

Comparison of Unscented Kalman Filter Design for Agricultural Anaerobic Digestion Model

Simon Hellmann^{*§}, Terrance Wilms[‡], Stefan Streif[§] and Sören Weinrich^{¶*}

^{*}DBFZ Deutsches Biomasseforschungszentrum, Leipzig, Germany, simon.hellmann@dbfz.de

[§]Chemnitz University of Technology, Chemnitz, Germany, stefan.streif@etit.tu-chemnitz.de

[‡] Technische Universität Berlin, Berlin, Germany, terrance.wilms@tu-berlin.de

[¶]Münster University of Applied Sciences, Münster, Germany, weinrich@fh-muenster.de

Abstract—Dynamic operation of biological processes, such as anaerobic digestion (AD), requires reliable process monitoring to guarantee stable operating conditions at all times. Unscented Kalman filters (UKF) are an established tool for nonlinear state estimation, and there exist numerous variants of UKF implementations, treating state constraints, improvements of numerical performance and different noise cases. So far, however, a unified comparison of proposed methods emphasizing the algorithmic details is lacking. The present study thus examines multiple unconstrained and constrained UKF variants, addresses aspects crucial for direct implementation and applies them to a simplified AD model. The constrained UKF considering additive noise delivered the most accurate state estimations. The long run time of the underlying optimization could be vastly reduced through pre-calculated gradients and Hessian of the associated cost function, as well as by reformulation of the cost function as a quadratic program. However, unconstrained UKF variants showed lower run times at competitive estimation accuracy. This study provides useful advice to practitioners working with nonlinear Kalman filters by paying close attention to algorithmic details and modifications crucial for successful implementation.

Index Terms—Process monitoring, nonlinear state estimation, sigmoid point Kalman filter, biogas technology, ADM1

I. INTRODUCTION

Anaerobic digestion (AD) is an established technology for the treatment of biogenic waste. In the AD process, organic matter is converted into biogas [1]. Demand-driven operation of biological processes such as AD requires reliable process monitoring to ensure stable operation [2]. As a means of process monitoring, Kalman filters have been examined in numerous studies for model-based online state estimation in various domains [3], [4]. In particular, the Unscented Kalman Filter (UKF) could be shown to be well suited for state estimation of nonlinear biological processes [5]–[7].

To this end, Kolas et al. (2009) [8] investigated various implementations of the UKF, involving different noise cases (additive and non-additive) as well as state constraints. More specifically, state constraints were addressed by adopting a nonlinear program (NLP) proposed by [9], and by reformulating the NLP as a quadratic program (QP) assuming linear output equations.

The authors are thankful for funding from German Federal Ministry of Food and Agriculture of the junior research group on simulation, monitoring and control of anaerobic digestion plants (grant no. 2219NR333). S. H. thanks Maik Gentsch for his crucial advice and encouragement, and Julius Frontzek for the insights he contributed through cubature Kalman filter design.

Weinrich and Nelles (2021) have recently proposed simplified AD models [10] derived from the highly complex Anaerobic Digestion Model No. 1 (ADM1) [11]. The potential of these ADM1 simplifications has been demonstrated in case studies addressing demand-oriented biogas production in lab- [12] and full-scale [13]. Moreover, the ADM1 simplifications have been shown to be locally observable, and thus appropriate to be applied in state estimation [14].

This paper compares different UKF designs for a simplified ADM1 model which is derived from [10]. By demonstrating multiple UKF implementations, we aim to provide insights into comparative performance of available algorithms, and thereby offer analytical, numerical and algorithmic guidance. The study thus also contributes to realizing model-based monitoring and control for demand-driven operation of AD plants.

II. UNSCENTED KALMAN FILTERING

In this work, we consider discrete-time stochastic systems

$$x_{k+1} = f(x_k, u_k) + v_k, \quad x_0 - \text{given} \quad (1a)$$

$$y_k = h(x_k) + w_k. \quad (1b)$$

with state variables $x \in \mathbb{R}^n$, control variables $u \in \mathbb{R}^p$, and measurement variables $y \in \mathbb{R}^q$. f can also represent the integration of continuous-time differential equations on a discrete time grid $t_k = k \Delta t$ with sample time Δt and $k \in \mathbb{N}_0$, see [9]. Process and measurement noise ($v \in \mathbb{R}^n$ and $w \in \mathbb{R}^q$) are assumed Gaussian and zero-mean with

$$E\{v(k)\} = 0, E\{w(k)\} = 0, \quad \forall k \quad (2a)$$

$$E\{v(k)v^T(l)\} = Q(k)\delta_{k,l}, \quad (2b)$$

$$E\{w(k)w^T(l)\} = R(k)\delta_{k,l}. \quad (2c)$$

Q and R are process and measurement noise covariance matrices and $\delta_{k,l}$ is the Kronecker delta. (1) shows the additive noise case. In case of non-additive noise, v_k and w_k are direct arguments of f and h , i.e., $f(x_k, u_k, v_k)$ and $h(x_k, w_k)$. The nominal linear time-variant equivalent of the nonlinear output equation (1b) is denoted as

$$y_k = C_k x_k. \quad (3)$$

Sigma point Kalman filters such as the UKF use scaled copies of the old estimate \hat{x}_{k-1} called sigma points to predict a-priori estimates \hat{x}_k^- (time update). These are in turn corrected with measurements y_k to deliver a-posteriori estimates \hat{x}_k (measurement update). The analogous procedure is applied for calculation of the state error covariance matrix P_{k-1} with corresponding prior P_{k-1}^- and posterior P_k . Each time step involves sigma points to be sampled from a scaled multivariate normal distribution with mean \hat{x}_{k-1} and covariance matrix proportionate to P_{k-1} [15].

A. Unconstrained Case

The basic concepts of the conventional unconstrained UKF are briefly summarized here. Sigma points χ_i are sampled around the state estimate \hat{x} which involves a scaling factor γ [15], given by

$$\gamma = \sqrt{n + \lambda} \quad \text{with} \quad (4a)$$

$$\lambda = \alpha^2(n + \kappa) - n. \quad (4b)$$

For Gaussian noise, nominal tuning parameter values are recommended by [8] as

$$[\alpha \quad \beta \quad \kappa] = [1 \quad 2 \quad 0]. \quad (5)$$

During time and measurement update, sigma points are aggregated as a weighted average with weights W^x for states and W^c for the state error covariance matrix [15]

$$W_0^x = \lambda / (n + \lambda), \quad (6a)$$

$$W_0^c = \lambda / (n + \lambda) + 1 - \alpha^2 + \beta, \quad (6b)$$

$$W_i^x = W_i^c = 1 / (2(n + \lambda)), \quad i = 1 \dots 2n. \quad (6c)$$

Augmentation: Non-additive process noise is incorporated by extending the matrix P with the process noise covariance matrix Q . Thereby the system state is augmented with zero-mean process noise [8] (denoted with superscript index a). This results in an augmented system order $L = 2n$

$$P_{k-1}^a = \begin{bmatrix} P_{k-1} & 0 \\ 0 & Q_{k-1} \end{bmatrix}, \quad (7a)$$

$$x_{k-1}^a = \begin{bmatrix} x_{k-1} \\ 0 \end{bmatrix}, \quad \chi_{k-1} = \begin{bmatrix} \chi_{k-1}^x \\ \chi_{k-1}^v \end{bmatrix}. \quad (7b)$$

Analogously, non-additive zero-mean process and measurement noise is incorporated by extending P with the process and measurement noise covariance matrices Q and R . This results in the fully augmented system order $L = 2n + q$

$$P_{k-1}^a = \begin{bmatrix} P_{k-1} & 0 & 0 \\ 0 & Q_{k-1} & 0 \\ 0 & 0 & R_{k-1} \end{bmatrix}, \quad x_{k-1}^a = \begin{bmatrix} x_{k-1}^T & 0 & 0 \end{bmatrix}^T, \quad (8a)$$

$$\chi_{k-1} = \left[(\chi_{k-1}^x)^T \quad (\chi_{k-1}^v)^T \quad (\chi_{k-1}^w)^T \right]^T. \quad (8b)$$

The distinction between nominal and augmented system order n and L is not addressed in [8] and is therefore clarified here. For the augmented and fully augmented version, computation of the weights (6) must be adjusted for the increased system order L . This ensures that the aggregation

from sigma points to estimates is maintained properly. By contrast, computation of the scaling factor (4) must still be conducted with the nominal system order n . Otherwise the effect of P_k during sampling of sigma points deviates from the additive noise version.

In [8] a numerically more robust reformulation is proposed for computing the a-posteriori estimates \hat{x}_k and P_k . This involves a separate update of the sigma point priors χ_k^{x-} through the innovation

$$\chi_{k,i}^x = \chi_{k,i}^{x-} + K_k \left(y_k - h(\chi_{k,i}^{x-}, \chi_{k,i}^w) \right), \quad i = 0 \dots 2L \quad (9)$$

and then to aggregate them to posteriors \hat{x}_k and P_k

$$\hat{x}_k = \sum_{i=0}^{2L} W_i^x \chi_{k,i}^x \quad (10a)$$

$$P_k = \sum_{i=0}^{2L} W_i^c (\chi_{k,i}^x - \hat{x}_k) (\chi_{k,i}^x - \hat{x}_k)^T. \quad (10b)$$

The derivation was conducted for the fully augmented case in [8]. For additive and augmented noise cases, $h(\chi_{k,i}^{x-}, \chi_{k,i}^w)$ in (9) must be replaced with $h(\chi_{k,i}^{x-})$. Further, computation of P_k must be slightly adjusted for additive noise

$$P_k = \sum_{i=0}^{2L} W_i^c (\chi_{k,i}^x - \hat{x}_k) (\chi_{k,i}^x - \hat{x}_k)^T + Q_k + K_k R_k K_k^T$$

with K_k according to [8, Tab. 5]. For augmented process noise, P_k is correctly computed as

$$P_k = \sum_{i=0}^{2L} W_i^c (\chi_{k,i}^x - \hat{x}_k) (\chi_{k,i}^x - \hat{x}_k)^T + K_k R_k K_k^T$$

with K_k according to [8, Tab. 6].

B. Constrained Case

Upper and lower bounds of a-posteriori estimates can be accounted for by clipping [8]. To account for nonlinear inequality constraints [8] adopted the NLP proposed by [9] but suggested to leave the scaling factor and weights as in (4) and (6), resulting in

$$\chi_{k,i}^x = \arg \min_{\chi_{k,i}^x} J_{k,i}^{NLP} \quad (11a)$$

$$\text{s.t. } \tilde{C}(\chi_{k,i}^x) \leq 0, \quad \text{where} \quad (11b)$$

$$J_{k,i}^{NLP} = (y_k - h(\chi_{k,i}^x))^T R_k^{-1} (y_k - h(\chi_{k,i}^x)) + (\chi_{k,i}^x - \chi_{k,i}^{x-})^T (P_k^-)^{-1} (\chi_{k,i}^x - \chi_{k,i}^{x-}). \quad (12)$$

The scope of this study is limited to linear inequality constraints. Therefore the nonlinear inequalities in (11b) read

$$\tilde{C}(\chi_{k,i}^x) = A \chi_{k,i}^x - b \leq 0 \quad (13)$$

with \tilde{C} and $b \in \mathbb{R}^{m \times 1}$, $A \in \mathbb{R}^{m \times n}$ and m as the number of constraints. Upper and lower bounds on a-posteriori estimates and sigma points can be included in the linear inequalities

(13). For linear output equations (3), [8] further showed that the NLP cost function (12) can be recast into the QP-form

$$J_{k,i}^{QP} = (\chi_{k,i}^x)^T \left(C_k^T R_k^{-1} C_k + (P_k^-)^{-1} \right) \chi_{k,i}^x + \quad (14)$$

$$- 2 \left(y_k^T R_k^{-1} C_k + (\chi_{k,i}^{x-})^T (P_k^-)^{-1} \right) \chi_{k,i}^x.$$

The NLP was solved in Matlab by using `fmincon`, and by using `quadprog` for the QP.

Remark: Note that (12) was proposed by [9] for the additive noise case, but [8] adopted it for the augmented noise cases also. Since (14) was derived in [8] considering the nominal output equation $h(x_k)$ for additive measurement noise, (14) also holds for additive noise. Lastly, as stated in [9] the posteriors \hat{x}_k and P_k must be computed from the solution of the NLP/QP as per (10).

III. IMPROVEMENTS OF NUMERICAL EFFICIENCY

The algorithmic implementations of this study involved numerically challenging operations, whose efficiency could be improved through modifications described in the following. The code was implemented in Matlab (Version R2022b) using the System Identification Toolbox (Version 10.0) [16].

A. Cholesky Decomposition

During each iteration of the UKF, the matrix square root of P_k needs to be computed, which is typically done by means of the Cholesky decomposition [15].

B. Square Root Version

The computational effort associated with computing the Cholesky factors can be reduced by directly updating the square root of P_k in each iteration. Therefore, the square root UKF was proposed in [15] for additive noise and is reported to show improved numerical stability compared with the conventional UKF [17].

C. Accelerating Optimization Through Gradients and Hessian

In Matlab, the standard solver for NLPs is `fmincon`, which by default approximates gradient and Hessian of the cost function through finite differences [18]. When providing analytic expressions of gradient and Hessian, computational efficiency as well as numerical robustness can be vastly increased. For this reason, these expressions are derived in the following.

The inequality constraints are merged into the cost function through Lagrange multipliers μ , delivering the Lagrangian

$$L_{k,i} = J_{k,i}^{NLP} + \mu^T (A \chi_{k,i}^x - b) \quad (15)$$

with $\mu \in \mathbb{R}^{m \times 1}$. The gradient of the Lagrangian reads

$$\frac{d}{d\chi_{k,i}^x} L_{k,i} = \frac{d}{d\chi_{k,i}^x} J_{k,i}^{NLP} + \mu^T A. \quad (16)$$

Further, the gradient of the cost function is computed as

$$\frac{d}{d\chi_{k,i}^x} J_{k,i}^{NLP} = \frac{\partial J_{k,i}^{NLP}}{\partial h} \frac{\partial h}{\partial \chi_{k,i}^x} + \frac{\partial J_{k,i}^{NLP}}{\partial \chi_{k,i}^x}, \quad \text{where (17a)}$$

$$\frac{\partial J_{k,i}^{NLP}}{\partial h} = -2 (y - h(\chi_{k,i}^x))^T R^{-1} \quad \text{and} \quad (17b)$$

$$\frac{\partial J_{k,i}^{NLP}}{\partial \chi_{k,i}^x} = 2 (\chi_{k,i}^x - \chi_{k,i}^{x-})^T (P_k^-)^{-1} \quad (17c)$$

with

$$\frac{\partial h}{\partial \chi_{k,i}^x} = \left. \frac{\partial h}{\partial x} \right|_{\chi_{k,i}^x}. \quad (18)$$

For linear output equations as in (3), (18) reduces to C_k . Finally, the Hessian of the Lagrangian reads

$$\frac{d^2}{d(\chi_{k,i}^x)^2} L_{k,i} = 2 (P_k^-)^{-1} + 2 \left(R_k^{-1} \frac{\partial h}{\partial \chi_{k,i}^x} \right)^T \frac{\partial h}{\partial \chi_{k,i}^x}. \quad (19)$$

IV. MODELLING OF THE ANAEROBIC DIGESTION PROCESS

The model equations are derived from the ADM1-R4 proposed by [10]. Water and nitrogen were omitted because they are quasi-autonomous states as shown in [14]. Furthermore, the gas phase was neglected to describe only the core of AD process, that is the degradation from macro nutrients to dissolved methane (CH_4) and carbon dioxide (CO_2).

The state vector comprises the mass concentrations (in kg m^{-3}) of the six states CH_4 , CO_2 , carbohydrates (ch), proteins (pr), lipids (li) and microbial biomass (bac)

$$x = [x_1 \ x_2 \ x_3 \ x_4 \ x_5 \ x_6]^T = [S_{\text{ch4}} \ S_{\text{co2}} \ X_{\text{ch}} \ X_{\text{pr}} \ X_{\text{li}} \ X_{\text{bac}}]^T.$$

The state differential equations read as follows:

$$\begin{aligned} \dot{x}_1 &= c_1 (\xi_1 - x_1) u + a_{11} c_2 x_3 + a_{12} c_3 x_4 + a_{13} c_4 x_5 \\ \dot{x}_2 &= c_1 (\xi_2 - x_2) u + a_{21} c_2 x_3 + a_{22} c_3 x_4 + a_{23} c_4 x_5 \\ \dot{x}_3 &= c_1 (\xi_3 - x_3) u - c_2 x_3 + a_{34} c_5 x_6 \\ \dot{x}_4 &= c_1 (\xi_4 - x_4) u - c_3 x_4 + a_{44} c_5 x_6 \\ \dot{x}_5 &= c_1 (\xi_5 - x_5) u - c_4 x_5 + a_{54} c_5 x_6 \\ \dot{x}_6 &= c_1 (\xi_6 - x_6) u + a_{61} c_2 x_3 + a_{62} c_3 x_4 + \\ &\quad + a_{63} c_4 x_5 - c_5 x_6. \end{aligned}$$

The dissolved gas concentrations of CH_4 and CO_2 as well as microbial biomass were assumed to be measurable:

$$y = \begin{bmatrix} x_1 \\ x_2 \\ x_6 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} x. \quad (20)$$

The substrate feed volume flow acts as the control variable u . Model parameters a , c and ξ were derived in the extended version of this manuscript [19], where the simulation scenario and the model are also described in more detail.

TABLE I
MODEL PARAMETERS USED FOR SYNTHETIC MEASUREMENT CREATION (TRUE VALUE) AND FOR UNSCENTED KALMAN FILTERING (UKF VALUE).

	c_2 [h ⁻¹]	c_3 [h ⁻¹]	c_4 [h ⁻¹]	c_5 [h ⁻¹]
true value	0.25	0.20	0.10	0.020
UKF value	0.32	0.26	0.13	0.026

TABLE II
OVERVIEW OF ALL IMPLEMENTED UKF VERSIONS WITH SHORT DESCRIPTION. EXPLANATIONS ARE GIVEN IN THE TEXT.

Description	Unconstrained		Constrained
Toolbox ^a	UKF-sysID		
Square root	UKF-SR	UKF-SR- γ	
Additive	UKF-add	UKF-add- γ	cUKF-add
Augmented	UKF-aug	UKF-aug- γ	cUKF-aug
Fully augm. ^b	UKF-fully-aug	UKF-fully-aug- γ	cUKF-fully-aug

^a Matlab System Identification Toolbox, ^b fully augmented

A. Simulation Scenario

A pilot-scale AD reactor with 100 L liquid volume was fed dynamically for one week with a substrate mix of maize silage and cattle manure, starting in steady state conditions.

Measurements were assumed to be taken every $\Delta t = 0.5$ h, resulting in $N = 337$ samples. Nominal measurements were superimposed with additive, zero-mean Gaussian noise with standard deviations σ_i as stated in [19]. To ensure comparability among all implemented UKFs, all of them were equally tuned as follows [20]:

$$\begin{aligned} x_0 &= [4.09, 10.52, 11.04, 2.57, 0.96, 2.02]^T \\ \hat{x}_0 &= [2.20, 19.30, 24.94, 2.22, 0.31, 2.64]^T \\ P_0 &= \text{diag}\{(\hat{x}_0 - x_0)^2\} \\ R &= 1.5 \cdot \text{diag}\{(\sigma_i)^2\} \\ Q &= \text{diag}\{[1, 1, 1, 1, 1, 1]\}. \end{aligned}$$

A plant-model mismatch as stated in Table I was assumed.

B. Normalized Root Mean Squared Error

To evaluate estimation accuracy, the normalized root mean squared error (NRMSE) between estimated and true values was used:

$$\text{NRMSE} = \frac{\sqrt{1/N \sum_i (\hat{x}_i - x_i)^2}}{1/N \sum_i x_i}. \quad (21)$$

This ratio can be computed both for states x and outputs y , and is denoted accordingly as NRMSE_x or NRMSE_y .

V. SIMULATION STUDIES

Manifold UKF variants have been implemented as summarized in Table II. They are classified as unconstrained and constrained UKFs. The former are described in Section V-A, the latter in Section V-B. Section V-C summarizes the performance of all implemented UKF variants.

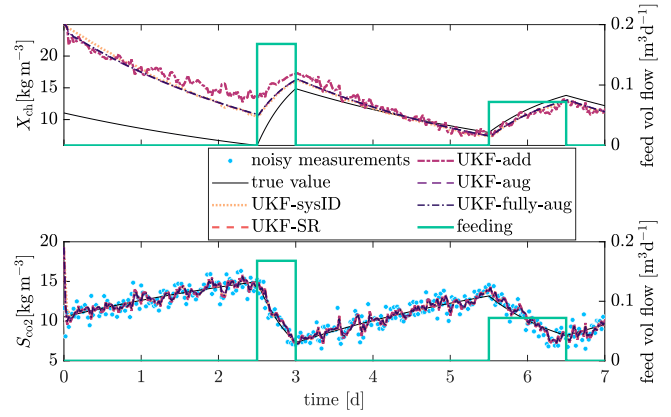


Fig. 1. Comparison of state estimation quality through different UKFs with nominal sigma point scaling. Top: Concentration of carbohydrates and corresponding estimations. Bottom: Concentration of dissolved CO₂, measurements and corresponding estimations.

TABLE III
COMPARISON OF UNCONSTRAINED UKF PERFORMANCE WITH DEFAULT SIGMA POINT SCALING ($\gamma = 2.4495$) ACCORDING TO (4) AND (5).

Algorithm	NRMSE_x^a	NRMSE_y^b	Run time [s]
UKF-sysID	0.4888	0.1152	1.73
UKF-SR	0.8533	0.1157	2.16
UKF-add	0.8533	0.1157	2.29
UKF-aug	0.3599	0.0934	3.66
UKF-fully-aug	0.3599	0.1081	4.14

^aaverage value of all non-measurable states ^baverage value of all measurable states

A. Unconstrained Case

Unconstrained UKF implementations using the default sigma point scaling are discussed first. Then the effect of reduced sigma point scaling is analyzed. UKF-sysID denotes the UKF implemented with the Matlab System Identification Toolbox assuming additive noise. It serves as a benchmark.

1) *Nominal Sigma Point Scaling*: The following algorithms were implemented and compared with the benchmark UKF-sysID, all considering nominal sigma point scaling (4) with default UKF tuning parameters (5), see Table II: square root UKF according to [21] (UKF-SR), as well as the unconstrained UKFs according to [8] assuming additive noise (UKF-add), non-additive process noise (UKF-aug) and non-additive process and measurement noise (UKF-fully-aug). Note that although the model used in this study assumes additive process and measurement noise, the augmented and fully augmented algorithm variants can still be applied.

Figure 1 compares state estimation performance of the unconstrained UKF variants by means of carbohydrates and CO₂ concentrations.

Smoothing of measurable states, such as dissolved CO₂ in the bottom of Figure 1, is very similar for all UKF versions. This is reinforced through nearly identical values of NRMSE_y , see Table III. The noise-free output is not met exactly, but the filters clearly smooth the noisy measurements, underlined by low NRMSE_y .

For the non-measurable states, such as carbohydrates in the top of Figure 1, all filters approach the true trajectory despite the initial estimation error. As of $t \approx 4$ d, estimations agree

TABLE IV
COMPARISON OF UNCONSTRAINED UKF PERFORMANCE WITH
MODIFIED SIGMA POINT SCALING ($\gamma = 1$).

Algorithm	NRMSE _x ^a	NRMSE _y ^b	Run time [s]
UKF-SR- γ	0.3733	0.0657	2.18
UKF-add- γ	0.3733	0.0657	2.15
UKF-aug- γ	0.3691	0.0647	3.84
UKF-fully-aug- γ	0.3695	0.1046	4.40

^aaverage value of all non-measurable states ^baverage value of all measurable states

well with true values. In light of a plant-model mismatch, it is plausible that they do not match exactly.

Nevertheless, convergence behavior of the algorithms differs. UKF-add and UKF-SR deliver identical results, and hence show overlapping graphs and the same NRMSE_x. They both do not involve augmentation, so essentially their flow of algorithmic operations is the same. Their identical performance emphasizes that the square root formulation of the additive UKF derived in [21] is equivalent to the conventional additive UKF. However, both UKF-add and UKF-SR show a slower convergence than UKF-sysID, clearly visible before the first feeding, and reflected in a higher NRMSE_x. Since UKF-sysID and UKF-SR are both based on the square root UKF [22], their graphs should match. However, they may deviate because UKF-sysID internally uses slightly different sigma point scaling and weighting than stated in [21]. Another reason might be different numerical performance. This is likely the case, especially because all three additive UKFs delivered negative state estimates for the lipids concentration (x_5) before the first feeding. To this end, UKF-add exhibited the lowest estimates of x_5 . Since negative concentrations are beyond the physically meaningful domain of the model, these negative estimates of x_5 might also influence the other state estimates.

By contrast, the augmented versions UKF-aug and UKF-fully-aug deliver no negative and much less noisy state estimates, see Figure 1. They approach the true trajectory faster and yield a lower NRMSE_x than UKF-sysID, see Table III. Run times of all additive UKF versions over the entire simulation horizon are in the same range of about 2 s. Among them, UKF-sysID is the fastest. This is plausible since it is a commercial implementation optimized for numerical efficiency. By comparison, run times of UKF-aug and UKF-fully-aug are clearly higher, reflecting the higher resulting system order, cf. (7) and (8).

2) *Reduced Sigma Point Scaling*: For all unconstrained UKF variants, except the benchmark, the sigma point scaling was reduced by changing the scaling factor γ from 2.4495 (according to nominal UKF scaling (4)) to 1. The resulting performance is illustrated in Figure 2, with corresponding graphs indicated by the name extension $-\gamma$. Performance improves especially for the additive variants UKF-add- γ and UKF-SR- γ . This manifests in lower NRMSE values than for nominal scaling and also than UKF-sysID. At the same time, about the same run times are maintained, as illustrated in Tables III and IV. Moreover, no negative values for estimated lipids concentrations (x_5) are obtained anymore.

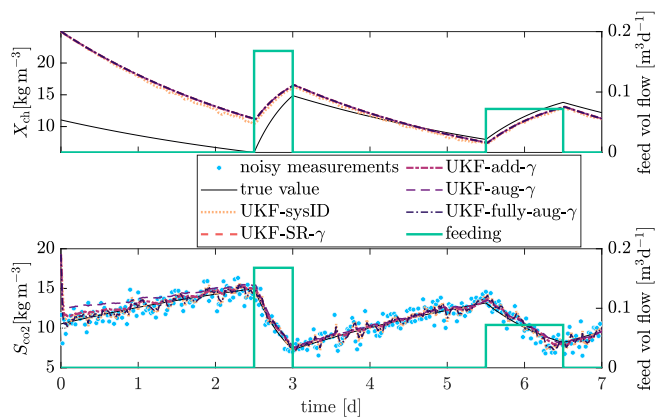


Fig. 2. Comparison of state estimation quality through different UKFs with modified sigma point scaling (reduced scaling factor $\gamma = 1$). For the benchmark UKF-sysID, conventional tuning was retained for comparison. Top: Concentration of carbohydrates and corresponding estimations. Bottom: Concentration of dissolved CO₂, measurements and corresponding estimations.

For the augmented versions, the positive effect of reducing γ is not as clear: NRMSE_y reduces for UKF-aug- γ , whereas NRMSE_x slightly increases. For the fully augmented version both NRMSE values remain almost unchanged at an acceptable level. We conclude that for the given model and simulation scenario, sigma point scaling according to [15] does not necessarily deliver the best possible estimation performance. This especially holds for the additive noise case.

B. Constrained Case

The algorithms presented so far did not explicitly account for state constraints. [8] suggested to introduce clipping in various locations of the unconstrained UKFs to address state estimates beyond physically meaningful bounds. However, clipping diminished the estimation quality in our case (results not shown). This behavior appears to be reasonable: abrupt clipping without simultaneously adapting the sigma point distribution distorts the unscented transformation. A remedy may lie in applying the truncation method [23] proposed for linear Kalman filters and extending it for UKFs. This was, however, not further pursued here.

By contrast, accounting for state constraints through solving the optimization problem (11a) could be shown to improve estimation performance especially for additive noise, which is explained in the following.

The constrained UKFs were implemented for all three noise cases, delivering additive (cUKF-add), augmented (cUKF-aug) and fully augmented cUKFs (cUKF-fully-aug). Note that nominal sigma point scaling (4) was applied. Furthermore, the optimization of each cUKF could be described by either the NLP (12) or the QP formulation (14) since the model output (20) is linear [8]. For the NLP, gradients and the Hessian were either approximated by finite differences (cUKF-NLP); by providing analytic expressions for the gradients (cUKF-NLP-grad); or both gradients and Hessian (cUKF-NLP-grad-hess). All setups of the optimization problems for a given noise case delivered the same estimations. Therefore, Figure 3 shows the cUKF performances only

TABLE V
COMPARISON OF CONSTRAINED UKF PERFORMANCE. RUN TIMES APPLY FOR NLP FORMULATION WITHOUT GRADIENTS AND HESSIAN.

Algorithm	NRMSE _x ^a	NRMSE _y ^b	Run time [s]
cUKF-add	0.2897	0.1040	54.17
cUKF-aug	0.6746	0.0902	129.57
cUKF-fully-aug	0.6345	0.0843	158.06

^aaverage value of all non-measurable states ^baverage value of all measurable states

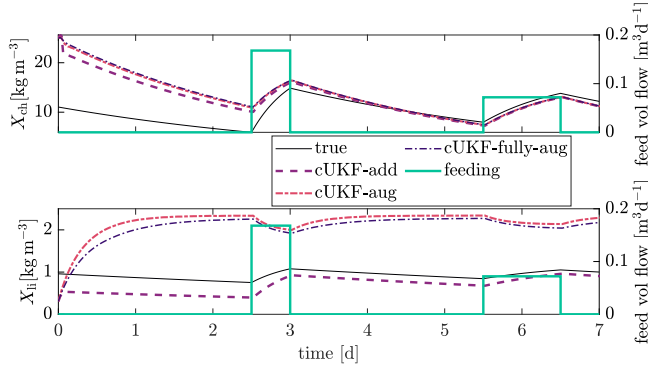


Fig. 3. Comparison of state estimation quality through different constrained UKFs. Top: Concentration of carbohydrates and corresponding estimations. Bottom: Concentration of lipids and corresponding estimations.

with respect to a given noise case. Furthermore, Table V summarizes corresponding NRMSE values. The stated run times apply for the NLP setup with finite differences.

The measurable states are smoothed comparably well as for the unconstrained UKFs, mirrored by NRMSE_y values in the same range as for nominal and reduced sigma point scaling, cf. Tables III, IV and V. For this reason, the graphs of CO₂ measurements and smoothed estimates are not shown here again.

Instead, Figure 3 illustrates estimation performance for the (non-measurable) carbohydrates and lipids concentrations X_{ch} and X_{li} . The latter exhibited negative values for all additive, unconstrained UKFs with nominal values of γ , see Section V-A1. It is clear that for carbohydrates, estimates of all three algorithms do not differ much, and they all approach the true trajectory well. This is reflected in very similar values of NRMSE_x for carbohydrates between 0.44 and 0.54.

By contrast, the lipids estimations differ significantly, Figure 3 bottom. This might, on the one hand, be attributed to the lower order of magnitude of X_{li} . [9] mentioned that for estimates close to the constraints (i.e. very low concentrations in our case), the optimization might result in a biased estimate. On the other hand, closely inspecting the algorithms of cUKF-aug and cUKF-fully-aug reveals a crucial aspect not addressed in [8]. Therein the authors adopt the NLP from [9] and apply it to the augmented cases, while in [9] it was formulated for the additive noise case. To this end, it remains unclear how cUKF-aug and cUKF-fully-aug effectively differ from each other aside from the augmentation, since the intermediate steps for computing the estimated output \hat{y} and the corresponding covariance matrix $P_{y_k y_k}$ do not come into effect in the constrained case. Instead, the outputs resulting from the updated sigma points $h(\chi_{k,i}^x)$

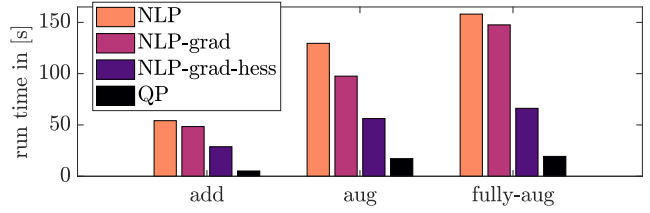


Fig. 4. Run times of cUKF versions with all three noise cases in different optimization setups.

are computed in each iteration of the optimization, (12). Additionally, [8] apply the nominal output $h(x_k)$ in the cost function of the fully augmented noise case, neglecting the non-additive formulation $h(x_k, v_k)$ used in the unconstrained UKFs.

We thus conclude that the NLP formulation (12) might only deliver reliable state estimates for the additive noise case (cUKF-add) for which it was originally proposed, especially for state estimates close to the bounds. This may also explain why the graphs of cUKF-aug and cUKF-fully-aug both show similarly poor lipids estimates in Figure 3. The poor estimates of X_{li} through cUKF-aug and cUKF-fully-aug also dominate the comparably high values of NRMSE_x in Table V.

Estimation results of all optimization setups for a given noise case are identical. This emphasizes that for linear output equations the QP reformulation of the NLP by [8] is indeed equivalent, and that the provision of gradients and Hessian increases numerical efficiency without jeopardizing accuracy.

Run times of the constrained UKFs are generally higher than for the unconstrained versions. However, they can be vastly reduced through a) analytic expressions of gradient and Hessian for the NLP, and b) through the QP formulation in case of linear outputs, as emphasized in Figure 4. Given the higher resulting system order caused by augmentation, it is plausible that numerical effort for cUKF-aug and cUKF-fully-aug is higher than for cUKF-add. In case run time is critical, a UKF formulation specifically designed for real-time applications was recently proposed by [4] and might present an alternative to the UKF designs discussed in this work.

C. Summary

Figure 5 compares the best versions of the different classes of implemented algorithms with respect to run time and estimation accuracy, expressed as the average NRMSE over both measurable and non-measurable states. The unconstrained UKFs showed faster run times than the constrained UKFs. The best accuracy was achieved with cUKF-add. However, the unconstrained UKFs UKF-aug- γ as well as the equivalent UKF-add- γ and UKF-SR- γ deliver almost the same accuracy with lower run times. Among the constrained UKFs, the augmented cUKF versions cUKF-aug and cUKF-fully-aug could not compete with the additive variant cUKF-add with respect to both run time and accuracy.

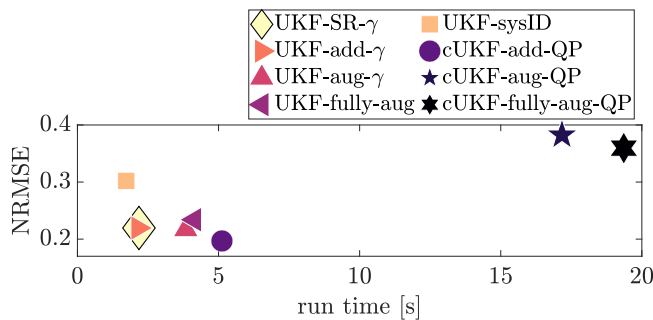


Fig. 5. Comparison of best-in-class implementations of UKFs by means of run time and estimation error.

VI. CONCLUSION

This study examined various UKF implementations and applied them to a simplified AD model. Crucial details of the underlying algorithms as well as modifications to leverage improved numerical performance were addressed. This provides useful hints for practitioners working with Kalman filters.

The best estimation accuracy was achieved with cUKF-add. The QP reformulation of the underlying NLP massively reduced the run time without compromising estimation accuracy. However, it is only applicable for models with linear output equations. For models involving nonlinear output equations, cUKF-NLP-add-grad-hess presents a useful alternative although the associated run time is much higher, see Figure 4.

If run time is critical, the unconstrained variants UKF-SR- γ , UKF-add- γ and UKF-aug- γ with reduced sigma point scaling represent competitive alternatives. The most convenient implementations might be the augmented unconstrained UKF-aug and UKF-fully-aug, which deliver acceptable estimations at low run times, even for nominal sigma point scaling.

As a last note, the present study was limited to simulation data and a reduced AD model with linear outputs. In future studies, the presented UKFs must therefore be applied to real measurement data and higher-order AD models involving nonlinear output equations such as those proposed by [10] or [24].

REFERENCES

- [1] S. Theuerl, C. Herrmann, M. Heiermann, P. Grundmann, N. Landwehr, U. Kreidenweis, and A. Prochnow, "The future agricultural biogas plant in germany: A vision," *Energies*, vol. 12, no. 3, p. 396, 2019.
- [2] P. Kazemi, J.-P. Steyer, C. Bengoa, J. Font, and J. Giralt, "Robust data-driven soft sensors for online monitoring of volatile fatty acids in anaerobic digestion processes," *Processes*, vol. 8, no. 1, p. 67, 2020.
- [3] M. Gentsch and R. King, "Real-time estimation of parameter maps," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 2391–2396, 2020.
- [4] T. Cantelobre, C. Chahbazian, A. Croux, and S. Bonnabel, "A real-time unscented kalman filter on manifolds for challenging auv navigation," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 2309–2316.
- [5] N. Raeyatdoost, M. Bongards, T. Bäck, and C. Wolf, "Robust state estimation of the anaerobic digestion process for municipal organic waste using an unscented kalman filter," *Journal of Process Control*, vol. 121, no. 1, pp. 50–59, 2023.

- [6] A. Kemmer, N. Fischer, T. Wilms, L. Cai, S. Groß, R. King, P. Neubauer, and M. N. Cruz Bournazou, "Nonlinear state estimation as tool for online monitoring and adaptive feed in high throughput cultivations," *Biotechnology and Bioengineering*, vol. 120, no. 11, pp. 3261–3275, 2023.
- [7] A. Tuveri, F. Pérez-García, P. A. Lira-Parada, L. Imsland, and N. Bar, "Sensor fusion based on extended and unscented kalman filter for bioprocess monitoring," *Journal of Process Control*, vol. 106, pp. 195–207, 2021.
- [8] S. Kolås, B. A. Foss, and T. S. Schei, "Constrained nonlinear state estimation based on the ukf approach," *Computers & Chemical Engineering*, vol. 33, no. 8, pp. 1386–1401, 2009.
- [9] P. Vachhani, S. Narasimhan, and R. Rengaswamy, "Robust and reliable estimation via unscented recursive nonlinear dynamic data reconciliation," *Journal of Process Control*, vol. 16, no. 10, pp. 1075–1086, 2006.
- [10] S. Weinrich and M. Nelles, "Systematic simplification of the anaerobic digestion model no. 1 (ADM1) - model development and stoichiometric analysis: Application," *Bioresource technology*, vol. 333, p. 125124, 2021.
- [11] D. J. Batstone, J. Keller, I. Angelidaki, S. V. Kalyuzhnyi, S. G. Pavlostathis, A. Rozzi, W. Sanders, H. Siegrist, and V. A. Vavilin, "The IWA Anaerobic Digestion Model No 1 (ADM1)," *Water Science and Technology*, vol. 45, no. 10, pp. 65–73, 2002.
- [12] S. Weinrich, E. Mauky, T. Schmidt, C. Krebs, J. Liebetrau, and M. Nelles, "Systematic simplification of the anaerobic digestion model no. 1 (ADM1) - laboratory experiments and model application," *Bioresource technology*, vol. 333, p. 125104, 2021.
- [13] E. Mauky, S. Weinrich, H.-J. Nägele, H. F. Jacobi, J. Liebetrau, and M. Nelles, "Model predictive control for demand-driven biogas production in full scale," *Chemical Engineering & Technology*, vol. 39, no. 4, pp. 652–664, 2016.
- [14] S. Hellmann, A.-J. Hempel, S. Streif, and S. Weinrich, "Observability and identifiability analyses of process models for agricultural anaerobic digestion plants," in *2023 24th International Conference on Process Control (PC)*. IEEE, 2023, pp. 84–89.
- [15] R. van der Merwe, "Sigma-point kalman filters for probabilistic inference in dynamic state-space models," Dissertation, Oregon Health & Science University, Portland, Oregon, USA, 2004.
- [16] The MathWorks, Inc., "Matlab version 9.13.0 (r2022b)," Natick, MA, USA, 2022.
- [17] F. de Vivo, A. Brandl, M. Battipede, and P. Gili, "Joseph covariance formula adaptation to square-root sigma-point kalman filters," *Nonlinear Dynamics*, vol. 88, no. 3, pp. 1969–1986, 2017.
- [18] The MathWorks, Inc., "Writing scalar objective functions," Matlab documentation: <https://www.mathworks.com/help/optim/ug/writing-scalar-objective-functions.html>, last visited 10/24, 2023.
- [19] S. Hellmann, T. Wilms, S. Streif, and S. Weinrich, "Comparative review of unscented kalman filter design for agricultural anaerobic digestion model," *arxiv preprint*, vol. arXiv:2310.15958, 2023.
- [20] R. Schneider and C. Georgakis, "How to not make the extended kalman filter fail," *Industrial & Engineering Chemistry Research*, vol. 52, no. 9, pp. 3354–3362, 2013.
- [21] R. van der Merwe and E. Wan, "The square-root unscented kalman filter for state and parameter-estimation," in *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.01CH37221)*. IEEE, 2001, pp. 3461–3464.
- [22] The MathWorks, Inc., "Extended and unscented kalman filter algorithms for online state estimation," Matlab documentation: <https://www.mathworks.com/help/control/ug/extended-and-unscented-kalman-filter-algorithms-for-online-state-estimation.html#bv4mkl>, last visited 10/24, 2023.
- [23] D. Simon and D. L. Simon, "Constrained kalman filtering via density function truncation for turbofan engine health estimation," *International Journal of Systems Science*, vol. 41, no. 2, pp. 159–171, 2010.
- [24] O. Bernard, Z. Hadj-Sadok, D. Dochain, A. Genovesi, and J. P. Steyer, "Dynamical model development and parameter identification for an anaerobic wastewater treatment process," *Biotechnology and Bioengineering*, vol. 75, no. 4, pp. 424–438, 2001.