless network measurements,” *IEEE Signal Processing Magazine*, vol. 22, pp. 41–53, 2005.

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

[16] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P.-J. Nordlund, “Particle filters for positioning, navigation and tracking,” *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 425–437, February 2002.



Fredrik Gustafsson is professor in Sensor Informatics at Department of Electrical Engineering, Linköping University, since 2005. He received the M.Sc. degree in electrical engineering 1988 and the Ph.D. degree in Automatic Control, 1992, both from Linköping University. During 1992–1999 he held various positions in automatic control, and 1999–2005 he had a professorship in Communication Systems. His research interests are in stochastic signal processing, adaptive filtering and change detection, with applications to communication, vehicular, airborne, and audio systems.

He has supervised 15 PhD, 20 licentiate and more than 190 master theses. He is the author of five books and over 190 conference papers, 60 journal papers and some twenty patents (h-index 31 in Google Scholar).

He is a co-founder of the companies *NIRA Dynamics AB*, developing an indirect tire pressure monitoring system, *Softube AB*, developing software emulators for guitar tube amplifiers and other music equipment, and *SenionLab AB*, developing indoor navigation solutions.

He was an associate editor for *IEEE Transactions of Signal Processing* 2000–2006 and is currently associate editor for *IEEE Transaction of Aerospace and Electronic Systems* and *EURASIP Journal on Advances in Signal Processing*. In 2004, he was awarded the Arnberg prize by the Royal Swedish Academy of Science (KVA) and in 2007 he was elected member of the Royal Academy of Engineering Sciences (IVA).



Gustaf Hendeby received his M.Sc. degree in Electrical Engineering and Applied Physics in 2002 and his Ph.D. in Automatic Control in 2008, both from Linköping University, Sweden. He remained at Linköping University as assistant professor until 2009 when he joined the German Research Center for Artificial Intelligence (DFKI). Since 2011 he works in the Competence Unit Informatics, Division of Information Systems at the Swedish Defense Research Agency (FOI) in Linköping, Sweden.

Dr. Hendeby's main research interests are stochastic signal processing, sensor fusion, and change detection, especially for nonlinear and non-Gaussian systems. He has experience in both theoretical analysis as well as practical implementation aspects.