A New Continuous Discrete Unscented Kalman Filter

Knudsen, Torben; Leth, John-Josef

Published in:
IEEE Transactions on Automatic Control

DOI (link to publication from Publisher):
10.1109/TAC.2018.2867325

Publication date:
2019

Document Version
Accepted author manuscript, peer reviewed version

Link to publication from Aalborg University

Citation for published version (APA):

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

Take down policy
If you believe that this document breaches copyright please contact us at vbn@aub.aau.dk providing details, and we will remove access to the work immediately and investigate your claim.
A New Continuous Discrete Unscented Kalman Filter
Torben Knudsen and John Leth

Abstract—The time and measurement update for the discrete time Kalman filter can be formulated in terms of conditional means and covariances. The unscented Kalman filter can be interpreted as calculating these conditional means and covariances by using the unscented transform. This approach can also be directly applied to nonlinear models as an alternative to the discrete time extended Kalman filter. In this paper, a novel method for computing the unscented Kalman filter for a nonlinear model with continuous time dynamics and discrete time measurements is presented. Compared to the existing approaches, this method is far simpler and less computationally demanding, and it performs at least as well.

Index Terms—Estimation and filtering, Unscented Kalman filter, Stochastic differential equation, Unscented transform, Continuous discrete estimation.

1. INTRODUCTION

Dynamical systems can be modeled as state space models, i.e., a first-order vector ordinary differential equation (ODE) and a static output equation. If there are measurement noise or unknown inputs, they can be modeled as (white noise) stochastic processes. Such a model is called a stochastic differential equation (SDE). Normally, the output vector and the possible input are assumed to be the only signals that can be measured. Based on these measurements, a state estimate is needed in many applications, e.g., forecasting, control, fault detection and system identification.

If both measurements and dynamics are in discrete time and the system is linear, the well-known Kalman filter [1], [2] is the optimal solution in the mean square error sense. This is abbreviated DD-KF, standing for discrete-discrete Kalman filter. A corresponding optimal solution called the CD-KF [3] exists for the CD (continuous-discrete) linear problem where the dynamic state is described by an SDE, and the measurements are at discrete times. For both DD and CD problems with nonlinear systems, the extended Kalman filter (EKF) has been very useful. The EKF simply linearizes the nonlinear system and uses the linearized system parameters where necessary in the TF. For DD nonlinear problems, [4] presents the so-called unscented Kalman filter (UKF), where the need for linearization is avoided by estimating the necessary covariances using the unscented transform (UT) [5], [6] and then directly using them in the DD-KF. This method is referred to as the discrete-discrete unscented Kalman filter (DD-UKF). One of the pioneers in this field compares DD-UKF to DD-EKF as follows: “It is more accurate, easier to implement, and uses the same order of calculations as linearization.” [7, p. 401]. A similar statement is found in [8, p. 158]. This is also our experience.

Inspired by the superior performance of DD-UKF, an algorithm, referred to as CD-UKF in this paper, has been suggested by [9] for computing the UKF in the continuous-discrete case. An earlier alternative to CD-UKF is the ensemble methods (EnKF) discussed by, e.g., [10] and [11] where Monte Carlo methods are used instead of UT methods, which are much more computationally demanding.

It is typical to split the state estimation into a time update and a measurement update. This approach is done in [9] and in this paper (see section II). The measurement update for CD-UKF is similar to that for DD-UKF. The difficulties lie in the time update. The method for the time update in [9] consists of the following steps:

1) Start with the CD system.
2) Obtain a DD system by discretizing the CD system using a small time step $\Delta t$.
3) Set up the DD-UKF equations for the above system.
4) Let the time step $\Delta t$ in the above equations tend to zero, resulting in a set of ordinary differential equations that then become the CD-UKF equations in [9].
5) The above so-called Moment Differential Equations (MDEs) [12] cannot be solved analytically; hence, a numerical method has to be used to integrate from one measurement time to the next.

See [9] for more details, especially regarding the time update by equations (34), (30) and (25) and the measurement update (27), where these equation numbers refer to [9].

The method of [9] is further elaborated by [13], [14]. In [15], a solution based on [9] is provided for a CD system extended with algebraic states. In [16], the time update is performed using the method from [9]; however, for the measurement update, alternative factorizations are suggested and analyzed. The most detailed analysis and development of methods for solving the MDEs numerically is found in the recent work [12]. The paper develops an adaptive solver with an automatic global error control and demonstrates its good performance.

In summary, a number of methods for solving the CD-UKF estimation problem numerically have been suggested. However, all of the above are based on solving the MDEs developed by [9].

For comparison with the method of [9], the time update steps of the method presented in this paper are as follows:

1) Start with the CD system.
2) Obtain a DD system by discretizing the CD system using a small time step $\Delta t$.
3) Use the unscented transform directly for the above system to calculate the necessary covariances and variances to advance from one measurement time to the next.

This is far simpler than the method of [9].

The main contribution of this paper is the new and simple CD-UKF method, which avoids the MDEs used in [9]. It is simpler to explain in an academic context, and it can be explained to practitioners without involving complicated mathematics related to SDEs.

Moreover, we remark that if it is applied to the Lorenz system, the CD-UKF method presented in this paper produces state and output estimates with the same precision as the CD-UKF method in [9] (see Table III) but outperforms it with respect to the accuracy of variance of the state estimate error (see Table IV). Finally, for this example, it runs more than 9 times faster than the CD-UKF method in [9] (see Table III). Using the results reported in [12], this difference might be reduced.

Notation: In what follows, the superscript $T$ indicates transposition, and $\triangleq$ denotes “defined by”. For random vectors $x$, $y$ and $z$, the conditional covariance matrix and conditional variance are denoted

$$
\text{Cov}(x, y|z) \triangleq E \left( (x - E(x|z))(y - E(y|z))^T | z \right),
$$

$$
\text{Cov}(x|z) \triangleq \text{Cov}(x, x|z),
$$

Torben Knudsen and John Leth are with section of Automation and Control, Aalborg University, Fredrik Bajers Vej 7C2-212 DK-9220 Aalborg, Denmark (tk, jjl)@es.aau.dk
where in the unconditional case, these are occasionally denoted by $C_y = \text{Cov}(x, y)$ and $C_{yx} = \text{Cov}(x, z)$, respectively. Finally, the estimate of a value $x$ is denoted by $\hat{x}$, and the estimation error by $\tilde{x} \triangleq x - \hat{x}$.

The remainder of this paper starts by presenting CD-KF for linear systems in section II. Section III presents CD-EKF for nonlinear systems. The UT is presented separately in section IV. The main contribution in the form of the new CD-UKF is developed in section V and compared with the alternative CD-UKF by [9] and CD-EKF in section VI based on a simulated Lorenz attractor example. Finally, section VII gives the conclusion.

II. THE CONTINUOUS–DISCRETE KF

For reference, the basic CD-KF for linear CD systems is first presented.

Consider the linear time varying CD system:

\begin{align}
    dx(t) &= (F(t)x(t) + B(t)u(t)) dt + dw(t), \\
    y(t_k) &= H(t_k)x(t_k) + D(t_k)u(t_k) + v(t_k), \\
    w(t) &\in W(Q(t)), \quad v(t_k) \in N(0, R(t_k)), \quad E\left(v(t_k)v(t_j)^T\right) = \Delta(t_i - t_j)R(t_j),
\end{align}

where $x \in \mathbb{R}^n$ is the state, $u \in \mathbb{R}^m$ is the input, $w \in \mathbb{R}^n$ is a Wiener process, $y \in \mathbb{R}^r$ is the output, and $v \in \mathbb{R}^r$ is the Gaussian measurement noise. The matrices $F, B, H, D$ are state space model parameters of suitable dimensions, $Q$ is the incremental process noise covariance, $R$ is the measurement noise covariance, and $\Delta$ is the discrete Dirac delta function. We remark that the SDE given by (1a) is to be understood in the sense of Ito (see [17]).

Given the measurements and the initial values,

\begin{align}
    Y_0 \triangleq y(0), y(t_1), y(t_2) \ldots y(t_k), \\
    \hat{x}(t_0) = \bar{x}_0, \quad P(t_0) = P_0,
\end{align}

the estimation problem is to find the estimate $\hat{x}(t_k)$ that minimizes the mean square error (MSE)

\begin{equation}
    E\left((x(t_k) - \hat{x}(t_k))^TA(x(t_k) - \hat{x}(t_k))^T\right),
\end{equation}

for any positive semi-definite $A$. It is well known that the general solution, which is also applicable to nonlinear systems, is the conditional mean value

\begin{equation}
    \hat{x}(t_k) = E\left(x(t_k)|Y_k^b\right).
\end{equation}

The CD-KF that solves the problem to the estimation problem for the linear time-varying CD system (1)-(2) can be formulated algorithmically as (5)-(6) below [3, Theo. 7.1]:

The measurement update at time $t_k$ is performed as follows:

\begin{align}
    K(t_k) &= P(t_k^+)^TH(t_k)^T(H(t_k)P(t_k^+)H(t_k)^T + R(t_k))^{-1}, \\
    \hat{x}(t_k^+) &= \hat{x}(t_k^-) + K(t_k)(y(t_k) - H(t_k)\hat{x}(t_k^-) - D(t_k)u(t_k)), \\
    P(t_k^+) &= (I - K(t_k)H(t_k))P(t_k^+)(I - K(t_k)H(t_k))^T + K(t_k)R(t_k)K(t_k)^T.
\end{align}

The time update from $t_k$ to $t_{k+1}$ is performed as follows:

\begin{align}
    \hat{x}(t_k) &= \hat{x}(t_k^+), \quad P(t_k) = P(t_k^+), \\
    \hat{x}(t) &= F(t)\hat{x}(t) + B(t)u(t), \\
    \hat{P}(t) &= F(t)\hat{P}(t)F(t)^T + Q(t), \\
    \hat{x}(t_{k+1}) &= \hat{x}(t_{k+1}), \quad P(t_{k+1}) = P(t_{k+1}),
\end{align}

where (6a) are the initial conditions at time $t_k$ for the ODE (6b)-(6c) used to obtain the time-updated results (6d) for time $t_{k+1}$.

It is, of course, important to mention that the terms in algorithm (5)-(6) have probabilistic interpretations:

\begin{align}
    \hat{x}(t_k^-) &\triangleq E\left(x(t_k)|Y_{k-1}^b\right), \\
    \hat{x}(t_k^+) &\triangleq E\left(x(t_k)|Y_k^b\right), \\
    P(t_k^-) &\triangleq \text{Cov}\left(x(t_k)|Y_{k-1}^b\right), \\
    \hat{x}(t_k^+) &\triangleq E\left((x(t_k) - \hat{x}(t_k^-))(x(t_k) - \hat{x}(t_k^-))^T|Y_{k-1}^b\right), \\
    P(t_k^-) &\triangleq \text{Cov}\left(x(t_k^+)|Y_k^b\right) = E\left((x(t_k) - \hat{x}(t_k^+))(x(t_k) - \hat{x}(t_k^+))^T|Y_k^b\right).
\end{align}

III. THE CONTINUOUS–DISCRETE EKF

Consider now, in place of (1a) and (1b), the nonlinear model (8) below, where the drift $f$ and output $h$ are nonlinear but the noise $v$ is still additive:

\begin{align}
    dx(t) &= f(x(t), u(t), t)dt + dw(t), \\
    y(t_k) &= h(x(t_k), u(t_k), t_k) + v(t_k).
\end{align}

The EKF is then derived from the KF by using the following heuristic principle: use the nonlinear relations when possible and the linearization otherwise. Thus, the following are changed in the nonlinear setting.

The measurement update at time $t_k$ (5b) is replaced by

\begin{equation}
    \hat{x}(t_k^-) = \hat{x}(t_k^-) + K(t_k)(y(t_k^-) - h(\hat{x}(t_k^-), u(t_k))).
\end{equation}

The time update from $t_k$ to $t_{k+1}$ (6b) is replaced by

\begin{equation}
    \hat{x}(t) = f(\hat{x}(t), u(t), t).
\end{equation}

In all the other equations, the linearized parameters (11) must be used.

\begin{align}
    F(t) &\triangleq \frac{\partial f}{\partial x}(\hat{x}(t), u(t), t), \\
    H(t) &\triangleq \frac{\partial h}{\partial x}(\hat{x}(t), u(t), t).
\end{align}

Unless explicitly mentioned, (5) and (6) are henceforth to be understood with the changes (9)-(11).

IV. UNSCENTED TRANSFORM

The basic new idea in the UKF is to use the UT to calculate conditional means and covariances in the measurement (5) and the time update (6). How this is done for the nonlinear CD problem is explained in section V. This section first explains the UT itself.

The problem solved by the UT is the basic probabilistic problem of calculating the second-order statistics of $x, y$, given the second-order statistics of $x$ and a relation $f$:

\begin{equation}
    y = f(x), \quad \mu_x = E(x), \quad C_x = \text{Cov}(x).
\end{equation}

The UT can formally be written as

\begin{equation}
    [\hat{\mu}_y, \hat{C}_{yx}, \hat{C}_y] = \text{UT}(f, \mu_x, C_x).
\end{equation}

From the mathematical point of view, this amounts to approximating integrals of the type

\begin{equation}
    \int_{\mathbb{R}^n} h(z)\phi_x(z)dz,
\end{equation}

where $\phi_x$ is the probability density function for $x$, and, e.g., $h = f$ in the case of $\hat{\mu}_y$. 0018-9286 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

Transactions on Automatic Control
For a nonlinear relation \( f \), there is no exact solution for even the second-order statistics. The EKF can be interpreted as using the linearization approach:

\[
y \approx f(\mu_x) + \nabla f(\mu_x)(x - \mu_x),
\]

implying that

\[
\hat{\mu}_y = f(\mu_x), \quad \hat{C}_{yy} = \nabla f(\mu_x)C_x, \\
\hat{C}_{yx} = \nabla f(\mu_x)C_x \nabla f(\mu_x)^T.
\]

A Monte Carlo approach would involve generating random realizations \( x_i \) based on the statistics of \( x \) and estimating the statistics for \( x, y \) as follows:

\[
\hat{\mu}_y = \frac{1}{N} \sum_{i=1}^{N} f(x_i), \quad \hat{\mu}_x = \frac{1}{N} \sum_{i=1}^{N} x_i, \\
\hat{C}_{yx} = \frac{1}{N} \sum_{i=1}^{N} (f(x_i) - \hat{\mu}_y)(x_i - \hat{\mu}_x)^T, \\
\hat{C}_y = \frac{1}{N} \sum_{i=1}^{N} (f(x_i) - \hat{\mu}_y)(f(x_i) - \hat{\mu}_y)^T,
\]

with \( N \) chosen to be sufficiently large.

The UT can be seen as a Monte Carlo method in which the values of \( x_i \) are not randomly drawn but constructed to obtain certain features \([18, p. 549-550]\). These constructed \( x_i \)s are called sigma points. In particular, if \( x \) is Gaussian, the eigenvectors of \( C_x \) can be used to construct the sigma points; see \([5]\) for details. An introduction to UT can be found in, e.g., \([4]\). There are various versions of UT with different features. The UT used here is given below, where \( n \) is the dimension of \( x \):

\[
\lambda = 2, \\
k = \sqrt{n + \lambda}, \\
u_i \triangleq \text{eigenvector } i \text{ for } C_x, \\
l_i \triangleq \text{eigenvector } i \text{ for } C_x, \\
x_i = \begin{cases} \mu_x, & i = 0 \\ \mu_x + ku_i \sqrt{l_i}, & 1 \leq i \leq n \\ \mu_x - ku_i \sqrt{l_i}, & n + 1 \leq i \leq 2n, \end{cases} \\
w_i = \begin{cases} \lambda \frac{1}{\lambda + 2n}, & i = 0 \\ \frac{1}{\lambda + 2(n+1)}, & 1 \leq i \leq 2n, \end{cases}
\]

\[
\hat{\mu}_y = \sum_{i=0}^{2n} w_i f(x_i), \\
\hat{C}_{yx} = \sum_{i=0}^{2n} w_i (f(x_i) - \hat{\mu}_y)(x_i - \hat{\mu}_x)^T, \\
\hat{C}_y = \sum_{i=0}^{2n} w_i (f(x_i) - \hat{\mu}_y)(f(x_i) - \hat{\mu}_y)^T.
\]

Note that \( \sum_{i=0}^{2n} w_i = 1 \). This construction can be shown to give exact results for linear and affine functions \( f(x) = Ax + b \). Here, eigenvectors are used to construct the sigma points. There are other ways, e.g., Cholesky factorization, which gives correct results for linear functions but different results for nonlinear functions, as shown in the test below.

The above UT has been tested and compared to the UT version by Rudolph van der Merwe and S. J. Julier, that uses Cholesky factorization \([7, 19]\). For reference, a Monte Carlo (MC) method using \( 10^4 \) random samples has also been included. The results for \( y = f(x) = x^2 + x \), i.e., squared length of \( x \), are shown in Table I. The chosen input statistics are \( E(x) = [1; 1] \) and \( \text{Cov}(x) = [1 \; 1 \; 1 \; 2] \) (using the Matlab notation). The MC method assumes a Gaussian distribution of \( x \). This squared length example is chosen because the theoretical results can be derived, as shown in Table I.

It is noted that the UT method \((15)\) is correct up to third-order moments \([7, 20]\). That is, the estimates of \( E(y) \) and \( \text{Cov}(y, x) \) agree with the (correct) theoretical results, while the estimate of \( \text{Cov}(y) \) does not, as it includes \( 4^{th} \) moments. This is true only in the case considered, where \( f \) is the square length of \( x \), which is Gaussian.

The UT method’s performance is also compared to that of the MC method. The two methods perform similarly, even though the MC method uses \( 10^5 \) points compared to 5 points used by the UT methods. The slightly better performance of the Cholesky-based method compared to the UT is not the case in general, as shown in Table II.

<table>
<thead>
<tr>
<th>Method</th>
<th>( E(y) )</th>
<th>( \text{Cov}(y) )</th>
<th>( \text{Cov}(y, x_1) )</th>
<th>( \text{Cov}(y, x_2) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>UT</td>
<td>5.00</td>
<td>39.0</td>
<td>4.00</td>
<td>6.00</td>
</tr>
<tr>
<td>UTCH</td>
<td>5.00</td>
<td>31.0</td>
<td>4.00</td>
<td>6.00</td>
</tr>
<tr>
<td>MC</td>
<td>5.04</td>
<td>33.9</td>
<td>3.99</td>
<td>6.03</td>
</tr>
<tr>
<td>Theo</td>
<td>5</td>
<td>34</td>
<td>4</td>
<td>6</td>
</tr>
</tbody>
</table>

Relative error

<table>
<thead>
<tr>
<th>Method</th>
<th>( E(y) )</th>
<th>( \text{Cov}(y) )</th>
<th>( \text{Cov}(y, x_1) )</th>
<th>( \text{Cov}(y, x_2) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>UT</td>
<td>0.0 - 0.147</td>
<td>0.0 - 0.0882</td>
<td>0.0 - 0.00428</td>
<td>0.0 - 0.00039</td>
</tr>
<tr>
<td>UTCH</td>
<td>0.0 - 0.147</td>
<td>0.0 - 0.0882</td>
<td>0.0 - 0.00428</td>
<td>0.0 - 0.00039</td>
</tr>
<tr>
<td>MC</td>
<td>0.00766</td>
<td>-0.00428</td>
<td>-0.00039</td>
<td>0.00467</td>
</tr>
</tbody>
</table>

**V. CONTINUOUS–DISCRETE UNSCENTED KALMAN FILTER**

The CD-UKF solves the same problem as the CD-EKF, i.e., estimating the state from the outputs of the system \((8)\). To understand the CD-UKF, it is necessary to follow the derivation of the (linear) CD-KF to recognize that the measurement update \((5b)\) and \((5c)\) originates from basic relation for Gaussian variables. Assume that \( x, y \) are Gaussian vectors, then

\[
E(x|y) = E(x) + \text{Cov}(x, y) \text{Cov}(y)^{-1}(y - E(y)), \quad (16a)
\]

\[
\text{Cov}(x|y) = \text{Cov}(x) - \text{Cov}(x, y) \text{Cov}(y)^{-1} \text{Cov}(y, x), \quad (16b)
\]

1 In Table II, the result of the MC method is used as a benchmark because calculating the necessary theoretical statistics for Table II can be very lengthy.
\[ x - E(x|y) \] and \( y \) are independent, (16c)
and the MSE optimal estimate of \( x \) given \( y \) is the conditional mean \( E(x|y) \). Note the important feature of (16c) that the estimation error is independent of the data on which it is based. This fact, and its implications, is used extensively below, e.g., in (18).

Now, let \( y \) denote some information that is already available and \( z \) some new information that is to be obtained. The extra condition \( z \) must then also be included in (16), resulting in

\[
\begin{align*}
E(x|y, z) &= E(x|z) \\
+ \text{Cov}(x, y|z) \text{Cov}(y|z)^{-1} (y - E(y|z)), \\
\text{Cov}(x|y, z) &= \text{Cov}(x|z) \\
- \text{Cov}(x, y|z) \text{Cov}(y|z)^{-1} \text{Cov}(y, x|z).
\end{align*}
\]

Introducing the estimation errors

\[
\tilde{y}(t_k) \triangleq y(t_k) - \hat{y}(t_k) \triangleq y(t_k) - E(\tilde{y}(t_k)|Y_{0}^{k-1}),
\]

\[
\tilde{x}(t_k) \triangleq x(t_k) - \hat{x}(t_k),
\]

then yields

\[
\begin{align*}
\text{Cov}(\tilde{y}(t_k)|Y_{0}^{k-1}) &= \text{Cov}(\tilde{y}(t_k)), \\
E(x(t_k)|\tilde{y}(t_k), Y_{0}^{k-1}) &= E(x(t_k)|Y_{0}^{k-1}) = \hat{x}(t_k), \\
\text{Cov}(x(t_k), \tilde{y}(t_k)|Y_{0}^{k-1}) &= E(\tilde{x}(t_k) - \hat{x}(t_k)) E(\tilde{y}(t_k) - \hat{y}(t_k)|Y_{0}^{k-1}) \\
&= E(\tilde{x}(t_k) - \hat{x}(t_k)) E(\tilde{y}(t_k) - \hat{y}(t_k)) \\
&= \text{Cov}(\tilde{x}(t_k), \tilde{y}(t_k)) = \text{Cov}(\tilde{y}(t_k), \tilde{x}(t_k))^T.
\end{align*}
\]

Using (17) with \( x, y, z \) replaced by \( x(t_k), \tilde{y}(t_k), Y_{0}^{k-1} \) and applying (7) then gives

\[
\begin{align*}
E(x(t_k)|\tilde{y}(t_k), Y_{0}^{k-1}) &= \hat{x}(t_k), \\
+ \text{Cov}(x(t_k), \tilde{y}(t_k)|Y_{0}^{k-1}) \text{Cov}(\tilde{y}(t_k)|Y_{0}^{k-1})^{-1} (y(t_k) - \hat{y}(t_k)), \\
\text{P}((t_k))^\dagger &= \text{P}(t_k) - \text{Cov}(x(t_k), \tilde{y}(t_k)|Y_{0}^{k-1}) \text{Cov}(\tilde{y}(t_k)|Y_{0}^{k-1})^{-1} \text{Cov}(\tilde{x}(t_k), x(t_k)|Y_{0}^{k-1}),
\end{align*}
\]

which, by (18) and (7), results in the following simpler formulas:

\[
\begin{align*}
\tilde{x}(t_k) &= \hat{x}(t_k) \\
+ \text{Cov}(\tilde{x}(t_k), \tilde{y}(t_k)) \text{Cov}(\tilde{y}(t_k)|Y_{0}^{k-1})^{-1} \tilde{y}(t_k), \\
\text{P}((t_k))^\dagger &= \text{P}(t_k) - \text{Cov}(\tilde{x}(t_k), \tilde{y}(t_k)) \text{Cov}(\tilde{y}(t_k)|Y_{0}^{k-1})^{-1} \text{Cov}(\tilde{x}(t_k), \tilde{y}(t_k)).
\end{align*}
\]

This completes the derivation of the basic formulas needed to explain the specific context in which the UT appears.

**Remark 1:** Equation (20) represents the measurement update (5), but in a more general formulation. Indeed, the standard formulation (5) of the linear CD-KF can be derived from (20) as follows:

\[
\begin{align*}
y(t_k) &= Hx(t_k) + Du(t_k) + v(t_k) \\
\hat{y}(t_k) &= H\hat{x}(t_k) + Du(t_k),
\end{align*}
\]

\[
\text{giving } \tilde{y}(t_k) = H\tilde{x}(t_k) + v(t_k) \text{ and thus } \text{Cov}(\tilde{x}(t_k), \tilde{y}(t_k)) = \text{Cov}(\tilde{x}(t_k), \tilde{y}(t_k))^T = \text{P}(t_k)H(t_k)^T.
\]
must represent the integration of the state SDE (29) from \( t_k \) to \( t_{k+1} \) 
starting from \( \dot{x}(t_k^+) \), \( P(t_k^+) \), given \( u(t) \), \( Q(t) \), \( t \in [t_k, t_{k+1}] \). 

\[
dx(t) = f(x(t), u(t))dt + dw(t). 
\]

(29)

The SDE (29) can be simulated by discretizing in time. In general, the specific integration algorithm should relate to the interpretation of the SDE in the sense of either Ito or Stratonovich. However, since the diffusion (equal to 1 here) is independent of \( x \), these two formalisms are identical [17, p.171]. The Euler-Maruyama integration [17], [21] may therefore be used as in (30).

\[
\begin{align*}
\delta t_i & \triangleq t_{i+1} - t_i, \ \delta w_i \triangleq w(t_{i+1}) - w(t_i), \\
x(t_{i+1}) - x(t_i) &= f(x(t_i), u(t_i))\delta t_i + \delta w_i, \\
&= \left( f(x(t_i), u(t_i)) + \frac{\delta w_i}{\delta t_i} \right) \delta t_i, \\
&= (f(x(t_i), u(t_i)) + n_i) \delta t_i, \\
n_i &\in \text{NID}(0, Q(t_i)\delta t_i^{-1}),
\end{align*}
\]

where \( i \) numbers the intermediate steps between \( t_k \) and \( t_{k+1}, \) and \( \text{NID} \) stands for Normally and Independently Distributed, i.e., white Gaussian noise. Similar to \( h_0 \) for the measurement update, an ancillary function \( f_a \) is defined to explain the use of the UT in the time update. Using (30), the final state \( x(t_{k+1}) \) is a function 

\[
x(t_{k+1}) = f_a([x(t_k), N^{k+1}_{k+1}], U^{t_{k+1}}_{t_k}),
\]

of the initial state \( x(t_k), \) stacked noise \( N^{k+1}_{k+1}, \) and input \( U^{t_{k+1}}_{t_k}. \) 

Note that \( N^{k+1}_{k+1} \) and \( U^{t_{k+1}}_{t_k} \) are the stacked noise \( n_i, \) and input \( u(t_i) \) for the intermediate time steps in the integration (30b), respectively. Additionally, note that \( x \) in (30)–(31) is an approximation of \( x \) in the SDE (29). Now, the UT transform can be used with the function \( f_a \) and the input statistics

\[
E\left(x(t_k) \right| N^{k+1})_{t_{k+1}} = \begin{pmatrix} \hat{x}(t_k^+) \\ 0 \\ 0 \end{pmatrix},
\]

(32a)

\[
\text{Cov}\left(x(t_k) \right| N^{k+1})_{t_{k+1}} = \begin{pmatrix} P(t_k^+) & 0 & \cdots & 0 \\ 0 & Q(t_{k+1})/\delta t & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & Q(t_{k+1})/\delta t \end{pmatrix},
\]

(32b)

where the time step \( \delta t \) is assumed constant for simplicity. The result of the UT is then the output-input statistics

\[
\hat{x}(t_k^+) \cdot P(t_k^+) = \text{Cov}(\hat{x}(t_k^+)),
\]

(33)

that are necessary in the UKF time update corresponding to (6).

The choice of sub-sampling time \( \delta t \) is important; it must be chosen to be sufficiently small to guarantee that the errors from discretization are smaller than those from estimation. However, there is a tradeoff, as the number of sigma points from (32) given by \( 2(k_{k+1}-1)/\delta t + 1 \) + 1 increases with decreasing \( \delta t. \) For stiff systems in particular, a variable step-size integration is advantageous [22]. The presented method can easily change the sub-sampling step size \( \delta t \) at sampling times \( t_k; \) however, changing it between sub-samplings appears to be difficult. A detailed study of the effect of sub-sampling time is outside the scope of this paper, instead, the reader is referred to [22].

In summary, this section shows how to use the UT to calculate all the conditional means, variances and covariances necessary for the KF. This is why the method of this paper is called CD-UKF.

VI. TEST OF THE CD-UKF

In this section, the CD-UKF developed in section V is tested by comparing it to two other methods using data from a simulated (nonlinear) Lorenz system. The first method is that described in [9], and the other method is the CD-EKF method presented in this paper. From the theory, it is known that the methods will work perfectly for linear models. In particular, the output prediction errors (residuals) must be white noise. To rule out algorithmic errors, the methods developed by the authors of this paper have been successfully tested for linear models (the test results are not included here).

The comparison with the CD-UKF method of [9] and the CD-EKF is performed on the Lorenz system (34) that is also used by others for assessing filters [10]. Moreover, the deterministic part uses the same parameters (34f) as the original work by Lorenz [23].

\[
dx = f(x)dt + dw, \quad y = h(x) + v,
\]

\[
f(x) = \begin{pmatrix} \sigma(x_2 - x_1) \\ x_1(x_2 - x_3) \end{pmatrix},
\]

(34c)

\[
h(x) = \begin{pmatrix} x_1 \\ x_3 \end{pmatrix},
\]

(34d)

\[
v \in \text{W}(0, Q), \quad v \in \text{NID}(0, R),
\]

(34e)

\[
\sigma = 10, \quad \rho = 28, \quad \beta = 8/3,
\]

(34f)

\[
Q = 4.5^2 I, \quad R = I,
\]

(34g)

\[
T_x = 0.01, \quad x(0) = (1 \ 1 \ 1)^T.
\]

(34h)

The simulation runs from time 0 with a simulation time step of 0.0001 and an output sampling time of 0.01, resulting in 701 samples. Initially, 50 subsamples for the time update step are used for the CD-UKF method in [9]; however, this number could be reduced to 10 before the estimation RMS error started to increase. The sub sampling used for the developed algorithm is then adjusted to obtain the same performance, resulting in 2 subsamples. This choice is based on many simulation runs; the results of one run are shown in Figure 1–3, while Table III contains the statistics for 100 simulations.

Figure 1 shows the simulated states in 3D in blue; the measurements \( y_1, y_2 \) of the states \( x_1, x_3 \) are shown in red in the \( x_1, x_3 \) plane. The estimates of the developed CD-UKF method are shown in green. The output prediction errors are observed to be perfectly white in Figure 2. This is not guaranteed theoretically, but it does show that the filter works nearly optimally at least regarding the “linear” correlation. The so-called lag dependence function (LDF) [24], [25] estimates auto correlations from data, while also accounting for nonlinear dependence. As this is appropriate for the nonlinear system, it is also calculated, but not shown. The result is similar to those in Figure 2 except the non-zero lag correlations are slightly larger, though still small. The state estimation errors are shown in Figure 3 to illustrate that the uncertainty of the states \( x_1, x_3 \) is smaller and relatively stable compared to \( x_2, \) as expected, as there is no measurement of \( x_2 \) (the following shorthand notation \( \hat{x}_i^k \triangleq \hat{x}(t_k^i) \) and \( \hat{y}_i^k \triangleq \hat{y}(t_k^i) \) is used in the figures and tables below).

Normal probability plots for the distribution of output and state prediction errors are shown in Figure 4–5. In the linear case, both will be Gaussian. For the Lorenz system, the output prediction errors appear to be nearly Gaussian, even though there are some deviations in the tails of the distribution. Such tail deviations are more pronounced for the state prediction errors, which then seem less Gaussian. This applies in particular to \( \hat{x}_2, \) which has no “direct” measurement.

Table III shows the primary comparison results with two decimals of precision (the normalized values in all tables are obtained using
State prediction errors with 2 standard deviations error bounds

Fig. 1. Simulated states (blue) and measurements (red) for the Lorenz system. The estimated states using the developed CD-UKF are also included and are shown in green.

Fig. 2. Whiteness test for residuals from the CD-UKF. Upper left and lower right subplots show autocorrelations for $\tilde{y}_1$ and $\tilde{y}_2$, respectively (the $p$ values are from the Portmanteau [26] whiteness test). The upper right plot shows the cross correlation between the two.

State prediction errors with 2 standard deviations error bounds

Fig. 3. State prediction errors (residuals) including 95% confidence limits from the CD-UKF, vs. time

Fig. 4. Normal probability plots for output prediction errors (residuals) from the CD-UKF.

Fig. 5. Normal probability plots for state prediction errors (residuals) from the CD-UKF.

precise values for the numerator and denominator). It provides the mean and standard deviation based on 100 simulations of RMS errors, and the computation time (see also the Appendix.) As explained above, the accuracy values of the filters are made similar by tuning the number of subsampling time steps. The RMS of the state estimates using only the measurement would be 1, $\infty$ and 1 for $x_1$, $x_2$, $x_3$, respectively, due to the measurement (34d). This agrees well with the results in Table III, as the RMS values are below 1, $\infty$ and 1, respectively. The only difference in the methods' performances is that CD-UKF and CD-EKF are at least 9 times faster than the version reported in [9], denoted here by CD-UKFSS.

In many applications, the accuracy of the variance $\text{Cov}(\tilde{x}_i(t_k^+))$ of the state estimate error $\tilde{x}_i(t_k^+)$ is as important as the state estimate itself. Therefore, this accuracy is also assessed by calculating the RMS of the normalized error

$$\tilde{x}_{n,i}(t_k^+) = \frac{\tilde{x}_i(t_k^+)}{\sqrt{P(t_k^+)}_{ii}}$$

(35)

with the estimated standard deviation $\sqrt{P(t_k^+)}_{ii}$ being the square root of the $i^{th}$ diagonal element of $P(t_k^+) = \text{Cov}(\tilde{x}_i(t_k^+)|Y_0^k)$. If
Comparing the performance of estimation methods for the Lorenz system. The means and standard deviations are based on 100 simulations. The first five columns (in blocks one and three) show RMS of state estimation errors and output prediction errors, and the sixth column shows the computation time.

<table>
<thead>
<tr>
<th>Method</th>
<th>Mean</th>
<th>Standard Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(\tilde{x}_{1}^{+})</td>
<td>(\tilde{x}_{2}^{+})</td>
</tr>
<tr>
<td>CD-UKFSS</td>
<td>0.59</td>
<td>1.39</td>
</tr>
<tr>
<td>CD-EKF</td>
<td>0.58</td>
<td>1.37</td>
</tr>
<tr>
<td>CD-UKF</td>
<td>0.58</td>
<td>1.37</td>
</tr>
<tr>
<td>Normalized</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CD-EKF/CD-UKFSS</td>
<td>0.99</td>
<td>0.98</td>
</tr>
<tr>
<td>CD-UKF/CD-UKFSS</td>
<td>0.99</td>
<td>0.98</td>
</tr>
<tr>
<td>CD-UKF/CD-EKF</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

**APPENDIX**

**TABLE VI**

RMS values of normalized state estimation errors. The minimum and maximum are based on 100 simulations.

<table>
<thead>
<tr>
<th>Method</th>
<th>Minimum</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(\tilde{x}_{1}^{+})</td>
<td>(\tilde{x}_{2}^{+})</td>
</tr>
<tr>
<td>CD-UKFSS</td>
<td>0.53</td>
<td>0.53</td>
</tr>
<tr>
<td>CD-EKF</td>
<td>0.53</td>
<td>0.53</td>
</tr>
<tr>
<td>CD-UKF</td>
<td>0.53</td>
<td>0.53</td>
</tr>
<tr>
<td>Normalized</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CD-EKF/CD-UKFSS</td>
<td>0.99</td>
<td>0.99</td>
</tr>
<tr>
<td>CD-UKF/CD-UKFSS</td>
<td>0.99</td>
<td>0.99</td>
</tr>
<tr>
<td>CD-UKF/CD-EKF</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

**REFERENCES**


