Accurate continuous-discrete unscented Kalman filtering for estimation of nonlinear continuous-time stochastic models in radar tracking

G.Yu. Kulikov, M.V. Kulikova

PII: S0165-1684(17)30130-5
DOI: 10.1016/j.sigpro.2017.04.002
Reference: SIGPRO 6448

To appear in: *Signal Processing*

Received date: 6 June 2016
Revised date: 8 March 2017
Accepted date: 3 April 2017

Please cite this article as: G.Yu. Kulikov, M.V. Kulikova, Accurate continuous-discrete unscented Kalman filtering for estimation of nonlinear continuous-time stochastic models in radar tracking, *Signal Processing* (2017), doi: 10.1016/j.sigpro.2017.04.002

Highlights

- The novel accurate continuous-discrete unscented Kalman filter is devised for treating stochastic models in radar tracking.
- The accurate continuous-discrete extended Kalman filter is revised for accurate estomation of stochastic models in radar tracking.
- The mixed-type accurate continuous-discrete extended-unscented Kalman filter is updated for raising its efficiency for radar tracking models.
- The accurate continuous-discrete extended and unscented Kalman filters are examined in estimating an aircraft executing a coordinated turn.

Accurate continuous-discrete unscented Kalman filtering for estimation of nonlinear continuous-time stochastic models in radar tracking

G. Yu. Kulikov[a,*], M. V. Kulikova[a]

[a] *CEMAT, Instituto Superior Técnico, Universidade de Lisboa, Av. Rovisco Pais, 1049-001 Lisboa, Portugal.*

Abstract

This paper presents a new state estimation technology grounded in the unscented Kalman filtering for nonlinear continuous-time stochastic systems. The resulting accurate continuous-discrete unscented Kalman filter is based on adaptive solvers with automatic global error control for treating numerically the moment differential equations arising in the mean and covariance calculation of propagated Gaussian density. It is intended for an accurate and robust state estimation in nonlinear continuous-discrete stochastic systems of various sorts, including in radar tracking models. This new filter is examined in severe conditions of tackling a seven-dimensional radar tracking problem, where an aircraft executes a coordinated turn. The latter is considered to be a challenging one for testing nonlinear filtering algorithms. For comparison, we also examine such efficient state estimators as the accurate continuous-discrete extended Kalman filter, the continuous-discrete unscented Kalman filter and the mixed-type accurate continuous-discrete extended-unscented Kalman filter designed earlier, but further modified in the present study. The comparison is fulfilled in terms of accuracy and efficiency of estimating the state in the mentioned air traffic control scenario.

Keywords: Nonlinear continuous-discrete stochastic state-space system, continuous-discrete unscented Kalman filter, accurate continuous-discrete extended Kalman filter, accurate continuous-discrete extended-unscented Kalman filter, accurate continuous-discrete unscented Kalman filter, radar tracking

1. Introduction

Many authors consider air/space traffic control scenarios as challenging tests for their nonlinear state estimators. For instance, Arasaratnam et al. [1] exhibit excellent properties of the introduced continuous-discrete cubature Kalman filter on a single-target single-sensor

*Corresponding author.

Email addresses: gkulikov@math.ist.utl.pt (G. Yu. Kulikov), maria.kulikova@ist.utl.pt (M. V. Kulikova)

tracking problem. The same test problem, but with a different process noise term, is employed for examination of various filtering and smoothing methods in [2]. Also, Särkkä [3] uses a reentry vehicle tracking for confirming advantages of his continuous-discrete unscented Kalman filter, and Teixeira et al. [4] conduct a comparison of their continuous-discrete extended and unscented Kalman filters on a problem of satellite trajectory estimation (also known as orbit determination). The purpose of this paper is to develop a new state estimation technology grounded in the unscented Kalman filtering, which is further termed the *Accurate Continuous-Discrete Unscented Kalman Filter* (ACD-UKF), for nonlinear continuous-time stochastic state-space systems and examine it in severe conditions of the air traffic control scenario in [1]. We also intend for a comparison of the accuracy and efficiency of this new filtering technique with those of the earlier designed *Accurate Continuous-Discrete Extended Kalman Filter* (ACD-EKF) [5–7], the *Continuous-Discrete Unscented Kalman Filter* (CD-UKF) grounded in the Itô-Taylor approximation of order 1.5 [7, 8] and the mixed-type *Accurate Continuous-Discrete Extended-Unscented Kalman Filter* (ACD-EUKF) [9]. However, the ACD-EKF and ACD-EUKF methods are amended as explained in Sections 2.2 and 2.3.

Our paper deals with nonlinear continuous-time mathematical models that obey the Itô-type *Stochastic Differential Equation* (SDE)

$$dX(t) = F\big(X(t)\big)dt + GdW(t), \quad t > 0, \tag{1}$$

where $X(t) \in \mathbb{R}^n$ is the n-dimensional real vector of system's state at time t, $F : \mathbb{R}^n \to \mathbb{R}^n$ is a drift function representing the dynamic behavior of the model, G is a constant square matrix of size n and $\{W(t),\, t > 0\}$ is the standard Brownian process with the identity diffusion matrix of the same size. The initial state X_0 of stochastic process (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 Gaussian distribution with mean $\bar{X}_0$ and covariance Π_0. Mathematical models of the form (1) are being utilized in many areas of study as diverse as target tracking, navigation, stochastic control, finance, and so on [1–4, 10–18].

The task of least-square state estimation in the SDE model (1) always compounds real measurements of some model's variables or their function (depending on the utilized technology) with computation of remaining (not measurable) parameters by an appropriate nonlinear filtering algorithm. It is usually assumed that the measurement 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 measurement Z_k to the state X_k of SDE (1) obeys the *measurement equation*

$$Z_k = h(X_k) + V_k, \quad k \geq 1, \tag{2}$$

where k stands for a discrete time index (i.e. X_k means the true state $X(t_k)$ at time t_k), $Z_k \in \mathbb{R}^m$ is the information available at time t_k, $h : \mathbb{R}^n \to \mathbb{R}^m$ is a sufficiently smooth function and the measurement noise $V_k \sim \mathcal{N}(0, R_k)$ with $R_k > 0$. All realizations of the noises $W(t)$, V_k and the initial state X_0 are taken from mutually independent normal distributions.

We point out that the equidistant fashion of measurement (2) is not obligatory in the methods presented in Section 2. When the sampling is irregular all our variable-stepsize

Kalman filters will be applied with no alteration in practice. Nevertheless, the equidistant sampling accepted in this paper is demanded in the state-of-the-art fixed-stepsize CD-UKF, because it is not able to accommodate automatically changes of the sampling rate, but the latter filter is also tested and compared to the variable-stepsize ones in Section 3.

The idea of accurate nonlinear Kalman filtering originates from [19], and now we are familiar with two efficient versions of the ACD-EKF grounded in the fourth- and sixth-order MDE solvers `NIRK4(2)M2` and `NIRK6(4)M2` with automatic local and global error controls [5–7]. The same approach yields also the rather efficient mixed-type ACD-EUKF algorithm designed for estimating nonlinear continuous-time stochastic state-space systems of the form (1), (2) with strongly nonlinear (and even nondifferentiable) observations in [9]. Here, we further advance the accurate continuous-discrete Kalman filtering theory for nonlinear stochastic systems and build the ACD-UKF method based on accurate numerical solutions of the *Moment Differential Equations* (MDEs) derived in [3]. Moreover, taking into account the higher efficiency of the numerical integration method `NIRK6(4)` exhibited in comparison to the fourth-order one `NIRK4(2)` in [20, 21], we restrict ourselves to the MDE solver `NIRK6(4)M2` for implementing this new ACD-UKF. In addition, we improve the earlier constructed ACD-EKF and ACD-EUKF methods by using more flexible and effective iterations for solving arisen nonlinear problems and also by relaxing our accuracy conditions in the utilized global error control mechanisms. All this allows the filtering time in radar tracking (and in other engineering tasks) to be reduced and, hence, the efficiency of these new versions of the ACD-EKF and ACD-EUKF to be raised, considerably.

2. Nonlinear Kalman filtering methods

In Section 2, we design the ACD-UKF state estimator for the continuous-discrete stochastic state-space system (1), (2) and also develop new versions of the ACD-EKF and ACD-EUKF methods, which are more efficient and robust in practical use. We begin with our ACD-UKF technique at first.

2.1. Accurate continuous-discrete unscented Kalman filter

Recently, it was shown that the accurate continuous-discrete nonlinear Kalman filtering presents quite a successful alternative to both the traditional EKF and the fixed-stepsize UKF for estimating stochastic mathematical models arisen in radar tracking and other engineering [5–9]. This method is grounded in an accurate numerical integration of MDEs for two moments of propagated Gaussian density. Therefore, to extend the idea of accurate nonlinear Kalman filtering to the more advanced ACD-UKF technique, we need first the MDEs corresponding to the Gaussian density propagated within the UKF method in time.

It is known that the UKF was designed originally for estimating discrete-time stochastic systems and is based on the *Unscented Transform* (UT) [22, 23]. The UT implies that the set of $2n+1$ deterministically selected sigma-points (vectors) $\mathcal{X}_i$ is chosen by the rule

$$\mathcal{X}_0 = \hat{X}, \quad \mathcal{X}_i = \hat{X} + \sqrt{\alpha+n}Se_i, \quad \mathcal{X}_{i+n} = \hat{X} - \sqrt{\alpha+n}Se_i, \quad i = 1, 2, \ldots, n, \qquad (3)$$

where e_i is the i-th unit coordinate vector in $\mathbb{R}^n$ and α is a predefined constant, for a given n-dimensional random variable $X \sim \mathcal{N}(\hat{X}, P = SS^\top)$ with S standing for the lower triangular Cholesky factor (square root) of the covariance matrix P. It is also possible to reduce the number of sigma-points to $n+1$ for the price of some accuracy and/or robustness loss [24–26]. However, those variants of the UT are not discussed in this paper. Below, we utilize the classical parametrization of the UT, i.e use the UT coefficients:

$$w_0^{(m)} = \frac{\lambda}{n+\lambda}, \quad w_0^{(c)} = \frac{\lambda}{n+\lambda} + 1 - \alpha^2 + \beta, \; w_i^{(m)} = w_i^{(c)} = \frac{1}{2n+2\lambda}, \; i = 1, 2, \ldots, 2n, \quad (4)$$

with the constants $\alpha = 1$, $\beta = 0$ and $\lambda = 3 - n$ [22]. An exhaustive study of other parameterizations of the UT and related issues is available in [26].

The sigma-vectors (3) and weights (4) allow the mean and covariance of the given Gaussian distribution to be calculated as follows:

$$\hat{X} = \sum_{i=0}^{2n} w_i^{(m)} \mathcal{X}_i, \quad (5)$$

$$P = \sum_{i=0}^{2n} w_i^{(c)} (\mathcal{X}_i - \hat{X})(\mathcal{X}_i - \hat{X})^\top. \quad (6)$$

The main property of the considered UT is that if one changes the given Gaussian distribution with mean $\hat{X}$ and covariance P by a sufficiently smooth nonlinear transformation $F(\cdot)$ then the mean and covariance of the transformed random variable $F(X)$ will be calculated approximately by the same formulas (5) and (6) but with the sigma-vectors $\mathcal{X}_i$ replaced by the transformed ones $F(\mathcal{X}_i)$ and with the mean $\hat{X}$ replaced by the mean evaluated for the transformed distribution by the transformed formula (5). It should also be noted that the UT approach provides a rather good accuracy of the mean and covariance computation of transformed symmetric random distributions (see, for instance, the proof in [27, Section 5.3] with some correction in [18, Section 5.5] and in [26]).

The important contribution of Särkkä [3] is that the evolution of the mean and covariance of the stochastic process determined by SDE (1) in time obeys the deterministic MDEs

$$\frac{d\hat{X}(t)}{dt} = F\big(\mathcal{X}(t)\big) W_m, \quad (7)$$

$$\frac{dP(t)}{dt} = \mathcal{X}(t) \mathcal{W} F^\top\big(\mathcal{X}(t)\big) + F\big(\mathcal{X}(t)\big) \mathcal{W} \mathcal{X}^\top(t) + Q \quad (8)$$

where the sigma-matrix $\mathcal{X}(t) := [\mathcal{X}_0(t), \ldots, \mathcal{X}_{2n}(t)]$ of size $n \times (2n+1)$ and with its columns $\mathcal{X}_i(t)$ denoting the sigma-vectors calculated by formulas (3) (with understanding that all the vectors and matrices depend on time now, and the lower triangular matrix S means the square root of the covariance $P(t)$ from the MDE (8)), $F(\mathcal{X}(t))$ denotes the sigma-matrix $\mathcal{X}(t)$ transformed by the drift function of the process equation (1), i.e. $F(\mathcal{X}(t)) := [F(\mathcal{X}_0(t)), \ldots, F(\mathcal{X}_{2n}(t))]$, $Q := GG^\top$, $\hat{X}(t)$ stands for the predicted state mean at time t.

The fixed $(2n+1)$-dimensional coefficient vector W_m and the square matrix $\mathcal{W}$ of size $2n+1$, both utilized in the MDEs (7), (8), depend on weights (4). They are defined by the following rule:

$$W_m := \big[w_0^{(m)}, \ldots, w_{2n}^{(m)}\big]^\top, \tag{9}$$

$$\mathcal{W} := \big(I_{2n+1} - [W_m, \ldots, W_m]\big)\mathrm{diag}\big\{w_0^{(c)}, \ldots, w_{2n}^{(c)}\big\}\big(I_{2n+1} - [W_m, \ldots, W_m]\big)^\top, \tag{10}$$

in which I_{2n+1} stands for the identity matrix of size $2n+1$, and the notation $\mathrm{diag}\{w_0^{(c)}, \ldots, w_{2n}^{(c)}\}$ implies the diagonal matrix of size $2n+1$ and with the given entries on its main diagonal. The MDEs (7), (8) are to be integrated in each sampling period $[t_{k-1}, t_k]$ and with the initial values $\hat{X}(t_{k-1}) = \hat{X}_{k-1|k-1}$, $P(t_{k-1}) = P_{k-1|k-1}$, which are the filtering solution at time t_{k-1}, for finding the predicted state expectation and covariance matrix at the next sampling instant t_k. More formally, their predicted values are taken to be $\hat{X}_{k|k-1} = \hat{X}(t_k)$ and $P_{k|k-1} = P(t_k)$. Then, the measurement update step of the presented CD-UKF is conducted in its usual form [3]. We point out that various solvers can be used for treating the MDEs (7), (8), which result in various implementations of this state estimation technique. For example, one implementation grounded in the built-in Matlab code `ode45` is presented in [4] and utilized for the orbit determination, there. Unfortunately, the local error control implemented in all Matlab solvers does not allow the MDEs (7), (8) to be solved with preassigned accuracy and, hence, the accuracy of state estimation may be compromised in that method. Therefore we elaborate further a variant of the CD-UKF grounded in the embedded pair `NIRK6(4)` with automatic global error control from [20], which is capable of solving the MDEs for user-supplied accuracy conditions. That is why our state estimator is termed the *Accurate Continuous-Discrete Unscented Kalman Filter* (ACD-UKF), below.

First of all we remark that, in contrast to MDEs used in the ACD-EKF and ACD-EUKF methods, which can be treated separately [5–7, 9], the MDEs (7), (8) underlying the ACD-UKF are strongly linked because their right-hand parts depend on the sigma-matrix $\mathcal{X}(t)$ and its columns relate to the expectation $\hat{X}(t)$ and the covariance $P(t)$ by means of formulas (3). Therefore these equations must be treated simultaneously in the numerical integration method `NIRK6(4)`. Besides, the MDE (7) is a vector equation, but the MDE (8) is a matrix one. Thus, their simultaneous treatment is complicated and the matrix MDE should be vectirized at first. Fortunately, such a transformation may be avoided with use of the equivalent Särkkä's formulation [3] of the above MDEs, but with respect to the sigma-points $\mathcal{X}_i(t)$, i.e.

$$\frac{d\mathcal{X}_i(t)}{dt} = F\big(\mathcal{X}(t)\big)W_m + \sqrt{3}\big[\mathbf{0}, S(t)\Phi(M(t)), -S(t)\Phi(M(t))\big]_i, \quad i = 0, 1, \ldots, 2n, \tag{11}$$

where the subscript i means also the i-th column in the right-hand side matrix of Eq. (11), $\mathbf{0}$ is the first column of zeros in the right-hand side matrix, $S(t)$ is the lower triangular factor in the Cholesky decomposition of the covariance $P(t)$, i.e $P(t) = S(t)S^\top(t)$, $M(t)$ is the matrix evaluated by by the formula

$$M(t) := S^{-1}(t)\left[\mathcal{X}(t)\mathcal{W}F^\top\big(\mathcal{X}(t)\big) + F\big(\mathcal{X}(t)\big)\mathcal{W}\mathcal{X}^\top(t) + Q\right]S^{-T}(t)$$

with all vectors and matrices defined after the MDE (8), and $\Phi(\cdot)$ is a matrix-valued function returning the lower triangular part of the argument matrix by the following rule:

$$\Phi_{lj}(M(t)) := \begin{cases} M_{lj}(t), & \text{if} \quad l > j, \\ M_{lj}(t)/2, & \text{if} \quad l = j, \\ 0, & \text{if} \quad l < j. \end{cases}$$

Further, we discuss an accurate solution method for treating Eqs (11), numerically.

We remark that the differential problem (11) is a system of $2n+1$ vector *Ordinary Differential Equations* (ODEs) of size n. Therefore it is a $(2n^2+n)$-dimensional ODE that can be presented in a concise form as follows:

$$\frac{d\mathbf{X}(t)}{dt} = \mathcal{F}\big(\mathbf{X}(t)\big) \tag{12}$$

with the $(2n^2+n)$-dimensional unknown vector $\mathbf{X}(t) := \big[\mathcal{X}_0^\top(t), \ldots, \mathcal{X}_{2n}^\top(t)\big]^\top$ and the right hand-side function $\mathcal{F}\big(\mathbf{X}(t)\big)$ unifying the right-hand sides of all Eqs (11) into the unique formula. We point out that the matrix $\mathcal{X}(t)$ and the vector $\mathbf{X}(t)$ consist of the same sigma-points $\mathcal{X}_i(t)$, $i = 0, 1, \ldots, 2n$, but differ in the shape of their storing.

Next, we have to discretize the nonlinear continuous-time problem (12) for its further numerical treatment in a sampling interval $[t_{k-1}, t_k]$. For that, we consider that some integration mesh $\{t_l\}_{l=0}^L := \{t_0 = t_{k-1}, t_{l+1} = t_l + \tau_l, l = 0, \ldots, L-1, t_L = t_k\}$ has already been introduced in the mentioned interval. We stress that the mesh $\{t_l\}_{l=0}^L$ is supposed to be variable. This means that the step size τ_l varies as much as it may be necessary. Later on, we explain how to generate such a mesh in each sampling period in automatic mode.

The discretization of ODE (12) is fulfilled by the Gauss-type *Nested Implicit Runge-Kutta* (NIRK) formula of order 6 [21, 28] on the prefixed mesh $\{t_l\}_{l=0}^L$ and yields nonlinear problems of the form

$$\begin{aligned}
\mathbf{X}_{l1}^2 &= a_{11}^2\mathbf{X}_l + a_{12}^2\mathbf{X}_{l+1} + \tau_l\big[d_{11}^2\mathcal{F}(\mathbf{X}_l) + d_{12}^2\mathcal{F}(\mathbf{X}_{l+1})\big], \\
\mathbf{X}_{l2}^2 &= a_{21}^2\mathbf{X}_l + a_{22}^2\mathbf{X}_{l+1} + \tau_l\big[d_{21}^2\mathcal{F}(\mathbf{X}_l) + d_{22}^2\mathcal{F}(\mathbf{X}_{l+1})\big], \\
\mathbf{X}_{l1}^3 &= a_{11}^3\mathbf{X}_l + a_{12}^3\mathbf{X}_{l+1} + \tau_l\big[d_{11}^3\mathcal{F}(\mathbf{X}_l) + d_{12}^3\mathcal{F}(\mathbf{X}_{l+1}) + d_{13}^3\mathcal{F}(\mathbf{X}_{l1}^2) + d_{14}^3\mathcal{F}(\mathbf{X}_{l2}^2)\big], \\
\mathbf{X}_{l2}^3 &= a_{21}^3\mathbf{X}_l + a_{22}^3\mathbf{X}_{l+1} + \tau_l\big[d_{21}^3\mathcal{F}(\mathbf{X}_l) + d_{22}^3\mathcal{F}(\mathbf{X}_{l+1}) + d_{23}^3\mathcal{F}(\mathbf{X}_{l1}^2) + d_{24}^3\mathcal{F}(\mathbf{X}_{l2}^2)\big], \\
\mathbf{X}_{l3}^3 &= a_{31}^3\mathbf{X}_l + a_{32}^3\mathbf{X}_{l+1} + \tau_l\big[d_{31}^3\mathcal{F}(\mathbf{X}_l) + d_{32}^3\mathcal{F}(\mathbf{X}_{l+1}) + d_{33}^3\mathcal{F}(\mathbf{X}_{l1}^2) + d_{34}^3\mathcal{F}(\mathbf{X}_{l2}^2)\big], \\
\mathbf{X}_{l+1} &= \mathbf{X}_l + \tau_l\big[b_1\mathcal{F}(\mathbf{X}_{l1}^3) + b_2\mathcal{F}(\mathbf{X}_{l2}^3) + b_3\mathcal{F}(\mathbf{X}_{l3}^3)\big]
\end{aligned} \tag{13}$$

where $\tau_l := t_{l+1} - t_l$ is the current step size of the mesh $\{t_l\}_{l=0}^L$, and the constant coefficients of the NIRK method (13) are: $b_1 = b_3 := 5/18$, $b_2 := 4/9$, $c_1^2 := (3-\sqrt{3})/6$, $c_2^2 := (3+\sqrt{3})/6$, $a_{11}^2 = a_{22}^2 := 1/2 + 2\sqrt{3}/9$, $a_{12}^2 = a_{21}^2 := 1/2 - 2\sqrt{3}/9$, $d_{11}^2 = -d_{22}^2 := (3+\sqrt{3})/36$, $d_{12}^2 = -d_{21}^2 := (-3+\sqrt{3})/36$, $c_1^3 := (5-\sqrt{15})/10$, $c_2^3 := 1/2$, $c_3^3 := (5+\sqrt{15})/10$, $a_{11}^3 = a_{32}^3 := (125+39\sqrt{15})/250$, $a_{12}^3 = a_{31}^3 := (125-39\sqrt{15})/250$, $a_{21}^3 = a_{22}^3 := 1/2$, $d_{11}^3 = -d_{32}^3 := (7+2\sqrt{15})/200$, $d_{12}^3 = -d_{31}^3 := (-7+2\sqrt{15})/200$, $d_{13}^3 = -d_{34}^3 := (18\sqrt{15}+15\sqrt{3})/1000$, $d_{14}^3 = -d_{33}^3 := (18\sqrt{15}-15\sqrt{3})/1000$, $d_{21}^3 = -d_{22}^3 := 1/32$, $d_{23}^3 = -d_{24}^3 := 3\sqrt{3}/32$. The

latter equation should be solved iteratively for the unknown vector $\mathbf{X}_{l+1}$ at each mesh node t_l, $l = 0, 1, \ldots, L-1$. Notice that formulas (13) present one step of the mentioned Gauss-type NIRK discretization formula in its nested 3-level form, which is preferable for practical use (see more details in [21, 28]).

We emphasize that the nonlinear problem (13) is large-scale because of the size of the original ODE (12). Therefore we have to pay special attention to its efficient numerical solution. As in [6], we apply a simplified Newton method of the form

$$\begin{aligned}
&\mathbf{X}_{l1}^{2,\ell} = a_{11}^2 \bar{\mathbf{X}}_l + a_{12}^2 \mathbf{X}_{l+1}^{\ell} + \tau_l \big[d_{11}^2 \mathcal{F}(\bar{\mathbf{X}}_l) + d_{12}^2 \mathcal{F}(\mathbf{X}_{l+1}^{\ell}) \big], \\
&\mathbf{X}_{l2}^{2,\ell} = a_{21}^2 \bar{\mathbf{X}}_l + a_{22}^2 \mathbf{X}_{l+1}^{\ell} + \tau_l \big[d_{21}^2 \mathcal{F}(\bar{\mathbf{X}}_l) + d_{22}^2 \mathcal{F}(\mathbf{X}_{l+1}^{\ell}) \big], \\
&\mathbf{X}_{l1}^{3,\ell} = a_{11}^3 \bar{\mathbf{X}}_l + a_{12}^3 \mathbf{X}_{l+1}^{\ell} + \tau_l \big[d_{11}^3 \mathcal{F}(\bar{\mathbf{X}}_l) + d_{12}^3 \mathcal{F}(\mathbf{X}_{l+1}^{\ell}) + d_{13}^3 \mathcal{F}(\mathbf{X}_{l1}^{2,\ell}) + d_{14}^3 \mathcal{F}(\mathbf{X}_{l2}^{2,\ell}) \big], \\
&\mathbf{X}_{l2}^{3,\ell} = a_{21}^3 \bar{\mathbf{X}}_l + a_{22}^3 \mathbf{X}_{l+1}^{\ell} + \tau_l \big[d_{21}^3 \mathcal{F}(\bar{\mathbf{X}}_l) + d_{22}^3 \mathcal{F}(\mathbf{X}_{l+1}^{\ell}) + d_{23}^3 \mathcal{F}(\mathbf{X}_{l1}^{2,\ell}) + d_{24}^3 \mathcal{F}(\mathbf{X}_{l2}^{2,\ell}) \big], \\
&\mathbf{X}_{l3}^{3,\ell} = a_{31}^3 \bar{\mathbf{X}}_l + a_{32}^3 \mathbf{X}_{l+1}^{\ell} + \tau_l \big[d_{31}^3 \mathcal{F}(\bar{\mathbf{X}}_l) + d_{32}^3 \mathcal{F}(\mathbf{X}_{l+1}^{\ell}) + d_{33}^3 \mathcal{F}(\mathbf{X}_{l1}^{2,\ell}) + d_{34}^3 \mathcal{F}(\mathbf{X}_{l2}^{2,\ell}) \big], \\
&\Big[I_{2n^2+n} - \frac{\tau_l}{6} \partial \mathcal{F}(\bar{\mathbf{X}}_l) \Big]^3 \big(\mathbf{X}_{l+1}^{\ell+1} - \mathbf{X}_{l+1}^{\ell} \big) = \bar{\mathbf{X}}_l - \mathbf{X}_{l+1}^{\ell} + \tau_l \sum_{i=1}^{3} b_i \mathcal{F}(\mathbf{X}_{li}^{3,\ell}), \ \ell = 0, 1, \ldots, \quad (14)
\end{aligned}$$

with the trivial initial guess $\mathbf{X}_{l+1}^0 = \bar{\mathbf{X}}_l$, but make important alterations to the earlier published iteration and to its stopping criterion as well.

First, we remark that one iterate of the simplified Newton method (14) implies that the single linear system of size $2n^2 + n$, whose coefficient matrix depends of the Jacobian $\partial \mathcal{F}(\bar{\mathbf{X}}_l)$ evaluated at the vector $\bar{\mathbf{X}}_l$ computed in the previous step of our method, is to be only solved. Moreover, in order to further simplify this iteration, we evaluate partially the Jacobian of the right-hand side function in ODE (12), i.e. without the second summand on the right-hand side of Eq. (11). More formally, we calculate the following matrix:

$$\partial \mathcal{F}(\bar{\mathbf{X}}_l) := \mathbf{1} \otimes \big[w_0^{(m)} J(\bar{\mathbf{X}}_{1:n,l}), \ldots, w_{2n}^{(m)} J(\bar{\mathbf{X}}_{2n^2+1:2n^2+n,l}) \big]$$

where $\mathbf{1}$ stands for the $(2n+1)$-dimensional unitary column-vector, the coefficients $w_i^{(m)}$, $i = 0, 1, \ldots, 2n$, are the weights from formula (4), $J(\cdot) := dF(\cdot)/dX(t)$ means the Jacobian of the drift function in SDE (1) evaluated at the corresponding vector, the first subscript $i : j$ implies that only the entries of the vector $\bar{\mathbf{X}}_l$ with indexes from i to j are taken, inclusively, and $\otimes$ denotes the Kronecker tensor product (see [29] for the definition and properties of the mentioned product; this product is also coded as the built-in function `kron` in Matlab).

Second, to amend the iteration presented in [6] and to make it more accurate and reliable, we replace the fixed number of iterates recommended in the cited paper with a variable number of iteration steps in method (14) determined automatically. In other words, we have to solve two tasks. The first one is to ensure that the output numerical solution $\bar{\mathbf{X}}_{l+1}$ is convergent of order 6 as the step size $\tau_l \to 0$. Our second intention is to provide some evidence that the iterative error of the simplified Newton iteration (14) is negligible in comparison to the discretization error of the NIRK formula (13). To achieve both goals, the following computation is performed. We make the first three iteration steps without

any error control. This is to ensure that the calculated numerical solution is convergent of order 6. Then, we begin monitoring the residual of the fourth iterate and so on. Thus, having computed the next iterate $\mathbf{X}_{l+1}^{\ell}$ (with $\ell = 4, 5, \ldots, 13$) we check its accuracy by the stopping criterion

$$\max_{i=1,\ldots,2n^2+n} \left\{ |\mathbf{X}_{i,l+1}^{\ell} - \mathbf{X}_{i,l+1}^{\ell-1}| / (1 + |\mathbf{X}_{i,l+1}^{\ell}|) \right\} \leq 0.1\epsilon_{loc} \tag{15}$$

where the first subscript i means the i-th entry in the corresponding vector and ϵ_{loc} is the upper bound for tolerated scaled local errors, which are discussed below. The stopping criterion (15) means that the scaled residual of iteration (14) is to be 10 times smaller than the maximum tolerated scaled local error of the discretization method (13). If this criterion holds for some iterate $\mathbf{X}_{l+1}^{\ell}$ with any $\ell = 4, 5, \ldots, 13$ we will stop our iteration and return the output solution $\bar{\mathbf{X}}_{l+1} = \mathbf{X}_{l+1}^{\ell}$ with the first $\ell \geq 4$ for which condition (15) holds. However, if the stopping criterion (15) is not satisfied for 10 successive iterates $\mathbf{X}_{l+1}^{\ell}$ the output solution $\bar{\mathbf{X}}_{l+1} = \mathbf{X}_{l+1}^{13}$ will be accepted as the outcome of the simplified Newton method (14) and, then, our local and global error controls will decide whether this solution is good enough.

Third, it is also important for efficiency of our ACD-UKF technique that the coefficient matrix $[I_{2n^2+n} - \tau_l \partial \mathcal{F}(\bar{\mathbf{X}}_l)/6]^3$ of the linear system in iteration (14) is not fully evaluated and the solution process is fulfilled as explained in [6].

Further, we elaborate the error control mechanism and the automatic mesh generation algorithm implemented in our ACD-UKF. We begin with evaluation of the scaled local and global errors of the discretization method (13).

The *local error* vector $\mathbf{le}_{l+1}$ associated with the output iterate $\bar{\mathbf{X}}_{l+1}$ of iteration (14) is evaluated by the formula

$$\mathbf{le}_{l+1} = \frac{\tau_l}{3}\Big[\frac{5}{6}\mathcal{F}(\bar{\mathbf{X}}_{l1}^3) - \frac{1}{2}\mathcal{F}(\bar{\mathbf{X}}_l) - \frac{2}{3}\mathcal{F}(\bar{\mathbf{X}}_{l2}^3) - \frac{1}{2}\mathcal{F}(\bar{\mathbf{X}}_{l+1}) + \frac{5}{6}\mathcal{F}(\bar{\mathbf{X}}_{l3}^3)\Big], \tag{16}$$

where the notation $\bar{\mathbf{X}}_{li}^3$ stands for the approximation to the third-level stage value $\mathbf{X}_{li}^3$, $i = 1, 2, 3$, calculated by the corresponding formulas of iteration (14) for the output solution $\bar{\mathbf{X}}_{l+1}$, and $\mathcal{F}(\cdot)$ implies the right-hand side of ODE (12). The absolute local error (16) is then scaled as follows:

$$|\mathbf{le}_{l+1}|_{sc} := \max_{i=1,2,\ldots,2n^2+n} \left\{ |\mathbf{le}_{i,l+1}| / (|\bar{\mathbf{X}}_{i,l+1}| + 1) \right\} \tag{17}$$

where the first subscript i stands for the i-th entry in the vector and n is the size of the given SDE model (1). The error $|\mathbf{le}_{l+1}|_{sc}$ is referred to as the *scaled local error* at the mesh node t_{l+1}.

It is known in literature that the notion of local error is weakly related to the actual accuracy of numerical integration of ODE (12) and insufficient for representing the magnitude of the true discretization error committed by the method (14). So one has to evaluate additionally the so-called *global error* [20, 30–33]. In this paper, we utilize the cheap technology designed for the embedded pair `NIRK6(4)` in [20]. For that, we merely sum up the

local error estimates (16). More formally, if we denote the global (true) error at the mesh node t_{l+1} by $\Delta\bar{\mathbf{X}}_{l+1}$ then the evaluation formula will be the following one:

$$\Delta\bar{\mathbf{X}}_{l+1} = \Delta\bar{\mathbf{X}}_l + \mathbf{le}_{l+1} \tag{18}$$

where the initial integration error $\Delta\bar{\mathbf{X}}_0$ is set to be zero. Again, the absolute global error (18) is scaled as follows:

$$|\Delta\bar{\mathbf{X}}_{l+1}|_{sc} := \max_{i=1,2,\ldots,2n^2+n}\big\{|\Delta\bar{\mathbf{X}}_{i,l+1}|/(|\bar{\mathbf{X}}_{i,l+1}|+1)\big\}. \tag{19}$$

The error $|\Delta\bar{\mathbf{X}}_{l+1}|_{sc}$ is referred to as the *scaled global error* at the mesh node t_{l+1}.

Having evaluated the scaled local and global errors we further need an algorithm of stepsize selection for adjusting the estimated errors to user-supplied accuracy requests in automatic mode. This task can be attacked by [20, Algorithm 3.2]. However, the cited algorithm is to be implemented for the scaled error estimates (17) and (19), below. Moreover, we take into account that, within the ACD-UKF method, one demands the accurate numerical solution at the end point of the integration interval $[t_{k-1}, t_k]$, only, i.e. at the next sampling time t_k. So the published stepsize selection is amended by relaxing its accuracy condition. Eventually, we arrive at the new state estimator presented as follows.

Initialization. Given the initial mean $\bar{X}_0$ and covariance Π_0, set $P_{0|0} := \Pi_0$, $\hat{X}_{0|0} := \bar{X}_0$ and $\epsilon_g := 10^{-4}$.

Loop Body. For $k := 1, 2, \ldots, K$, where K is the number of sampling instants in the integration interval of SDE (1), do:

Time Update: Given $\hat{X}_{k-1|k-1}$ and $P_{k-1|k-1}$ at time t_{k-1}, compute the predicted state mean $\hat{X}_{k|k-1}$ and covariance matrix $P_{k|k-1}$ at time t_k. For that, we apply the Cholesky decomposition $P_{k-1|k-1} = S_{k-1}S_{k-1}^\top$ and determine the sigma-points $\mathcal{X}_i(t_{k-1})$, $i = 0, 1, \ldots, 2n$, by formulas (3) at time t_{k-1}. Then, we form the $(2n^2+n)$-dimensional initial vector $\mathbf{X}_{k-1} := [X_0^\top(t_{k-1}), \ldots, X_{2n}^\top(t_{k-1})]^\top$ and set $\epsilon_{loc} := \epsilon_g^{5/4}$, $\tau_{max} := 0.1$, $M := 1$, and perform the following calculation:

1) While $M = 1$ do;
2) $l := 0$, $M := 0$, $\tau_0 := \min\{0.001, \delta\}$, $t_0 := t_{k-1}$, $\bar{\mathbf{X}}_0 := \mathbf{X}_{k-1}$, $\Delta\bar{\mathbf{X}}_0 := 0$, $|\Delta\bar{\mathbf{X}}|_{max} := 0$;
3) While $(t_l < t_k)$ & $(|\Delta\bar{\mathbf{X}}_l|_{sc} \le 100\epsilon_g)$ do;
4) $t_{l+1} := t_l + \tau_l$, compute $\bar{\mathbf{X}}_{l+1}$, $\bar{\mathbf{X}}^3_{l1}$, $\bar{\mathbf{X}}^3_{l2}$, $\bar{\mathbf{X}}^3_{l3}$ by (14), (15) and $|\mathbf{le}_{l+1}|_{sc}$ by (16), (17);
5) $\tau_l^* := \min\Big\{1.5, 0.8\big(\epsilon_{loc}/|\mathbf{le}_{l+1}|_{sc}\big)^{1/5}\Big\}\tau_l$;
6) If $|\mathbf{le}_{k+1}|_{sc} > \epsilon_{loc}$,
then $\tau_l := \tau_l^*$;
else do;
7) Evaluate the scaled global error $|\Delta\bar{\mathbf{X}}_{l+1}|_{sc}$ by (18), (19);
8) $|\Delta\bar{\mathbf{X}}|_{max} := \max\{|\Delta\bar{\mathbf{X}}|_{max}, |\Delta\bar{\mathbf{X}}_{l+1}|_{sc}\}$;
9) $\tau_{l+1} := \min\{\tau_l^*, t_k - t_{l+1}, \tau_{max}\}$;
10) $l := l + 1$;
end{else};

end{while};
11) If $|\Delta\bar{\mathbf{X}}_l|_{sc} > \epsilon_g$,
then $M := 1$;
12) If $M = 1$,
then $\epsilon_{loc} := \big(0.8\epsilon_g/|\Delta\bar{\mathbf{X}}|_{max}\big)^{5/4}\epsilon_{loc}$;
end{while};
13) Stop.

Having calculated the solution vector $\bar{\mathbf{X}}_L$ containing all sigma-points $\mathcal{X}_i(t_L)$, $i = 0, 1, \ldots, 2n$, evaluated at the time t_L, where the subscript L marks the last node in the generated mesh $\{t_l\}_{l=0}^{L}$ (i.e. $t_L \equiv t_k$), we find the predicted state expectation and covariance matrix by means of formulas (3), i.e. $\hat{X}_{k|k-1} = \mathcal{X}_0(t_L)$ and $S_{i,k} = [\mathcal{X}_i(t_L) - \mathcal{X}_0(t_L)]/\sqrt{3}$, $i = 1, \ldots, n$, where $S_{i,k}$ stands for the i-th column in the lower triangular matrix S_k, which is the Cholesky factor of the matrix $P_{k|k-1}$. Thus, $P_{k|k-1} = S_k S_k^{\top}$.

Measurement Update: Having derived the predicted state mean $\hat{X}_{k|k-1}$, the predicted covariance matrix $P_{k|k-1}$ and the sigma-vectors $\mathcal{X}_i(t_k)$, $i = 0, 1, \ldots, 2n$, at the sampling time t_k, we form the sigma-matrix $\mathcal{X}(t_k) := [\mathcal{X}_0(t_k), \ldots, \mathcal{X}_{2n}(t_k)]$ of size $n \times (2n+1)$ and proceed as follows:

1. The sigma-matrix $\mathcal{X}(t_k)$ calculated in the time update is transformed then by the function $h(\cdot)$ of Eq. (2). This results in the $(n \times (2n+1))$-matrix
$$\mathcal{Z}(t_k) := h\big(\mathcal{X}(t_k)\big) := \big[h\big(\mathcal{X}_0(t_k)\big), \ldots, h\big(\mathcal{X}_{2n}(t_k)\big)\big]. \tag{20}$$
2. Therefore the predicted measurement mean is given by
$$\hat{Z}_{k|k-1} := \mathcal{Z}(t_k) W_m \tag{21}$$
and the innovations covariance matrix obeys the formula
$$P_{zz,k|k-1} := \mathcal{Z}(t_k)\mathcal{W}\mathcal{Z}^{\top}(t_k) + R_k. \tag{22}$$
We recall that the matrix R_k means the covariance of the measurement noise V_k in Eq. (2) and the coefficient vector W_m and the matrix $\mathcal{W}$ are defined by formulas (9) and (10), respectively.
3. Next, we evaluate the cross-covariance matrix
$$P_{xz,k|k-1} := \mathcal{X}(t_k)\mathcal{W}\mathcal{Z}^{\top}(t_k) \tag{23}$$
and estimate the continuous-discrete Kalman gain
$$\mathbf{W}_k := P_{xz,k|k-1} P_{zz,k|k-1}^{-1}. \tag{24}$$
4. Finally, the filtering state and covariance matrix are calculated at the sampling time t_k by the following formulas:
$$\hat{X}_{k|k} := \hat{X}_{k|k-1} + \mathbf{W}_k(Z_k - \hat{Z}_{k|k-1}), \tag{25}$$
$$P_{k|k} := P_{k|k-1} + \mathbf{W}_k P_{zz,k|k-1} \mathbf{W}_k^{\top}. \tag{26}$$

In the above ACD-UKF algorithm, the required accuracy is set to be $\epsilon_g := 10^{-4}$ for computing the sigma-points $\mathcal{X}_i(t_L)$, $i = 0, 1, \ldots, 2n$, by the embedded pair `NIRK6(4)` at the last node t_L of the generated mesh. So it may be altered by the user. We also remark that the variable $|\Delta\bar{\mathbf{X}}|_{max}$ evaluates the maximum scaled global error estimate (19) accumulated in the current numerical integration run.

2.2. Accurate continuous-discrete extended Kalman filter

This section is devoted to a revision of the `NIRK6(4)M2`-based *Accurate Continuous-Discrete Extended Kalman Filter* (ACD-EKF) derived originally in [7]. It outlines briefly the cited state estimator and focuses only on particulars amended in the utilized iteration and global error control mechanism, as those implemented in the ACD-UKF. These alterations are intended for boosting the efficiency of state estimation with our revised ACD-EKF.

At the heart of any ACD-EKF method are the *Moment Differential Equations* (MDEs) written for SDE (1) in the following form:

$$\frac{d\hat{X}(t)}{dt} = F\big(\hat{X}(t)\big), \tag{27}$$

$$\frac{dP(t)}{dt} = J\big(\hat{X}(t)\big)P(t) + P(t)J^{\top}\big(\hat{X}(t)\big) + Q \tag{28}$$

where $F(\cdot)$ is the drift function of the process equation (1), $Q := GG^{\top}$, $\hat{X}(t)$ stands for the predicted state mean at time t and the Jacobian $J\big(\hat{X}(t)\big) := dF\big(\hat{X}(t)\big)/d\hat{X}(t)$ is evaluated at the estimated state trajectory. The MDEs (27), (28) are to be integrated in the sampling period $[t_{k-1}, t_k]$ and with the initial values $\hat{X}(t_{k-1}) = \hat{X}_{k-1|k-1}$, $P(t_{k-1}) = P_{k-1|k-1}$, which are the filtering solution at time t_{k-1}, for finding the predicted state expectation and covariance matrix at the next sampling instant t_k. More formally, their predicted values are taken to be $\hat{X}_{k|k-1} = \hat{X}(t_k)$ and $P_{k|k-1} = P(t_k)$.

The main task related to the formulated MDEs is their accurate numerical solution because the accuracy of numerical integration of Eqs. (27), (28) feeds into the quality of practical state estimation within the ACD-EKF method. As in the ACD-UKF algorithm presented in Section 2.1, we fix first a mesh $\{t_l\}_{l=0}^{L}$ in the sampling interval $[t_{k-1}, t_k]$ and apply then the Gauss-type NIRK formula of order 6 in [21, 28] for yielding the corresponding nonlinear problem at each mesh node. To our benefit and in contrast to the MDEs (7), (8) arisen within the ACD-UKF technique, we see that the MDEs (27), (28) can be easily decoupled, i.e. the MDE (27) does not depend on Eq. (28), and, hence, can be solved, separately. Now taking into account that only the first equation in these MDEs is nonlinear and, hence, is to be treated with special care, we apply the mentioned discretization method

for solving it, numerically, and arrive at the discrete-time nonlinear problem of the form

$$\begin{aligned}
\hat{X}^2_{l1} &= a^2_{11}\hat{X}_l + a^2_{12}\hat{X}_{l+1} + \tau_l\big[d^2_{11}F(\hat{X}_l) + d^2_{12}F(\hat{X}_{l+1})\big],\\
\hat{X}^2_{l2} &= a^2_{21}\hat{X}_l + a^2_{22}\hat{X}_{l+1} + \tau_l\big[d^2_{21}F(\hat{X}_l) + d^2_{22}F(\hat{X}_{l+1})\big],\\
\hat{X}^3_{l1} &= a^3_{11}\hat{X}_l + a^3_{12}\hat{X}_{l+1} + \tau_l\big[d^3_{11}F(\hat{X}_l) + d^3_{12}F(\hat{X}_{l+1}) + d^3_{13}F(\hat{X}^2_{l1}) + d^3_{14}F(\hat{X}^2_{l2})\big],\\
\hat{X}^3_{l2} &= a^3_{21}\hat{X}_l + a^3_{22}\hat{X}_{l+1} + \tau_l\big[d^3_{21}F(\hat{X}_l) + d^3_{22}F(\hat{X}_{l+1}) + d^3_{23}F(\hat{X}^2_{l1}) + d^3_{24}F(\hat{X}^2_{l2})\big],\\
\hat{X}^3_{l3} &= a^3_{31}\hat{X}_l + a^3_{32}\hat{X}_{l+1} + \tau_l\big[d^3_{31}F(\hat{X}_l) + d^3_{32}F(\hat{X}_{l+1}) + d^3_{33}F(\hat{X}^2_{l1}) + d^3_{34}F(\hat{X}^2_{l2})\big],\\
\hat{X}_{l+1} &= \hat{X}_l + \tau_l\big[b_1F(\hat{X}^3_{l1}) + b_2F(\hat{X}^3_{l2}) + b_3F(\hat{X}^3_{l3})\big],
\end{aligned} \tag{29}$$

which should be iterated for an approximate state expectation $\hat{X}_{l+1}$ at each node of the prefixed mesh $\{t_l\}_{l=0}^L$. In formula (29), $\tau_l := t_{l+1} - t_l$ means the current step size of this mesh and the numerical values of all constant coefficients of method (29) are given after formula (13). Then, the discrete-time nonlinear problem (29) is iterated numerically by the simplified Newton method

$$\begin{aligned}
\hat{X}^{2,\ell}_{l1} &= a^2_{11}\hat{X}_l + a^2_{12}\hat{X}^\ell_{l+1} + \tau_l\big[d^2_{11}F(\hat{X}_l) + d^2_{12}F(\hat{X}^\ell_{l+1})\big],\\
\hat{X}^{2,\ell}_{l2} &= a^2_{21}\hat{X}_l + a^2_{22}\hat{X}^\ell_{l+1} + \tau_l\big[d^2_{21}F(\hat{X}_l) + d^2_{22}F(\hat{X}^\ell_{l+1})\big],\\
\hat{X}^{3,\ell}_{l1} &= a^3_{11}\hat{X}_l + a^3_{12}\hat{X}^\ell_{l+1} + \tau_l\big[d^3_{11}F(\hat{X}_l) + d^3_{12}F(\hat{X}^\ell_{l+1}) + d^3_{13}F(\hat{X}^{2,\ell}_{l1}) + d^3_{14}F(\hat{X}^{2,\ell}_{l2})\big],\\
\hat{X}^{3,\ell}_{l2} &= a^3_{21}\hat{X}_l + a^3_{22}\hat{X}^\ell_{l+1} + \tau_l\big[d^3_{21}F(\hat{X}_l) + d^3_{22}F(\hat{X}^\ell_{l+1}) + d^3_{23}F(\hat{X}^{2,\ell}_{l1}) + d^3_{24}F(\hat{X}^{2,\ell}_{l2})\big],\\
\hat{X}^{3,\ell}_{l3} &= a^3_{31}\hat{X}_l + a^3_{32}\hat{X}^\ell_{l+1} + \tau_l\big[d^3_{31}F(\hat{X}_l) + d^3_{32}F(\hat{X}^\ell_{l+1}) + d^3_{33}F(\hat{X}^{2,\ell}_{l1}) + d^3_{34}F(\hat{X}^{2,\ell}_{l2})\big],\\
\Big[I_n - \frac{\tau_l}{6}J(\hat{X}_l)\Big]^3\big(\hat{X}^{\ell+1}_{l+1} - \hat{X}^\ell_{l+1}\big) &= \hat{X}_l - \hat{X}^\ell_{l+1} + \tau_l\sum_{i=1}^3 b_iF(\hat{X}^{3,\ell}_{li}),\ \ell = 0,1,2,\ldots,
\end{aligned} \tag{30}$$

at every mesh point t_l, as explained in Section 2.1. In other words, we perform the first three iteration steps with the initial guess $\hat{X}^0_{l+1} = \hat{X}_l$ and without any error control. This is to ensure that the outcome numerical solution is convergent of order 6. Then, we begin monitoring the residual of the fourth iterate and so on. So, having computed the next iterate $\hat{X}^\ell_{l+1}$ (with $\ell = 4, 5, \ldots, 13$) we check its accuracy by the stopping criterion

$$\max_{i=1,\ldots,n}\Big\{|\hat{X}^\ell_{i,l+1} - \hat{X}^{\ell-1}_{i,l+1}|/(1+|\hat{X}^\ell_{i,l+1}|)\Big\} \le 0.1\epsilon_{loc} \tag{31}$$

where the first subscript i means the i-th entry in the corresponding vector and ϵ_{loc} stands for the upper bound restricting the tolerated scaled local errors. The stopping criterion (31) implies that the scaled residual of iteration (30) must be 10 times smaller than this upper bound for the tolerated scaled local errors of method (29). If condition (31) holds for some iterate $\hat{X}^\ell_{l+1}$ with any $\ell = 4, 5, \ldots, 13$ we will stop our iteration and return the output solution $\hat{X}_{l+1} = \hat{X}^\ell_{l+1}$ with the first $\ell \ge 4$ for which condition (31) holds. However, if the stopping criterion (31) is not satisfied for 10 successive iterates $\hat{X}^\ell_{l+1}$ the output solution $\hat{X}_{l+1} = \hat{X}^{13}_{l+1}$ will be accepted as the outcome of the simplified Newton method (30) and,

then, our local and global error controls will decide whether this solution is good enough. It is also worthwhile to recall that the maximum efficiency of our filtering method will be achieved if the matrix $[I_n - \tau_l J(\hat{X}_l)/6]^3$ of the linear system in iteration (30) is not fully evaluated and the solution process is organized as explained in [6].

In addition, it should be noted that iteration (30) arisen within the ACD-EKF is much cheaper than iteration (14) arisen within the ACD-UKF because the size of linear problems treated in the first iteration is n, whereas the size of linear problems solved in the second one is $2n^2 + n$. That is why the ACD-EKF is expected to be less time-consuming in comparison to the ACD-UKF. On the other hand, we have to emphasize the higher accuracy of the latter filter as its advantage.

The local and global error control mechanisms implemented in our revised filter are similar to those in the ACD-UKF. The sole difference is the size of evaluated and regulated errors. More formally, the *local error* le_{l+1} associated with the output iterate $\hat{X}_{l+1}$ of iteration (30) is given by the formula

$$le_{l+1} = \frac{\tau_l}{3}\Big[\frac{5}{6}F(\hat{X}^3_{l1}) - \frac{1}{2}F(\hat{X}_l) - \frac{2}{3}F(\hat{X}^3_{l2}) - \frac{1}{2}F(\hat{X}_{l+1}) + \frac{5}{6}F(\hat{X}^3_{l3})\Big], \tag{32}$$

where the notation $\hat{X}^3_{li}$ stands for the approximation to the third-level stage value $\hat{X}^3_{li}$, $i = 1, 2, 3$, computed by the corresponding formulas of iteration (30) for the output solution $\hat{X}_{l+1}$, and $F(\cdot)$ implies the right-hand side of MDE (27). The absolute local error (27) is then scaled as follows:

$$|le_{l+1}|_{sc} := \max_{i=1,2,\ldots,n}\big\{|le_{i,l+1}|/(|\hat{X}_{i,l+1}| + 1)\big\}. \tag{33}$$

The subscript i in (33) stands for the i-th entry in each vector and n is the size of the given SDE (1). The error $|le_{l+1}|_{sc}$ is referred to as the *scaled local error* of our ACD-EKF method at time t_{l+1}.

As in the ACD-UKF, the global (or true) error of the revised filter is estimated on the generated mesh $\{t_l\}_{l=0}^{L}$ by the formula

$$\Delta\hat{X}_{l+1} = \Delta\hat{X}_l + le_{l+1} \tag{34}$$

where the notation $\Delta\hat{X}_{l+1}$ stands for the absolute global error evaluated at time t_{l+1}, and the initial integration error $\Delta\hat{X}_0$ is set to be zero. Then, the global error (34) is scaled as follows:

$$|\Delta\hat{X}_{l+1}|_{sc} := \max_{i=1,2,\ldots,n}\big\{|\Delta\hat{X}_{i,l+1}|/(|\hat{X}_{i,l+1}| + 1)\big\} \tag{35}$$

where, as customary, the first subscript i means the i-th entry in the corresponding vector. The error (35) is referred to as the *scaled global error* of the ACD-EKF at the mesh node t_{l+1}.

Next, having integrated accurately the state expectation equation (27) at the mesh node t_{l+1}, we further treat numerically the covariance matrix equation (28). For that, we take into

account the linearity of this MDE and apply the corresponding part of Mazzoni's scheme [34], which, for Eq. (28), reads

$$P_{l+1} = M_{l+1/2} P_l M_{l+1/2}^\top + \tau_l K_{l+1/2} Q K_{l+1/2}^\top \tag{36}$$

where $Q := GG^\top$, $t_{l+1/2} := t_l + \tau_l/2$ is the mid-point of the current step, and the variable matrices are evaluated at the mid-time $t_{l+1/2}$ as follows:

$$K_{l+1/2} := \Big[I_n - \frac{\tau_l}{2} J(\hat{X}_{l+1/2})\Big]^{-1}, \quad M_{l+1/2} := K_{l+1/2}\Big[I_n + \frac{\tau_l}{2} J(\hat{X}_{l+1/2})\Big]. \tag{37}$$

In formula (36), we have also taken into account the constant form of the diffusion matrix G in SDE (1). The Jacobian $J(\hat{X}_{l+1/2})$ on the right-hand sides of formulas (37) is evaluated at the mid-point expectation, which is represented with high accuracy by the stage value $\hat{X}_{l2}^3$ in iteration (30) (see [20, Theorem 3.1]). So, we merely accept $\hat{X}_{l+1/2} = \hat{X}_{l2}^3$. In addition, we do not calculate and control the error of the numerical scheme (36) because of the linearity of the covariance matrix equation (28). This saves the execution time of our code, dramatically. We consider that a good mesh $\{t_l\}_{l=0}^L$ is generated in each sampling interval $[t_{k-1}, t_k]$ by the stepsize selection algorithm implemented for treating the nonlinear MDE (27), only. Eventually, we arrive at the updated filter presented as follows.

Initialization. Given the initial mean $\bar{X}_0$ and covariance Π_0, set $P_{0|0} := \Pi_0$, $\hat{X}_{0|0} := \bar{X}_0$ and $\epsilon_g := 10^{-4}$.

Loop Body. For $k := 1, 2, \ldots, K$, where K is the number of sampling instants in the integration interval of SDE (1), do:

Time Update: Given $\hat{X}_{k-1|k-1}$ and $P_{k-1|k-1}$ at time t_{k-1}, compute the predicted state mean $\hat{X}_{k|k-1}$ and covariance matrix $P_{k|k-1}$ at time t_k. For that, we set $\epsilon_{loc} := \epsilon_g^{5/4}$, $\tau_{max} := 0.1$, $M := 1$, and perform the calculation:

1) While $M = 1$ do;
2) $l := 0$, $M := 0$, $\tau_0 := \min\{0.001, \delta\}$, $t_0 := t_{k-1}$, $\hat{X}_0 := \hat{X}_{k-1|k-1}$, $\Delta\hat{X}_0 := 0$, $|\Delta\hat{X}|_{max} := 0$, $P_0 := P_{k-1|k-1}$;
3) While $(t_l < t_k)$ & $(|\Delta\hat{X}_l|_{sc} \le 100\epsilon_g)$ do;
4) $t_{l+1} := t_l + \tau_l$, compute $\hat{X}_{l+1}$, $\hat{X}_{l1}^3$, $\hat{X}_{l2}^3$, $\hat{X}_{l3}^3$ by (30), (31) and $|le_{l+1}|_{sc}$ by (32), (33);
5) $\tau_l^* := \min\Big\{1.5, 0.8\big(\epsilon_{loc}/|le_{l+1}|_{sc}\big)^{1/5}\Big\}\tau_l$;
6) If $|le_{k+1}|_{sc} > \epsilon_{loc}$,
then $\tau_l := \tau_l^*$;
else do;
7) Evaluate the scaled global error $|\Delta\hat{X}_{l+1}|_{sc}$ by (34), (35);
8) $|\Delta\hat{X}|_{max} := \max\{|\Delta\hat{X}|_{max}, |\Delta\hat{X}_{l+1}|_{sc}\}$;
9) Compute the covariance matrix P_{l+1} by (36), (37);
10) $\tau_{l+1} := \min\{\tau_l^*, t_k - t_{l+1}, \tau_{max}\}$;
11) $l := l + 1$;
end{else};

end{while};
12) If $|\Delta\hat{X}_l|_{sc} > \epsilon_g$,
then $M := 1$;
13) If $M = 1$,
then $\epsilon_{loc} := \big(0.8\epsilon_g/|\Delta\hat{X}|_{max}\big)^{5/4}\epsilon_{loc}$;
end{while};
14) Stop.

The numerical solutions $\hat{X}_L$ and P_L, with the subscript L denoting the last node in the generated mesh $\{t_l\}_{l=0}^{L}$ ($t_L \equiv t_k$), are taken as the outcome of the solver `NIRK6(4)M2` applied to MDEs (27), (28) for computing the predicted state mean $\hat{X}_{k|k-1} := \hat{X}_L$ and covariance matrix $P_{k|k-1} := P_L$ with the scaled global error not exceeding ϵ_g.

Measurement Update: Having computed the predicted values $\hat{X}_{k|k-1}$ and $P_{k|k-1}$, we find then the filtering state mean $\hat{X}_{k|k}$ and covariance matrix $P_{k|k}$ at the time instant t_k as follows:

$$R_{e,k} := R_k + H_k P_{k|k-1} H_k^{\top}, \quad K_{f,k} := P_{k|k-1} H_k^{\top} R_{e,k}^{-1}, \tag{38}$$

$$\hat{X}_{k|k} := \hat{X}_{k|k-1} + K_{f,k} e_k, \quad e_k := Z_k - h\big(\hat{X}_{k|k-1}\big), \quad P_{k|k} := P_{k|k-1} - K_{f,k} H_k P_{k|k-1} \tag{39}$$

where the *observation matrix* H_k is derived by linearization of the function $h(\cdot)$ in Eq. (2) around the predicted state mean $\hat{X}_{k|k-1}$, i.e. $H_k := dh(\hat{X}_{k|k-1})/dX_k$, and $e_k \sim \mathcal{N}(0, R_{e,k})$.

2.3. Mixed-type accurate continuous-discrete extended-unscented Kalman filter

Section 2.3 elaborates a new version of the mixed-type *Accurate Continuous-Discrete Extended-Unscented Kalman Filter* (ACD-EUKF) presented in [9] in the first time. In contrast to the cited paper, we devise here a more effective `NIRK6(4)M2`-based state estimator. The idea behind the mixed-type Kalman filtering implies that the time update in such a state estimator is done by the corresponding part of the ACD-EKF whereas the measurement update is conducted with use of the UT. In other words, in the ACD-EKF algorithm discussed in Section 2.2, formulas (38), (39) should be replaced with the UKF measurement update (20)–(26). This allows the accurate and cheap prediction of system's state mean and covariance in the first filter to be unified with the accurate measurement update of the ACD-UKF designed in Section 2.1. All this yields the following ACD-EUKF method.

Initialization. Given the initial mean $\bar{X}_0$ and covariance Π_0, set $P_{0|0} := \Pi_0$, $\hat{X}_{0|0} := \bar{X}_0$ and $\epsilon_g := 10^{-4}$.

Loop Body. For $k := 1, 2, \ldots, K$, where K is the number of sampling instants in the integration interval of SDE (1), do:

Time Update: Given $\hat{X}_{k-1|k-1}$ and $P_{k-1|k-1}$ at time t_{k-1}, compute the predicted state mean $\hat{X}_{k|k-1}$ and covariance matrix $P_{k|k-1}$ at time t_k. For that, we set $\epsilon_{loc} := \epsilon_g^{5/4}$, $\tau_{max} := 0.1$, $M := 1$, and perform the calculation:

1) While $M = 1$ do;
2) $l := 0$, $M := 0$, $\tau_0 := \min\{0.001, \delta\}$, $t_0 := t_{k-1}$, $\hat{X}_0 := \hat{X}_{k-1|k-1}$, $\Delta\hat{X}_0 := 0$, $|\Delta\hat{X}|_{max} := 0$,

$P_0 := P_{k-1|k-1}$;
3) While $(t_l < t_k)$ & $(|\Delta\hat{X}_l|_{sc} \le 100\epsilon_g)$ do;
4) $t_{l+1} := t_l + \tau_l$, compute $\hat{X}_{l+1}$, $\hat{X}^3_{l1}$, $\hat{X}^3_{l2}$, $\hat{X}^3_{l3}$ by (30), (31) and $|le_{l+1}|_{sc}$ by (32), (33);
5) $\tau_l^* := \min\left\{1.5, 0.8\big(\epsilon_{loc}/|le_{l+1}|_{sc}\big)^{1/5}\right\}\tau_l$;
6) If $|le_{k+1}|_{sc} > \epsilon_{loc}$,
then $\tau_l := \tau_l^*$;
else do;
7) Evaluate the scaled global error error $|\Delta\hat{X}_{l+1}|_{sc}$ by (34), (35);
8) $|\Delta\hat{X}|_{max} := \max\{|\Delta\hat{X}|_{max}, |\Delta\hat{X}_{l+1}|_{sc}\}$;
9) Compute the covariance matrix P_{l+1} by (36), (37);
10) $\tau_{l+1} := \min\{\tau_l^*, t_k - t_{l+1}, \tau_{max}\}$;
11) $l := l + 1$;
end{else};
end{while};
12) If $|\Delta\hat{X}_l|_{sc} > \epsilon_g$,
then $M := 1$;
13) If $M = 1$,
then $\epsilon_{loc} := \big(0.8\epsilon_g/|\Delta\hat{X}|_{max}\big)^{5/4}\epsilon_{loc}$;
end{while};
14) Stop.

The numerical solutions $\hat{X}_L$ and P_L, with the subscript L denoting the last node in the generated mesh $\{t_l\}_{l=0}^{L}$ ($t_L \equiv t_k$), are taken as the outcome of the solver `NIRK6(4)M2` applied to MDEs (27), (28) for computing the predicted state mean $\hat{X}_{k|k-1} := \hat{X}_L$ and covariance matrix $P_{k|k-1} := P_L$ with the scaled global error not exceeding ϵ_g.

Measurement Update: Having derived the predicted state mean $\hat{X}_{k|k-1}$ and covariance matrix $P_{k|k-1}$, we proceed as follows.

1. The Cholesky decomposition is applied for factorizing the predicted covariance matrix by the formula
$$P_{k|k-1} = S_{k|k-1}S_{k|k-1}^{\top} \tag{40}$$
where $S_{k|k-1}$ is the lower triangular factor. Then, we calculate the sigma-vectors $\mathcal{X}_{i,k|k-1}$, $i = 0, 1, \ldots, 2n$, by formulas (3) and with use of the square root matrix $S_{k|k-1}$ in decomposition (40). These sigma-vectors $\mathcal{X}_{i,k|k-1}$ constitute the sigma-matrix $\mathcal{X}_{k|k-1} := [\mathcal{X}_{0,k|k-1}, \ldots, \mathcal{X}_{2n,k|k-1}]$ of size $n \times (2n+1)$.
2. The sigma-matrix $\mathcal{X}_{k|k-1}$ is further transformed by the function $h(\cdot)$ of Eq. (2), i.e.
$$\mathcal{Z}_{k|k-1} := h\big(\mathcal{X}_{k|k-1}\big) := \big[h\big(\mathcal{X}_0(t_k)\big), \ldots, h\big(\mathcal{X}_{2n}(t_k)\big)\big]. \tag{41}$$
3. Therefore the predicted measurement mean is given by
$$\hat{Z}_{k|k-1} := \mathcal{Z}_{k|k-1}W_m \tag{42}$$

and the innovations covariance matrix obeys the formula

$$P_{zz,k|k-1} := \mathcal{Z}_{k|k-1}\mathcal{W}\mathcal{Z}_{k|k-1}^{\top} + R_k. \tag{43}$$

We recall that the matrix R_k means the covariance of the measurement noise V_k in Eq. (2) and the coefficient vector W_m and the matrix $\mathcal{W}$ are defined by formulas (9) and (10), respectively.

4. Next, we evaluate the cross-covariance matrix

$$P_{xz,k|k-1} := \mathcal{X}_{k|k-1}\mathcal{W}\mathcal{Z}_{k|k-1}^{\top} \tag{44}$$

and estimate the continuous-discrete Kalman gain

$$\mathbf{W}_k := P_{xz,k|k-1}P_{zz,k|k-1}^{-1}. \tag{45}$$

5. Finally, the filtering state and covariance matrix are calculated at the sampling time t_k as follows:

$$\hat{X}_{k|k} := \hat{X}_{k|k-1} + \mathbf{W}_k(Z_k - \hat{Z}_{k|k-1}), \tag{46}$$

$$P_{k|k} := P_{k|k-1} + \mathbf{W}_k P_{zz,k|k-1}\mathbf{W}_k^{\top}. \tag{47}$$

Notice that the measurement update formulas (40)–(47) are equivalent to those of the ACD-EUKF published in [9]. However, we have utilized the result of Särkkä [3] for presenting our measurement update in the concise and straight forward for a Matlab implementation form in this paper.

3. Simulation results and discussion

Section 3 intends for a numerical examination of the variable-stepsize state estimators designed in Section 2 and for their comparison to earlier developed filters in severe conditions of the air traffic control scenario in [1]. All the methods are coded in Matlab. The filters of Section 2 are abbreviated to `ACD-UKF`, `ACD-EKF6n` and `ACD-EUKF6n`, respectively. The letter "`n`" in the latter two acronyms refers to the revised ACD-EKF and ACD-EUKF estimators presented in Sections 2.2 and 2.3, respectively, whereas the acronyms `ACD-EKF6o` and `ACD-EUKF6o` stand for the old variants published in [7, 9]. However, these earlier ACD-EKF and ACD-EUKF versions are also grounded in the MDE solver `NIRK6(4)M2` and implemented as explained in [6]. They are utilized for observing advantages coming from the implemented new iteration and relaxed global error control mechanism. In addition, we examine two fixed-stepsize filters `CD-UKF256`, `CD-UKF512` and one variable-stepsize filter `CD-UKFm`. The letter "`m`" in the last acronym suggests that this filter is based on the Matlab code `ode45` with the local error tolerance set to be 10^{-12}, i.e. precisely as in [4]. We stress that `ode45` is applied for solving MDEs (7), (8), numerically. The acronym `CD-UKF256` refers to the m-step *Continuous-Discrete Unscented Kalman Filter* (CD-UKF) with $m = 256$ equidistant subdivisions of each sampling interval. It is implemented as described in [7, 9].

The listed filters are applied to tackling the seven-dimensional radar tracking problem, where an aircraft executes a coordinated turn. This problem is described with all particulars in [7, 9]. Below, we follow the cited papers for outlining it briefly.

Aircraft's dynamics (i.e. a coordinated turn in the horizontal plane) obeys the SDE model (1) with the state vector $X(t) := [x(t), \dot{x}(t), y(t), \dot{y}(t), z(t), \dot{z}(t), \omega(t)]^T \in \mathbb{R}^7$, where $x(t)$, $y(t)$, $z(t)$ and $\dot{x}(t)$, $\dot{y}(t)$, $\dot{z}(t)$ stand for its positions and corresponding velocities in the Cartesian coordinates at time t and $\omega(t)$ is the (nearly) constant turn rate, and with the vector-valued drift function $F(\cdot) := [\dot{x}(t), -\omega(t)\dot{y}(t), \dot{y}(t), \omega(t)\dot{x}(t), \dot{z}(t), 0, 0]^T \in \mathbb{R}^7$. The stochastic noise term $W(t)$ is also 7-dimensional with all entries $\{w_i(t),\, t > 0\}$, $i = 1, 2, \ldots, 7$, being mutually independent Brownian processes with zero mean and unit covariance in this SDE model. The latter simulates unpredictable errors due to turbulence, wind force and so on. The driving noise diffusion matrix G is constant, diagonal and given by the formula $G := \mathrm{diag}\{0, \sigma_1, 0, \sigma_1, 0, \sigma_1, \sigma_2\}$ with $\sigma_1 = \sqrt{0.2}$ m/s and $\sigma_2 = 0.007^\circ$/s.

The measurement equation in the air traffic control scenario under consideration is non-linear, discrete-time and fixed by formula (2) with the available measurement information $Z_k := [r_k, \theta_k, \phi_k]^T \in \mathbb{R}^3$ and the utilized measurement function

$$h(\cdot) := \begin{bmatrix} \sqrt{x_k^2 + y_k^2 + z_k^2} \\ \tan^{-1}(y_k/x_k) \\ \tan^{-1}\Big(z_k/\sqrt{x_k^2 + y_k^2}\Big) \end{bmatrix} \in \mathbb{R}^3$$

where the coordinates x_k, y_k, z_k stand for the aircraft's position at time t_k. We emphasize that the used measurement information corresponds to the radar located at the origin. Also, this radar is supposed to be equipped for measuring the range r, the azimuth angle θ and the elevation angle ϕ. In addition, the radar measurements are corrupted by the white noise $V_k \sim \mathcal{N}(0, R)$ with the constant covariance matrix $R := \mathrm{diag}\{\sigma_r^2, \sigma_\theta^2, \sigma_\phi^2\}$ where $\sigma_r = 50$ m, $\sigma_\theta = 0.1^\circ$, $\sigma_\phi = 0.1^\circ$.

In this paper, we simulate the aircraft dynamics for 150 s and consider the most difficult case of angular velocity $\omega = 6^\circ$/s, only. The "true" and filtering aircraft states are calculated for the sampling periods $\delta = 5, 10, \ldots, 30$ s. These "true" states are computed by the Euler-Maruyama method with step size equal to 0.0005. They are used for generating the measurement history and for evaluating the *Accumulated Root Mean Square Errors* (ARMSE) in position, in velocity and in turn rate, denoted as ARMSE_p, ARMSE_v and ARMSE_ω, below. These errors are calculated as follows:

$$\mathrm{ARMSE}_p := \Big[\frac{1}{100K}\sum_{l=1}^{100}\sum_{k=1}^{K}\big(x_{ref,l}(t_k) - \hat{x}_{k|k,l}\big)^2 + \big(y_{ref,l}(t_k) - \hat{y}_{k|k,l}\big)^2 + \big(z_{ref,l}(t_k) - \hat{z}_{k|k,l}\big)^2\Big]^{1/2},$$

$$\mathrm{ARMSE}_v := \Big[\frac{1}{100K}\sum_{l=1}^{100}\sum_{k=1}^{K}\big(\dot{x}_{ref,l}(t_k) - \hat{\dot{x}}_{k|k,l}\big)^2 + \big(\dot{y}_{ref,l}(t_k) - \hat{\dot{y}}_{k|k,l}\big)^2 + \big(\dot{z}_{ref,l}(t_k) - \hat{\dot{z}}_{k|k,l}\big)^2\Big]^{1/2},$$

$$\mathrm{ARMSE}_\omega := \Big[\frac{1}{100K}\sum_{l=1}^{100}\sum_{k=1}^{K}\big(\omega_{ref,l}(t_k) - \hat{\omega}_{k|k,l}\big)^2\Big]^{1/2}$$

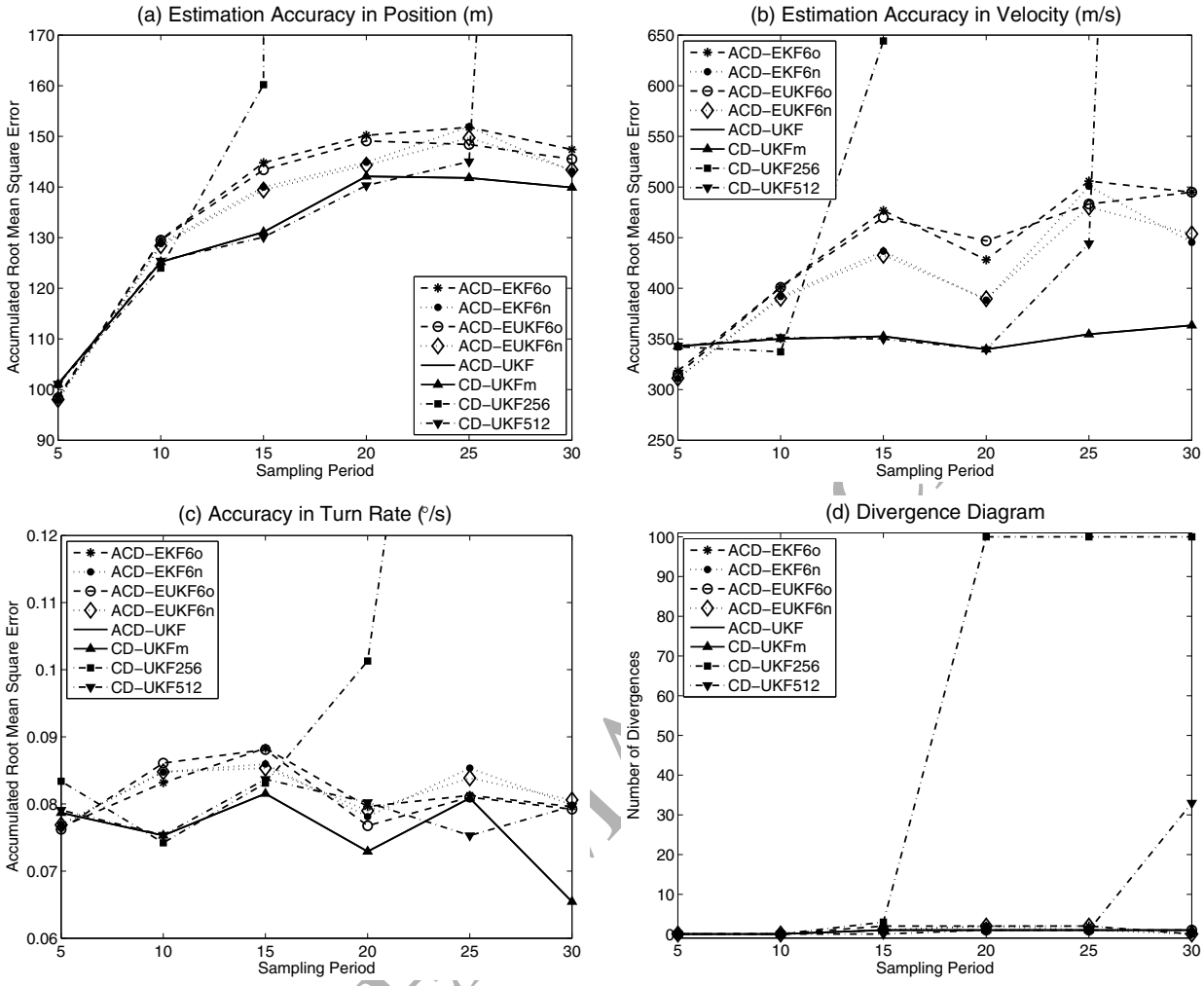

Figure 1: The accuracy comparison of the new and earlier developed filters in the air traffic control scenario when $\omega = 6^\circ/\mathrm{s}$

where the subscript ref refers to the "true" stochastic states (reference solution) computed by the Euler-Maruyama method, l marks the corresponding Monte Carlo run (out of 100 independent simulations), k denotes the particular sampling time t_k and K stands for the total number of sampling instants in the simulation interval $[0, 150]$, depending on the sampling rate δ. In addition, we show a divergence diagram of the filtering methods examined within the above air traffic control scenario out of 100 Monte Carlo runs of the corresponding codes. According to Arasaratnam et al. [1], divergence is declared when the position error

$$\left[(x_{ref,l}(t_k) - \hat{x}_{k|k,l})^2 + (y_{ref,l}(t_k) - \hat{y}_{k|k,l})^2 + (z_{ref,l}(t_k) - \hat{z}_{k|k,l})^2\right]^{1/2} > 500\,\mathrm{m}$$

at any sampling instant $t_k \in [0, 150]$. Further details on calculation of the ARMSE's and divergence numbers under consideration can be found in [7, 9].

Accuracies of all the filters achieved in our air traffic control scenario test as well as the

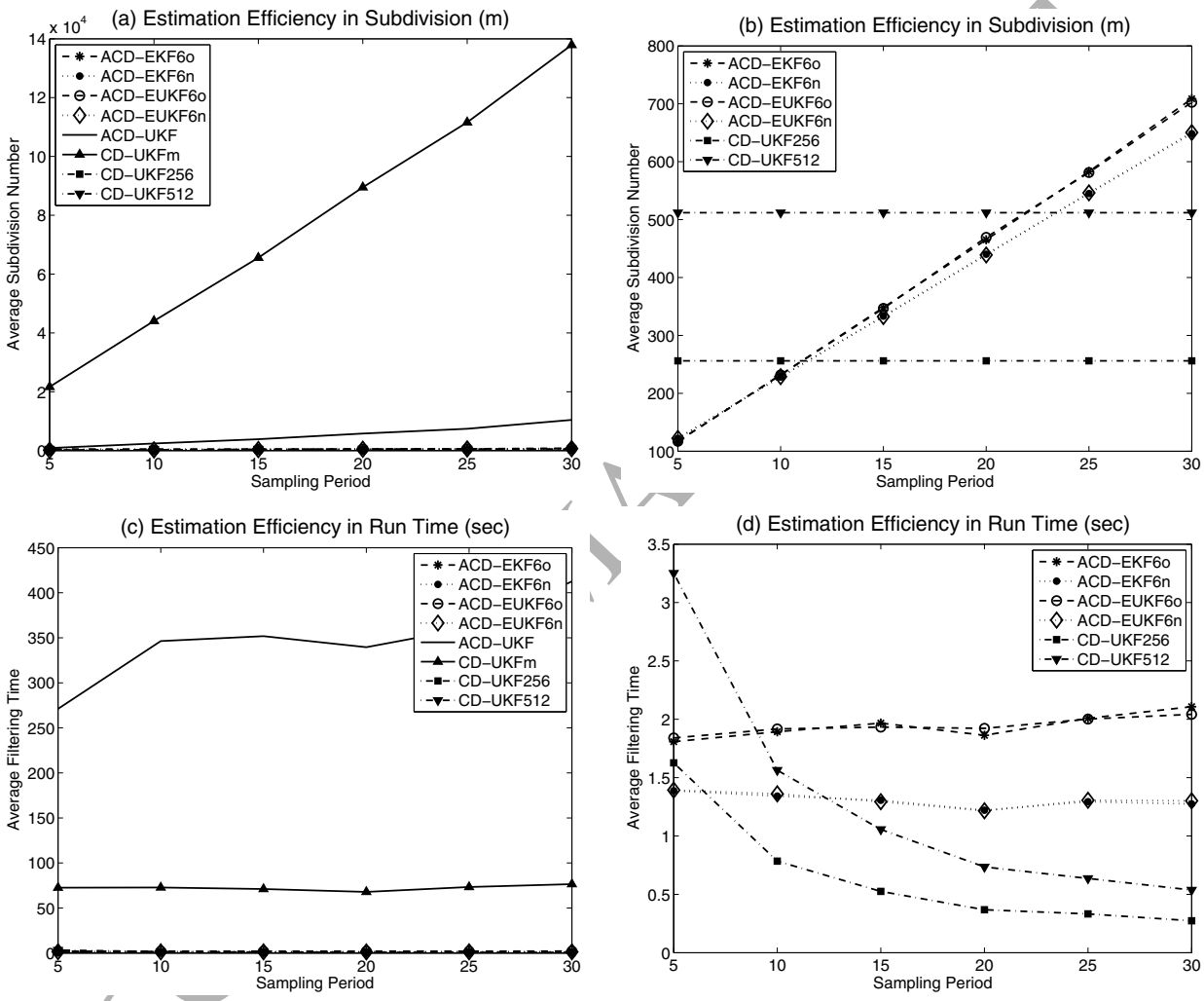

Figure 2: The efficiency comparison of the new and earlier developed filters in the air traffic control scenario when $\omega = 6^\circ/\mathrm{s}$

outcome divergence diagram are exhibited in Fig. 1. The presented plots suggest that the variable-stepsize state estimators can automatically accommodate changes in the length of sampling intervals and work accurately for the δ's considered in our numerical experiment. In contrast, the fixed-stepsize filters `CD-UKF256` and `CD-UKF512` will fulfil well only if the sampling period is short enough. For the first method, it is when $\delta \leq 10$ s. Then, the accuracy of `CD-UKF256` deteriorates at $\delta = 15$ s, and its ARMSE_p and ARMSE_v exceed 10^{+5} for $\delta \geq 20$ s. This state estimator also exhibits the 100% divergence when $\delta \geq 20$ s. The second filter `CD-UKF512` estimates better. It is accurate for any $\delta \leq 25$ s. However, its ARMSE_p and ARMSE_v grow with a factor of 10 at $\delta = 30$ s. The divergence number of `CD-UKF512` also boosts to 33 out of 100 Monte Carlo runs for this longest δ.

As we already said, all the variable-stepsize filters estimate rather accurately the state in the air traffic control scenario under consideration (see Fig. 1). However, it is observed clearly that, in general, the minimum errors are achieved with `ACD-UKF` and `CD-UKFm`, especially for the longer waiting times. Besides, the fixed-stepsize filters `CD-UKF256` and `CD-UKF512` are equally accurate but only when their discretization errors are negligible and do not compromise the accuracy of state estimation, i.e. when the sampling periods are rather short. All this is evident and explained by the fact that the UKF-based state estimators provide the third order approximation to the conditional mean of the propagated Gaussian distribution, whereas the EKF-based methods enjoy only the first order approximation of the filtering expectation (see, for instance, [18, 27]). On the other hand, the errors committed by the EKF-based state estimators are reasonable and their sole ARMSE_v may deteriorate with a factor of about 1.5 for the longer sampling periods. Within this EKF-based group of filters, we see that `ACD-EKF6n`, `ACD-EUKF6n` evidently outperform their earlier versions `ACD-EKF6o`, `ACD-EUKF6o`, respectively. We also remark that the errors committed by the ACD-EUKF methods are slightly smaller than those of the ACD-EKF state estimators. The latter is in line with the earlier results reported in [9].

Next, we discuss efficiency issues for our filters. More precisely, we look at the *average filtering time* of 100 Monte Carlo runs and at the *average subdivision number* (m) of the sampling interval used by the state estimators under examination. We point out that the average subdivision is equal to the value of m in the m-step CD-UKF, and this number is not predefined and chosen automatically in the remaining variable-stepsize methods. These data are displayed in Fig. 2, where the left-hand plots present the full information and the right-hand diagrams expose the same values but without the run time and committed subdivisions of `ACD-UKF` and `CD-UKFm`. The latter allow the efficiency of the other filters to be clearly assessed. It should also be mentioned that while assessing the average filtering time we look at the relative values, only. In other words, we are interested in a qualitative analysis and ignore the absolute time expenditures. This is because of Matlab calculations being very slow and not competitive to precompiled codes.

The efficiency in subdivision plots says that the largest amounts of steps are committed by `CD-UKFm`. This is because of the extremely stringent accuracy condition accepted in this filter. Then, it is followed by `ACD-UKF` (see Fig. 2(a)). The latter is also evident since the size of arisen ODE (12) is 105 in the considered radar tracking scenario and the method's step size is regulated by the maximum error committed in all entries of the state vector.

All the EKF-based state estimators regulate the step size by the maximum error committed in the predicted expectation vector of size 7. That is why they fulfil much smaller steps while solving the corresponding MDEs in each sampling interval. Furthermore, the updated ACD-EKF and ACD-EUKF methods also utilize smaller numbers of steps in comparison to those of their old versions (see Fig. 2(b)) and, hence, demand less execution time (see Fig. 2(d)).

Eventually, we conclude that `ACD-UKF` and `CD-UKFm` are the most accurate but time-consuming filters with some advantage of the second one (see Fig. 2(c)). This is because of the explicit fashion of the Matlab code `ode45` implemented therein. Thus, our future task is to raise efficiency of `ACD-UKF` with use of cheaper ODE solvers, but with global error control, especially those grounded in Runge-Kutta, general linear and peer methods published recently in [31–33, 35–44].

4. Concluding remarks

This paper has presented an extension of the accurate Gaussian filtering technique devised recently in [6, 7, 19] towards efficient UKF-based state estimators for treating continuous-time nonlinear stochastic systems with discrete measurements arising in target tracking modeling. Our idea of the ACD-UKF method is grounded in accurate numerical solutions of the differential equations which describe the propagation of the sigma-vectors of the unscented transform in time and are derived by Särkkä [3]. The latter filter has been implemented by means of the Gauss-type NIRK formula of orders 6 with automatic scaled local and global error controls. This ODE solver, which originates from [20, 21, 28], has been modified for calculating the requested sigma-points with negligible errors. The implemented automatic numerical integration error control contributes positively to the accuracy of practical state estimation of continuous-discrete stochastic systems arisen in radar tracking theory.

In addition, our research has revised the earlier version of the ACD-EKF method designed and tested extensively in [5–8]. Again, the latter filter is grounded in the same Gauss-type NIRK formula of orders 6 with automatic local and global error control mechanisms utilized for solving the state expectation equation together with a part of Mazzoni's scheme used for treating the covariance matrix equation. Most importantly, this new ACD-EKF variant distinguishes from its earlier version in the number of conducted iteration steps for solving arisen nonlinear problems (i.e. the previous ACD-EKF implementation fulfils strictly four iterates per step of the utilized MDE solver whereas the amended one exploits a variable number of iterates determined at each step of the MDE solver, automatically) and in the utilized error control mechanism. More precisely, our old filters demand the computed state expectation trajectory to be accurate at all nodes of the generated mesh $\{t_l\}_{l=0}^{L}$ in the cited papers. In contrast, the new one requires the preassigned accuracy of numerical integration to hold at the end-point t_L, only, i.e. at the next sampling time t_k. All this has been shown to improve the accuracy and efficiency of practical state estimation in target tracking modeling.

The above-mentioned improvement has also been implemented in the mixed-type Kalman filtering method designed earlier in the UKF framework in [9]. We stress that the idea of the mixed-type filtering is powerful and allows the best features of the accurate continuous-discrete extended and unscented Kalman filters to be unified within the unique method. More precisely, its time updates are done as those in the ACD-EKF whereas the measurement updates are conducted by means of the sufficiently accurate unscented transform of the UKF technique (see [9] for further particulars of this filter).

The efficiency of the novel ACD-UKF, ACD-EKF and ACD-EUKF algorithms has been confirmed numerically within the air traffic control scenario in [1], which is considered to be a challenging test problem for practical examination of nonlinear filters. In particular, our variable-stepsize methods outperform evidently the state-of-the-art fixed-stepsize UKF state estimators [7] in the accuracy of state estimation, especially for sufficiently long sampling periods. Nevertheless, among the devised filtering techniques, the highest state estimation accuracy is achieved with the ACD-UKF method. It is also more robust to error accumulation in comparison to the ACD-EKF and ACD-EUKF methods within the air traffic control scenario with low sampling rates [45].

On the other hand, this filter is quite time-consuming because of the size of the MDEs utilized for propagating the sigma-vectors of the unscented transform in time. The latter MDEs are to be solved accurately in each sampling interval. Thus, a further effort is needed for raising efficiency of such numerical solutions. Finally, our paper shows how advanced ODE solvers with automatic stepsize selection and error control facilities, especially those grounded in Runge-Kutta, general linear and peer methods published recently in [20, 21, 28, 31, 35–44], can contribute to solving difficult problems in applied radar tracking.

Acknowledgements

The authors acknowledge the support from Portuguese National Funds through the *Fundação para a Ciência e a Tecnologia* (FCT) within projects UID/Multi/04621/2013, SFRH/BPD/64397/2009 and the *Investigador FCT 2013* programme. They are also grateful to the anonymous referees for their valuable remarks and comments on the paper.

References

[1] I. Arasaratnam, S. Haykin, T. R. Hurd, Cubature Kalman filtering for continuous-discrete systems: Theory and simulations, IEEE Trans. Signal Process. 58 (10) (2010) 4977–4993.

[2] S. Särkkä, J. Sarmavuori, Gaussian filtering and smoothing for continuous-discrete dynamic systems, Signal Processing 93 (2013) 500–510.

[3] S. Särkkä, On unscented Kalman filter for state estimation of continuous-time nonlinear systems, IEEE Trans. Automat. Contr. 52 (9) (2007) 1631–1641.

[4] B. O. S. Teixeira, M. A. Santillo, R. S. Erwin, D. S. Bernstein, Spacecraft tracking using sampled-data Kalman filters, IEEE Contr. Syst. Mag. 28 (4) (2008) 78–94.

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

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

[7] G. Yu. Kulikov, M. V. Kulikova, The accurate continuous-discrete extended Kalman filter for radar tracking, IEEE Trans. Signal Process. 64 (4) (2016) 948–958.
[8] G. Yu. Kulikov, M. V. Kulikova, Accurate state estimation in the Van der Vusse reaction, in: Proceedings of the 2014 IEEE Multi-Conference on Systems and Control, 2014, pp. 759–764.
[9] M. V. Kulikova, G. Yu. Kulikov, A mixed-type accurate continuous-discrete extended-unscented Kalman filter for target tracking, in: Proceedings of the 2015 European Control Conference, 2015, pp. 2824–2829.
[10] K. J. Aström, Introduction to Stochastic Control Theory, Academic Press, New York, 1970.
[11] A. H. Jazwinski, Stochastic Processes and Filtering Theory, Academic Press, New York, 1970.
[12] J. L. Crassidis, J. L. Junkins, Optimal Estimation of Dynamic Systems, CRC Press LLC, New York, 2004.
[13] Y. Bar-Shalom, X.-R. Li, T. Kirubarajan, Estimation with Applications to Tracking and Navigation, Wiley, New York, 2001.
[14] M. S. Grewal, A. P. Andrews, Kalman Filtering: Theory and Practice, Prentice Hall, New Jersey, 2001.
[15] M. S. Grewal, L. R. Weill, A. P. Andrews, Global Positioning Systems, Inertional Navigation and Integration, Wiley, New York, 2001.
[16] L. Aggoun, R. J. Elliot, Measure Theory and Filtering, Cambridge University Press, Cambridge, U.K., 2005.
[17] F. L. Lewis, Optimal Estimation: with an Introduction to Stochastic Control Theory, John Wiley & Sons, New York, 1986.
[18] S. Särkkä, Bayesian Filtering and Smoothing, Cambridge University Press, Cambridge, U.K., 2013.
[19] G. Yu. Kulikov, M. V. Kulikova, Accurate numerical implementation of the continuous-discrete extended Kalman filter, IEEE Trans. Automat. Contr. 59 (1) (2014) 273–279.
[20] G. Yu. Kulikov, Cheap global error estimation in some Runge-Kutta pairs, IMA J. Numer. Anal. 33 (1) (2013) 136–163.
[21] G. Yu. Kulikov, Embedded symmetric nested implicit Runge-Kutta methods of Gauss and Lobatto types for solving stiff ordinary differential equations and Hamiltonian systems, Comput. Math. Math. Phys. 55 (6) (2015) 983–1003.
[22] S. J. Julier, J. K. Uhlmann, H. F. Durrant-Whyte, A new method for the nonlinear transformation of means and covariances in filters and estimators, IEEE Trans. Automat. Contr. 45 (3) (2000) 477–482.
[23] S. J. Julier, J. K. Uhlmann, Unscented filtering and nonlinear estimation, Proceedings of the IEEE 92 (3) (2004) 401–422.
[24] S. J. Julier, J. K. Uhlmann, Reduced sigma point filters for the propagation of means and covariances through nonlinear transformations, in: Proceedings of the American Control Conference, 2002, pp. 887–892.
[25] H. M. Menegaz, J. Y. Ishihara, G. A. Borges, New minimum sigma set for unscented filtering, Int. J. Robust Nonlinear Control 25 (17) (2015) 3286–3298.
[26] H. M. Menegaz, J. Y. Ishihara, G. A. Borges, A. N. Vargas, A systematization of the unscented Kalman filter theory, IEEE Trans. Automat. Contr. 60 (10) (2015) 2583–2598.
[27] E. A. Wan, R. Van der Merwe, The unscented Kalman filter, in: *S. Haykin ed.* Kalman Filtering and Neural Networks, John Wiley & Sons, Inc., New York, 2001, pp. 221–280.
[28] G. Yu. Kulikov, S. K. Shindin, Adaptive nested implicit Runge-Kutta formulas of Gauss type, Appl. Numer. Math. 59 (3-4) (2009) 707–722.
[29] P. Lancaster, Theory of matrices, Academic Press, New York, 1970.
[30] G. Yu. Kulikov, Global error control in adaptive Nordsieck methods, SIAM J. Sci. Comput. 34 (2) (2012) A839–A860.
[31] G. Yu. Kulikov, R. Weiner, Global error estimation and control in linearly-implicit parallel two-step peer W-methods, J. Comput. Appl. Math. 236 (6) (2011) 1226–1239.
[32] G. Yu. Kulikov, R. Weiner, A singly diagonally implicit two-step peer triple with global error control for stiff ordinary differential equations, SIAM J. Sci. Comput. 37 (3) (2015) A1593–A1613.
[33] R. Weiner, G. Yu. Kulikov, S. Beck, J. Bruder, New third- and fourth-order singly diagonally implicit

two-step peer triples with local and global error controls for solving stiff ordinary differential equations, J. Comput. Appl. Math. 316 (2017) 380–391.

[34] T. Mazzoni, Computational aspects of continuous-discrete extended Kalman filtering, Comput. Statist. 23 (4) (2008) 519–539.

[35] J. C. Butcher, Numerical Methods for Ordinary Differential Equations, John Wiley and Sons, Chichester, 2008.

[36] Z. Jackiewicz, General Linear Methods for Ordinary Differential Equations, John Wiley and Sons, Hoboken, 2009.

[37] G. Yu. Kulikov, R. Weiner, Variable-stepsize interpolating explicit parallel peer methods with inherent global error control, SIAM J. Sci. Comput. 32 (4) (2010) 1695–1723.

[38] S. González-Pinto, D. Hernández-Abreu, J. I. Montijano, An efficient family of strongly A-stable Runge-Kutta collocation methods for stiff systems and DAEs. Part I: Stability and order results, J. Comput. Appl. Math. 234 (2010) 1105–1116.

[39] S. González-Pinto, D. Hernández-Abreu, J. I. Montijano, An efficient family of strongly A-stable Runge-Kutta collocation methods for stiff systems and DAEs. Part II: Convergence results, Appl. Numer. Math. 62 (2012) 1349–1360.

[40] B. A. Schmitt, R. Weiner, Parallel two-step W-methods with peer variables, SIAM J. Numer. Anal. 42 (2004) 265–286.

[41] B. A. Schmitt, R. Weiner, H. Podhaisky, Multi-implicit peer two-step W-methods for parallel time integration, BIT 45 (2005) 197–217.

[42] R. Weiner, B. A. Schmitt, H. Podhaisky, S. Jebens, Superconvergent explicit two-step peer methods, J. Comput. Appl. Math. 223 (2009) 753–764.

[43] R. Weiner, G. Yu. Kulikov, H. Podhaisky, Variable-stepsize doubly quasi-consistent parallel explicit peer methods with global error control, Appl. Numer. Math. 62 (10) (2012) 1591–1603.

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

[45] M. V. Kulikova, G. Yu. Kulikov, On computational robustness of accurate continuous-discrete unscented Kalman filtering for target tracking models, in: Proceedings of the 2016 European Control Conference, 2016, pp. 1129–1134.