Computers and Chemical Engineering 33 (2009) 1386–1401

Contents lists available at ScienceDirect

Computers and Chemical Engineering

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

Constrained nonlinear state estimation based on the UKF approach

S. Kolås[a,b,*], B.A. Foss[a], T.S. Schei[c]

[a] Departmentof Engineering Cybernetics, NTNU, N-7491 Trondheim, Norway
[b] Hydro Aluminium AS, Technology & Competence, N-6884 Øvre Årdal, Norway
[c] Cybernetica AS, N-7038 Trondheim, Norway

ARTICLE INFO

Article history:
Received 21 March 2008
Received in revised form 14 December 2008
Accepted 5 January 2009
Available online 31 January 2009

Keywords:
Unscented
Kalman filter
Nonlinear state estimation
Constrained nonlinear systems

ABSTRACT

In this paper we investigate the use of an alternative to the extended Kalman filter (EKF), the unscented Kalman filter (UKF). First we give a broad overview of different UKF algorithms, then present an extension to the ensemble of UKF algorithms, and finally address the issue of how to add constraints using the UKF approach. The performance of the constrained approach is compared with EKF and a selection of UKF algorithms on nonlinear process systems with multimodal probability density functions. The conclusion is that with an algebraic reformulation of the correction part, the reformulated UKF shows strong performance on our selection of nonlinear constrained process systems.

1. Introduction

Nonlinear state estimation is a broad field[1] which includes many algorithms such as Moving Horizon Estimation, the Particle filter, the Ensemble Kalman filter, the Unscented Kalman filter and the Extended Kalman filter, just to mention some. The extended Kalman filter (EKF), which was originally proposed by Stanley Schmidth in 1967 (Bellatoni & Dodge, 1967) in order to apply the Kalman filter to nonlinear spacecraft navigation problems, is probably the most widely used method in applied nonlinear state estimation. However, several authors have experienced shortcomings applying the EKF to systems with severe nonlinearity and/or constraints (see e.g. Julier & Uhlmann, 1994; Julier, Uhlmann, & Durrant-Whyte, 1995; Schei, 1997; Nørgaard, Poulsen, & Ravn, 2000; Rao, 2000; Hasseltine & Rawlings, 2003; Bizup & Brown, 2003; Chen, Lang, Bakshi, Goel, & Ungarla, 2007; Vachhani, Narasimhan, & Rengaswamy, 2006; Kandepu, Foss, & Imsland, 2008). The shortcomings are related to difficulties in determining the Jacobians, errors introduced by linearization and/or the ability to deal with systems with multimodal or asymmetric probability density functions (pdf).

State estimation may introduce some challenges in systems based on first principles. In e.g. chemical processes one may wish to estimate concentrations. Mathematically the model may allow negative concentration, whilst this is not physically possible. A first principle model is also often nonlinear, and theory based on Gaussian noise may not be applicable, since Gaussian noise propagated through a nonlinear model is no longer Gaussian. Further, nonlinearities may be severe such that theory based on linearizations are too inaccurate and provide poor estimation accuracy. At last the nonlinear system may have a skew or multimodal probability density function. If $\hat{x}_k$ is an estimate of concentrations in a physical system, we may introduce constraints in order to force the estimates to have physical meaning. Hence the use of constraints may be important.

If handling of constraints is unavoidable, the EKF has some limitations in propagating the constraints both through the state and covariance calculations. To overcome limitations in the EKF, several approaches and alternatives have been suggested. In this paper, we focus on the Unscented Kalman filter (UKF). In particular several variants of the UKF algorithm with constraint handling are investigated on cases where the EKF has shown bad performance. It is, however, outside the scope of the paper to include an in-depth discussion on MHE and Particle filters beyond the fact that we use examples on which such algorithms have been evaluated, see Hasseltine and Rawlings (2003) and Rawlings and Bakshi (2006).

Literature discussing constraint implementation in the UKF approach is rather limited. In Julier and Uhlmann (1994) the idea of including constraints in the UKF approach is in fact listed as a general possibility, but without being further discussed. The main contributions on constraint handling are Vachhani et al. (2006) and Kandepu et al. (2008). Further, Li and Leung (2004) use equality constraints on the corrected estimate, and Julier and Laviola (2007) discuss two methods for nonlinear equality constraint handling.

* Corresponding author at: Department of Engineering Cybernetics, NTNU, N-7491 Trondheim, Norway.
E-mail addresses: steinar.kolas@itk.ntnu.no, steinar.kolas@hydro.com (S. Kolås).

[1] For a review of nonlinear state estimation see Daum (1986) and Daum (2005).

0098-1354/$ – see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compchemeng.2009.01.012

This paper is organized as follows. First we introduce the Kalman filter-philosophy based on the work by Rudolph Kalman (Kalman, 1960), before we introduce the reader to the UKF and present several algorithms based on this approach. Subsequently we propose an extension to the UKF framework by reformulating the correction step. This reformulation, combined with different constraint handling techniques, results in good performance on the selected cases. We also show that the selection of the square root algorithm may be of importance when it comes to convergence speed of the UKF. Further, we extend the work of Vachhani et al. (2006) to Nonlinear Programming (NLP) UKF by presenting the full state estimation algorithm for alternative formulations of the UKF. Moreover, we show how this algorithm can be simplified to a Quadratic Program (QP) UKF given some additional system constraints. Various simulation examples (taken from Hasseltine & Rawlings, 2003), are used to show how a selection of the algorithms performs before a discussion and conclusions end the paper.

2. System description

In this work we address the general discrete time nonlinear system given by the state space formulation

$$x_k = f(x_{k-1}, u_{k-1}, v_{k-1}) \qquad x_0 - \text{given} \tag{1}$$

$$y_{k-1} = h(x_{k-1}, w_{k-1}) \tag{2}$$

where x_k denotes the system states, y_k the model output (or measurements), u_k deterministic control inputs, v_k system state noise input (stationary or time-varying stochastic), w_k output (measurement) noise (stationary or time-varying stochastic), and $f(\cdot)$ and $h(\cdot)$ are nonlinear functions. Note that $f(\cdot)$ can also be considered as the integration of the continuous-time transition function over a unit sample-time interval (see e.g. Vachhani et al., 2006). All variables are vectors with appropriate dimensions. Linear systems are given by

$$f(x_k, u_k, v_k) = A_k x_k + B_k u_k + C_k v_k \tag{3}$$

$$h(x_k, w_k) = D_k x_k + w_k \tag{4}$$

covering both linear time-varying (LTV) systems and linear time-invariant (LTI) systems. A_k, B_k, C_k and D_k are matrices with appropriate dimensions describing the linear relationship between the variables. Estimated values are denoted with a $\hat{}$, typically $\hat{x}_k$, $\hat{y}_k$, etc. Further, the simulator and the estimator share the same model, and hence there is no modelling error, although model errors are introduced through the noise inputs v_k and w_k.

3. The Kalman-approach to state estimation

In 1960 Rudolph Kalman (1960) presented a recursive state estimation method, which has become known as the Kalman filter. The important contribution was the fact that a recursive algorithm could be used to accurately compute the first and second order moments (mean and covariance) of a linear system corrupted by Gaussian white noise on its system and output models. However, it may be worth mentioning that Kalman did not restrict the Kalman filter to linear time invariant systems with Gaussian distributions, nor did he assume any specific probability distribution (Julier & Uhlmann, 2004). Kalman's assumptions were that the system random variables could be consistently estimated by sequentially updating the first and second order moments (mean and covariance), and that the estimator was on the linear form

$$\hat{x}_k = \hat{x}_k^- + K_k \cdot e_k \tag{5}$$

The Kalman filter consists of two parts; a forward prediction part and a correction part. The prediction part computes a predicted (a prior) estimate of the first and second order moments at time k given information up until $k-1$ denoted by $\hat{x}_k^-$ and $P_{x_k}^-$. The correction part computes the corrected (posterior) estimates $\hat{x}_k$ and P_{x_k} using available data at time k. In (5) the predicted estimate ($\hat{x}_k^-$) is updated by a linear gain (K_k) times an error (e_k). The error part is the deviation between an output value(s) (y_k) and the prediction of the said output(s) ($\hat{y}_k$). Hence, (5) can be formulated as

$$\hat{x}_k = \hat{x}_k^- + K_k(y_k - \hat{y}_k) \tag{6}$$

where the linear gain (K_k), or the so-called Kalman gain, which for the algorithms discussed in this paper, can be shown to yield the solution

$$K_k = P_{x_k y_k} P_{y_k y_k}^{-1} \tag{7}$$

where $P_{x_k y_k}$ and $P_{y_k y_k}$ are the cross covariance and output covariance, respectively.

All Kalman filters do follow the prediction-correction structure. As a vehicle to compare alternative algorithms, we have divided the algorithm into steps as shown in Table 1. In the tables, used throughout this paper to describe the algorithms, we have chosen to keep the column labeled 'Algorithm step' the same for all algorithms, although all steps do not appear in all algorithms. This is because we think this will help in comparing the steps involved in each algorithm. If a step is not applicable to an algorithm, it is indicated by an '–' in the right most column.

Note that all algorithms must be initialized with first and second order moment state information. The prediction and correction part of an algorithm is covered by steps 1–6 and steps 7–12, respectively.

Table 1
The table explains the meaning of each step used throughout the paper.

Step no.	Algorithm step	Meaning
1	χ_{k-1}	Sigma points (see Chapter 4.3)
2	$\chi_{k,i}^{x^-}$	Propagated state sigma points
3	$\hat{x}_k^-$	Predicted state estimate
4	$P_{x_k}^-$	Predicted covariance
5	$\gamma_{k,i}$	Propagated output sigma points
6	$\hat{y}_k$	Predicted output
7	$P_{y_k y_k}$	Output covariance
8	$P_{x_k y_k}$	Cross covariance
9	K_k	Kalman gain
10	$\chi_{k,i}^{x}$	Corrected sigma points
11	$\hat{x}_k$	Corrected state estimate
12	P_{x_k}	Corrected covariance

4. Recursive state estimators

The Kalman-filter structure is widely applied for nonlinear state estimation. The subsequent sections describe the state estimators studied in this work. First the Kalman filter (KF) and the Extended Kalman filter (EKF) are introduced to serve as a comparison for the Jacobian free algorithms to follow. Note that the formulation of both the KF and the EKF may be unfamiliar, since several algebraic steps are kept in order to show the 1:1 relationship of the algorithms. This results in seemingly more computational steps than in the common formulation of KF/EKF.

4.1. Kalman filter

If one assume the linear time invariant (LTV) system given by (3) and (4) with the system covariance Q_k and output covariance R_k

Table 2
The KF algorithm.

χ_{k-1}	–
$\chi^{x-}_{k,i}$	–
$\hat{x}^-_k =$	$A_{k-1}\hat{x}_{k-1} + B_{k-1}u_{k-1} + C_{k-1}v_{k-1}$
$P^-_{x_k} =$	$A_{k-1}P_{\hat{x}_{k-1}}A^T_{k-1} + C_{k-1}Q_{k-1}C^T_{k-1}$
$\gamma_{k,i}$	–
$\hat{y}_k =$	$D_{k-1}\hat{x}_{k-1} + w_{k-1}$
$P_{y_k y_k} =$	$D_k P^-_{x_k} D^T_k + R_k$
$P_{x_k y_k} =$	$P^-_{x_k} D^T_k$
$K_k =$	$P_{x_k y_k} P^{-1}_{y_k y_k}$
$\chi^x_{k,i}$	–
$\hat{x}_k =$	$\hat{x}^-_k + K_k(y_k - \hat{y}_k)$
$P_{x_k} =$	$(I - K_k D_k)P^-_{x_k}$

Table 3
The EKF algorithm.

χ_{k-1}	–
$\chi^{x-}_{k,i}$	Calculate Jacobians $\nabla f_{\hat{x}_{k-1}}, \nabla f_{v_{k-1}}$
$\hat{x}^-_k =$	$f(\hat{x}_{k-1}, u_{k-1}, v_{k-1})$
$P^-_{x_k} =$	$\nabla f_{\hat{x}_{k-1}} P_{x_{k-1}} \nabla f^T_{\hat{x}_{k-1}} + \nabla f_{v_{k-1}} Q_{k-1} \nabla f^T_{v_{k-1}}$
$\gamma_{k,i}$	Calculate Jacobians $\nabla h_{\hat{x}^-_k}, \nabla h_{w_k}$
$\hat{y}_k =$	$h(\hat{x}^-_k, w_k)$
$P_{y_k y_k} =$	$\nabla h_{\hat{x}^-_k} P^-_{x_k} \nabla h^T_{\hat{x}^-_k} + \nabla h_{w_k} R_k \nabla h^T_{w_k}$
$P_{x_k y_k} =$	$P^-_{x_k} \nabla h^T_{x_k}$
$K_k =$	$P_{x_k y_k} P^{-1}_{y_k y_k}$
$\chi^x_{k,i}$	–
$\hat{x}_k =$	$\hat{x}^-_k + K_k(y_k - \hat{y}_k)$
$P_{x_k} =$	$(I - K_k \nabla h_{\hat{x}^-_k})P^-_{x_k}$

and corrupted by Gaussian noise $v_k = N(\bar{v}, Q)$, $w_k = N(\bar{w}, R)$, ($\bar{v} = \bar{w} = 0$), the Kalman filter approximates the said system as given in Table 2.

The EKF is based on (cf. Table 3) the nonlinear system given by (1) and (2) with the same noise models as for the KF.

One should notice the slightly different definition of the predicted output between the KF and the EKF. We have used the Jacobians $\nabla f_{\hat{x}_{k-1}} = (\partial/\partial x)f(x,u,v)|_{x=\hat{x}_{k-1}, u=u_{k-1}, v=v_{k-1}}$, $\nabla f_{v_{k-1}} = (\partial/\partial v)f(x,u,v)|_{x=\hat{x}_{k-1}, u=u_{k-1}, v=v_{k-1}}$, $\nabla h_{\hat{x}^-_k} = (\partial/\partial x)h(x,w)|_{x=\hat{x}^-_k, w=w_{k-1}}$ and $\nabla h_{w_k} = (\partial/\partial w)h(x,w)|_{x=\hat{x}^-_k, w=w_{k-1}}$. Since the Jacobian $\nabla h_{\hat{x}^-_k}$ and ∇h_{w_k} are calculated around $\hat{x}^-_k$ (the predicted state estimate) this yields a difference since the KF uses $\hat{x}_{k-1}$ (the corrected state estimate). Note also that we have placed the calculation of the Jacobians in the steps $\chi^{x-}_{k,i}$ and $\gamma_{k,i}$ to illustrate that the calculation of the Jacobians, with respect to performance, can be compared to the sigma points step used in the UKF algorithms presented later.

4.2. The unscented Kalman filter (UKF)

The Unscented Kalman[2] filter, proposed by Julier and Uhlmann (1994),[3] is based on the intuition that "*it is easier to approximate a probability distribution than a nonlinear function*" (Julier, Uhlmann, & Durrant-Whyte, 2000; Julier, 2002). By using the Unscented transform to compute the mean and covariance, the Unscented Kalman filter avoids the need to use Jacobians in the algorithm. The probability distribution is approximated by a set of deterministic points which captures the mean and covariance of the distribution. These points, called sigma points, are then processed through the nonlinear model of the system, producing a set of propagated sigma points. By choosing appropriate weights, the weighted average and the weighted outer product of the transformed points provides an estimate of the mean (for example $\hat{x}^-_k$) and covariance (for example $P^-_{x_k}$) of the transformed distribution. To elaborate, a sigma point set, χ^x_k is a matrix which contains states or sigma points $\chi^x_k = [\chi^x_{k,1} \quad \cdots \quad \chi^x_{k,l}]$, where l typically equals $2n+1$ if $x_k \in R^n$.

Passing the vector of sigma points $\chi^x_{k-1,i}$ through a nonlinear function results in the propagated sigma points

$$\chi^x_{k,i} = g(\chi^x_{k-1,i}) \tag{8}$$

and the approximated mean $\hat{x}_k$ and the covariance P_{x_k} can be calculated as

$$\hat{x}_k = \sum_{i=1}^{2n} W^x_i \cdot \chi^x_{k,i} \tag{9}$$

$$P_{x_k} = \sum_{i=1}^{2n} W^c_i (\chi^x_{k,i} - \hat{x}_k)(\chi^x_{k,i} - \hat{x}_k)^T \tag{10}$$

Several UKF algorithms have been proposed (for an overview see van der Merwe, 2004) and they differ mainly in how the noise (system noise and output noise) is injected into the system, how the set of sigma point is selected and how the weights are calculated. There are also UKF algorithms based on square root formulations (see van der Merwe, 2004) and UKF algorithms addressing nonlinear state model and linear output (measurement) model (also known as Rao–Blackwellized UKF's) (Briers, Maskell, & Wright, 2003; Hao, Xiong, Sun, & Wang, 2007). Neither the square root UKF nor the Rao–Blackwellized UKF algorithms have our attention in this work.

In the next chapters we will present several UKF algorithms, but before they are presented, some words about the selection of the sigma points and the weights may be appropriate.

In the original work by Julier and Uhlmann (1994) a symmetric sigma point set was proposed, resulting in $2n+1$ sigma points where $x_k \in R^n$. However, the calculation of the sigma points is considered the most computational demanding steps in the UKF, resulting in an effort to reduce the number of sigma points. This resulted in the 'Simplex Sigma point set'/'Minimal skew simplex sigma point set' (Julier & Uhlmann, 2002, 2004), the 'Spherical simplex sigma point set' (Julier, 2003) and the 'Higher order unscented filter' (Tenne & Singh, 2003). Throughout this paper we are using the symmetric sigma point scheme as a basis.

Having the mean and covariance available, the symmetric sigma point set χ_k is selected by constructing the matrix

$$\chi_k = \begin{bmatrix} \hat{x}_k + \Gamma\sqrt{P_{x_k}} & \hat{x}_k - \Gamma\sqrt{P_{x_k}} \end{bmatrix} \tag{11}$$

where Γ is a scaling factor. Note also that the notation $\hat{x}_k \pm \Gamma\sqrt{P_{x_k}}$ means that the vector $\hat{x}_k$ is added/subtracted to each column in the matrix $\Gamma\sqrt{P_{x_k}}$. In Julier and Uhlmann (1994) they added the freedom to include in χ_k any multiples of the mean $\hat{x}_k$, since it would not affect the mean, only the scaling factor Γ (Julier & Uhlmann, 1994) leading to the sigma point calculation

$$\chi_k = \begin{bmatrix} \hat{x}_k & \hat{x}_k + \Gamma\sqrt{P_{x_k}} & \hat{x}_k - \Gamma\sqrt{P_{x_k}} \end{bmatrix} \tag{12}$$

In the UKF literature, (12) is the most applied method in selecting the sigma point set. We will later introduce an UKF algorithm based

[2] In the original work of Julier and Uhlmann (1994) the Unscented Kalman filter was named 'The New Filter'.

[3] We refer to this document as the first work of UKF, since it was submitted to IEEE Transactions on Automatic Control in 1994, although not accepted before in 2000.

on the sigma point selection given by (11). Further, in the mentioned literature, three different ways of calculating the weights are proposed. We will, however, try to merge these into one set of equations and suggest different sets of parameters in order to achieve each of them. From this, a general scaling factor Γ can be formulated as

$$\Gamma = \sqrt{n+\lambda} \tag{13}$$

where

$$\lambda = \kappa \tag{14}$$

is suggested in the original work of Julier and Uhlmann (1994). Julier and Uhlmann (1994) and Julier et al. (2000) proposed

$$\kappa = 3 - n \tag{15}$$

if the system is Gaussian.[4] However care should be taken selecting κ, since $\kappa < 0$ may lead to negative definite covariance calculations.

Motivated by making the scales independent of the system size, the Scaled Unscented Transform was introduced (Julier, 2002), and further extended by van der Merwe (2004) suggesting

$$\lambda = \alpha^2(n+\kappa) - n \tag{16}$$

The weights W_i^x and W_i^c are calculated as

$$\begin{aligned} W_0^x &= \lambda/(n+\lambda) \\ W_0^c &= \lambda/(n+\lambda) + (1-\alpha^2+\beta) \\ W_i^x &= W_i^c = 1/(2(n+\lambda)) \end{aligned} \tag{17}$$

van der Merwe (2004) proposes

$$[\,\alpha \quad \beta \quad \kappa\,] = [\,1 \quad 2 \quad 0\,] \tag{18}$$

if the system is Gaussian. Note that with

$$[\,\alpha \quad \beta\,] = [\,1 \quad 0\,] \tag{19}$$

one achieves the original proposed scaling factor and weights as proposed in the original work of Julier and Uhlmann (1994). When using (11) as the basis for the sigma point selection

$$[\,\alpha \quad \beta \quad \kappa\,] = [\,1 \quad 0 \quad 0\,] \tag{20}$$

should be used according to (Simon, 2006) in applying (17). Calculating the sigma points involves computing the matrix square root[5] of $\sqrt{P_{x_k}}$, and any matrix square root method can be used (Julier & Uhlmann, 2004). The Cholesky decomposition is the preferred method, producing a lower triangular matrix. However, one should also consider using the symmetric square root (producing a symmetric matrix), since the choice of matrix square root affects the errors in the higher order terms by adjusting the way in which the errors between the approximated Taylor series and the true Taylor series are distributed among the different states (Julier et al., 2000). As will be demonstrated in the simulation chapter, this may affect convergence performance.

4.2.1. UKF with additive noise

Assume the system given by (1) and (2) with the system covariance Q_k and output covariance R_k and corrupted by additive Gaussian noise $v_k = N(\bar{v}, Q)$, $w_k = N(\bar{w}, R)$. The UKF for this system is given in Table 4. Note that $\chi_{k-1} = \chi_{k-1}^x$. Note also that the notation $\hat{x}_{k-1} \pm \Gamma\sqrt{P_{x_{k-1}}}$ means that the vector $\hat{x}_{k-1}$ is added/subtracted to each column in the matrix $\Gamma\sqrt{P_{x_{k-1}}}$.

Table 4
The UKF algorithm proposed by Simon (2006).

$\chi_{k-1} =$	$\left[\hat{x}_{k-1} + \Gamma\sqrt{P_{x_{k-1}}} \quad \hat{x}_{k-1} - \Gamma\sqrt{P_{x_{k-1}}}\right]$
$\chi_{k,i}^{x-} =$	$f(\chi_{k-1,i}^x, u_{k-1})$
$\hat{x}_k^- =$	$\sum_{i=1}^{2n} W_i^x \cdot \chi_{k,i}^{x-}$
$P_{x_k}^- =$	$\sum_{i=1}^{2n} W_i^c(\chi_{k,i}^{x-} - \hat{x}_k^-)(\chi_{k,i}^{x-} - \hat{x}_k^-)^T + Q_k$
$\gamma_{k,i} =$	$h(\chi_{k,i}^{x-})$
$\hat{y}_k =$	$\sum_{i=1}^{2n} W_i^x \cdot \gamma_{k,i}$
$P_{y_k y_k} =$	$\sum_{i=1}^{2n} W_i^c(\gamma_{k,i} - \hat{y}_k)(\gamma_{k,i} - \hat{y}_k)^T + R_k$
$P_{x_k y_k} =$	$\sum_{i=1}^{2n} W_i^c(\chi_{k,i}^{x-} - \hat{x}_k^-)(\gamma_{k,i} - \hat{y}_k)^T$
$K_k =$	$P_{x_k y_k} P_{y_k y_k}^{-1}$
$\chi_{k,i}^x$	–
$\hat{x}_k =$	$\hat{x}_k^- + K_k(y_k - \hat{y}_k)$
$P_{x_k} =$	$P_{x_k}^- - K_k P_{y_k y_k} K_k^T$

Note the noise is assumed additive.

Table 5
The UKF algorithm with additive noise ((Julier et al., 1995); van der Merwe, 2004).

$\chi_{k-1} =$	$\left[\hat{x}_{k-1} \quad \hat{x}_{k-1} + \Gamma\sqrt{P_{x_{k-1}}} \quad \hat{x}_{k-1} - \Gamma\sqrt{P_{x_{k-1}}}\right]$
$\chi_{k,i}^{x-} =$	$f(\chi_{k-1,i}^x, u_{k-1})$
$\hat{x}_k^- =$	$\sum_{i=0}^{2n} W_i^x \cdot \chi_{k,i}^{x-}$
$P_{x_k}^- =$	$\sum_{i=0}^{2n} W_i^c(\chi_{k,i}^{x-} - \hat{x}_k^-)(\chi_{k,i}^{x-} - \hat{x}_k^-)^T + Q_k$
$\gamma_{k,i} =$	$h(\chi_{k,i}^{x-})$
$\hat{y}_k =$	$\sum_{i=0}^{2n} W_i^x \cdot \gamma_{k,i}$
$P_{y_k y_k} =$	$\sum_{i=0}^{2n} W_i^c(\gamma_{k,i} - \hat{y}_k)(\gamma_{k,i} - \hat{y}_k)^T + R_k$
$P_{x_k y_k} =$	$\sum_{i=0}^{2n} W_i^c(\chi_{k,i}^{x-} - \hat{x}_k^-)(\gamma_{k,i} - \hat{y}_k)^T$
$K_k =$	$P_{x_k y_k} P_{y_k y_k}^{-1}$
$\chi_{k,i}^x$	–
$\hat{x}_k =$	$\hat{x}_k^- + K_k(y_k - \hat{y}_k)$
$P_{x_k} =$	$P_{x_k}^- - K_k P_{y_k y_k} K_k^T$

The algorithm in Table 4 uses (11) in selecting the sigma point set χ_{k-1} and an optional step using the updated propagated sigma points calculated as $\chi_k^{x-} = \left[\hat{x}_k^- + \Gamma\sqrt{P_{x_k}^-} \quad \hat{x}_k^- - \Gamma\sqrt{P_{x_k}^-}\right]$ before propagating $\chi_{k,i}^{x-}$ through the output function calculating $\gamma_{k,i}$.

In Julier et al. (1995), (12) is used when selecting the sigma point set, thus leading to $2n+1$ sigma points, as opposed to $2n$ sigma points used in the algorithm described in Table 4.

Based on the work by Julier et al. (1995) and van der Merwe (2004) the algorithm with augmented sigma points is as shown in Table 5.

As in the algorithm by in Table 4, an updated sigma point set can be calculated before propagating the sigma points through the output function. van der Merwe (2004) proposes an alternative method, and claims that using the updated sigma points removes any third order moments from the original propagated sigma point set $\chi_{k,i}^{x-}$. Hence, if this is a problem, one may consider the method proposed by van der Merwe (2004).

4.2.2. UKF with augmented system noise

The same system as in Section 4.2.1 is considered. In the original work of Julier and Uhlmann (1994), the system noise is not assumed additive, but is augmented into the sigma point set. The

[4] It can be shown that $(n+\kappa) = 3$ minimizes the difference between the moments in the Taylor series of the standard Gaussian and the sigma points up to the fourth order (Julier et al., 2000).

[5] Matrix square root is a method such that $P = \sqrt{P}(\sqrt{P})^T$.

Table 6
The UKF algorithm with augmented process noise (Julier & Uhlmann, 1994).

$\chi_{k-1} =$	$\left[\hat{x}^a_{k-1} \quad \hat{x}^a_{k-1} + \Gamma\sqrt{P^a} \quad \hat{x}^a_{k-1} - \Gamma\sqrt{P^a}\right]$
$\chi^{x-}_{k,i} =$	$f(\chi^x_{k-1,i}, u_{k-1}, \chi^v_{k-1,i})$
$\hat{x}^-_k =$	$\sum_{i=0}^{2n} W^x_i \cdot \chi^{x-}_{k,i}$
$P^-_{x_k} =$	$\sum_{i=0}^{2n} W^c_i (\chi^{x-}_{k,i} - \hat{x}^-_k)(\chi^{x-}_{k,i} - \hat{x}^-_k)^T$
$\gamma_{k,i} =$	$h(\chi^{x-}_{k,i})$
$\hat{y}_k =$	$\sum_{i=0}^{2n} W^x_i \cdot \gamma_{k,i}$
$P_{y_k y_k} =$	$\sum_{i=0}^{2n} W^c_i (\gamma_{k,i} - \hat{y}_k)(\gamma_{k,i} - \hat{y}_k)^T + R_k$
$P_{x_k y_k} =$	$\sum_{i=0}^{2n} W^c_i (\chi^{x-}_{k,i} - \hat{x}^-_k)(\gamma_{k,i} - \hat{y}_k)^T$
$K_k =$	$P_{x_k y_k} P^{-1}_{y_k y_k}$
$\chi^x_{k,i}$	–
$\hat{x}_k =$	$\hat{x}^-_k + K_k(y_k - \hat{y}_k)$
$P_{x_k} =$	$P^-_{x_k} - K_k P_{y_k y_k} K^T_k$

Table 7
The fully augmented UKF algorithm (van der Merwe, 2004).

$\chi_{k-1} =$	$\left[\hat{x}^a_{k-1} \quad \hat{x}^a_{k-1} + \Gamma\sqrt{P^a} \quad \hat{x}^a_{k-1} - \Gamma\sqrt{P^a}\right]$
$\chi^{x-}_{k,i} =$	$f(\chi^x_{k-1,i}, u_{k-1}, \chi^v_{k-1,i})$
$\hat{x}^-_k =$	$\sum_{i=0}^{2n} W^x_i \cdot \chi^{x-}_{k,i}$
$P^-_{x_k} =$	$\sum_{i=0}^{2n} W^c_i (\chi^{x-}_{k,i} - \hat{x}^-_k)(\chi^{x-}_{k,i} - \hat{x}^-_k)^T$
$\gamma_{k,i} =$	$h(\chi^{x-}_{k,i}, \chi^w_{k,i})$
$\hat{y}_k =$	$\sum_{i=0}^{2n} W^x_i \cdot \gamma_{k,i}$
$P_{y_k y_k} =$	$\sum_{i=0}^{2n} W^c_i (\gamma_{k,i} - \hat{y}_k)(\gamma_{k,i} - \hat{y}_k)^T$
$P_{x_k y_k} =$	$\sum_{i=0}^{2n} W^c_i (\chi^{x-}_{k,i} - \hat{x}^-_k)(\gamma_{k,i} - \hat{y}_k)^T$
$K_k =$	$P_{x_k y_k} P^{-1}_{y_k y_k}$
$\chi^x_{k,i}$	–
$\hat{x}_k =$	$\hat{x}^-_k + K_k(y_k - \hat{y}_k)$
$P_{x_k} =$	$P^-_{x_k} - K_k P_{y_k y_k} K^T_k$

reason is that by augmenting the assumed system noise into the sigma point set, the noise is also accounted for in the mean value (Julier & Uhlmann, 1994). The augmentation is done by merging the system noise covariance matrix Q_k and the covariance matrix P_{x_k} such that

$$P^a = \begin{bmatrix} P_{x_{k-1}} & 0 \\ 0 & Q_{k-1} \end{bmatrix} \tag{21}$$

This produces more sigma points, and Zandt (2001) demonstrates that increasing the number of sigma points may increase the accuracy of the UKF algorithm, however at the expense of increased computational cost. The UKF with augmented system noise that approximates the said system is given in Table 6. Note that the sigma points $\chi_{k-1} = [\chi^{Tx}_{k-1} \chi^{Tv}_{k-1}]^T$ and the augmented state vector $\hat{x}^a_{k-1} = [\hat{x}^T_{k-1} 0]^T$. Note also that the notation $\hat{x}^a_{k-1} \pm \Gamma\sqrt{P^a}$ means that the vector $\hat{x}^a_{k-1}$ is added/subtracted to each column of $\Gamma\sqrt{P^a}$.

4.2.3. Fully augmented UKF

As in the previous sections, we consider the nonlinear system given by (1) and (2) corrupted by Gaussian noise $v_k = N(\bar{v}, Q)$, $w_k = N(\bar{w}, R)$. In the work of van der Merwe (2004), both the system noise and the output noise is augmented into the sigma point set. This is done by merging the system noise covariance matrix Q_k and the output noise covariance matrix R_k with the covariance matrix P_{x_k} such that

$$P^a = \begin{bmatrix} P_{x_{k-1}} & 0 & 0 \\ 0 & Q_{k-1} & 0 \\ 0 & 0 & R_{k-1} \end{bmatrix} \tag{22}$$

The UKF is shown in Table 7. Note that $\chi_{k-1} = [\chi^{Tx}_{k-1} \quad \chi^{Tv}_{k-1} \quad \chi^{Tw}_{k-1}]^T$ and the augmented state vector $\hat{x}^a_{k-1} = [\hat{x}^T_{k-1} 0 0]^T$.

4.2.4. The reformulated UKF

Motivated by improved numerical behavior and flexible constraint handling, we propose an alternative UKF-formulation for the nonlinear system with noise inputs (assuming noise corrupted by Gaussian noise). By focusing on the sigma points and defining

$$\chi^x_{k,i} = \chi^{x-}_{k,i} + K_k(y_k - h(\chi^{x-}_{k,i}, \chi^w_{k,i})) \tag{23}$$

it can be shown that

$$\hat{x}_k = \sum_{i=0}^{2n} W^x_i \cdot \chi^x_{k,i} = \hat{x}^-_k + K_k(y_k - \hat{y}_k) \tag{24}$$

and that

$$P_{x_k} = \sum_{i=0}^{2n} W^c_i (\chi^x_{k,i} - \hat{x}_k)(\chi^x_{k,i} - \hat{x}_k)^T = P^-_{x_k} - K_k P_{y_k y_k} K^T_k \tag{25}$$

A proof is provided in Appendix A. The augmented (based on (22)) reformulated UKF is given in Table 8.

Although we have used the fully augmented UKF approach in this algorithm, it can be applied to all the other algorithms previously mentioned by proper adoption of the steps $\chi^x_{k,i}$, $\hat{x}_k$ and P_{x_k} as defined in Table 8. Note that if the reformulated correction steps are applied to the algorithm given in Table 4, P_{x_k} is guaranteed to be positive semi-definite with α, β, κ as in (20), since $W^c_i > 0$. This may not be the case in the other algorithms, since W^c_0 may be negative. Further, P_{x_k} may not be positive semi-definite if the standard calculation $P_{x_k} = P^-_{x_k} - K_k P_{y_k y_k} K^T_k$ is used due to round off errors. Therefore the reformulated correction steps may be preferred in

Table 8
The fully augmented UKF algorithm with reformulated correction steps.

$\chi_{k-1} =$	$\left[\hat{x}^a_{k-1} \quad \hat{x}^a_{k-1} + \Gamma\sqrt{P^a} \quad \hat{x}^a_{k-1} - \Gamma\sqrt{P^a}\right]$
$\chi^{x-}_{k,i} =$	$f(\chi^x_{k-1,i}, u_{k-1}, \chi^v_{k-1,i})$
$\hat{x}^-_k =$	$\sum_{i=0}^{2n} W^x_i \cdot \chi^{x-}_{k,i}$
$P^-_{x_k} =$	$\sum_{i=0}^{2n} W^c_i (\chi^{x-}_{k,i} - \hat{x}^-_k)(\chi^{x-}_{k,i} - \hat{x}^-_k)^T$
$\gamma_{k,i} =$	$h(\chi^{x-}_{k,i}, \chi^w_{k,i})$
$\hat{y}_k =$	$\sum_{i=0}^{2n} W^x_i \cdot \gamma_{k,i}$
$P_{y_k y_k} =$	$\sum_{i=0}^{2n} W^c_i (\gamma_{k,i} - \hat{y}_k)(\gamma_{k,i} - \hat{y}_k)^T$
$P_{x_k y_k} =$	$\sum_{i=0}^{2n} W^c_i (\chi^{x-}_{k,i} - \hat{x}^-_k)(\gamma_{k,i} - \hat{y}_k)^T$
$K_k =$	$P_{x_k y_k} P^{-1}_{y_k y_k}$
$\chi^x_{k,i} =$	$\chi^{x-}_{k,i} + K_k(y_k - \gamma_{k,i})$
$\hat{x}_k =$	$\sum_{i=0}^{2n} W^x_i \cdot \chi^x_{k,i}$
$P_{x_k} =$	$\sum_{i=0}^{2n} W^c_i (\chi^x_{k,i} - \hat{x}_k)(\chi^x_{k,i} - \hat{x}_k)^T$

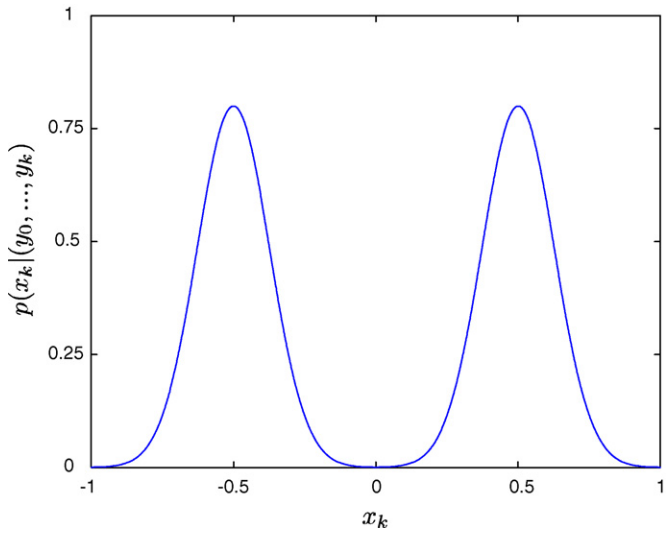

Fig. 1. An illustration of a multimodal probability density function (pdf).

any of the UKF algorithms due to better numerical behavior, but at the expense of computational load.

5. Constrained state estimation

State estimation is challenging in nonlinear systems based on first principles. In chemical processes for instance one may wish to estimate concentrations. An estimator may however give rise to negative concentration estimates even though this is physically impossible. Another issue is the fact that algorithms based on Gaussian noise may not be applicable to nonlinear systems since Gaussian noise propagated through a nonlinear model is distorted. Further, severe nonlinearities may prevent the use of theory based on linearization due to poor estimation accuracy. Finally the nonlinear system may have a skew or multimodal probability density function (pdf).

One way to interpret the Kalman filter (Simon, 2006) is that it solves the maximum probabilities problem

$$\hat{x}_k = arg\max_{x_k} pdf(x_k|(y_1^T \cdots y_k^T)) \tag{26}$$

given x_0, w_k and v_k Gaussian, i.e. $\hat{x}_k$ is the value of x_k that maximizes $pdf(x_k|(y_1^T \cdots y_k^T))$. Then one may ask what happens if one uses the Kalman filter on a system with a multimodal pdf? A multimodal pdf is illustrated in Fig. 1. Intuitively, applying any Kalman filter on a system illustrated in Fig. 1, given (26) is true, there may exist two solutions, one for $\hat{x}_k < 0$ and one for $\hat{x}_k > 0$.

The estimate ($\hat{x}_k$) of e.g. concentrations in a physical system may be constrained in order to force the estimates to have a physical meaning. Hence, constraint handling may be an important part of any estimation algorithm for nonlinear systems.

As mentioned earlier, the literature addressing constraints in the UKF approach is rather limited. In Julier and Uhlmann (1994) the idea of constraining the sigma points is in fact listed as a general possibility, but is however not further discussed. In the following chapters we will turn our focus to constraint implementation.

5.1. Heuristic constraint handling

5.1.1. Constraint methods

A common method for implementing constraints in the KF and EKF algorithms is known as clipping, where the corrected state estimate $\hat{x}_k$ is set equal to some predefined bounds d_k if outside these limits (Hasseltine & Rawlings, 2003; Simon, 2006; Kandepu et al., 2007).

Table 9
The constraint candidates.

χ_{k-1}	CC_1
$\chi_{k,i}^{x-}$	CC_2
$\hat{x}_k^-$	CC_3
$P_{x_k}^-$	–
$\gamma_{k,i}$	$CC_{4,5}$
$\hat{y}_k$	CC_6
$P_{y_k y_k}$	–
$P_{x_k y_k}$	–
K_k	–
$\chi_{k,i}^{x}$	CC_7
$\hat{x}_k$	CC_8
P_{x_k}	–

In Simon and Simon (2005, 2006) and in Simon (2006) projection methods for Kalman filtering are studied. The unconstrained state estimate $\hat{x}_k$ is projected onto some constrained set $\tilde{x}_k$ by solving the following convex QP problem

$$\min_{\tilde{x}_k}(\tilde{x}_k - \hat{x}_k)^T W_k(\tilde{x}_k - \hat{x}_k) \qquad \text{s.t.} \quad D_k\tilde{x}_k \le d_k \tag{27}$$

(27) can be rewritten as the QP problem

$$\min_{\tilde{x}_k}(\tilde{x}_k^T W_k \tilde{x}_k - 2\hat{x}_k^T W\tilde{x}_k) \qquad \text{s.t} \quad D_k\tilde{x}_k \le d_k \tag{28}$$

By choosing $W_k = I$ we obtain a least squares method, and by choosing $W_k = (P_{x_k})^{-1}$ we get a maximum likelihood estimate (Simon & Simon, 2005). What is worth mentioning is that if $W_k = I$ and $D_k = I$, which correspond to a constraint on each $\hat{x}_k$, the solution of (27) is $\tilde{x}_k^T = d_k$ in the case when $\hat{x}_k$ violates the constraints d_k (see also Simon, 2006). In this case solving (28) gives the same solution as clipping, and we may conclude that clipping is optimal in such cases.[6]

Other literature discussing constraint implementation in the UKF approach, apart from Vachhani et al. (2006) and Kandepu et al. (2008) is found in Li and Leung (2004) where they used equality constraints on the corrected estimate, and Julier and Laviola (2007) where two methods for nonlinear equality constraints are proposed.

5.1.2. Constraint candidates

Based on the algorithm steps we are able to identify constraint candidates as given in Table 9.

By constraint candidates (CC) we think of steps in the algorithm which may be suitable for some constraint handling, e.g. constraining CC_1 means that χ_{k-1} is constrained according to some set. Kandepu et al. (2008) proposed the constraint candidates CC_1, CC_2, CC_3, CC_5, CC_6 and CC_8 using the UKF algorithm as given in Table 7. Further they showed the effect of using the constraint candidate CC_1 on a CSTR case.

By using the algorithm as in Table 4 we have the opportunity to extend the list of constraint candidates, by constraining the updated sigma points before they are propagated through the output function. This is indicated by constraint candidate CC_4.

The proposed UKF-formulation of the correction steps as given in Table 8 provides the possibility to further extend the list of con-

[6] Considering implementation aspects this may have some implications, since the clipping may produce lower computationally load than solving a QP-problem.

straint candidates by constraining each $\chi^x_{k,i}$ before the corrected estimate $\hat{x}_k$ with its associated covariance P_{x_k} is calculated. This is indicated by the constraint candidate CC_7. It is worth noticing that constraining the state estimate $\hat{x}_k$ (CC_8) has no direct impact on the covariance P_{x_k} in all the presented algorithms, except for the NLP UKF, QP UKF (see Section 5.2) and the reformulated UKF where the constrained estimate will have a direct impact on the associated covariance. This aspect, that the constraints are propagated directly through the succeeding covariance calculations, is in our view one of the most important aspects of using the proposed reformulated UKF-algorithm.

5.1.3. Constraints implementation

A logical choice when implementing constraints on the sigma points (CC_1), the propagated sigma points (CC_2) and the updated sigma points (CC_4 and CC_7), is to apply clipping after the calculation step, since no covariance is associated with these constraint candidates. Clipping (or minimum square if there is a algebraic relationship between the constraints) could also be applied to the other constraint candidates (CC_3, CC_6 and CC_8). Alternatively, by instead utilizing the covariance in the constraints calculations, the constrained $\tilde{x}^-_k$, $\hat{y}_k$ and $\hat{x}_k$ can be calculated by the maximum likelihood approach as

$$\min_{\tilde{x}^-_k}(\tilde{x}^-_k - \hat{x}^-_k)^T (P^-_{x_k})^{-1}(\tilde{x}^-_k - \hat{x}^-_k) \quad \text{s.t.} \quad D_k\tilde{x}^-_k \le d_{\hat{x}^-_k} \tag{29}$$

$$\min_{\tilde{y}_k}(\tilde{y}_k - \hat{y}_k)^T P^{-1}_{y_k y_k}(\tilde{y}_k - \hat{y}_k) \quad \text{s.t.} \quad D_{\tilde{y}_k}\tilde{y}_k \le d_{\hat{y}_k} \tag{30}$$

$$\min_{\tilde{x}_k}(\tilde{x}_k - \hat{x}_k)^T P^{-1}_{x_k}(\tilde{x}_k - \hat{x}_k) \quad \text{s.t.} \quad D_k\tilde{x}_k \le d_{\hat{x}_k} \tag{31}$$

The choice of which method to use, least squares or maximum likelihood, depends on the system and available computational resources.

5.2. The NLP/QP-UKF

The Kalman-approach assumes a linear correction of the predicted estimate, see (5). In Vachhani et al. (2006) they use the UKF formulation in Julier et al. (1995), i.e. the algorithm as by Table 5 with $[\alpha \quad \beta \quad \kappa]$ as in (19), and propose a nonlinear correction by solving an NLP. A special property of their NLP algorithm is a recalculation of the weights W if the sigma points χ_{k-1} violates some constraints.

In this work we propose an NLP for the correction step as in Vachhani et al. (2006), but without recalculating the weights W. Further we present the NLP UKF for the fully augmented scaled UKF. Assuming the nonlinear system given by (1) and (2) corrupted by Gaussian noise, the algorithm is given as in Table 10.

By choosing

$$J=(y_k - h(\chi^x_{k,i}))^T R_k^{-1}(y_k - h(\chi^x_{k,i})) + (\chi^x_{k,i} - \chi^{x-}_{k,i})^T (P^-_{x_k})^{-1}(\chi^x_{k,i} - \chi^{x-}_{k,i}) \tag{32}$$

as in Vachhani et al. (2006), this leads to a NLP problem. However, by assuming a linear output model, as by (4) it can be shown, see Appendix A.3, that the problem reduces to a QP-problem. In this case J becomes

$$J = \chi^{x^T}_{k,i}(D_k^T R_k^{-1} D_k + (P^-_k)^{-1})\chi^x_{k,i} - 2(y_k^T R_k^{-1} D_k + \chi^{x-T}_{k,i}(P^-_k)^{-1})\chi^x_{k,i} \tag{33}$$

Intuitively, (32) and (33) balances the correction between the output y_k and the propagated state sigma points $\chi^{x-}_{k,i}$ according to the output uncertainty R_k and the predicted covariance $P^-_{x_k}$. Hereafter we will refer to the algorithm using (32) as the NLP UKF and the algorithm using (33) as the QP UKF. The QP UKF formulation drastically reduces the computational load compared to the NLP UKF, see Appendix B, since a QP solver generally is less computationally demanding then an NLP solver.

Both the NLP UKF and the QP UKF minimize J with respect to $\chi^x_{k,i}$ such that some constraints are fulfilled. The constraints can typically be given by (34)

$$\min_{\chi^x_{k,i}} J \quad \text{s.t.} \quad x_{i_{low}} \le \chi^x_{k,i} \le x_{i_{upp}},\ H_i\chi^x_{k,i} \le b_i \tag{34}$$

Note that $\chi^x_{k,i}$ can be calculated directly after computing $P^-_{x_k}$. The model output sigma point set $\gamma_{k,i}$ can then be calculated utilizing the updated sigma points $\chi^x_{k,i}$, i.e $\gamma_{k,i} = h(\chi^x_{k,i}, \chi^w_{k,i})$. Note also that the steps regarding $\chi^x_{k,i}$, $\hat{x}_k$ and P_{x_k} as defined in Table 10 could be applied to all the other UKF algorithms as well, e.g. to the algorithm as in Table 4.

Table 10
The fully augmented NLP/QP UKF algorithm.

$\chi_{k-1} =$	$\left[\hat{x}_{a_{k-1}} \hat{x}_{a_{k-1}} + \Gamma\sqrt{P_a} \ldots \hat{x}_{a_{k-1}} - \Gamma\sqrt{P_a}\right]$
$\chi^{x-}_{k,i} =$	$f(\chi^x_{k-1,i}, u_{k-1}, \chi^v_{k-1,i})$
$\hat{x}^-_k =$	$\sum_{i=0}^{2n} W^x_i \cdot \chi^{x-}_{k,i}$
$P^-_{x_k} =$	$\sum_{i=0}^{2n} W^c_i(\chi^{x-}_{k,i} - \hat{x}^-_k)(\chi^{x-}_{k,i} - \hat{x}^-_k)^T$
$\gamma_{k,i} =$	$h(\chi^{x-}_{k,i}, \chi^w_{k,i})$
$\hat{y}^-_k =$	$\sum_{i=0}^{2n} W^x_i \cdot \gamma_{k,i}$
$P_{y_k y_k} =$	–
$P_{x_k y_k} =$	–
$K_k =$	–
$\chi^x_{k,i}$	$\min_{\chi^x_{k,i}} J$
$\hat{x}_k =$	$\sum_{i=0}^{2n} W^x_i \cdot \chi^x_{k,i}$
$P_{x_k} =$	$\sum_{i=0}^{2n} W^c_i(\chi^x_{k,i} - \hat{x}_k)(\chi^x_{k,i} - \hat{x}_k)^T$

6. Notes regarding the algorithms

We have so far presented several algorithms based on the Kalman-approach. It may be worth mentioning that all the unconstrained algorithms reduces to the linear Kalman filter for linear systems with Gaussian noise. Further we have pointed to several ways of implementing constraints in the presented algorithms. Although the UKF algorithms have been presented as independent algorithms, we will emphasize the fact that the conceptual ideas from one algorithm could easily be used in one of the others, i.e. the reformulated steps, the NLP/QP steps and the constraint handling can be applied to any of the presented UKF algorithms in any combination. For example one can combine the UKF algorithm in Table 4 with constraints in the prediction steps and NLP/QP in the correction steps to facilitate nonlinear update in this step. In the simulation chapter to come we will show some examples of such combinations.

7. Simulation studies

Several authors have investigated the performance of the UKF and compared it with the EKF or other estimation algorithms, see Julier and Uhlmann (1994), Julier et al. (1995), Julier, Uhlmann, and Durrant-Whyte (1997), Zandt (2001), Julier and Uhlmann (2002),

Akin et al. (2003), Laviola (2003), Julier and Uhlmann (2004), van der Merwe (2004), Romanenko and Castro (2004), Romanenko et al (2004), Rawlings and Bakshi (2006), Xiong et al. (2006), Xiong et al. (2007), Pieper (2007) and Kandepu et al. (2008).

In the following we present some of the presented algorithms' ability to handle problems where the EKF suffers from convergence properties, or even fail to converge. We do not consider the KF, since we study nonlinear systems only in this simulation study.

The case selection is based on cases in Hasseltine and Rawlings (2003) used to indicate the performance of the Moving Horizon Estimation (MHE) approach and at the same time show the limitations using the EKF. The key challenge in the selected cases is that they all have a multi-modal pdf, where some solutions are physically valid, and some not.

7.1. Case '2-state CSTR'

7.1.1. Case description

Consider the gas-phase, reversible reaction (Hasseltine & Rawlings, 2003)

$$2A \xrightarrow{k_r} B \qquad k_r = 0.16 \tag{35}$$

with stoichiometric matrix

$$s = [s_1 \quad s_2] = [-2 \quad 1] \tag{36}$$

and reaction rate

$$r = k_r P_A^2 \tag{37}$$

The state and output vectors are defined as

$$x = \begin{bmatrix} P_A \\ P_B \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, y = [1 \quad 1]x \tag{38}$$

where P_A and P_B are the partial pressures. It is assumed that the ideal gas law holds (high temperature, low pressure), and that the reaction occurs in a well-mixed, isothermal batch reactor. From first principles, the continuous model for this system is

$$\dot{x}(t) = f_c(x(t)) = s^T r(t) \tag{39}$$

and the output model

$$y(t) = h_c(x(t)) = [1 \quad 1]x(t) \tag{40}$$

A discrete analytical solution of (39) is

$$x_{1,k+1} = \left(\frac{1}{x_{1,k}} - s_1 k_r \Delta t\right)^{-1} = \frac{x_{1,k}}{1 - s_1 k_r \Delta t\, x_{1,k}} = \frac{x_{1,k}}{1 + 2k_r \Delta t\, x_{1,k}} \tag{41}$$

$$x_{2,k+1} = x_{2,k} + \frac{s_2 k_r}{s_1 k_r}(x_{1,k+1} - x_{1,k}) = x_{2,k} + \frac{k_r \Delta t\, x_{1,k}}{1 + 2k_r \Delta t\, x_{1,k}} \tag{42}$$

where Δt is the integration step length. Further it is assumed that the system experience Gaussian noise both in the states and in the outputs, given by $v_k \sim N(0, Q_k)$ and $w_k \sim N(0, R_k)$. The discrete system becomes

$$x_{k+1} = f(x_k) + v_k \tag{43}$$

and the discrete model of (40)

$$y_k = h(x_k) + w_k = [1 \quad 1]x_k + w_k \tag{44}$$

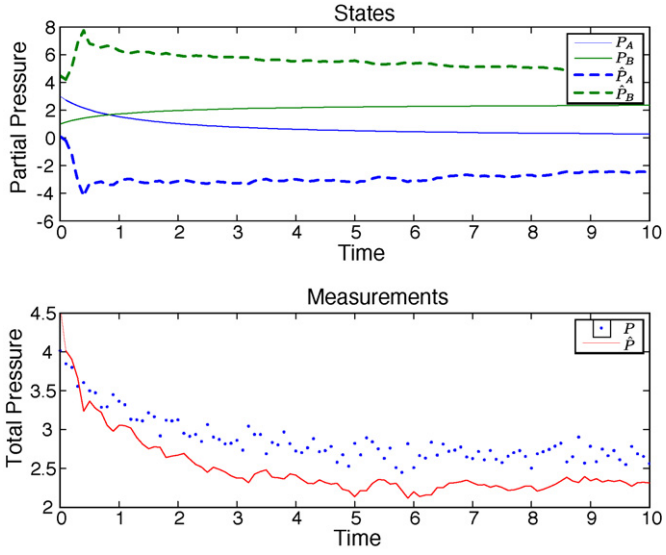

Fig. 2. Unconstrained EKF.

The parameters used for this system are

$$\Delta t = t_{k+1} - t_k = 0.1 \qquad P_0 = \begin{bmatrix} 6^2 & 0 \\ 0 & 6^2 \end{bmatrix}$$
$$Q_k = \begin{bmatrix} 0.001^2 & 0 \\ 0 & 0.001^2 \end{bmatrix} \qquad R_k = 0.1^2 \tag{45}$$
$$x_0 = \begin{bmatrix} 3 \\ 1 \end{bmatrix} \qquad \hat{x}_0 = \begin{bmatrix} 0.1 \\ 4.5 \end{bmatrix}$$

Note that the initial guess for the states ($\hat{x}_0$), is very poor. This simple example is used by several authors in order to investigate estimator performance, see Hasseltine and Rawlings (2003), Vachhani et al. (2006), Rawlings and Bakshi (2006), and Kandepu et al. (2008). The reason why this problem is interesting is that the estimator may experience a multimodal pdf, which may lead to unphysical estimates.

7.1.2. Simulation results

In the following chapters we investigate some of the previous described algorithms applied on the 2-state CSTR case. Note that all the parameters are as described in the case description above for all the presented algorithms and that the noise sequences are identical in all simulations. Note also that we have used the exact solution to (39). By using Euler or Runge–Kutta integration schemes, the results will differ slightly from the results presented below. This is especially the case in the unconstrained cases. However, the main characteristics are maintained whichever integration scheme is used. We have chosen to be true to the source of these examples (Hasseltine & Rawlings, 2003), and have used the same parameters to achieve comparable results.

7.1.2.1. EKF. Fig. 2 shows the results of the simulation using unconstrained EKF as by Table 3, with numerically derived Jacobians.

As Fig. 2 shows, the unconstrained EKF fails to converge to the true states within the given time frame.[7] These results are in agreement with the results of Hasseltine and Rawlings (2003) and Kandepu et al. (2008). The reason why the EKF fails is that while the negative pressure is unphysical, the unconstrained estimator allows the estimate to enter regions where the partial pressure

[7] Actually the EKF will converge very slowly, but one need to run the simulation approximately 1000 samples.

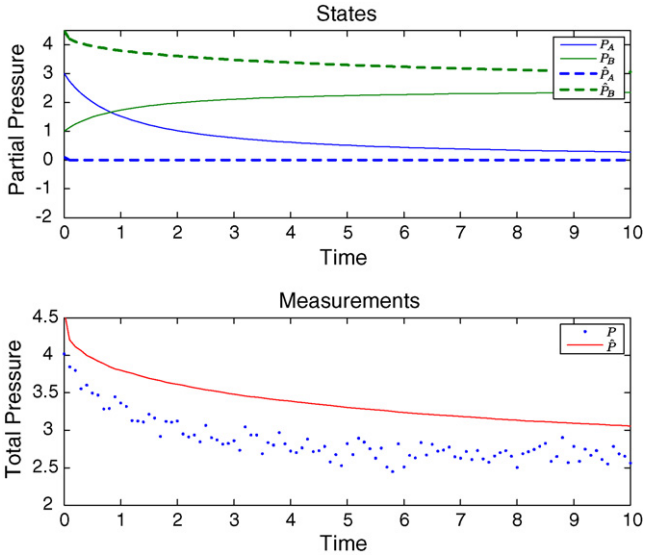

Fig. 3. Constrained EKF.

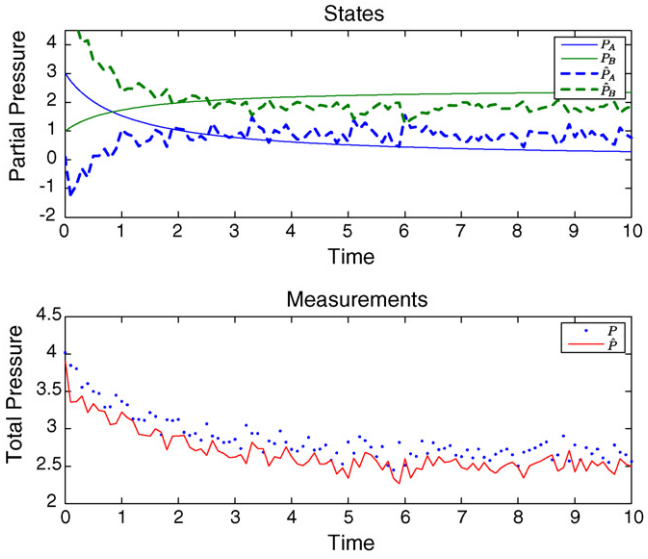

Fig. 4. Unconstrained fully augmented UKF.

may be negative. The behavior of the estimator in this case is best explained "*as a poor initial guess leading to an errant region of attraction*" (Hasseltine & Rawlings, 2003). The "*errant region of attraction*" exist due to the multimodal properties of this system's pdf, see Hasseltine and Rawlings (2003) for a broader discussion of why the EKF fails.

Fig. 3 shows the results of the simulation using EKF with constraints. The constraint is implemented as clipping the corrected state estimate (CC_8), and the constraints are such that $\hat{x}_k \geq 0$.

As Fig. 3 shows, the constrained EKF fails to converge to the true states. These results are in agreement with the results of Hasseltine and Rawlings (2003) and Kandepu et al. (2008). By clipping the state $\hat{x}_k$, it is restricted to a valid physical region, but the knowledge about the constraints is not propagated into the covariance, and the hence accuracy of the approximated covariance matrix P_{x_k} is questionable. As Fig. 3 indicates, the performance of the clipped EKF is rather poor.

7.1.2.2. UKF. In the following chapters a selection of the presented Jacobian free algorithms is tested on the 2-state CSTR case. For the UKF algorithms using (12) for sigma point selection, (18) is used for α, β and κ. For the UKF algorithms using (11) for the sigma point selection, the values given by (20) is used for α, β and κ.

First we investigate the unconstrained fully augmented UKF given in Table 7 and using the Cholesky square root algorithm. Fig. 4 shows the results of the simulation.

As Fig. 4 shows, this algorithm fails to converge to the true states within the given time frame. These results are in agreement with the results of Kandepu et al. (2008). Assuming (26) also is valid for the UKF, the unconstrained UKF as well as the EKF has to deal with multiple optima, and we believe this is the reason why the unconstrained UKF algorithms suffer poor performance on this case.

Next, we extend the fully augmented UKF as in Table 7 with constraint handling and Cholesky square root. The constraints is such that $\chi^x_{k-1} \geq 0$ (CC_1). Fig. 5 shows the results of the simulation.

As Fig. 5 shows, this algorithm converges to the true states after approximately 25 samples regards ($t = 2.5$ s). These results are in agreement with the results achieved by Kandepu et al. (2008). By constraining $\chi^x_{k-1} \geq 0$ (CC_1) the sigma points are restricted to physical valid values. Also, by inspecting the structure of the UKF algorithm, one sees (e.g. see Table 7) that constraining the sigma points not only propagates the constraints to the mean value, but also influences the covariance calculations. By this the covariance is related to the mean value, and the unphysical region of attraction is limited. Note that since the sigma points are propagated through the model after they are constrained, constraining the sigma points only (CC_1) does not generally guarantee that the state estimates does not enter an unphysical value.

All the UKF algorithms converge to the true state when constraining the sigma points $\chi^x_{k-1} \geq 0$ (CC_1). Although the unconstrained UKF in Table 4 with symmetric square root calculation may converge, it is possible to increase the convergence performance. By using the reformulated correction steps from the algorithm in Table 8 and constraint handling by using CC_1 ($\chi^x_{k-1} \geq 0$) and CC_7 ($\chi^x_k \geq 0$) applied on the non-augmented algorithm in Table 4, we get the results as given in Fig. 6.

In fact, the combination of constraint handling and the reformulated steps makes this algorithm to almost converge after only 2 samples (0.2 s) for this particular noise sequence.

We now investigate the QP UKF by combining Table 10 and (33) with the algorithm presented in Table 4, again using the symmetric square root formulation.

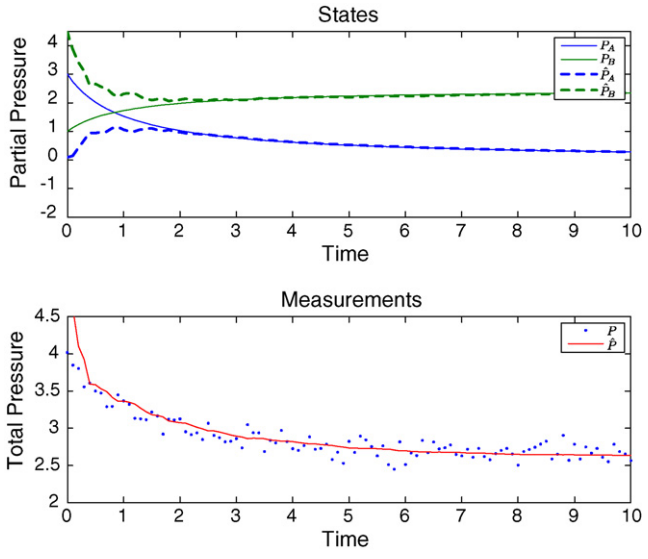

Fig. 5. Constrained ($\chi^x_{k-1} \geq 0$ (CC_1)) fully augmented UKF.

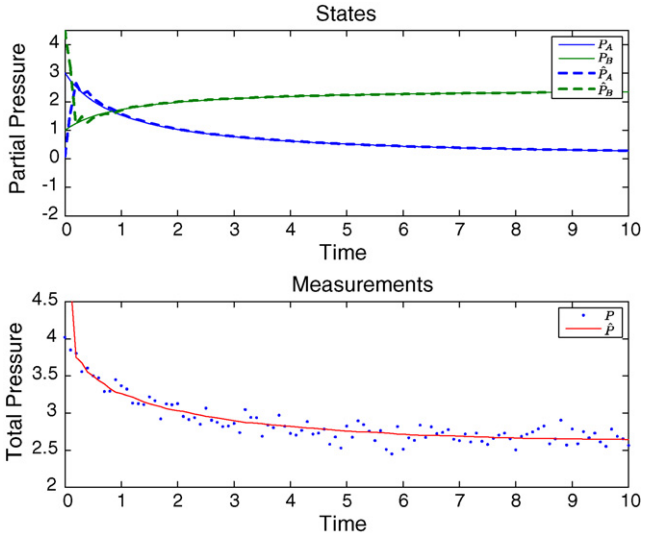

Fig. 6. Constrained (CC_1 and CC_7) non-augmented UKF as by D. Simon.

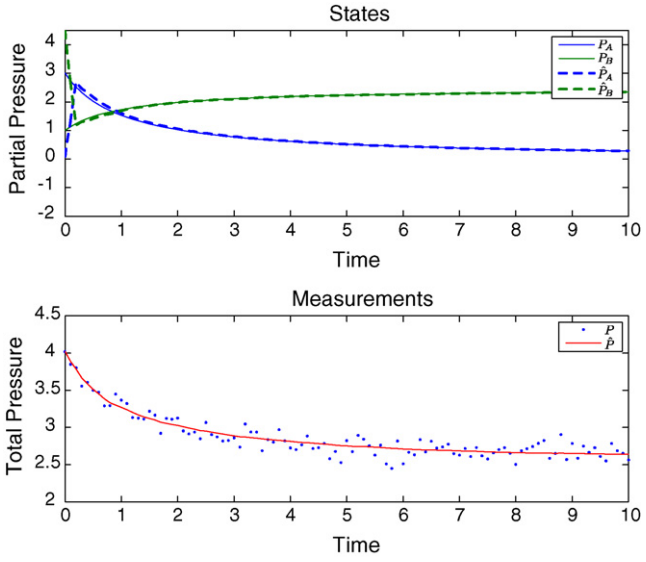

Fig. 7. QP-UKF using symmetric square root calculations.

The system under investigation has a linear output model, and hence the results of the QP UKF are identical to the NLP UKF (Table 10). The results of the simulation are shown in Fig. 7.

As Fig. 7 shows, the NLP/QP-algorithm converge after 2 samples (0.2 s). Notice also the lack of the initial spike in the output. This is because the output in this situation is selected using the optimized sigma point set $\chi^{x}_{k,i}$ instead of $\chi^{x-}_{k,i}$.

Although the QP UKF and NLP UKF produces the same estimate in the case of a linear output model, there is a huge difference when it comes to computational load. Used on the 2-state CSTR problem, the QP UKF is approximately 8 times faster than the NLP UKF[8] when using the algorithm in Table 4 as a basis, see Appendix B. Using the QP UKF described in Table 10 is approximately 11 times faster than the NLP UKF. In this case we also experience best convergence performance using the symmetric square root and not the Cholesky square root algorithm.

[8] Implemented in Matlab using the quadprog() for the QP problem and fmincon() for the NLP problem.

7.2. Case '3-state batch reactor'

7.2.1. Case description

Consider the gas-phase, reversible reaction (Hasseltine & Rawlings, 2003)

$$A \underset{k_2}{\overset{k_1}{\leftrightarrows}} B + C \tag{46}$$

$$2B \underset{k_4}{\overset{k_3}{\leftrightarrows}} C \tag{47}$$

$$k = [\, k_1 \quad k_2 \quad k_3 \quad k_4 \,] = [\, 0.5 \quad 0.05 \quad 0.2 \quad 0.01 \,] \tag{48}$$

with stoichiometric matrix

$$v = \begin{bmatrix} -1 & 1 & 1 \\ 0 & -2 & 1 \end{bmatrix} \tag{49}$$

and reaction rate

$$r = \begin{bmatrix} k_1 c_A - k_2 c_B c_C \\ k_3 c_B^2 - k_4 c_C \end{bmatrix} \tag{50}$$

The state and output vectors are defined as

$$x = \begin{bmatrix} c_A \\ c_B \\ c_C \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}, \qquad y = [RT \quad RT \quad RT]x, \qquad RT = 32.84 \tag{51}$$

where c_i is the concentration of the species. It is assumed that the ideal gas law holds, and that the reaction occurs in a well-mixed, constant volume, isothermal batch reactor. From first principles, the continuous model for this system is

$$\dot{x}(t) = f_c(x(t)) = v^T r(t) \tag{52}$$

and the output (measurement) model

$$y(t) = h(x(t)) = [RT \quad RT \quad RT]x(t) \tag{53}$$

The model is discretized by using Runge–Kutta 4th order method. Further it is assumed that the system experiences Gaussian noise both in the state and in the measurements, respectively $w_k \sim N(0, Q_k)$ and $v_k \sim N(0, R_k)$. The discrete system is given by

$$x_{k+1} = f(x_k) + w_k \tag{54}$$

$$y_k = h(x_k) + v_k = [RT \quad RT \quad RT]x_k + v_k \tag{55}$$

The parameters used for this system are

$$\Delta t = t_{k+1} - t_k = 0.25 \qquad P_0 = \begin{bmatrix} 0.5^2 & 0 & 0 \\ 0 & 0.5^2 & 0 \\ 0 & 0 & 0.5^2 \end{bmatrix}$$

$$Q_k = \begin{bmatrix} 0.001^2 & 0 & 0 \\ 0 & 0.001^2 & 0 \\ 0 & 0 & 0.001^2 \end{bmatrix} \qquad R_k = 0.25^2 \tag{56}$$

$$x_0 = \begin{bmatrix} 0.5 \\ 0.05 \\ 0 \end{bmatrix} \qquad \hat{x}_0 = \begin{bmatrix} 0 \\ 0 \\ 4 \end{bmatrix}$$

It may be noted that the initial guess for the estimator ($\hat{x}_0$), is very poor. Again the estimator may encounter a multi-modal pdf, which may lead to unphysical estimates.

7.2.2. Simulation results

In the next chapters we present some results of our investigation on the 3-state CSTR case. Note that all the parameters are as described in the case description above for all the presented algorithms. Also the noise sequences w_k and v_k are identical in all simulations. The performance of the EKF and the constrained (clipped) EKF used on this case is well documented in Hasseltine and Rawlings (2003), and since our results is in full agreement with

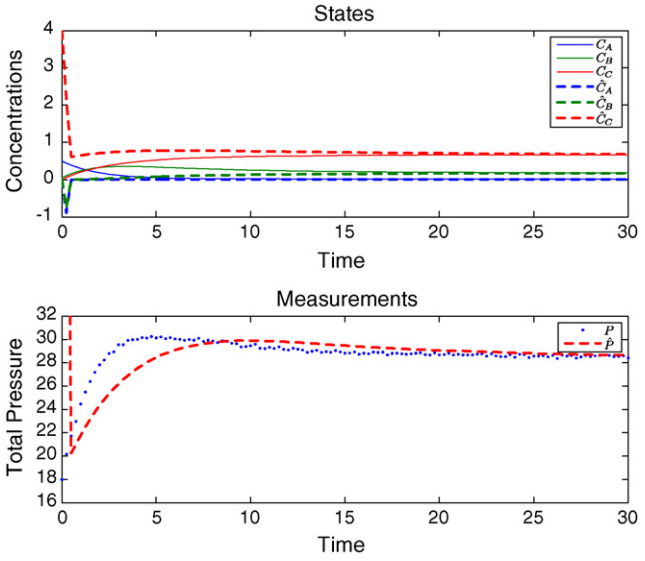

Fig. 8. The figure shows the performance of the UKF algorithm based on the algorithm in Table 7 with constraint handling using CC_1.

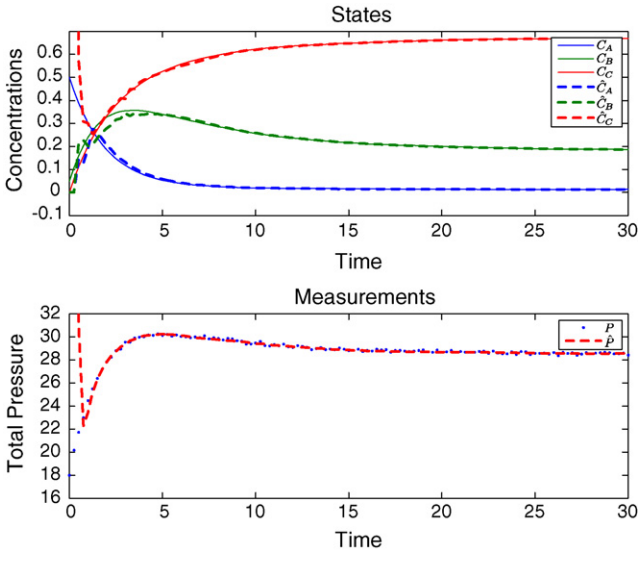

Fig. 9. The figure shows the performance of the UKF algorithm based on the algorithm in Table 8 with constraints. The constraints used are CC_1, CC_3 and CC_7.

those results regarding the EKF, we have chosen not to include them here.

Further, none of the unconstrained UKF algorithms presented in this work converged within the time limit. Hence, the following results are focused on the constrained approach.

The fully augmented UKF as in Table 7 is first applied, using the Cholesky square root algorithm, constrained sigma points (CC_1) and $[\alpha \quad \beta \quad \kappa]$ as in (18).

As Fig. 8 shows, the performance is rather poor when it comes convergence speed. Note that all of the constrained UKF algorithms converge within the time limit, but uses unacceptable long time when constraining only the sigma point set by $\chi^x_{k-1} \geq 0$ (CC_1).

Second, we investigate the reformulated fully augmented UKF (Table 8) with the Cholesky square root algorithm and constrained sigma points (CC_1 and CC_7), constrained predicted state estimates (CC_3) and

$$\begin{bmatrix} \alpha & \beta & \kappa \end{bmatrix} = \begin{bmatrix} 0.7 & 0 & 3-n \end{bmatrix} \tag{57}$$

By using the constraints

$$\begin{aligned} \chi^x_{k-1} &\geq 0\,(CC_1) \\ \chi^x_{k,i} &\geq 0\,(CC_7) \\ x^-_k &\leq [\infty \quad \infty \quad 4]^T\,(CC_3) \end{aligned} \tag{58}$$

we got the results as shown in Fig. 9. Convergence to the correct states is fast.

The results of the other UKF algorithms, using the reformulated correction steps, the same $[\alpha \quad \beta \quad \kappa]$-set as defined by (57)[9] and constraints as defined in (58), have all similar performance.

Third, we study the NLP/QP UKF with additive system noise, i.e. the algorithm in Table 5 with $\chi^x_{k,i}$, $\hat{x}_k$ and P_{x_k} as in Table 10. Further, the symmetric square root algorithm is used, constrained sigma points (CC_1), constrained predicted state estimate x^-_k (CC_3) and

$$[\alpha \quad \beta \quad \kappa] = [1 \quad 10 \quad 0] \tag{59}$$

By using the constraints

$$\chi^x_{k-1} \geq 0\,(CC_1) \tag{60}$$

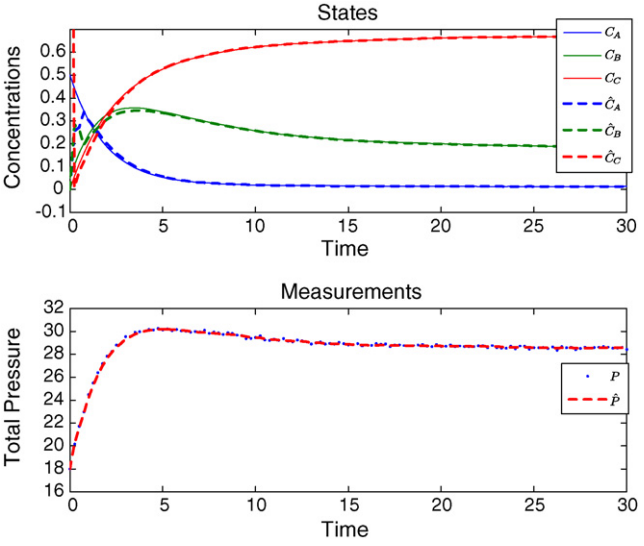

Fig. 10. The figure shows the performance of the QP UKF algorithm based on the algorithm in Table 5 with the correction steps as in Table 10. The constraints used are CC_1, CC_3 and CC_7.

$$x^-_k \leq [\infty \quad \infty \quad 4]^T\,(CC_3) \tag{61}$$

and minimize $\chi^x_{k,i}$ such that

$$0 \leq \chi^x_{k,i} \leq [\infty \quad \infty \quad 4]^T\,(\text{i.e. } CC_7 \text{ by QP}) \tag{62}$$

we got the results[10] as shown in Fig. 10.

As Fig. 10 shows, the performance is further improved. Note that the use of the constraints (CC_1) and (CC_3) was required in order to achieve these results, as well as some tuning of the α, β and κ (for a broader general discussion of the effect of α, β and κ see Julier & Uhlmann, 2002; Julier & Uhlmann, 2004).

[9] Note that the algorithms using Eq. (11) for the sigma point selection, uses (20) as their $[\alpha \quad \beta \quad \kappa]$-set.

[10] The constraints CC_7 was found by using process knowledge combined with trail and error.

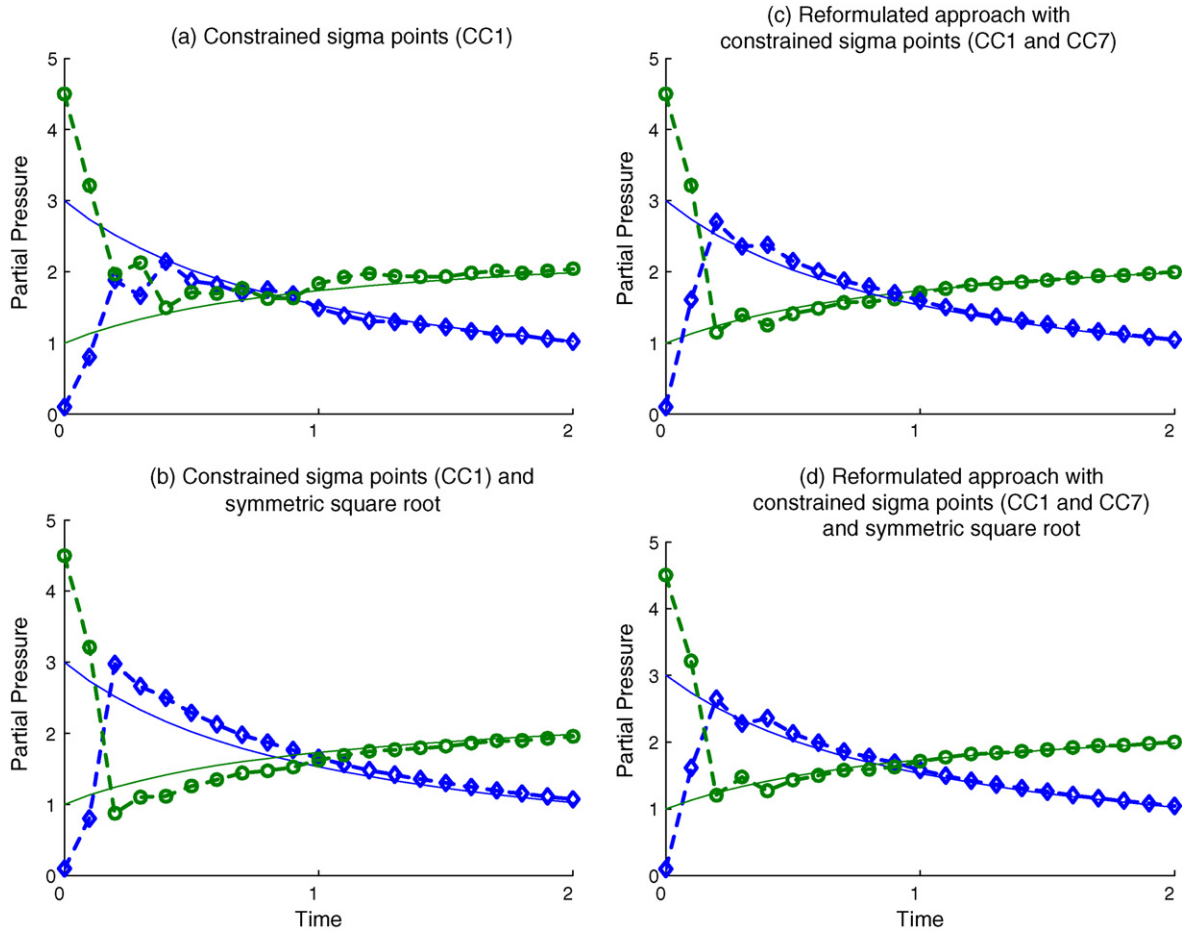

Fig. 11. A comparison of the effect of the different formulation alternatives. The figure shows four sub-figures with the states of the 20 first samples of the 2-state CSTR case, using the non-augmented UKF as in Table 4, weights (W) by (17) and constraining $\chi^x_{k-1} \geq 0$ (CC_1). In (a) the Cholesky square root algorithm is used. (b) The change in the estimates from (a) as the Cholesky square root algorithm is replaced with a symmetric square root. (c) The change in the estimates from sub-figure (a) as the reformulated correction steps is introduced along with constraining $\chi^x_{k,i} \geq 0$. (d) the change in the estimates from sub-figure (a) as the Cholesky square root is replaced with a symmetric square root and the reformulated correction steps is introduced constraining $\chi^x_{k,i} \geq 0$ (CC_7).

8. Discussion

We have shown several UKF based algorithms and proposed some methods to implement constraints using these algorithms. Further we have indicated that the convergence performance of the algorithms is not only different, but for the same algorithm might be influenced by the selection of the square root algorithm,[11] at which algorithm step the constraints are placed, and the use of the reformulated correction steps. Fig. 11 further illustrates this.

As Fig. 11 demonstrates there is an effect of using both the reformulated correction steps and the symmetric square root calculation in the 2-state CSTR case with respect to convergence speed.

Compared to the results in Kandepu et al. (2008) and the results presented in Fig. 5, we have shown that it is possible to decrease the convergence time from 25 (2.5 s) samples to approximately 2 samples (0.2 s) (see Fig. 11d). Compared to the unconstrained EKF, which converges in approximately 1000 samples with this particular noise sequence, this indicates the superior performance of the suggested constrained UKF-approach applied to the 2-state CSTR case.[12]

Also, by applying UKF to the 3-state case, we have demonstrated that the constrained reformulated approach achieves good performance. The constrained UKF, constraining only $\chi^x_{k-1} \geq 0$ (CC_1), suffered from poor performance with respect to convergence speed, and converged after approximately 70 samples. The QP UKF converged after approximately 3 samples, and the reformulated fully augmented constrained UKF after approximately 5 samples.

However, care should be taken. Applying constraints as suggested in this paper may introduce singularities and even indefinite covariance matrices. This may happen e.g. if all the sigma points of one state are outside the constraints at the same time as the estimated state also is outside the constraint. Strategies to identify and overcome such situations may be required. Several strategies could be applied, but it is beyond the scope (and space) of this paper to enter that discussion.

By introducing both a reformulation of the correction steps and the use of constraints in the UKF approach, the UKF may not be considered 'unscented' any more. To distinguish from the original UKF, it is maybe more correct to refer to the versions utilizing constraints and/or reformulated corrections steps as the Constrained Unscented Kalman filter (CUKF).

Further, Tables 11–13 in Appendix B indicates that adding complexity to the original UKF algorithms have a computational cost, but as Tables 14 and 15 in the appendix indicates, with a poten-

[11] See Appendix B.
[12] See Appendix B for performance measurements regard the 'Case 2A-¿B'.

tial increase in estimation accuracy, even though the most complex not necessarily gives the most accurate estimation. Given a system, which algorithm to select is a trial and error process.

9. Conclusions

We have in this paper given a broad overview of several UKF based nonlinear estimation algorithms as an alternative to the EKF, suggested a reformulation of the correction step which can be applied to all of the presented UKF algorithms, presented a QP formulation of the NLP UKF (which also can be applied to all of the presented UKF algorithms) and proposed alternatives to realize constraints within the UKF approach.

Further we have demonstrated the superior performance of the constrained UKF approach over the EKF approach applied to both a '2-state CSTR' example system and a '3-state batch' example system.

Acknowledgement

The financial support of Hydro Aluminium ASA is gratefully acknowledged.

Appendix A. Algebraic proofs

A.1. Proof of reformulated corrected state estimate

We want to show that

$$\hat{x}_k = \sum_{i=0}^{2n} W_i^x \cdot \chi_{k,i}^x = \hat{x}_k^- + K_k(y_k - \hat{y}_k^-) \tag{63}$$

Define

$$\chi_{k,i}^x = \chi_{k,i}^{x-} + K_k(y_k - h(\chi_{k,i}^{x-}\chi_{k,i}^w)) \tag{64}$$

then

$$\begin{aligned}
\hat{x}_k &= \sum_{i=0}^{2n} W_i^x \cdot \chi_{k,i}^x \\
&= \sum_{i=0}^{2n} W_i^x(\chi_{k,i}^{x-} + K_k(y_k - h(\chi_{k,i}^{x-}\chi_{k,i}^w))) \\
&= \sum_{i=0}^{2n} W_i^x \chi_{k,i}^{x-} + \sum_{i=0}^{2n} W_i^x K_k(y_k - h(\chi_{k,i}^{x-}\chi_{k,i}^w)) \\
&= \sum_{i=0}^{2n} W_i^x \chi_{k,i}^{x-} + K_k \sum_{i=0}^{2n} W_i^x(y_k - h(\chi_{k,i}^{x-}\chi_{k,i}^w)) \\
&= \sum_{i=0}^{2n} W_i^x \chi_{k,i}^{x-} + K_k \left(\sum_{i=0}^{2n} W_i^{x-} y_k - \sum_{i=0}^{2n} W_i^x h(\chi_{k,i}^{x-}\chi_{k,i}^w) \right) \\
&= \hat{x}_k^- + K_k(y_k - \hat{y}_k)
\end{aligned} \tag{65}$$

Note $\sum_{i=0}^{2n} W_i^x = 1$ and the other definitions as given by Table 8 for the reformulated UKF.

A.2. Proof of the reformulated corrected covariance

We want to show that

$$P_{x_k} = \sum_{i=0}^{2n} W_i^c(\chi_{k,i}^x - \hat{x}_k)(\chi_{k,i}^x - \hat{x}_k)^T = P_{x_k}^- - K_k P_{y_k y_k} K_k^T \tag{66}$$

where

$$P_{x_k}^- > 0 \tag{67}$$

Define

$$\chi_{k,i}^x = \chi_{k,i}^{x-} + K_k(y_k - h(\chi_{k,i}^{x-}\chi_{k,i}^w)) = \chi_{k,i}^{x-} + K_k(y_k - \gamma_{k,i}) \tag{68}$$

Table 11
The table shows the average time to run 10 simulations of the '2-state CSTR' case using unconstrained versions of the algorithms.

	Ch [s]	Sy [s]	Ch+Re [s]	Sy+Re [s]
EKF (Table 3)	0.4	0.4	0.4	0.4
UKF D. Simon (Table 4)	0.4	0.5	0.4	0.5
UKF non-aug (Table 5)	0.5	0.6	0.6	0.6
UKF Q-aug (Table 6)	0.7	0.8	0.8	0.9
UKF fully-aug (Table 7)	0.8	0.9	0.9	0.9

Table 12
The table shows the average time to run 10 simulations of the '2-state CSTR' case using constrained versions of the UKF algorithms.

	Ch [s]	Sy [s]	Ch+Re [s]	Sy+Re [s]
UKF D. Simon (Table 4)	0.8	0.9	1.2	1.2
UKF non-aug (Table 5)	1.0	1.0	1.4	1.5
UKF Q-aug (Table 6)	1.7	1.6	2.4	2.5
UKF fully-aug (Table 7)	1.9	2.0	2.9	2.9

Table 13
The table shows the average time to run 10 simulations of the '2-state CSTR' case using the QP/NLP UKF algorithms.

	QP Ch [s]	QP Sy [s]	NLP Ch [s]	NLP Sy [s]
UKF D. Simon (Table 4)	1.3	1.3	12.2	12.1
UKF non-aug (Table 5)	1.6	1.6	15.1	17.4
UKF Q-aug (Table 6)	2.6	2.6	29.6	29.0
UKF fully-aug (Table 7)	3.0	3.1	35.9	38.3

Table 14
The table shows the average square root error based on 100 Monte Carlo simulations of the '2-state CSTR' case using constrained versions of the UKF algorithms.

	Ch	Sy	Ch+Re	Sy+Re
UKF D. Simon	$[0.37\ 0.43]^T$	$[0.36\ 0.41]^T$	$[0.32\ 0.42]^T$	$[0.32\ 0.42]^T$
UKF non-aug	$[0.42\ 0.48]^T$	$[0.40\ 0.46]^T$	$[0.33\ 0.45]^T$	$[0.33\ 0.45]^T$
UKF Q-aug	$[0.49\ 0.55]^T$	$[0.45\ 0.51]^T$	$[0.41\ 0.51]^T$	$[0.40\ 0.49]^T$
UKF fully-aug	$[0.52\ 0.58]^T$	$[0.46\ 0.52]^T$	$[0.44\ 0.53]^T$	$[0.42\ 0.52]^T$

Table 15
The table shows the average square root estimation error based on 100 Monte Carlo simulations of the '2-state CSTR' case using constrained versions of the UKF algorithms.

	QP Ch	QP Sy	NLP Ch	NLP Sy
UKF D. Simon	$[0.34\ 0.40]^T$	$[0.34\ 0.41]^T$	$[0.34\ 0.41]^T$	$[0.34\ 0.41]^T$
UKF non-aug	$[0.35\ 0.41]^T$	$[0.35\ 0.42]^T$	$[0.35\ 0.41]^T$	$[0.35\ 0.41]^T$
UKF Q-aug	$[0.41\ 0.48]^T$	$[0.40\ 0.46]^T$	$[0.42\ 0.48]^T$	$[0.41\ 0.48]^T$
UKF fully-aug	$[0.45\ 0.52]^T$	$[0.43\ 0.50]^T$	$[0.45\ 0.52]^T$	$[0.43\ 0.50]^T$

then, by using the definitions given in Table 8 for the reformulated UKF we obtain

$$
\begin{aligned}
P_{x_k} &= \sum_{i=0}^{2n} W_i^c(\chi_{k,i}^x - \hat{x}_k)(\chi_{k,i}^x - \hat{x}_k)^T \\
&= \sum_{i=0}^{2n} W_i^c(\chi_{k,i}^x - (\hat{x}_k^- + K_k(y_k - \hat{y}_k^-)))(\chi_{k,i}^x - (\hat{x}_k^- + K_k(y_k - \hat{y}_k^-)))^T \\
&= \sum_{i=0}^{2n} W_i^c(\chi_{k,i}^{x-} + K_k(y_k - \gamma_{k,i}) - \hat{x}_k^- - K_k(y_k - \hat{y}_k^-)) \cdot (\chi_{k,i}^{x-} + K_k(y_k - \gamma_{k,i}) - \hat{x}_k^- - K_k(y_k - \hat{y}_k^-))^T \\
&= \sum_{i=0}^{2n} W_i^c((\chi_{k,i}^{x-} - \hat{x}_k^-) + K_k(y_k - y_k - \gamma_{k,i} + \hat{y}_k^-)) \cdot ((\chi_{k,i}^{x-} - \hat{x}_k^-) + K_k(y_k - y_k - \gamma_{k,i} + \hat{y}_k^-))^T \\
&= \sum_{i=0}^{2n} W_i^c((\chi_{k,i}^{x-} - \hat{x}_k^-) + K_k(\gamma_{k,i} + \hat{y}_k^-)) \cdot ((\chi_{k,i}^{x-} - \hat{x}_k^-) + K_k(\gamma_{k,i} + \hat{y}_k^-))^T \\
&= \sum_{i=0}^{2n} W_i^c((\chi_{k,i}^{x-} - \hat{x}_k^-)(\chi_{k,i}^{x-} - \hat{x}_k^-)^T - (\chi_{k,i}^{x-} - \hat{x}_k^-)(\gamma_{k,i} - \hat{y}_k^-)^T K_k^T - K_k(\gamma_{k,i} - \hat{y}_k^-)(\chi_{k,i}^{x-} - \hat{x}_k^-)^T + K_k(\gamma_{k,i} - \hat{y}_k^-)(\gamma_{k,i} - \hat{y}_k^-)^T K_k^T) \\
&= \sum_{i=0}^{2n} W_i^c(\chi_{k,i}^{x-} - \hat{x}_k^-)(\chi_{k,i}^{x-} - \hat{x}_k^-)^T - \sum_{i=0}^{2n} W_i^c(\chi_{k,i}^{x-} - \hat{x}_k^-)(\gamma_{k,i} - \hat{y}_k^-)^T K_k^T - K_k \sum_{i=0}^{2n} W_i^c(\gamma_{k,i} - \hat{y}_k^-)(\chi_{k,i}^{x-} - \hat{x}_k^-)^T + K_k \sum_{i=0}^{2n} W_i^c(\gamma_{k,i} - \hat{y}_k^-)(\gamma_{k,i} - \hat{y}_k^-)^T K_k^T \\
&= P_{x_k}^- - P_{x_k y_k} K_k^T - K_k P_{x_k y_k}^T + K_k P_{y_k y_k} K_k^T \\
&= P_{x_k}^- - P_{x_k y_k} P_{y_k y_k}^{-1} P_{x_k y_k}^T - P_{x_k y_k} P_{y_k y_k}^{-1} P_{x_k y_k}^T + P_{x_k y_k} P_{y_k y_k}^{-1} P_{y_k y_k} P_{y_k y_k}^{-1} P_{x_k y_k}^T \\
&= P_{x_k}^- - P_{x_k y_k} P_{y_k y_k}^{-1} P_{x_k y_k}^T - P_{x_k y_k} P_{y_k y_k}^{-1} P_{x_k y_k}^T + P_{y_k y_k} P_{y_k y_k}^{-1} P_{x_k y_k}^T \\
&= P_{x_k}^- - P_{x_k y_k} P_{y_k y_k}^{-1} P_{x_k y_k}^T \\
&= P_{x_k}^- - P_{x_k y_k} P_{y_k y_k}^{-1} P_{y_k y_k} P_{y_k y_k}^{-1} P_{x_k y_k}^T \\
&= P_{x_k}^- - K_k P_{y_k y_k} K_k^T
\end{aligned}
\tag{69}
$$

A.3. Proof of the QP-formulation

Vachhani et al. (2006) suggested to use an NLP to calculate an updated sigma point set. An adoption of Vachhani's approach leads to the NLP formulation

$$
\min_{\chi_{k,i}^{x+}} \left\{ (y_k - h(\chi_{k,i}^x))^T R_k^{-1} (y_k - h(\chi_{k,i}^x)) + (\chi_{k,i}^x - \chi_{k,i}^{x-})^T (P_{x_k}^-)^{-1} (\chi_{k,i}^x - \chi_{k,i}^{x-}) \right\} \tag{70}
$$

If the output function $h(\cdot)$ is linear we can formulate the problem

$$
\begin{aligned}
&\min_{\chi_{k,i}^{x+}} \left\{ (y_k - D_k \chi_{k,i}^x)^T R_k^{-1} (y_k - D_k \chi_{k,i}^x) + (\chi_{k,i}^x - \chi_{k,i}^{x-})^T (P_{x_k}^-)^{-1} (\chi_{k,i}^x - \chi_{k,i}^{x-}) \right\} \\
&= \min_{\chi_{k,i}^{x+}} \left\{ (y_k^T R_k^{-1} - \chi_{k,i}^{Tx} D_k^T R_k^{-1})(y_k - D_k \chi_{k,i}^x) + (\chi_{k,i}^{Tx} (P_{x_k}^-)^{-1} - \chi_{k,i}^{Tx-} (P_{x_k}^-)^{-1})(\chi_{k,i}^x - \chi_{k,i}^{x-}) \right\} \\
&= \min_{\chi_{k,i}^{x+}} \left\{ y_k^T R_k^{-1} y_k - \chi_{k,i}^{Tx} D_k^T R_k^{-1} y_k - y_k^T R_k^{-1} D_k \chi_{k,i}^x + \chi_{k,i}^{Tx} D_k^T R_k^{-1} D_k \chi_{k,i}^x + \chi_{k,i}^{Tx} (P_{x_k}^-)^{-1} \chi_{k,i}^x \right. \\
&\quad \left. - \chi_{k,i}^{Tx-} (P_{x_k}^-)^{-1} \chi_{k,i}^x - \chi_{k,i}^{Tx} (P_{x_k}^-)^{-1} \chi_{k,i}^{x-} + \chi_{k,i}^{Tx-} (P_{x_k}^-)^{-1} \chi_{k,i}^{x-} \right\}
\end{aligned}
\tag{71}
$$

Assume symmetric positive definite R_k and P_k and get

$$
\begin{aligned}
&\min_{\chi_{k,i}^{x+}} \left\{ y_k^T R_k^{-1} y_k - 2 y_k^T R_k^{-1} D_k \chi_{k,i}^x + \chi_{k,i}^{Tx} D_k^T R_k^{-1} D_k \chi_{k,i}^x + \chi_{k,i}^{Tx} (P_{x_k}^-)^{-1} \chi_{k,i}^x \right. \\
&\quad \left. - 2 \chi_{k,i}^{Tx-} (P_{x_k}^-)^{-1} \chi_{k,i}^x + \chi_{k,i}^{Tx-} (P_{x_k}^-)^{-1} \chi_{k,i}^{x-} \right\}
\end{aligned}
\tag{72}
$$

However minimizing (72) is the same as minimizing the QP-problem

$$
\begin{aligned}
&\min_{\chi_{k,i}^{x+}} \left\{ (\chi_{k,i}^{Tx} D_k^T R_k^{-1} D_k \chi_{k,i}^x - 2 y_k^T R_k^{-1} D_k \chi_{k,i}^x) + (\chi_{k,i}^{Tx} (P_{x_k}^-)^{-1} \chi_{k,i}^x - 2 \chi_{k,i}^{Tx-} (P_{x_k}^-)^{-1} \chi_{k,i}^x) \right\} \\
&= \min_{\chi_{k,i}^{x+}} \left\{ \chi_{k,i}^{Tx} \underbrace{(D_k^T R_k^{-1} D_k + (P_{x_k}^-)^{-1})}_{H_k} \chi_{k,i}^x - 2 \underbrace{(y_k^T R_k^{-1} D_k + \chi_{k,i}^{Tx-} (P_{x_k}^-)^{-1})}_{f_{k_i}^T} \chi_{k,i}^x \right\} \\
&= \min_{\chi_{k,i}^{x+}} \left\{ \chi_{k,i}^{Tx} H_k \chi_{k,i}^x - 2 f_{k_i}^T \chi_{k,i}^x \right\} \\
&= \min_{\chi_{k,i}^{x+}} \left\{ \frac{1}{2} \chi_{k,i}^{Tx} \underbrace{2 H_k}_{\tilde{H}_k} \chi_{k,i}^x - \underbrace{2 f_{k_i}^T}_{\tilde{f}_{k_i}^T} \chi_{k,i}^x \right\} \\
&= \min_{\chi_{k,i}^{x+}} \left\{ \frac{1}{2} \chi_{k,i}^{Tx} \tilde{H}_k \chi_{k,i}^x - \tilde{f}_{k_i}^T \chi_{k,i}^x \right\}
\end{aligned}
\tag{73}
$$

Appendix B. Performance measures

All simulations and development of the algorithms was done in Matlab Release 14. The computer used was a Dell Precision M70 with Intel Pentium M processor running at 2.13 Ghz and 2 GB RAM.

B.1. Computational performance

As a guide to compare and determine the performance of the various algorithms we measured the average time to run 10 simulations of the case '2-state CSTR'. The results are shown in Tables 11–13. The notation in the tables is as follows: Column labeled Ch: Simulated using Cholesky matrix square root. Column labeled Sy: Simulated using symmetric matrix square root. Column labeled Ch+Re: Simulated using Cholesky matrix square root and

the reformulated correction steps. Column labeled Sy+Re: Simulated using symmetric matrix square root and the reformulated correction steps. All values are in seconds [s].

Note that the results in Table 11 are performance measures only, since none of the unconstrained algorithms gave acceptable performance with respect to the estimation accuracy. The results in Table 11 indicate that some of the UKF algorithms in its plain version are not necessarily more computational demanding than the EKF, given the tested case. However, as the number of the sigma points increase, the computational demand increases as well.

Note that for the simulation results shown in Table 12 the constraints used are $\chi^x_{k-1} \geq 0$ (CC_1) in all the algorithms. For the simulations using the reformulated correction steps, the constraints $\chi^x_k \geq 0$ (CC_7) are also used. The notation in the table is as follows: Column labeled QP Ch: Simulated using Cholesky square root and the QP version of the correction steps. Column labeled QP Sy: Simulated using symmetric square root and the QP version of the correction steps. Column labeled NLP Ch: Simulated using Cholesky square root and the NLP version of the correction steps. Column labeled NLP Sy: Simulated using Symmetric matrix square root and the NLP version of the correction steps. All values are in seconds [s]

Note also that for the simulation results shown in Table 13 the constraints used are $\chi^x_{k-1} \geq 0$ (CC_1) in all the algorithms. For the QP UKF's the cost function used is as by (33). For the NLP UKF's the cost function used is as by (32).

Tables 11–13 indicates that adding complexity to the UKF algorithms clearly has a computational cost, but as will be shown below, with a potential increase in estimation accuracy.

B.2. Estimation performance

As a guide to compare and determine the performance of the various algorithms we measured the average square root error of the true and estimated state by 100 Monte Carlo simulations of the case '2-state CSTR'. The error was measured as

$$\bar{e} = \frac{1}{100}\sum_{j=1}^{100}(\tilde{e}_{i,j}); \quad i \in [1,2] \tag{74}$$

$$\tilde{e}_{i,j} = \frac{1}{100}\sum_{k=1}^{100}\sqrt{(x_{i,k}-\hat{x}_{i,k})(x_{i,k}-\hat{x}_{i,k})^T}; \quad i \in [1,2] \tag{75}$$

where $\tilde{e}$ is a vector for the average square root error in simulation run j, and $\bar{e}$ is a vector containing the average value of the states of all the average square root errors.

The results are shown in Tables 14 and 15.[13] The notation in the table is as follows: Column labeled Ch: Simulated using Cholesky square root. Column labeled Sy: Simulated using symmetric square root. Column labeled Ch+Re: Simulated using Cholesky square root and the reformulated correction steps. Column labeled Sy+Re: Simulated using symmetric matrix square root and the reformulated correction steps.

Note that for the simulation results shown in Table 14 the constraints used are $\chi^x_{k-1} \geq 0$ (CC_1) in all the algorithms. For the simulations using the reformulated correction steps, the constraints $\chi^x_k \geq 0$ (CC_7) are also used. The notation in the table is as follows: Column labeled QP Ch: Simulated using Cholesky square root and the QP version of the correction steps. Column labeled QP Sy: Simulated using Symmetric square root and the QP version of the correction steps. Column labeled NLP Ch: Simulated using Cholesky square root and the NLP version of the correction steps. Column labeled NLP Sy: Simulated using Symmetric matrix square root and the NLP version of the correction steps.

Note also that for the simulation results shown in Table 13 the constraints used are $\chi^x_{k-1} \geq 0$ (CC_1) in all the algorithms. For the QP UKF's the cost function used is as by (33). For the NLP UKF's the cost function used is as by (32).

Tables 11–13 indicates that adding complexity to the original UKF algorithms have a computational cost, but as Tables 14 and 15 indicates, with a potential increase in estimation accuracy, even though the most complex not necessarily gives the most accurate estimation. Given a system, which algorithm to select is a trial and error process.

[13] Notation: UKF D. Simon means the algorithm in Table 4. UKF Non-aug means the algorithm in Table 5. UKF Q-aug means the algorithm in Table 6. UKF Fully-aug means the algorithm in Table 7.

References

Akin, B., Orguner, U., & Ersak, A. (2003). State estimation of induction motor using Unscented Kalman Filter. In *Proceedings of 2003 IEEE Conference on Control Applications, 2* (pp. 915–919).

Bellatoni, J., & Dodge, K. (1967). A square root formulation of the Kalman Schmidt filter. *AIAA Journal*, *5*, 1309–1314.

Bizup, D. F., & Brown, D. E. (2003). The over-extended Kalman filter—dont use it! In *Proceedings of the Sixth International Conference of, 1* (pp. 40–46).

Briers, M., Maskell, S. R., & Wright, R. (2003). A Rao–Blackwellised un-scented Kalman filter. In *Proceedings of the 6th International Conference of Information Fusion, 1* (pp. 55–61).

Chen, W., Lang, L., Bakshi, B. R., Goel, P. K., & Ungarla, S. (2007). Bayesian Estimation via Sequential Monte Carlo Sampling – Constrained Dynamic Systems. *Automatica*, *43*, 1615–1622.

Daum, F. E. (1986). Beyond Kalman filters: Practical design of nonlinear filters. *SPIE*, *2561*, 252–262.

Daum, F. E. (2005). Nonlinear filters: Beyond the Kalman filters. *IEEE AE System Magazine*, *20*(8), 57–69.

Hao, Y., Xiong, Z., Sun, F., & Wang, X. (2007). Comparison of un-scented Kalman filters. In *Proceedings of the 2007 IEEE ICMA, University of Wisconsin-Madison, August 2007* (pp. 895–899).

Hasseltine, E. L., & Rawlings, J. B. (2003). *A critical evaluation of extended Kalman filtering and moving horizon estimation. TWMCC*. Technical report number 2002–03.

Julier, J., & Laviola, J. J. (2007). On Kalman filtering with nonlinear equality constraints. *IEEE Transactions on Signal Processing*, *55*(6), 2774–2784.

Julier, S. J. (2002). The scaled unscented transform. In *Proceedings of the American Control Conference, 6* (pp. 4555–4559).

Julier, S. J. (2003). The spherical simplex unscented transformation. In *Proceedings of the American Control Conference, June 4–6* (pp. 2430–2434).

Julier, S. J., & Uhlmann, J. K. (1994). *A general method for approximating nonlinear transformation of probability distributions*. Technical report. RRG, Dept. of Engineering Science, University of Oxford. http://www.robots.ox.ac.uk/~siju/work/publications/lettersize/Unscented.zip (August 1994).

Julier, S. J., & Uhlmann, J. K. (2002). Reduced sigma point filters for the propagation of means and covariances through nonlinear transformations. In *Proceedings of the American Control Conference, May 8–10* (pp. 887–892).

Julier, S. J., & Uhlmann, J. K. (March 2004). Unscented filtering and nonlinear estimation. *Proceedings IEEE*, *92*(3), 401–422.

Julier, S. J., Uhlmann, J. K., & Durrant-Whyte, H. (1995). A new approach for filtering nonlinear system. In *Proceedings of the American Control Conference* (pp. 1628–1632).

Julier, S. J., Uhlmann, J. K., & Durrant-Whyte, H. (1997). A new extension of the Kalman filter to nonlinear systems. In *Proceedings of Aerospace/Sense: The 11th International Symposium On Aerospace/Defence Sensing, Simulation and Controls* (pp. 1628–1632).

Julier, S. J., Uhlmann, J. K., & Durrant-Whyte, H. (2000). A new method for the nonlinear transformation of means and covariances in filters and estimators. *IEEE Trans. on Automatic Control*, *45*(3), 477–482.

Kalman, R. (1960). A new approach to linear filtering and prediction problems. *Transaction of the ASME-Journal of Basic Engineering*, *82*(Series D), 35–45.

Kandepu, R., Foss, B., & Imsland, L. (2008). Applying the unscented Kalman filter for nonlinear state estimation. *Journal of Process Control*, *18*, 753–768.

Laviola, J. J. (2003). A comparison of unscented and extended Kalman filtering for estimating quaternion motion. In *American Control Conference, 2003. Proceedings, 3* (pp. 2435–2440).

Li, W., & Leung, H. (2004). Simultaneous registration and fusion of multiple dissimilar sensors for cooperative driving. *IEEE Transactions on Intelligent Transportations Systems*, *5*(2), 84–98.

Nørgaard, M., Poulsen, N. K., & Ravn, O. (2000). New developments in state estimation for nonlinear systems. *Automatica*, *36*, 1627–1638.

Pieper, R. J. B. (2007). *Comparing estimation algorithms for camera position and orientation*. Department of Electrical Engineering, Linköpings universitet, SE581 83 Linköping, Sweden.

Rao, C. V. (2000). *Moving horizon strategies for the constrained monitoring and control of nonlinear discrete-time systems*. PhD thesis. Chemical Engineering, University of Wisconsin-Madison.

Rawlings, J. B., & Bakshi, B. R. (2006). Particle filter and moving horizon estimation. *Computers and Chemical Engineering*, *30*, 1529–1541.

Romanenko, A., & Castro, J. A. A. M. (2004). The unscented filter as an alternative to the EKF for nonlinear state estimation: A simulation case study. *Computers and Chemical Engineering*, *28*, 347–355.

Romanenko, A., Santos, L. O., & Afonso, P. A. F. N. A. (2004). Unscented Kalman filtering of a simulated pH system. *Industrial Engineering and Chemistry Research*, *43*, 7531–7538.

Schei, T. S. (1997). A finite-difference method for linearization in nonlinear algorithms. *Automatica*, *33*(11), 2053–2058.

Simon, D. (2006). *Optimal state estimation*. Wiley. ISBN: 10-0-471-70858-5

Simon, D., & Simon, D. L. (2005). Aircraft turbofan engine health estimation using constrained Kalman filtering. *ASME Journal of Engineering for Gas Turbine and Power*, *127*, 323–328.

Simon, D., & Simon, D. L. (2006). Kalman filtering with inequality constraints for turbofan engine health estimation. *IEEE Proceedings Control Theory and Applications*, *153*(3), 371–378.

Tenne, D., & Singh, T. (2003). The higher order unscented filter. In *Proceedings of the American Control Conference, 3* (pp. 2441–2446).

Vachhani, P., Narasimhan, S., & Rengaswamy, R. (2006). Robust and reliable estimation via unscented recursive nonlinear dynamic data reconciliation. *Journal of Process Control*, *16*, 1075–1086.

van der Merwe, R. (2004). *Sigma-point Kalman filters for probabilistic inference in dynamic state-space models*. PhD dissertation. OGI School of Science and Engineering at Oregon Health and Science University.

Xiong, K., Chang, H. Y., & Chan, C. W. (2006). Performance evaluation of UKF-based nonlinear filtering. *Automatica*, *42*, 261–270.

Xiong, K., Chang, H. Y., & Chan, C. W. (2007). Authors reply to "Comments on. Performance evaluation of UKF-based nonlinear filtering". *Automatica*, *43*, 569–570.

Zandt, J. R. V. (2001). A more robust unscented transform. *SPIE Proceedings of AeroSense: Signal and Data Processing of Small Targets*, *4473*, 371–380.