Gennady Yu. Kulikov* and Maria V. Kulikova

The continuous–discrete extended Kalman filter revisited

Abstract: This paper elaborates a new approach to nonlinear filtering grounded in an accurate implementation of the continuous–discrete extended Kalman filter for estimating stochastic dynamic systems. It implies that the moment differential equations for calculation of the predicted state mean and error covariance of propagated Gaussian density are solved accurately, i.e., with negligible errors. The latter allows the total error of the extended Kalman filter to be reduced significantly and results in a new accurate continuous–discrete extended Kalman filtering method. In addition, this filter exploits the scaled local and global error controls to avoid any comparison of different physical units. The designed state estimator is compared numerically with continuous–discrete unscented and cubature Kalman filters to expose its practical efficiency. The problem of long waiting times (i.e., infrequent measurements) arisen in chemical and other engineering is also addressed.

Keywords: Stochastic continuous–discrete system, continuous–discrete unscented Kalman filter, continuous–discrete cubature Kalman filter, accurate continuous–discrete extended Kalman filter, adaptive MDE solver with scaled local and global error controls.

MSC 2010: 65L05, 93E11

DOI: 10.1515/rnam-2017-0003
Received May 19, 2016; accepted November 30, 2016

1 Introduction

Mathematical models in many areas of study as diverse as target tracking, navigation, stochastic control, chemistry and finance (see, for example, [3, 4, 7–9, 11, 12, 25–27, 29, 32]) are written in the form of Itô-type *stochastic differential equation* (SDE)

$$\mathrm{d}x(t) = F\big(x(t), u(t)\big)\mathrm{d}t + G(t)\mathrm{d}w(t), \qquad t > 0 \tag{1.1}$$

where $x(t) \in \mathbb{R}^{n_1}$ is the n_1-dimensional vector of system state at time t, $u(t) \in \mathbb{R}^{n_2}$ is an optional measurable input (this also includes possible control inputs) at time t, $F : \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \to \mathbb{R}^{n_1}$ is a nonlinear sufficiently smooth function representing the dynamic behaviour of the model, $G(t)$ is a time-variant matrix of size $n_1 \times q$ and $\{w(t),\ t > 0\}$ is a Brownian process with square diffusion matrix $Q(t) > 0$ of size q. The initial state x_0 of stochastic process (1.1) is a random variable. More precisely, $x_0 \sim \mathcal{N}(\bar{x}_0, \Pi_0)$ with $\Pi_0 > 0$, where the notation $\mathcal{N}(\bar{x}_0, \Pi_0)$ stands for the normal distribution with mean $\bar{x}_0$ and covariance Π_0.

The task of least-square state estimation in model (1.1) always compounds real measurements of some model variables or their function (depending on the utilized technology) with computation of remaining (not measurable) parameters by a nonlinear filter. It is usually assumed that the observation information arrives discretely and in equidistant intervals of size $\delta = t_k - t_{k-1}$. This time interval δ is called the *sampling period* (or *waiting time*) in filtering theory. The relation of the observation z_k to the state vector x_k of SDE (1.1) obeys the *measurement equation*

$$z_k = h(x_k) + v_k, \qquad k \geqslant 1 \tag{1.2}$$

where k stands for a discrete time index (i.e., x_k means $x(t_k)$), $z_k \in \mathbb{R}^m$ is the information available at time instant t_k, $h : \mathbb{R}^{n_1} \to \mathbb{R}^m$ is a differentiable function and the measurement noise $v_k \sim \mathcal{N}(0, R_k)$ with $R_k > 0$.

***Corresponding Author: Gennady Yu. Kulikov:** CEMAT, Instituto Superior Técnico, Universidade de Lisboa, Av. Rovisco Pais, 1049-001 Lisboa, Portugal. E-mail: gkulikov@math.ist.utl.pt
Maria V. Kulikova: CEMAT, Instituto Superior Técnico, Universidade de Lisboa, Av. Rovisco Pais, 1049-001 Lisboa, Portugal

All realizations of the process noise $w(t)$, the measurement noise v_k and the initial state x_0 are assumed to be taken from mutually independent Gaussian distributions.

Historically, the first state estimator for the continuous–discrete stochastic state-space system (1.1)–(1.2) was built in the *extended Kalman filtering* (EKF) framework and grounded in the stochastic Euler discretization method, as presented, for instance, in [11]. More formally, the Euler–Maruyama scheme is applied to the SDE (1.1) on a time interval $[t_{k-1}, t_k]$ to derive the discrete-time system

$$x_k = x_{k-1} + \delta F\big(x_{k-1}, u(t_{k-1})\big) + G(t_{k-1})\tilde{w}_{k-1} \tag{1.3}$$

where $\delta = t_k - t_{k-1}$ and $\tilde{w}_{k-1} \sim \mathcal{N}(0, \delta Q(t_{k-1}))$. Then, it follows from (1.3) that taking the expectation yields

$$\mathbf{E}\{x_k\} = \mathbf{E}\{x_{k-1}\} + \delta\, \mathbf{E}\big\{F\big(x_{k-1}, u(t_{k-1})\big)\big\} \tag{1.4}$$

with $\mathbf{E}\{x(t_k)\} := \hat{x}(t_k)$ and $\mathbf{E}\{x(t_{k-1})\} := \hat{x}(t_{k-1})$. The state vector x_{k-1} is independent of the noise $\tilde{w}_{k-1}$. Therefore the associated covariance is determined by the formula

$$\mathbf{var}\{x_k\} = \mathbf{var}\big\{x_{k-1} + \delta F\big(x_{k-1}, u(t_{k-1})\big)\big\} + \delta G(t_{k-1})Q(t_{k-1})G^T(t_{k-1}). \tag{1.5}$$

Further, the EKF method implies that the moment equations (1.4) and (1.5) are solved approximately on each sampling interval $[t_{k-1}, t_k]$ by means of the first-order Taylor expansion of the nonlinear drift function $F\big(x(t), u(t)\big)$ around the filtering state estimate $\hat{x}_{k-1|k-1}$ at the time t_{k-1}. Substituting the mentioned expansion into formulas (1.4) and (1.5) yields the *time-update* step of the classical EKF method:

$$\begin{aligned} \hat{x}_{k|k-1} &= \hat{x}_{k-1|k-1} + \delta F\big(\hat{x}_{k-1|k-1}, u(t_{k-1})\big) && (1.6)\\ P_{k|k-1} &= \big[I_{n_1} + \delta \partial_x F\big(\hat{x}_{k-1|k-1}, u(t_{k-1})\big)\big]\, P_{k-1|k-1} \\ &\quad \times \big[I_{n_1} + \delta \partial_x F\big(\hat{x}_{k-1|k-1}, u(t_{k-1})\big)\big]^T + \delta G(t_{k-1})Q(t_{k-1})G^T(t_{k-1}) && (1.7) \end{aligned}$$

where I_{n_1} stands for the identity matrix of size n_1, and

$$\partial_x F\big(\hat{x}_{k-1|k-1}, u(t_{k-1})\big) := \partial F\big(\hat{x}_{k-1|k-1}, u(t_{k-1})\big)/\partial x(t) \tag{1.8}$$

means the partial derivative (Jacobian) of the nonlinear function $F\big(x(t), u(t)\big)$ with respect to $x(t)$ evaluated at $\big(\hat{x}_{k-1|k-1}, u(t_{k-1})\big)$.

The *measurement-update* step of this EKF is performed in its usual form, i.e. after arrival of a new measurement information z_k, one calculates the following:

$$R_{e,k} = R_k + H_k P_{k|k-1} H_k^T, \qquad K_k = P_{k|k-1} H_k^T R_{e,k}^{-1} \tag{1.9}$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k e_k, \qquad e_k = z_k - h(\hat{x}_{k|k-1}), \qquad P_{k|k} = P_{k|k-1} - K_k H_k P_{k|k-1} \tag{1.10}$$

where the Jacobian $H_k := \mathrm{d}h(\hat{x}_{k|k-1})/\mathrm{d}x_k$ is evaluated at the predicted state mean $\hat{x}_{k|k-1}$ from formula (1.6) and $e_k \sim \mathcal{N}(0, R_{e,k})$ are *innovations* of the Kalman filter. Eventually, the *linear least-square estimate* $\hat{x}_{k|k}$ of the system state $x(t_k)$ based on measurements $\{z_1, \dots, z_k\}$ is determined. The EKF variant (1.6)–(1.10) is the simplest but successful state estimator that has been utilized by practitioners for decades (see further details in [6, 7, 11, 23, 28]).

Despite EKF popularity, this method has been criticized on its performance for radar tracking [1, 2] and for state estimation in chemical engineering [9, 12, 26, 27, 29, 32]. More precisely, Arasaratnam *et al.* [2] exhibit that the EKF does not work properly in the air traffic control scenario considered in the cited paper and loses the recently designed *cubature Kalman filter* (CKF) [1] and *unscented Kalman filter* (UKF) [13, 14]. Haseltine and Rawlings [9] report that their EKF fails for two types of chemical reactors meaning that wrong steady-states are calculated and negative concentrations are observed after convergence, which are of no physical sense. Jørgensen [12] claims that his EKF is not able to reconstruct offset free concentrations in the Van der Vusse reaction scenario on the basis of temperature measurements, only. It is also found out that the EKF may fail for nonlinear systems with infrequent observations [29].

All this may be true, but only partially. The notion of EKF does not imply the unique technique, as presented above, and refers to a class of various methods with different properties. For instance, Frogerais et al. [5] consider and examine five EKF implementations on two nonlinear test problems. They elaborate at least two approaches for constructing EKF algorithms, which can lead to a great variety of state estimators grounded in ordinary or stochastic differential equation numerical schemes. Therefore the above-cited criticism might not mean that the EKF technology fails itself, and says only that an unfortunate version of this method has been used. Further, we discuss a state estimator which resolves many problems associated with EKF in literature. It is also shown that our EKF can even outperform the recently designed CKF and UKF methods in chemical engineering models.

Following Frogerais *et al.* [5], we identify two principal ways of EKF implementation, namely, the *continuous–discrete* and *discrete–discrete* EKFs. The first approach implies that the discretization operation is performed in a deterministic setting, whereas the second one does it in a stochastic setting. This difference is crucial because it is explained in [17] that the deterministic discretization error can be easily regulated in automatic mode and made negligible in contrast to the stochastic discretization error arising in the discrete-discrete approach. That is why we consider that the *continuous–discrete EKF* (CD-EKF) is more accurate and, hence, preferable for practical use. In addition, it works for stochastic models with sparse measurements [17] and resolves the inconsistency of the demanded sampling frequency and practically available measurements mentioned in [29].

The CD-EKF designed in [17] is flexible and accurate, but time-consuming. It is based on the embedded Runge–Kutta pair NIRK4(2) with global error control from [16], which is applied for a simultaneous solution of the predicted state mean and error covariance differential equations. That solver controls the absolute local and global errors, which, in turn, imply that different physical units are compared. The latter is not supported by practitioners and must be amended to undimensional (scaled) local and global error controls. Additionally, the cited CD-EKF does not ensure the positive semi-definiteness of the error covariance matrix, but this may be crucial for a proper performance of the EKF technique.

The listed drawbacks of the published CD-EKF are resolved by a separate treatment of the moment differential equations. More precisely, the predicted state expectation equation (which is the sole nonlinear equation in this system) is treated accurately by means of the NIRK4(2) (or NIRK6(4)) method designed in [16]. However, the absolute local and global error controls implemented there are replaced with the scaled local and global error controls developed in [21, 31]. The predicted covariance differential equation is integrated numerically by the corresponding part of Mazzoni's scheme [24], which ensures the positive semi-definiteness of computed covariance in exact arithmetic. In other words, we build the hybrid triples (denoted further as NIRK4(2)M2 and NIRK6(4)M2) with the scaled local and global error control mechanisms to solve accurately the moment differential equations arising in the framework of the CD-EKF and, then, design efficient state estimators for practical use. Our new methods resolve many problems associated with EKF and, hence, can be widely applied in practice. They can even outperform modern state estimators based on the CKF [1] and on the UKF [13, 14]. Below, we present the new CD-EKF methods grounded in the NIRK4(2)M2 and NIRK6(4)M2 MDE solvers at large and outline briefly other state estimators also examined in this paper.

2 Accurate continuous–discrete extended Kalman filter

The CD-EKF is based on replacement of values of the state mean and error covariance matrix predicted by formulas (1.6), (1.7) with values satisfying the *moment differential equations* (MDEs)

$$\hat{x}'(t) = F(\hat{x}(t), u(t)) \tag{2.1}$$

$$P'(t) = \partial_x F(\hat{x}(t), u(t))P(t) + P(t)\partial_x F^T(\hat{x}(t), u(t)) + G(t)Q(t)G^T(t) \tag{2.2}$$

at sampling times [11, 23]. Here, $\partial_x F(\hat{x}(t), u(t))$ stands for the Jacobian of the function $F(\hat{x}(t), u(t))$ defined by (1.8), $G(t)$ is the matrix from the stochastic noise term of SDE (1.1), $Q(t)$ is the covariance matrix of the zero-mean Gaussian white-noise process $w(t)$, $\hat{x}(t)$ means the state expectation of the system state vector $x(t)$

at time t (i.e., $x(t)$ is a solution to SDE (1.1)), and $u(t)$ is the known input (i.e., a known function of t). The matrix $P(t)$ in formula (2.2) has the physical meaning of being the variance of the state prediction error, i.e., $x(t) - \hat{x}(t)$, and, then, has to be positive semi-definite.

One utilizes values of the filtering state mean and error covariance matrix at the sampling time t_{k-1} as initial values of MDEs (2.1)–(2.2) set in the time interval $[t_{k-1}, t_k]$, i.e., $\hat{x}(t_{k-1}) = \hat{x}_{k-1|k-1}$, $P(t_{k-1}) = P_{k-1|k-1}$. Having solved the derived initial value problem in this sampling interval one yields predicted values of the state mean and error covariance matrix at the next sampling time t_k as follows: $\hat{x}_{k|k-1} = \hat{x}(t_k)$, $P_{k|k-1} = P(t_k)$. Then, after arrival of a new measurement z_k, the standard measurement-update formulas (1.9)–(1.10) are applied to determine the linear least-square estimate $\hat{x}_{k|k}$ of the system state $x(t_k)$ at the time t_k.

The CD-EKF is rather efficient, but still has some drawback. The predicted state mean and error covariance matrix of this filter are the exact solutions to the MDEs (2.1)–(2.2) formulated in each sampling interval. Unfortunately, these exact solutions are hardly available in practice because of nonlinearity of equation (2.1). So one has to apply a numerical (discretization) method for calculating approximations to the requested values. Certainly, some discretization error is always introduced in this step of the algorithm.

When one solves the initial value problem (2.1)–(2.2) on a prefixed mesh (for one step or for several steps) the method will compute the numerical solution with unpredictable error. It may be small or large. This depends on the MDEs (2.1)–(2.2), on the predefined mesh and on the size δ of the interval $[t_{k-1}, t_k]$. The main difficulty is that it is not possible to prefix a mesh which is optimal for all initial value problems arising in practical state estimation tasks. A better solution is to request the method itself to generate an optimal mesh so that a prefixed level of accuracy is achieved automatically. In other words, the user limits only the magnitude of tolerated errors and the solver (i.e., the discretization scheme and the error evaluation and control algorithm) generates a mesh corresponding to the set accuracy condition in automatic mode. Thus, one does not know *a priori* how many steps will be fulfilled, but the error of numerical integration will correspond to the preassigned level. This is a more complicated solution method. However, it increases the accuracy and reliability of state estimation, significantly. It leads to the new concept of *accurate continuous–discrete extended Kalman filtering* (ACD-EKF) [17, 19, 20, 22]. Now we improve the ACD-EKF by implementing the more advanced MDE solvers NIRK4(2)M2 and NIRK6(4)M2 with scaled local and global error controls.

It is further assumed that we have completed the state estimation at the time instant t_{k-1} and calculated the filtering state expectation $\hat{x}_{k-1|k-1}$ and the filtering error covariance matrix $P_{k-1|k-1}$. Our task is to compute a numerical solution to the MDEs (2.1)–(2.2) with the initial values $\hat{x}(t_{k-1}) = \hat{x}_{k-1|k-1}$, $P(t_{k-1}) = P_{k-1|k-1}$ at the next sampling time instant t_k. For that, we suppose first that a subdivision (*mesh*) $\{t_l\}_{l=0}^{L}$ has already been fixed in the integration interval $[t_{k-1}, t_k]$. We point out that this mesh may be variable. Later in Section 2, we explain how to generate it in automatic mode.

The application of the first hybrid triple NIRK4(2)M2 to the state expectation equation (2.1) results in its following discretization:

$$\begin{aligned}
\hat{x}_{l1}^2 &= a_{11}^2\hat{x}_l + a_{12}^2\hat{x}_{l+1} + \tau_l\big[d_{11}^2 F(\hat{x}_l, u_l) + d_{12}^2 F(\hat{x}_{l+1}, u_{l+1})\big]\\
\hat{x}_{l2}^2 &= a_{21}^2\hat{x}_l + a_{22}^2\hat{x}_{l+1} + \tau_l\big[d_{21}^2 F(\hat{x}_l, u_l) + d_{22}^2 F(\hat{x}_{l+1}, u_{l+1})\big]\\
\hat{x}_{l+1} &= \hat{x}_l + \frac{\tau_l}{2}\big[F(\hat{x}_{l1}^2, u_{l1}^2) + F(\hat{x}_{l2}^2, u_{l2}^2)\big]
\end{aligned} \tag{2.3}$$

on the mesh $\{t_l\}_{l=0}^{L}$. Here, $\tau_l := t_{l+1} - t_l$ denotes the variable step size of this mesh, and the constant coefficients of the Gauss-type *nested implicit Runge–Kutta* (NIRK) formula (2.3) of order 4 are:

$$c_1^2 := (3-\sqrt{3})/6, \qquad c_2^2 := (3+\sqrt{3})/6, \qquad a_{11}^2 := 1/2 + 2\sqrt{3}/9$$
$$a_{12}^2 := 1/2 - 2\sqrt{3}/9, \qquad a_{21}^2 := 1/2 - 2\sqrt{3}/9, \qquad a_{22}^2 := 1/2 + 2\sqrt{3}/9$$
$$d_{11}^2 := (3+\sqrt{3})/36, \qquad d_{12}^2 := (\sqrt{3}-3)/36, \qquad d_{21}^2 := (3-\sqrt{3})/36$$
$$d_{22}^2 := -(3+\sqrt{3})/36.$$

The stage values $\hat{x}_{lj}^2$ of method (2.3) imply approximations to the state expectations $\hat{x}(t_{lj}^2)$ at the time points $t_{lj}^2 := t_l + c_j^2\tau_l$, and the measurable input $u_{lj}^2 := u(t_{lj}^2)$, $j = 1, 2$. We recall that $u(t)$ is a known function of time.

As customary, the function $F(\cdot)$ represents the right-hand side of MDE (2.1) or, which is the same, the drift function in SDE (1.1). Eventually, the nonlinear problem (2.3) is to be iterated for an approximate expectation $\hat{x}_{l+1}$ at every node of the mesh $\{t_l\}_{l=0}^{L}$. We do not discuss further its efficient solution method because it is presented in [22, Sec. IIIA] in detail.

The hybrid triple NIRK6(4)M2 is grounded in the Gauss-type NIRK formula of order 6, which, for equation (2.1), reads

$$\begin{aligned}
\hat{x}_{l1}^{2} &= a_{11}^{2}\hat{x}_l + a_{12}^{2}\hat{x}_{l+1} + \tau_l\big[d_{11}^{2}F(\hat{x}_l,u_l) + d_{12}^{2}F(\hat{x}_{l+1},u_{l+1})\big]\\
\hat{x}_{l2}^{2} &= a_{21}^{2}\hat{x}_l + a_{22}^{2}\hat{x}_{l+1} + \tau_l\big[d_{21}^{2}F(\hat{x}_l,u_l) + d_{22}^{2}F(\hat{x}_{l+1},u_{l+1})\big]\\
\hat{x}_{l1}^{3} &= a_{11}^{3}\hat{x}_l + a_{12}^{3}\hat{x}_{l+1} + \tau_l\big[d_{11}^{3}F(\hat{x}_l,u_l) + d_{12}^{3}F(\hat{x}_{l+1},u_{l+1})\\
&\qquad + d_{13}^{3}F(\hat{x}_{l1}^{2},u_{l1}^{2}) + d_{14}^{3}F(\hat{x}_{l2}^{2},u_{l2}^{2})\big]\\
\hat{x}_{l2}^{3} &= a_{21}^{3}\hat{x}_l + a_{22}^{3}\hat{x}_{l+1} + \tau_l\big[d_{21}^{3}F(\hat{x}_l,u_l) + d_{22}^{3}F(\hat{x}_{l+1},u_{l+1})\\
&\qquad + d_{23}^{3}F(\hat{x}_{l1}^{2},u_{l1}^{2}) + d_{24}^{3}F(\hat{x}_{l2}^{2},u_{l2}^{2})\big]\\
\hat{x}_{l3}^{3} &= a_{31}^{3}\hat{x}_l + a_{32}^{3}\hat{x}_{l+1} + \tau_l\big[d_{31}^{3}F(\hat{x}_l,u_l) + d_{32}^{3}F(\hat{x}_{l+1},u_{l+1})\\
&\qquad + d_{33}^{3}F(\hat{x}_{l1}^{2},u_{l1}^{2}) + d_{34}^{3}F(\hat{x}_{l2}^{2},u_{l2}^{2})\big]\\
\hat{x}_{l+1} &= \hat{x}_l + \tau_l\big[b_1F(\hat{x}_{l1}^{3},u_{l1}^{3}) + b_2F(\hat{x}_{l2}^{3},u_{l2}^{3}) + b_3F(\hat{x}_{l3}^{3},u_{l3}^{3})\big]
\end{aligned} \tag{2.4}$$

with $l = 0, 1, \dots, L-1$, where $\tau_l := t_{l+1} - t_l$ is again the variable step size of the mesh $\{t_l\}_{l=0}^{L}$, and the constant coefficients of method (2.4) are:

$$\begin{gathered}
b_1 = b_3 := 5/18, \qquad b_2 := 4/9, \qquad c_1^2 := (3-\sqrt{3})/6, \qquad c_2^2 := (3+\sqrt{3})/6\\
a_{11}^2 = a_{22}^2 := 1/2 + 2\sqrt{3}/9, \qquad a_{12}^2 = a_{21}^2 := 1/2 - 2\sqrt{3}/9, \qquad d_{11}^2 = -d_{22}^2 := (3+\sqrt{3})/36\\
d_{12}^2 = -d_{21}^2 := (-3+\sqrt{3})/36\\
c_1^3 := (5-\sqrt{15})/10, \qquad c_2^3 := 1/2, \qquad c_3^3 := (5+\sqrt{15})/10, \qquad a_{11}^3 = a_{32}^3 := (125+39\sqrt{15})/250\\
a_{12}^3 = a_{31}^3 := (125-39\sqrt{15})/250, \qquad a_{21}^3 = a_{22}^3 := 1/2, \qquad d_{11}^3 = -d_{32}^3 := (7+2\sqrt{15})/200\\
d_{12}^3 = -d_{31}^3 := (-7+2\sqrt{15})/200, \qquad d_{13}^3 = -d_{34}^3 := (18\sqrt{15}+15\sqrt{3})/1000\\
d_{14}^3 = -d_{33}^3 := (18\sqrt{15}-15\sqrt{3})/1000, \qquad d_{21}^3 = -d_{22}^3 := 1/32, \qquad d_{23}^3 = -d_{24}^3 := 3\sqrt{3}/32.
\end{gathered}$$

The nonlinear problem (2.4) is iterated for an approximate state mean $\hat{x}_{l+1}$ by using the simplified Newton scheme presented in [20, Sec. 2.3] at large.

Both NIRK4(2)M2 and NIRK6(4)M2 methods solve equation (2.2) by means of the *modified implicit midpoint rule* [24], which reads

$$P_{l+1} = M_{l+1/2}P_lM_{l+1/2}^T + \tau_lK_{l+1/2}G(t_{l+1/2})Q(t_{l+1/2})G^T(t_{l+1/2})K_{l+1/2}^T \tag{2.5}$$

where $t_{l+1/2} := t_l + \tau_l/2$, and the variable matrices $K_{l+1/2}$ and $M_{l+1/2}$ evaluated at the mid-point $t_{l+1/2}$ are:

$$\begin{aligned}
K_{l+1/2} &:= \Big[I_{n_1} - \frac{\tau_l}{2}\partial_xF(\hat{x}_{l+1/2},u_{l+1/2})\Big]^{-1}\\
M_{l+1/2} &:= K_{l+1/2}\Big[I_{n_1} + \frac{\tau_l}{2}\partial_xF(\hat{x}_{l+1/2},u_{l+1/2})\Big].
\end{aligned} \tag{2.6}$$

However, there is some difference in using formulas (2.5), (2.6) in NIRK4(2)M2 and NIRK6(4)M2. The reason is that the mid-point state mean $\hat{x}_{l+1/2}$ is not available in the NIRK method (2.3) and, hence, one has to apply Mazzoni's interpolation [24]:

$$\hat{x}_{l+1/2} = \frac{1}{2}\left[\hat{x}_l^4 + \hat{x}_{l+1}^4 - \frac{\tau_l^2}{4}\partial_xF(\hat{x}_l^4,u_l)F(\hat{x}_l^4,u_l)\right]. \tag{2.7}$$

Note that formula (2.7) is not expensive because $\hat{x}_l^4$, $\hat{x}_{l+1}^4$, $F(\hat{x}_l^4,u_l)$, and $\partial_xF(\hat{x}_l^4,u_l)$ have already been computed when solving the nonlinear problem (2.3) (see [22, Sec. IIIA]). Alternatively, NIRK6(4)M2 needs no interpolation for finding the mid-point state expectation since $\hat{x}_{l+1/2} = \hat{x}_{l2}^{3,4}$, where $\hat{x}_{l2}^{3,4}$ is the stage value iterate explained in [20, Sec. 2.3]. We stress that formula (2.5) ensures the positive semi-definiteness of derived error covariance P_{l+1} in exact arithmetic.

Next, we consider an automatic algorithm for generating the mesh $\{t_l\}_{l=0}^{L}$. This mesh is generated so that the scaled global error does not exceed the user-supplied tolerance ε_g at all mesh nodes. Therefore we have to evaluate first scaled values of the local and global errors in our MDE solvers NIRK4(2)M2 and NIRK6(4)M2.

The *local error* le_{l+1} is calculated in the first triple by the formula

$$le_{l+1} = \frac{\tau_l}{2}\Big[F(\hat{x}_{l1}^{2,4}, u_{l1}^2) + F(\hat{x}_{l2}^{2,4}, u_{l2}^2) - F(\hat{x}_l^4, u_l) - F(\hat{x}_{l+1}^4, u_{l+1})\Big] \tag{2.8}$$

where $\hat{x}_l^4$, $\hat{x}_{l+1}^4$ mean approximations to the solution from the NIRK formula (2.3) and $\hat{x}_{l1}^{2,4}$, $\hat{x}_{l2}^{2,4}$ are its stage values computed as explained in [22, Sec. IIIA]. Then, the local error (2.8) is measured as follows:

$$|le_{l+1}|_{\mathrm{sc}} := \max_{i=1,2,\dots,n_1} \{|(le_{l+1})_i|/(|(\hat{x}_{l+1}^4)_i| + 1)\} \tag{2.9}$$

where the subscript i stands for the i-th entry in the corresponding vector and n_1 is the size of MDE (2.1). The error defined by formula (2.9) is referred to as the *scaled local error* at the mesh node t_{l+1}. The *global error* $\Delta\hat{x}_{l+1}$ at t_{l+1} is evaluated by the formula

$$\Delta\hat{x}_{l+1} = \Delta\hat{x}_l + le_{l+1} \tag{2.10}$$

where the initial integration error $\Delta\hat{x}_0$ is set to be zero [16, 17]. Again, the global error (2.10) is scaled as follows:

$$|\Delta\hat{x}_{l+1}|_{\mathrm{sc}} := \max_{i=1,2,\dots,n_1} \{|(\Delta\hat{x}_{l+1})_i|/(|(\hat{x}_{l+1}^4)_i| + 1)\} \tag{2.11}$$

where the subscript i means the i-th entry in the corresponding vector. Thus, the magnitude $|\Delta\hat{x}_{l+1}|_{\mathrm{sc}}$ is referred to as the *scaled global error* at the mesh node t_{l+1}.

Similarly, within NIRK6(4)M2, the local error is calculated by

$$le_{l+1} = \frac{\tau_l}{3}\left[\frac{5}{6}F(\hat{x}_{l1}^{3,4}, u_{l1}^3) - \frac{1}{2}F(\hat{x}_l^4, u_l) - \frac{2}{3}F(\hat{x}_{l2}^{3,4}, u_{l2}^3) - \frac{1}{2}F(\hat{x}_{l+1}^4, u_{l+1}) + \frac{5}{6}F(\hat{x}_{l3}^{3,4}, u_{l3}^3)\right]$$

in which the predicted state means $\hat{x}_l^4$, $\hat{x}_{l+1}^4$ and the stage values $\hat{x}_{l1}^{3,4}$, $\hat{x}_{l2}^{3,4}$, $\hat{x}_{l3}^{3,4}$ come from the iteration discussed in [20, Sec. 2.3]. Formulas (2.9)–(2.11) are further applied for computing scaled values of the local and global errors in NIRK6(4)M2.

Having evaluated the mentioned errors we utilize Algorithm 3.2 in [16], but with some changes and implemented for the scaled error estimates (2.9) and (2.11), to generate the mesh $\{t_l\}_{l=0}^{L}$ in the interval $[t_{k-1}, t_k]$ in automatic mode. Now we suppose that the filtering state mean $\hat{x}_{k-1|k-1}$ and the filtering error covariance matrix $P_{k-1|k-1}$ have been computed at the time instant t_{k-1}. Then, their estimation is fulfilled within the ACD-EKF method at the next sampling instant t_k as follows.

Time Update: Set $\varepsilon_{\mathrm{loc}} := \varepsilon_g^{(s-1)/(s-2)}$, $\tau_0 := \min\{0.01, \delta\}$, $\tau_{\max} := 0.1$, $M := 1$, and perform

```
1. While $M = 1$ do;
2.     $l := 0,\ M := 0,\ t_0 := t_{k-1},\ \hat{x}_0^4 := \hat{x}_{k-1|k-1},\ \Delta\hat{x}_0 := 0,\ P_0 := P_{k-1|k-1}$;
3.     While $(t_l < t_k)$ & $(|\Delta\hat{x}_l|_{\mathrm{sc}} \leqslant 10\varepsilon_g)$ do;
4.         $t_{l+1} := t_l + \tau_l$, compute $\hat{x}_{l+1}^4$ and $|le_{l+1}|_{\mathrm{sc}}$;
5.         $\tau_l^* := \min\left\{1.5, 0.8(\varepsilon_{\mathrm{loc}}/|le_{l+1}|_{\mathrm{sc}})^{1/(s-1)}\right\}\tau_l$;
6.         If $|le_{l+1}|_{\mathrm{sc}} > \varepsilon_{\mathrm{loc}}$,
             then $\tau_l := \tau_l^*$;
             else do;
7.             Evaluate $|\Delta\hat{x}_{l+1}|_{\mathrm{sc}}$;
8.             If $|\Delta\hat{x}_{l+1}|_{\mathrm{sc}} > \varepsilon_g$,
                 then $M := 1$;
9.             If $M = 0$,
                    then compute the matrix $P_{l+1}$;
10.                 $\tau_{l+1} := \min\{\tau_l^*, t_k - t_{l+1}, \tau_{\max}\}$;
11.                 $l := l + 1$;
                end else;
        end while;
12.     If $M = 1$,
            then $\varepsilon_{\mathrm{loc}} := \left(0.8\varepsilon_g / \max_l |\Delta\hat{x}_l|_{\mathrm{sc}}\right)^{(s-1)/(s-2)} \varepsilon_{\mathrm{loc}}$;
    end while;
13. Stop.
```

The numerical solutions $\hat{x}_L^4$ and P_L, where the subscript L marks the last node in the generated mesh $\{t_l\}_{l=0}^L$ (i.e., $t_L \equiv t_k$), are taken as the output of the triple NIRK4(2)M2 (or NIRK6(4)M2) applied to MDEs (2.1)–(2.2) for calculating the predicted state expectation $\hat{x}_{k|k-1} := \hat{x}_L^4$ and the predicted covariance matrix $P_{k|k-1} := P_L$ with the scaled global error not exceeding the user-supplied tolerated error bound ε_g. This is the sole parameter that has to be set by the user, and which is chosen to be 10^{-4} in the numerical experiments, below. One can alter the global error tolerance ε_g depending on the requested accuracy of state estimation.

Measurement Update: Having computed the predicted state mean $\hat{x}_{k|k-1}$ and error covariance matrix $P_{k|k-1}$, one determines the filtering state mean $\hat{x}_{k|k}$ and error covariance matrix $P_{k|k}$ by means of the measurement-update formulas (1.9)–(1.10).

It should be noted that the presented ACD-EKF algorithm covers both NIRK4(2)M2- and NIRK6(4)M2-based state estimators. Thus, for NIRK4(2)M2, one sets $s = 4$ and uses proper formulas for computing the corresponding numerical solution to MDEs (2.1)–(2.2) and its errors, as explained above. The other state estimator is obtained by setting $s = 6$ in the above algorithm and utilizing formulas related to the triple NIRK6(4)M2 for calculating the numerical solution and its scaled local and global errors (2.9)–(2.12). The notation $\max_l |\Delta\hat{x}_l|_{\mathrm{sc}}$ stands for the maximum scaled global error evaluated in the current numerical integration run.

3 Continuous–discrete cubature and unscented Kalman filters

In contrast to the presented ACD-EKF, which uses variable meshes, the *continuous–discrete cubature Kalman filter* (CD-CKF) [2] is grounded in the Itô-Taylor expansion of order 1.5 (IT-1.5) applied to SDEs for converting them to stochastic difference equations on equidistant meshes. Then, having completed this transformation and assumed that all the conditional densities are Gaussian-distributed, the filtering solution to the obtained discrete-time stochastic model further reduces to approximation of the arisen Gaussian-weighted integrals by cubature rules. The cited paper recommends an implementation of the third-degree spherical-radial cubature rule (with all necessary details and explanation) in this setting.

The *continuous–discrete unscented Kalman filter* (CD-UKF) designed in [18] is also a fixed-stepsize state estimator based on the same IT-1.5. To introduce both CD-CKF and CD-UKF, we remark at first that the cited papers deal with a particular case of SDE (1.1) where the diffusion matrix of the stochastic noise term is constant, i.e., $G(t) \equiv G$ and $Q(t) \equiv Q$. So, to simplify the presentation, we impose the mentioned condition and discuss these two state estimators for SDEs of the form

$$\mathrm{d}x(t) = F(x(t), u(t))\mathrm{d}t + G\mathrm{d}w(t), \quad t > 0 \tag{3.1}$$

because the chemical model examined in Section 4 is covered by SDE (3.1). In general, the considered discretization scheme does not work for the stochastic model (1.1) and must be replaced with the more complicated formula in [15, Sec. 10.4].

Thus, having divided the sampling interval $[t_{k-1}, t_k]$ in m equal subintervals and applied the IT-1.5 [2, Formula (18)] to SDE (3.1), we arrive at a discrete-time stochastic model, which is estimated then either by the CKF [1] or by the additive (zero-mean) noise case UKF [30, Table 7.3] with $\alpha = 1$, $\beta = 0$, and $\lambda = 3 - n_1$. In addition, taking into account existence of numerically sensitive operations involved in the CD-CKF, we utilize its square-root version [2, Appendix B]. The CD-UKF is implemented in the original (non-square-root) form. Further details of these CD-CKF and CD-UKF methods can be found in [18, Sec. IIIB, IIIC].

Finally, we point out that the quality of any continuous–discrete nonlinear state estimator depends mainly on errors of two sorts, namely, on the error in capturing the nonlinear dynamics (i.e., the *discretization error*) and on the error in approximating the moments of Gaussian density (i.e., the *moment approximation error*). Both errors influence the accuracy and reliability of implemented continuous–discrete nonlinear Kalman filter, considerably.

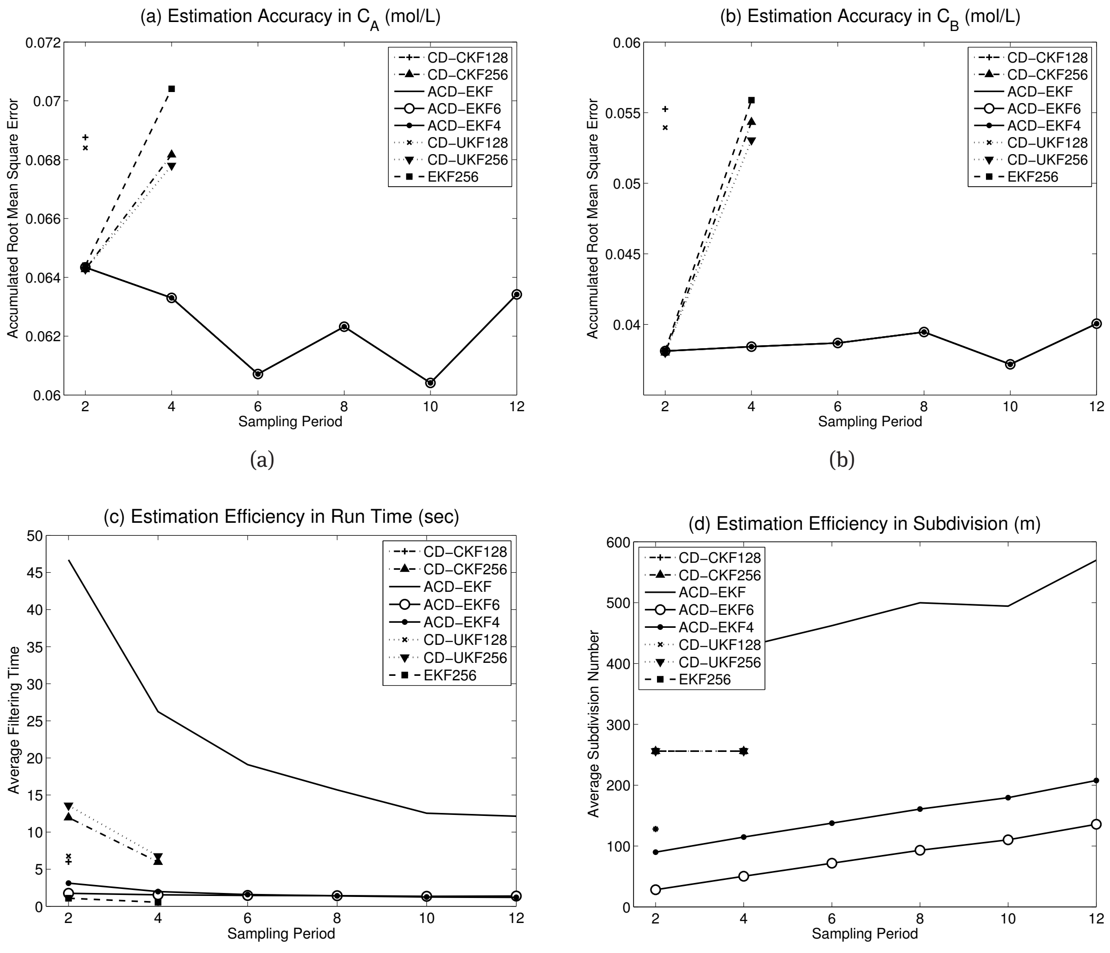
Our ACD-EKF clearly outperforms the fixed-stepsize CD-CKF and CD-UKF algorithms in terms of committed discretization errors. The latter state estimation techniques calculate the predicted state mean and error covariance on a prefixed equidistant mesh, which depends on the number m of utilized extra mesh nodes, and, hence, with unpredictable errors. In other words, the CD-CKF and/or CD-UKF with a fixed m can work successfully for one model and fail for others, and this may hardly be predicted *a priori*. To address the stated problem, we have developed the ACD-EKF methods, which generate variable meshes in each sampling interval $[t_{k-1}, t_k]$ such that a preassigned level ε_g of accuracy is achieved, automatically. More precisely, the user limits the maximum magnitude ε_g of the scaled global errors tolerated in the numerical solution of the MDEs and the ACD-EKF solver yields the predicted state mean vector and covariance matrix calculated for the set accuracy condition in automatic mode. Certainly, this is a more sophisticated filtering technique, but it increases the accuracy and reliability of state estimation, essentially [17, 18, 20, 22]. On the other hand, the moment approximation errors of the CD-CKF and CD-UKF are smaller than that of the ACD-EKF [13, 14, 30]. Thus, when the discretization errors of the fixed-stepsize filters are negligible they will be expected to produce more accurate results than our new ACD-EKF methods will do.

4 Numerical example

The elaborated filters are examined on one SDE model in chemical engineering. We recall that the CD-CKF and CD-UKF are m-step methods. Therefore they are marked, for instance, as CD-CKF128 or CD-UKF256 in Fig. 1 and below. The first abbreviation stands for the CD-CKF with $m = 128$ subdivisions of each sampling interval, and the second one means the CD-UKF with $m = 256$. The traditional EKF method (1.6)–(1.10) is also coded and run, but it is implemented within the m-step fashion, as the other fixed-stepsize filters. The latter estimator is abbreviated to EKF256. We remark that this EKF is tested only for $m = 256$ since it is grounded in the Euler-Maruyama scheme of order 0.5. That is why the traditional EKF is not competitive to the CD-CKF and CD-UKF for the same number m of steps on the chemical system under consideration. The ACD-EKF is a variable-stepsize filter. Thus, we abbreviate the NIRK4(2)M2-based ACD-EKF to ACD-EKF4. Similarly, ACD-EKF6 implies the NIRK6(4)M2-based ACD-EKF. For comparison, we also examine the earlier version of the ACD-EKF from [17]. It is abbreviated to ACD-EKF. All our ACD-EKF methods are implemented with the same accuracy condition $\varepsilon_g := 10^{-4}$. The filters under examination are coded and run in MATLAB.

The chosen test problem is the *Van der Vusse* benchmark example [12]. It models the reaction of four species denoted as A, B, C, and D. The desired product is B, while C and D are unwanted by-products. This reaction is conducted in a continuously stirred tank reactor (CSTR) with a cooling jacket and presented by the SDE (3.1) where the state vector is $(c_A(t), c_B(t), T(t), T_J(t))^T \in \mathbb{R}^4$ with $c_A(t)$ and $c_B(t)$ denoting concentrations of the species A and B at time t and with $T(t)$ and $T_J(t)$ standing for temperatures of the CSTR and the cooling jacket, respectively. The full mathematical description of this Van der Vusse example and the used measurement equation can be found in [18, 20]. The given chemical model is corrupted by a disturbance in the feed concentration of A. More precisely, we assume that the constant parameter c_{A0} is increased by 100% at time $t = 50$ hr, as in the cited papers. The entire simulation time is 120 hr.

For testing, we generate first a reference solution and true measurements for each size δ of the sampling period fixed in this experiment by solving the Van der Vusse model with the Euler–Maruyama method and the fixed step size equal to 0.0001. Then, we conduct 100 Monte Carlo runs to evaluate the *accumulated root mean square error* (ARMSE) in the first two entries of the state vector (i.e., in the estimated concentrations



(a)

(b)

(c)

(d)

Figure 1. A comparison of the m-step EKF, CD-CKF, CD-UKF methods and the variable-stepsize ACD-EKFs in the stochastic Van der Vusse reaction scenario.

of the species A and B), for the waiting times $\delta = 2, 4, 6, 8, 10, 12$ hr. For instance, the ARMSE_{c_A}, i.e., the ARMSE in the estimated concentration c_A, is evaluated by

$$\mathrm{ARMSE}_{c_A} := \left[\frac{1}{100K} \sum_{l=1}^{100} \sum_{k=1}^{K} \left(c_A^{\mathrm{ref},l}(t_k) - \hat{c}_A^{k|k,l}\right)^2 \right]^{1/2}$$

where the superscript ‘ref’ stands for the computed reference stochastic solution, l marks the corresponding Monte Carlo simulation, k denotes a particular sampling time t_k and $K := [120/\delta]$ (with $[\cdot]$ standing for the integer part of the number) implies the total number of sampling instants for the fixed value of δ. These errors are exposed in Fig. 1ab. They show the estimation accuracy of all the filters under examination. In addition, Fig. 1cd exhibits the *average filtering time* of 100 Monte Carlo runs and the *average subdivision number* (m) of the sampling interval, respectively. We stress that m is predefined in the fixed-stepsize filters, only. The latter diagrams allow the efficiency of the utilized filters to be assessed.

Figure 1 displays that all the fixed-stepsize filters (CD-UKF128, CD-CKF128, EKF256, CD-CKF256, CD-UKF256) succeed for small sizes of the sampling period, i.e., for $\delta = 2, 4$ hr, only. When the waiting time $\delta \geqslant 6$ hr they fail to return a numerical answer. That is why no data are exhibited for these filters in Fig. 1 when $\delta \geqslant 6$ hr. Among these fixed-stepsize filters, the most accurate one is CD-UKF256, followed by CD-CKF256 and, then, by EKF256. The remaining two filters work for the single $\delta = 2$ hr, and again CD-UKF128 is slightly more accurate than CD-CKF128 (see Fig. 1ab). On the other hand, Fig. 1c shows that the cheapest method is EKF256, followed by CD-CKF128 and CD-UKF128 and, then, by CD-CKF256 and CD-UKF256. In other words, we conclude that the fixed-stepsize filters may be used in the Van der Vusse reaction scenario when the waiting time δ is reasonably short. Additionally, EKF256 provides the comparable (to CD-CKF256 and CD-UKF256) accuracy of the state estimation, but much cheaper.

Our variable-stepsize filters (ACD-EKF, ACD-EKF4, ACD-EKF6) succeed for all δ. Their ARMSEs are undistinguishable and vary slightly in Fig. 1ab. This says that the ACD-EKF methods are insensitive to the size of the sampling interval and, hence, they can work for sufficiently long waiting times as well. So they are a proper means for attacking the problem of infrequent measurements stated in chemical engineering by Soroush [29]. In addition, our simulation shows that all the filters perform well for sufficient short waiting times, as that considered in [12], and reconstruct the offset free concentrations on the basis of the temperature measurements, only. Thus, this result resolves also the criticism reported in the cited paper.

The efficiency plot Fig. 1c exposes that the average filtering time of ACD-EKF4 and ACD-EKF6 is much less than that of ACD-EKF. This is because the earlier designed filter uses much more steps to integrate numerically on each sampling interval in comparison to the other variable-stepsize filters considered in this paper (see Fig. 1d). In addition, our six-order method ACD-EKF6 outperforms slightly the fourth-order one ACD-EKF4 because it is equally accurate but less time-consuming, especially for $\delta = 2$ hr (see Fig. 1c). It is also interesting to note that ACD-EKF6, which is the most efficient one among the adaptive filters, is almost as efficient as EKF256, but the latter state estimator is much less accurate. Thus, ACD-EKF6 can be a method of choice in practical state estimation tasks.

5 Concluding remarks

The present paper elaborates such a simple but still efficient state estimation technique as the EKF. It has been used successfully in many areas of study for decades. However, despite its efficiency the EKF has been criticized recently on its performance for a number of mathematical models in literature. On the other hand, we have shown that the better implementation of this method allows some cited criticism to be resolved. The latter creates the necessary background of using the EKF for estimation of complicated continuous–discrete stochastic state-space systems.

We have observed that our version of the EKF, i.e., the ACD-EKF, is even competitive to the contemporary fixed-stepsize CD-CKF and CD-UKF methods. As we mentioned above, the discretization and moment approximation errors determine the total accuracy of practical nonlinear state estimators, and the first one dominates evidently in majority of continuous–discrete stochastic models with strong nonlinearity. That is why the variable-stepsize ACD-EKF can be more efficient for such sort of state-space systems than the above-discussed m-step CD-CKF and CD-UKF methods, where the discretization error is unpredictable and may be of any size, depending on the value of m. Besides, our new technique is more convenient for practical utilization because it does not require any manual tuning in contrast to the fixed-stepsize filters where a proper number m of the sampling period subdivisions must be *a priori* identified and set by the user. The latter is a nontrivial and time-consuming task even for experienced users.

We stress that the power of our filters is grounded in adaptivity of the implemented MDE solvers with the scaled local and global error controls designed in this paper. It is sufficient to fix the requested accuracy by setting the corresponding value of the parameter ε_g in the ACD-EKF method and the code will compute the predicted moments of propagated Gaussian density with the scaled global error corresponding to the user-supplied bound ε_g in automatic mode. In other words, our ACD-EKFs are self-turned algorithms, which generate automatically optimal sampling interval subdivisions depending on the imposed accuracy condition. In addition, it is worthwhile to point out that the positive semi-definiteness of the predicted covariance is ensured in our methods in exact arithmetic. We emphasize that these two important properties are not provided by conventional general purpose codes as, for instance, built-in MATLAB ODE solvers [10, Section 12.2], which can be applied for treating the mentioned MDEs [19]. The earlier version of the ACD-EKF method presented in [17] is also not capable for preserving the positive semi-definiteness of the predicted covariance. That is why the newly designed ACD-EKFs are superior for practical use.

Finally, we have to remark on complexity of the fixed-stepsize CD-CKF and CD-UKF methods discussed in the paper. We recall that the CKF and UKF are considered to be derivative free methods in contrast to the EKF. However, this conclusion does not hold for continuous-time stochastic systems because the application of the IT-1.5 discretization formula demands the differential operators $\mathbb{L}_0$ and $\mathbb{L}_j$ to be evaluated at each step of the CD-CKF and CD-UKF (see further details in [2, 18]). Such calculations are more complicated and time-consuming than the evaluation of the standard Jacobian requested in our ACD-EKF methods. Note that the latter can even be fulfilled numerically (and in automatic mode) in contrast to the evaluation of the operators $\mathbb{L}_0$ and $\mathbb{L}_j$ made by hand. It should also be remarked that the formulas presented for calculation of these operators in [2, 18] work for SDEs with the constant diffusion matrix G (i.e., for mathematical models of the form (3.1)), only. Thus, the need for estimating the more general SDE (1.1) complicates essentially the evaluation of $\mathbb{L}_0$ and $\mathbb{L}_j$, as explained in [15, Sec. 10.4], and makes it hardly applicable in practice. In contrast, our ACD-EKFs treat both SDE models (1.1) and (3.1) in the same way, i.e., without an extra effort from the user. Thus, ACD-EKF4 and ACD-EKF6 possess an obvious applied potential and seem to be a good practical alternative to the fixed-stepsize EKF, CD-CKF and CD-UKF, especially in the situation when a quick sampling is technically impossible (or too expensive) or the sampling is irregular.

Funding: This work was supported by Portuguese National Fund (Fundação para a Ciência e a Tecnologia) within projects UID/Multi/04621/2013, SFRH/BPD/64397/2009 and the Investigador FCT 2013 programme.

References

[1] I. Arasaratnam and S. Haykin, Cubature Kalman filters. *IEEE Trans. Automat. Contr.* **54** (2009), 1254–1269.

[2] I. Arasaratnam, S. Haykin, and T. R. Hurd, Cubature Kalman filtering for continuous–discrete systems: Theory and simulations. *IEEE Trans. Signal Process.* **58** (2010), 4977–4993.

[3] Y. Bar-Shalom, X.-R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*. Wiley, New York, 2001.

[4] J. L. Crassidis and J. L. Junkins, *Optimal Estimation of Dynamic Systems*. CRC Press LLC, New York, 2004.

[5] P. Frogerais, J.-J. Bellanger and L. Senhadji, Various ways to compute the continuous–discrete extended Kalman filter. *IEEE Trans. Automat. Contr.* **57** (2012), 1000–1004.

[6] G. C. Goodwin and K. S. Sin, *Adaptive Filtering Prediction and Control*. Prentice-Hall, Englewood Cliffs, New Jersey, 1984.

[7] M. S. Grewal and A. P. Andrews, *Kalman Filtering: Theory and Practice*. Prentice Hall, New Jersey, 2001.

[8] M. S. Grewal, L. R. Weill, and A. P. Andrews, *Global Positioning Systems, Inertional Navigation and Integration*. Wiley, New York, 2001.

[9] E. L. Haseltine and J. B. Rawlings, Critical evaluation of extended Kalman filtering and moving-horizon estimation. *Ind. Eng. Chem. Res.* **44** (2005), 2451–2460.

[10] D.J. Higham and N.J. Higham, *MATLAB Guide*, SIAM, Philadelphia, 2005.

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

[12] J. B. Jørgensen, A critical discussion of the continuous–discrete extended Kalman filter. In: *European Congress of Chemical Engineering - 6*, *Copenhagen*, *Denmark*, 2007. (Available at http://www.nt.ntnu.no/users/skoge/prost/proceedings/ecce6_sep07/upload/3520.pdf).

[13] S. J. Julier and J. K. Uhlmann, Unscented filtering and nonlinear estimation. *Proc. of the IEEE* **92** (2004), 401–422.

[14] S. J. Julier, J. K. Uhlmann, and H. F. Durrant-Whyte, A new method for the nonlinear transformation of means and covariances in filters and estimators. *IEEE Trans. Automat. Contr.* **45** (2000), 477–482.

[15] P. E. Kloeden and E. Platen, *Numerical Solution of Stochastic Differential Equations*. Springer, Berlin, 1999.

[16] G. Yu. Kulikov, Cheap global error estimation in some Runge–Kutta pairs. *IMA J. Numer. Anal.* **33** (2013), 136–163.

[17] G. Yu. Kulikov and M. V. Kulikova, Accurate numerical implementation of the continuous–discrete extended Kalman filter. *IEEE Trans. Automat. Contr.* **59** (2014), 273–279.

[18] G. Yu. Kulikov and M. V. Kulikova, Accurate state estimation in the Van der Vusse reaction. In: *Proc. of the 2014 IEEE Multi-Conference on Systems and Control*, pp. 759–764, Oct. 2014.

[19] G. Yu. Kulikov and M. V. Kulikova, The accurate continuous–discrete extended Kalman filter for continuous-time stochastic systems. *Russ. J. Numer. Anal. Math. Modelling* **30** (2015), 239–249.

[20] G. Yu. Kulikov and M. V. Kulikova, High-order accurate continuous–discrete extended Kalman filter for chemical engineering. *European J. Control* **21** (2015), 14–26.

[21] G. Yu. Kulikov and R. Weiner, A singly diagonally implicit two-step peer triple with global error control for stiff ordinary differential equations. *SIAM J. Sci. Comput.* **37** (2015), A1593–A1613.

[22] M. V. Kulikova and G. Yu. Kulikov, Square-root accurate continuous–discrete extended Kalman filter for target tracking. In: *Proc. of the 52-nd IEEE Conference on Decision and Control*, pp. 7785–7790, Dec. 2013.

[23] F. L. Lewis, *Optimal Estimation: with an Introduction to Stochastic Control Theory*. John Wiley & Sons, New York, 1986.

[24] T. Mazzoni, Computational aspects of continuous–discrete extended Kalman filtering. *Comput. Statist.* **23** (2008), 519–539.

[25] B. Øksendal, *Stochastic Differential Equations: An Introduction with Applications*, Springer, New York, 2003.

[26] A. Romanenko and J. A. A. M. Castro, The unscented filter as an alternative to the EKF for nonlinear state estimation: a simulation case study. *Comput. Chem. Eng.* **28** (2004), 347–355.

[27] A. Romanenko, L. O. Santos, and P. A. F. N. A. Afonso, Unscented Kalman filtering of a simulated pH system. *Ind. Eng. Chem. Res.* **43** (2004), 7531–7538.

[28] D. Simon, *Optimal State Estimation: Kalman, H Infinity and Nonlinear Approaches*. Wiley, Hoboken, New Jersey, 2006.

[29] M. Soroush, State and parameter estimation and their applications in process control. *Comput. Chem. Eng.* **23** (1998), 229–245.

[30] E. A. Wan and R. Van der Merwe, The unscented Kalman filter. In: *Kalman Filtering and Neural Networks* (Ed. S. Haykin). John Wiley & Sons, Inc., New York, 2001, pp. 221–280.

[31] R. Weiner and G. Yu. Kulikov, Local and global error estimation and control within explicit two-step peer triples. *J. Comput. Appl. Math.* **262** (2014), 261–270.

[32] D. I. Wilson, M. Agarwal, and D. W. T. Rippin, Experiences implementing the extended Kalman filter on an industrial batch reactor. *Comput. Chem. Eng.* **22** (1998), 1653–1672.