Parametric Bayesian Filters for Nonlinear Stochastic Dynamical Systems: A Survey

Paweł Stano*, Zsófia Lendek, Jelmer Braaksma, Robert Babuška, Cees de Keizer and Arnold J. den Dekker

***Abstract*—Nonlinear stochastic dynamical systems are commonly used to model physical processes. For linear and Gaussian systems, the Kalman Filter is optimal in minimum mean squared error sense. However, for nonlinear or non-Gaussian systems the estimation of states or parameters is a challenging problem. Furthermore, it is often required to process data online. Therefore, apart from being accurate, the feasible estimation algorithm also needs to be fast. In this paper we review Bayesian filters which possess the aforementioned properties. Each filter is presented in an easy to implement algorithmic form. We focus on parametric methods, among which we distinguish three types of filters: filters based on analytical approximations (Extended Kalman Filter, Iterated Extended Kalman Filter), filters based on statistical approximations (Unscented Kalman Filter, Central Difference Filter, Gauss-Hermite Filter), and filters based on the Gaussian Sum Approximation (Gaussian Sum Filter). We discuss each of these filters, and compare them on illustrative examples.**

I. INTRODUCTION

The concept of *filtering* has been studied for decades in various engineering problems that require extracting information of interest from an uncertain or changing environment. A filter is a *recursive algorithm* designed for a case where the complete knowledge of the relevant signal characteristics is not available [1]. The main purpose of a filter is to utilize the available information about the process of interest in order to obtain an estimate of certain variables that cannot be measured directly.

In this paper we analyze filters designed for nonlinear discrete-time continuous-state dynamical systems. These are used to model, among others, physical [2], chemical [3], biological [4], or economic [5] processes. Usually, in each of these cases, one is interested in continuous-time phenomena, often governed by (partial) differential equations. However, due to the complexity of these models and the limited computational power available, a number of simplifications are required in order to obtain an efficient solution. Discrete-time systems provide such a simplification since in this framework time is represented by the monotonic set of discrete time steps that allows recursive filtering of the process of interest. Since the discretized system is only an approximation of the original one, there is always a certain degree of uncertainty incorporated into the model, which depends on the discretization technique that was applied [6]. Other possible approximation is done by replacing the detailed deterministic (dynamical) relations with probabilistic approximations which, if appropriately chosen, further simplify the system. However, this comes with the price of increased uncertainty of the model (see [7] and references therein).

The main objective of this paper is to review and discuss the filtering methods that are commonly applied to nonlinear stochastic dynamical systems. Among many techniques dealing with this subject we can distinguish: grid-based methods [8]–[10] designed for dynamical systems defined on finite state space, point-mass methods [11]–[13] that are based on grid approximation of the continuous state space, Beneš-Daum filters [14]–[16] derived for a specific class of nonlinear systems with linear observations, parametric methods [17]–[20], i.e., methods for which the estimation problem has a solution in a finite dimensional parameter space, nonparametric methods based on numerical integrations via Monte Carlo approach such as Particle Filters [9], [21]–[23] or Ensemble Kalman Filters [24]–[27] popular in data assimilation problems, and more [28]–[30]. Throughout the years, each of these approaches lead to a development of a multitude of algorithms. Detailed analysis of such a vast number of estimation techniques is a monumental task. Therefore, in this article, we focus only on the parametric filters. We present filtering algorithms and investigate their properties and their feasibility for online applications.

To help the reader better understand the properties of the filters discussed we analyze their performance in four numerical experiments. For these experiments we use popular systems that have been extensively studied in the literature.

This paper is organized as follows. First, in Section II, we state the general *Bayesian Filtering* (BF) problem. Next, we proceed to the solutions of this problem. We start from the simplest estimation methods that are applicable to simple dynamical systems, and step by step continue to advanced filters suitable for more complex estimation problems. In Section III we present a class of analytical approximators. Section IV deals with a class of statistical approximators. In Section V we discuss filters that are based on Gaussian Sum approximation. Section VI concludes the paper.

II. GENERAL PROBLEM FORMULATION: BAYESIAN FILTERING

In this section we formulate the *Bayesian filtering* (BF) framework for nonlinear and non-Gaussian dynamical systems. This will be used as a basis for each filter presented in the remainder of this paper.

First, let us establish the notation utilized throughout the paper. For each time instant $k = 1, 2, \ldots$ we use:

- bold lower case font to denote random

P. Stano, Zsófia Lendek, R. Babuška and Arnold J. den Dekker are with Delft Center for Systems and Control Delft University of Technology, Mekelweg 2, 2628 CD Delft, The Netherlands, P.M.Stano@tudelft.nl

Zsófia Lendek is also with the Department of Automation, Technical University of Cluj-Napoca, Memorandumului 28, 400114 Cluj-Napoca, Romania.

J. Braaksma and C. de Keizer are with IHC Systems B.V. P.O. Box 41, 3360 AA Sliedrecht, The Netherlands.

variables $(\mathbf{x}_k, \mathbf{y}_k, \mathbf{v}_k, \mathbf{w}_k)$, or functions of random variables $(\mathbf{f}_k, \mathbf{h}_k)$,
- bold capital letters to denote matrices $\left(\mathbf{R}_k, \mathbf{Q}_k, \mathbf{P}_{k|k}, \mathbf{F}_k, \mathbf{H}_k\right)$
- italic font to denote scalars $\left(\omega^i, n, \kappa, q_l, h\right)$,
- bold italic lower case font to denote deterministic vectors $\left(\boldsymbol{u}_k, \hat{\boldsymbol{x}}_k, \boldsymbol{x}_k^i\right)$ or the realizations $(\boldsymbol{y}_k, \boldsymbol{x}_k)$ of the random variables $(\mathbf{y}_k, \mathbf{x}_k)$,
- calligraphic upper case letters to denote deterministic sets, e.g., $\mathcal{Y}_k = \{\boldsymbol{y}_i, i = 1, ..., k\}$.

A. Bayesian Filtering

First, let us define the probabilistic state-space system, which serves as a framework for the BF problem. The probabilistic state-space framework relates the process model describing the evolution of the states in time, the observation model relating the noisy measurements of the system to the actual state, and the initial state of the system. In discrete-time, at each time instant $k = 1, 2, ...$, the probabilistic state-space description is given by

the state model

$$\mathbf{x}_{k+1} = \mathbf{f}_k(\mathbf{x}_k, \mathbf{v}_k), \tag{1}$$

the observation model

$$\mathbf{y}_k = \mathbf{h}_k(\mathbf{x}_k, \mathbf{w}_k), \tag{2}$$

and the initial condition

$$\mathbf{x}_0 \sim p_0, \tag{3}$$

where $\mathbf{x}_k \in \mathbb{R}^n$ and $\mathbf{y}_k \in \mathbb{R}^p$ are random variables corresponding to the state model and the measurement model at time step k, respectively. $\mathbf{v}_k \in \mathbb{R}^d$ and $\mathbf{w}_k \in \mathbb{R}^l$ are uncorrelated random variables, which represent the system noise and the measurement noise at time step k, and are independent of the distribution of the initial state p_0. Throughout the paper we assume that $\mathbf{f}_k : \mathbb{R}^n \times \mathbb{R}^d \to \mathbb{R}^n$ is a known nonlinear function that models the evolution of the state $\mathbf{x}_k$ affected by the random variable $\mathbf{v}_k$, and that $\mathbf{h}_k : \mathbb{R}^n \times \mathbb{R}^l \to \mathbb{R}^p$ is known nonlinear function that relates the observation variable $\mathbf{y}_k$ to the state variable $\mathbf{x}_k$ at time step k under the disturbances caused by the noise $\mathbf{w}_k$. Furthermore, we assume that the distributions of state and observation noise $\mathbf{v}_k$ and $\mathbf{w}_k$ are known for all $k \geq 1$.

Note that in some applications the functions $\mathbf{f}_k$ or $\mathbf{h}_k$ might depend on uncertain parameters [31]. In such cases it is possible to learn the dynamics of the system online with the Expectation-Maximization algorithm [31]–[33].

Given the sequence of measurements up to time step k, i.e., $\mathcal{Y}_k = \{\boldsymbol{y}_i, i = 1, ..., k\}$, and the initial knowledge of the state distribution p_0, the objective of the estimation is to find a certain *probability density function* (PDF). For dynamical systems we distinguish three classical estimation problems [34]–[37]:

1) *m-step smoothing*: estimation of $p(\mathbf{x}_{k-m}|\mathcal{Y}_k)$,
2) *m-step prediction*: estimation of $p(\mathbf{x}_{k+m}|\mathcal{Y}_k)$,
3) *filtering*: estimation of $p(\mathbf{x}_k|\mathcal{Y}_k)$.

One can distinguish three types of smoothing algorithms: fixed-point smoothers, which estimate a state at a fixed point of time using a growing number of measurements, fixed-interval smoothers, which estimate states within a fixed time interval using all the measurements from the same time interval, and fixed-lag smoothers, which estimate states with a fixed time delay. The overview of these methods is out of scope of this paper, instead readers interested in smoothing methods are referred to [28], [36]–[41].

The prediction problem is closely related to the filtering problem. In fact, finding the *m*-step predictor can always be done by iterating the prediction step of a given filter. Thus, no specialized algorithms are needed for this.

In this paper we restrict our analysis to BFs, which recursively solve the filtering problem for the system (1)–(3) in two steps. First, during the *prediction step*, the state model (1) and the density $p(\mathbf{x}_{k-1}|\mathcal{Y}_{k-1})$ are used to derive the *predicted state density* via the Chapman-Kolmogorov equation:

$$p(\mathbf{x}_k|\mathcal{Y}_{k-1}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1}) p(\mathbf{x}_{k-1}|\mathcal{Y}_{k-1}) d\mathbf{x}_{k-1}, \tag{4}$$

where the *transition density* $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ is determined by the known statistics of $\mathbf{v}_{k-1}$ and the transformation $\mathbf{f}_{k-1}$.

The prediction step is followed by the *update step* where the most recent measurement $\mathbf{y}_k$ is combined with the predicted state density $p(\mathbf{x}_k|\mathcal{Y}_{k-1})$ using the observation model (2). The desired *posterior* PDF $p(\mathbf{x}_k|\mathcal{Y}_k)$ is computed via Bayes' rule:

$$p(\mathbf{x}_k|\mathcal{Y}_k) = \frac{p(\boldsymbol{y}_k|\mathbf{x}_k) p(\mathbf{x}_k|\mathcal{Y}_{k-1})}{\int p(\boldsymbol{y}_k|\mathbf{x}_k) p(\mathbf{x}_k|\mathcal{Y}_{k-1}) d\mathbf{x}_k}. \tag{5}$$

Note that most real-life applications do not require a PDF but rather a concrete point estimate of a state. The posterior PDF contains all the information required for computing an optimal point estimate $\hat{\boldsymbol{x}}_k$ of the state with respect to a predefined criterion. In general, the choice of the criterion is an important (and non trivial) problem. An incorrectly chosen $\hat{\boldsymbol{x}}_k$ might lead to a significant decrease in the filter's performance as, e.g., in multi-target tracking applications [42], [43]. Two of the most popular estimators [8], [44], [45] are the *minimum mean-square error* (MMSE) estimator and the *maximum a posteriori* (MAP) estimator. The MMSE estimate is computed as the conditional mean of $\mathbf{x}_k$ given $\mathcal{Y}_k$

$$\hat{\boldsymbol{x}}_k^{\text{MMSE}} = \mathbb{E}(\mathbf{x}_k|\mathcal{Y}_k) = \int \mathbf{x}_k p(\mathbf{x}_k|\mathcal{Y}_k) d\mathbf{x}_k.$$

The MAP estimate is given by the vector that maximizes the posterior density, i.e., it is a solution of the optimization problem

$$\hat{\boldsymbol{x}}_k^{\text{MAP}} = \arg\max_{\mathbf{x}_k} p(\mathbf{x}_k|\mathcal{Y}_k).$$

Note that the MAP estimate is not unique if the posterior PDF achieves the maximal value in multiple points (e.g., the PDF of the uniform distribution).

For systems with linear dynamics and additive Gaussian noises [46]–[49] the posterior PDF is also Gaussian [50] and is computed in a closed from by the *Kalman Filter* (KF) [51]. The KF is an unbiased estimator [52] that is optimal in the MMSE and the MAP sense (for the KF the MMSE estimate and the MAP estimate are identical). Despite restrictive assumptions the KF continues to be very popular among practitioners. In particular the problem of tuning the

parameters of the KF in case of model uncertainty has attracted much attention. In [53] methods for unbiased, consistent and asymptotically normal estimators of covariances $\mathrm{cov}(\mathbf{v}_k)$ and $\mathrm{cov}(\mathbf{w}_k)$ have been developed. These algorithms have been further improved in [54]–[57]. The effect of prefiltering upon the estimates of covariances has been studied in [58]. It has beed noted that precise knowledge of $\mathrm{cov}(\mathbf{v}_k)$ is more critical [48], [55], [59] than knowledge of $\mathrm{cov}(\mathbf{w}_k)$, which often can be derived from sensors specifications.

Unfortunately, for general nonlinear systems there exists no closed form solution to the filtering problem. Thus, in most cases we need to rely on approximations of the true posterior PDF which lead to suboptimal solutions [60], [61]. In the following sections we present suboptimal parametric filters.

B. Performance Evaluation

It is possible to asses the achievable performance of a given nonlinear filter by computing the (sampled) variance of the estimator and comparing it with the *Posterior Cramér-Rao Bound* (PCRB). The PCRB gives a lower bound on the *mean squared error* (MSE) for any estimator of a random variable, see Chapter 2.4 of [62]. Thus, it is a generalization of the classical *Cramér-Rao Bound* (CRB), see Chapter 32 of [63] or [64], which bounds the MSE of estimators of deterministic variables. The PCRB is derived for the system (1)–(3) and it is independent of the filter applied to the system. Thus, the PCRB serves as a benchmark for comparing the performance of the nonlinear filters [65]. Applications of the CRB to continuous-time nonlinear filtering are discussed in [66] whereas [67] focuses on discrete-time nonlinear filtering.

In what follows $\mathbf{x}_{1:k}$ and $\mathbf{y}_{1:k}$ denote random variables $(\mathbf{x}_1, ..., \mathbf{x}_k)$ and $(\mathbf{y}_1, ..., \mathbf{y}_k)$, respectively. The PCRB is a lower bound on the mean squared estimation error defined by

$$\mathbb{E}_{\mathbf{y}_{1:k},\mathbf{x}_{1:k}}\left(\left(\hat{\boldsymbol{x}}_{1:k}-\mathbf{x}_{1:k}\right)\left(\hat{\boldsymbol{x}}_{1:k}-\mathbf{x}_{1:k}\right)^{T}\right), \tag{6}$$

where $\mathbb{E}_{\mathbf{y}_{1:k},\mathbf{x}_{1:k}}$ denotes the expectation taken with respect to the random variables $\mathbf{y}_{1:k}$ and $\mathbf{x}_{1:k}$, and $\hat{\boldsymbol{x}}_{1:k}$ is an estimator of $\mathbf{x}_{1:k}$ which depends on the observation $\mathbf{y}_{1:k}$.

The lower bound on (6) is given by the inverse $(\mathbf{J}_{1:k})^{-1}$, i.e.,

$$\mathbb{E}_{\mathbf{y}_{1:k},\mathbf{x}_{1:k}}\left(\left(\hat{\boldsymbol{x}}_{1:k}-\mathbf{x}_{1:k}\right)\left(\hat{\boldsymbol{x}}_{1:k}-\mathbf{x}_{1:k}\right)^{T}\right)\geq\left(\mathbf{J}_{1:k}\right)^{-1}, \tag{7}$$

with the $nk \times nk$ information matrix $\mathbf{J}_{1:k}$ defined by [67]

$$\mathbf{J}_{1:k}:=\mathbb{E}_{\mathbf{y}_{1:k},\mathbf{x}_{1:k}}\left(-\Delta_{\mathbf{x}_{1:k}}^{\mathbf{x}_{1:k}}\log p\left(\mathbf{x}_{1:k},\mathbf{y}_{1:k}\right)\right), \tag{8}$$

where $\Delta_x^y = \nabla_x(\nabla_y)^T$ is a second-order derivative operator and $p(\mathbf{x}_{1:k}, \mathbf{y}_{1:k})$ is a joint density of the random variable $(\mathbf{x}_{1:k}, \mathbf{y}_{1:k})$.

Equations (7)–(8) give the lowest bound on the MSE of an estimator of the whole trajectory $\hat{\boldsymbol{x}}_{1:k}$. However, by (8), computation of the right-hand side of (7) requires inverting the large $nk \times nk$ matrix $\mathbf{J}_{1:k}$, which is undesirable from the numerical perspective. Fortunately, it has been shown [67] that it is possible to compute the PCRB recursively for each single-step estimator $\hat{\boldsymbol{x}}_k$

$$\mathbb{E}_{\mathbf{y}_{1:k},\mathbf{x}_{1:k}}\left(\left(\hat{\boldsymbol{x}}_{k}-\mathbf{x}_{k}\right)\left(\hat{\boldsymbol{x}}_{k}-\mathbf{x}_{k}\right)^{T}\right)\geq\left(\mathbf{J}_{k}\right)^{-1}, \tag{9}$$

where $\mathbf{J}_k$ is $n \times n$ matrix that can be computed recursively by solving Riccati-like equations:

$$\mathbf{J}_{k+1}=\mathbf{D}_k^{22}-\left(\mathbf{D}_k^{12}\right)^T\left(\mathbf{J}_k+\mathbf{D}_k^{11}\right)^{-1}\mathbf{D}_k^{12}, \tag{10}$$

where

$$\mathbf{D}_k^{11}:=\mathbb{E}_{\mathbf{x}_{1:k}}\left(-\Delta_{\mathbf{x}_k}^{\mathbf{x}_k}\log p\left(\mathbf{x}_{k+1}|\mathbf{x}_k\right)\right) \tag{11a}$$

$$\mathbf{D}_k^{12}:=\mathbb{E}_{\mathbf{x}_{1:k}}\left(-\Delta_{\mathbf{x}_k}^{\mathbf{x}_{k+1}}\log p\left(\mathbf{x}_{k+1}|\mathbf{x}_k\right)\right) \tag{11b}$$

$$\mathbf{D}_k^{22}:=\mathbb{E}_{\mathbf{x}_{1:k}}\left(-\Delta_{\mathbf{x}_{k+1}}^{\mathbf{x}_{k+1}}\log p\left(\mathbf{x}_{k+1}|\mathbf{x}_k\right)\right) + \mathbb{E}_{\mathbf{y}_{1:k},\mathbf{x}_{1:k}}\left(-\Delta_{\mathbf{x}_{k+1}}^{\mathbf{x}_{k+1}}\log p\left(\mathbf{y}_{k+1}|\mathbf{x}_{k+1}\right)\right). \tag{11c}$$

The iteration (10) is initialized with matrix $\mathbf{J}_0$, which is calculated from the initial condition (3)

$$\mathbf{J}_0:=\mathbb{E}_{\mathbf{x}_0}\left(-\Delta_{\mathbf{x}_0}^{\mathbf{x}_0}\log p_0\left(\mathbf{x}_0\right)\right).$$

Thus, dealing with large matrices is avoided.

Note that the PCRB implementation requires the derivatives in (11) to be evaluated in the true state $\mathbf{x}_k$ and $\mathbf{x}_{k+1}$ [65], [68]. Alternatively, the PCRB can be approximated by evaluating $\mathbf{D}_k^{11}, \mathbf{D}_k^{12}$ and $\mathbf{D}_k^{22}$ in the estimate of the state [68]. It has been argued that such an approximated PCRB can also be used as a performance measure of nonlinear filters [68]. In some online applications the use of the *Conditional PCRB*, which depends on the actual realization $\mathcal{Y}_k$ of the random variable $\mathbf{y}_{1:k}$, is preferable over the standard PCRB [69]. Another interesting class of PCRB used in target tracking applications and designed for systems with uncertainty about measurements origin has been studied in [70]–[72]. Recursive algorithms for computing the PCRB for prediction, filtering and smoothing estimation problems are discussed in [40].

The PCRB provides a very useful performance measure for nonlinear filters. Another important property of the discussed algorithms is stability. In case of nonlinear filtering the stability analysis requires an advanced measure theoretic approach that is out of scope of this paper. A comprehensive overview of the stability properties and asymptotic analysis of nonlinear filtering methods is given in [73], [74].

III. ANALYTICAL APPROXIMATIONS: EKF, IEKF

In this section we describe the *Extended Kalman Filter* (EKF) and its modification, the *Iterated Extended Kalman Filter* (IEKF). Both filters are analytical methods because the approximations of (4)-(5) are derived using the Taylor series expansion, a method that exploits the analytical structure of functions $\mathbf{f}_k$ and $\mathbf{h}_k$.

A. Extended Kalman Filter

The EKF is one of the most popular modifications of the KF and is designed to estimate the states of a nonlinear system. The main idea of the EKF algorithm is that at each time step the nonlinear state (1) and observation (2) models can be analytically approximated in order to obtain a linear system.

For sufficiently smooth functions $\mathbf{f}_k$ and $\mathbf{h}_k$ given the previous state estimate $\hat{\boldsymbol{x}}_{k-1|k-1}$ and covariance $\mathbf{P}_{k-1|k-1}$ the EKF approximates the right-hand sides of (1)–(2) with the first-order Taylor series expansion around the points $(\hat{\boldsymbol{x}}_{k-1|k-1}, \mathbf{0})$ and $(\mathbf{f}_{k-1}(\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{0}), \mathbf{0})$, respectively [48], [75]:

$$\begin{aligned}\mathbf{x}_k \approx{}& \mathbf{f}_{k-1}(\hat{\boldsymbol{x}}_{k-1|k-1}, \mathbf{0}) + \mathbf{F}_{k-1}(\Delta\mathbf{x}_{k-1} - \hat{\boldsymbol{x}}_{k-1|k-1}) \\ &+ \mathbf{V}_{k-1}\left(\Delta\mathbf{v}_{k-1}\right) \qquad (12a)\\ \mathbf{y}_k \approx{}& \mathbf{h}_k(\mathbf{f}_{k-1}(\hat{\boldsymbol{x}}_{k-1|k-1}, \mathbf{0}), \mathbf{0}) \\ &+ \mathbf{H}_k(\Delta\mathbf{x}_k - \mathbf{f}_{k-1}(\hat{\boldsymbol{x}}_{k-1|k-1}, \mathbf{0})) + \mathbf{W}_k\left(\Delta\mathbf{w}_k\right) \qquad (12b)\end{aligned}$$

for every $\Delta\mathbf{x}_{k-1}, \Delta\mathbf{x}_k, \Delta\mathbf{v}_{k-1}, \Delta\mathbf{w}_k$ where

1) $\mathbf{F}_{k-1}$ is the Jacobian matrix of the partial derivatives of $\mathbf{f}_{k-1}$ with respect to the state variable $\mathbf{x}$, evaluated at the point $(\hat{\boldsymbol{x}}_{k-1|k-1}, \mathbf{0})$:
$$\mathbf{F}_{k-1} = \frac{\partial \mathbf{f}_{k-1}}{\partial \mathbf{x}}(\hat{\boldsymbol{x}}_{k-1|k-1}, \mathbf{0}), \qquad (13)$$
2) $\mathbf{V}_{k-1}$ is the Jacobian matrix of the partial derivatives of $\mathbf{f}_{k-1}$ with respect to the noise variable $\mathbf{v}$, evaluated at the point $(\hat{\boldsymbol{x}}_{k-1|k-1}, \mathbf{0})$:is
$$\mathbf{V}_{k-1} = \frac{\partial \mathbf{f}_{k-1}}{\partial \mathbf{v}}(\hat{\boldsymbol{x}}_{k-1|k-1}, \mathbf{0}), \qquad (14)$$
3) $\mathbf{H}_k$ is the Jacobian matrix of the partial derivatives of $\mathbf{h}_k$ with respect to the state variable $\mathbf{x}$, evaluated at the point $(\mathbf{f}_{k-1}(\hat{\boldsymbol{x}}_{k-1|k-1}, \mathbf{0}), \mathbf{0})$:
$$\mathbf{H}_k = \frac{\partial \mathbf{h}_k}{\partial \mathbf{x}}(\mathbf{f}_{k-1}(\hat{\boldsymbol{x}}_{k-1|k-1}, \mathbf{0}), \mathbf{0}), \qquad (15)$$
4) $\mathbf{W}_k$ is the Jacobian matrix of the partial derivatives of $\mathbf{h}_k$ with respect to the noise variable $\mathbf{w}$, evaluated at the point $(\mathbf{f}_{k-1}(\hat{\boldsymbol{x}}_{k-1|k-1}, \mathbf{0}), \mathbf{0})$:
$$\mathbf{W}_k = \frac{\partial \mathbf{h}_k}{\partial \mathbf{w}}(\mathbf{f}_{k-1}(\hat{\boldsymbol{x}}_{k-1|k-1}, \mathbf{0}), \mathbf{0}). \qquad (16)$$

It can be easily seen that the right-hand sides of both (12a) and (12b) are Gaussian random variables. Therefore, the predicted state and the posterior state densities are given by

$$p(\mathbf{x}_k|\mathcal{Y}_{k-1}) = \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k-1}, \mathbf{P}_{k|k-1}), \qquad (17a)$$
$$p(\mathbf{x}_k|\mathcal{Y}_k) = \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k}, \mathbf{P}_{k|k}), \qquad (17b)$$

with the means and the covariances as in Algorithm 1.

The approximations (12) are accurate only if the following three assumptions hold:

I. the noises $\mathbf{v}_k$, and $\mathbf{w}_k$ are lightly tailed, i.e., the norms of the covariance matrices $\mathbf{V}_k$, and $\mathbf{W}_k$ are small, and
II. the estimate $\hat{\mathbf{x}}_{k-1|k-1}$ is approximately equal to the actual state of the system at time step $k-1$,
III. the functions $\mathbf{f}_k$ and $\mathbf{h}_k$ do not exhibit severe nonlinear behavior.

The first two postulates, together with the fact that $\mathbb{E}[\mathbf{v}_k] = \mathbf{0}$ and $\mathbb{E}[\mathbf{w}_k] = \mathbf{0}$ justify the Taylor expansions around the aforementioned points, whereas the third one allows one to truncate the infinite series after the first derivative term.

Note that as far as real systems are concerned Postulate I seems reasonable. Indeed, in most applications the process and the measurement noises are bounded within narrow intervals [8], [76]. Postulate II simply states that the estimator is accurate, meaning unbiased, and precise, meaning with small covariance matrix.

Postulate III is more critical. To understand why, recall that the approximation (12a) models the predicted state as a Gaussian random variable, whereas in reality a variable after the nonlinear transformation $\mathbf{f}_k$ is no longer normally distributed. In case of mild nonlinearities (different measures of nonlinearity are reported in [77], [78]), the transformed variable can be accurately approximated by a Gaussian distribution. However, for a system that exhibits a strong nonlinear behavior the approximation is no longer feasible and might result in an inconsistent estimator. The influence of the linearization errors on the final EKF performance has been extensively studied in the literature [8], [79]–[81].

Note that the EKF requires the covariance matrices $\mathbf{Q}_k$ and $\mathbf{R}_k$. They can be derived from stochastic properties of the noises $\mathbf{v}_k$, and $\mathbf{w}_k$ or, if these are unknown, tuned from data [82], [83].

Algorithm 1 Extended Kalman Filter

Require: $\mathbf{P}_{k-1|k-1}$, $\hat{\boldsymbol{x}}_{k-1|k-1}$, $\mathbf{Q}_{k-1}$, and $\mathbf{R}_k$
Prediction step:
Compute matrices $\mathbf{F}_{k-1}$, and $\mathbf{V}_{k-1}$ according to (13)–(14)
Compute the predicted mean $\hat{\boldsymbol{x}}_{k|k-1}$:
$\hat{\boldsymbol{x}}_{k|k-1} = \mathbf{f}_{k-1}(\hat{\boldsymbol{x}}_{k-1|k-1}, \mathbf{0})$
Compute the predicted covariance $\mathbf{P}_{k|k-1}$:
$\mathbf{P}_{k|k-1} = \mathbf{F}_{k-1}\mathbf{P}_{k-1|k-1}\mathbf{F}_{k-1}^T + \mathbf{V}_{k-1}\mathbf{Q}_{k-1}\mathbf{V}_{k-1}^T$
Update step:
Compute matrices $\mathbf{H}_k$, and $\mathbf{W}_k$ according to (15)–(16)
Compute the Kalman gain $\mathbf{K}_k$:
$\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{H}_k^T\left(\mathbf{H}_k\mathbf{P}_{k|k-1}\mathbf{H}_k^T + \mathbf{W}_k\mathbf{R}_k\mathbf{W}_k^T\right)^{-1}$
Compute the estimated mean $\hat{\boldsymbol{x}}_{k|k}$:
$\hat{\boldsymbol{x}}_{k|k} = \hat{\boldsymbol{x}}_{k|k-1} + \mathbf{K}_k\left(\boldsymbol{y}_k - \mathbf{h}_k(\hat{\boldsymbol{x}}_{k|k-1}, \mathbf{0})\right)$
Compute the estimated covariance $\mathbf{P}_{k|k}$:
$\mathbf{P}_{k|k} = \left(\mathbf{I} - \mathbf{K}_k\mathbf{H}_k\right)\mathbf{P}_{k|k-1}$

B. Iterated Extended Kalman Filter

In order to improve the EKF the *Iterated Extended Kalman Filter* (IEKF) has been developed [84], [85]. This algorithm has a strong resemblance to the conventional EKF. In fact, for both filters the linearization of the prediction function $\mathbf{F}_k$ is derived in the same manner, and they differ only in the way in which the updated estimate is computed. The IEKF assumes that the measurement model is such that for every time step k the noise variable $\mathbf{w}_k$ can be explicitly expressed as a function of $\mathbf{y}_k$ and $\mathbf{x}_k$, i.e., for each k there exists function $\mathbf{g}_k$ such that:

$$\mathbf{w}_k = \mathbf{g}_k(\mathbf{y}_k, \mathbf{x}_k).$$

If the observation model has additive linear noises, i.e.,

$$\mathbf{h}_k(\mathbf{x}_k, \mathbf{w}_k) = \mathbf{h}_k(\mathbf{x}_k) + \mathbf{H}_k\mathbf{w}_k,$$

with an invertible matrix $\mathbf{H}_k$, then

$$\mathbf{g}_k(\mathbf{y}_k, \mathbf{x}_k) = \left(\mathbf{H}_k\right)^{-1}\left(\mathbf{y}_k - \mathbf{h}_k(\mathbf{x}_k)\right),$$

which is the scaled difference between the measured and the predicted variables. The IEKF linearizes $\mathbf{g}_k$ around the

updated state estimate $\hat{\boldsymbol{x}}_{k|k}$ rather than around the predicted state estimate $\hat{\boldsymbol{x}}_{k|k-1}$ as the EKF does. This is achieved by the following iteration (hence the name): the algorithm starts with a linearized model around the predicted estimate $\hat{\boldsymbol{x}}_{k|k-1}$, and uses it to compute the updated state estimate $\hat{\boldsymbol{x}}^1_{k|k}$. Then, the function $\mathbf{g}_k$ is linearized around this newly obtained vector, and the new updated state estimate $\hat{\boldsymbol{x}}^2_{k|k}$ is derived. This procedure is repeated until the iteration step i_0 is reached such that $\|\hat{\boldsymbol{x}}^{i_0}_{k|k} - \hat{\boldsymbol{x}}^{i_0-1}_{k|k}\| < \epsilon$, where ϵ is a predefined small number. This iteration, which is equivalent to Gauss-Newton method [86], is presented in Algorithm 2. For the detailed derivation of the update algorithm see Section 3.4 of [84].

Algorithm 2 IEKF: Update Iteration

Require: ϵ, $\mathbf{P}_{k|k-1}$, $\hat{\boldsymbol{x}}_{k|k-1}$, $\mathbf{R}_k$, $\boldsymbol{y}_k$
  Set the initial estimate: $\hat{\boldsymbol{x}}^0_{k|k} = \hat{\boldsymbol{x}}_{k|k-1}$
  Set the initial counter: $i = 0$
  repeat
    Augment the counter: $i = i + 1$
    Linearize the error model : $\mathbf{H}^i_k = \frac{\partial \mathbf{g}_k}{\partial x}\left(\boldsymbol{y}_k, \hat{\boldsymbol{x}}^{i-1}_{k|k}\right)$
    Compute the Kalman gain:
    $\mathbf{K}^i_k = \mathbf{P}_{k|k-1}\left(\mathbf{H}^i_k\right)^T\left(\mathbf{H}^i_k\mathbf{P}_{k|k-1}\left(\mathbf{H}^i_k\right)^T + \mathbf{R}_k\right)^{-1}$
    Update the estimate:
    $\hat{\boldsymbol{x}}^i_{k|k} = \hat{\boldsymbol{x}}_{k|k-1} - \mathbf{K}^i_k\left(\mathbf{g}_k\left(\mathbf{y}_k, \hat{\boldsymbol{x}}^{i-1}_{k|k}\right) + \mathbf{H}^i_k\left(\hat{\boldsymbol{x}}_{k|k-1} - \hat{\boldsymbol{x}}^{i-1}_{k|k}\right)\right)$
  until $\|\hat{\boldsymbol{x}}^i_{k|k} - \hat{\boldsymbol{x}}^{i-1}_{k|k}\| < \epsilon$
  Set: $i_0 = i$
  Set the updated estimate: $\hat{\boldsymbol{x}}_{k|k} = \boldsymbol{x}^{i_0}_{k|k}$
  Set the covariance of the updated estimate:
  $\mathbf{P}_{k|k} = \left(\mathbf{I} - \mathbf{K}^{i_0}_k\mathbf{H}^{i_0}_k\right)\mathbf{P}_{k|k-1}$

Note that in the case of a linear observation model with additive noises the IEKF is reduced to the standard EKF. The disadvantage of the IEKF is that, due to the internal loop, it is numerically more involved than the EKF. Also, it has been argued that both the IEKF and the EKF perform similarly if the state is only partially observable [87]. Informative examples of applications and comparison of the performance of the two filters are discussed in [87], [88].

C. Other EKF-like Algorithms

The accuracy of the EKF can be further improved by the addition of higher-order terms in approximation (12). Better accuracy comes with the price of increased computational burden. Furthermore, although the higher-order filters reduce the bias of the estimators [89], in general they cannot produce unbiased estimates [79].

Other variations of the EKF have recently been developed algorithms that avoid gradient computations [90], [91]. Regarding these filters, two approaches can be distinguished: implicit methods, and explicit methods. In the implicit approach the problem of calculating a Jacobian is replaced by the one of finding a solution of an analytical equation (see [90] and references therein). In the explicit approach the nonlinear operator is linearized by means of Euler or Newmark expansion [91]. As presented in [90], [91], in certain situations, e.g. in case of jumps in parameter values, those filters achieve better performance than the conventional EKF.

D. Example

To illustrate the difference in the performance between the EKF and the IEKF let us investigate a simple two-dimensional nonlinear system defined by

$$\mathbf{x}_{k+1}(1) = 0.1\left(\mathbf{x}_k(1)\right)^2 - 2\mathbf{x}_k(1) + 20 + \mathbf{v}_k(1), \tag{18a}$$
$$\mathbf{x}_{k+1}(2) = \mathbf{x}_k(1) + 0.3\mathbf{x}_k(2) - 3 + \mathbf{v}_k(2), \tag{18b}$$

and

$$\mathbf{y}_k(1) = \left(\mathbf{x}_k(1)\right)^2 + \left(\mathbf{x}_k(2)\right)^2 + \mathbf{w}_k(1), \tag{19a}$$
$$\mathbf{y}_k(2) = 3\left(\mathbf{x}_k(2)\right)^2/\mathbf{x}_k(1) + \mathbf{w}_k(2) \tag{19b}$$

Equations (18)–(19) constitute a system that is a modification of the case studied in [87]. The system is nonlinear in both the state model (the second order term in (18a)) and the observation model. Furthermore, both the state and the observation models are influenced by mutually independent additive Gaussian noises $\mathbf{v}_k$ and $\mathbf{w}_k$ with covariance matrices $\mathbf{Q}_k = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$ and $\mathbf{R}_k = \begin{bmatrix} 1 & 0 \\ 0 & 10 \end{bmatrix}$ respectively.

For the purpose of comparison, starting from the initial state $\mathbf{x}_0 = [10\ 10]^T$, we have generated a random state trajectory $\boldsymbol{x}_{1:20} = (\boldsymbol{x}_1, ..., \boldsymbol{x}_{20})$ with the corresponding observations $\mathcal{Y}_{20}$ according to (18) and (19) respectively. Figure 1 compares the estimates obtained by the EKF and the IEKF aiming to reproduce the trajectory $\boldsymbol{x}_{1:20}$ from the generated measurements $\mathcal{Y}_{20}$. Both filters are initialized from the actual state of the system, i.e., from $\mathbf{x}_0 = [10\ 10]^T$ each having the same initial uncertainty about the true state $\mathbf{P}_0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$. Furthermore, the parameter ϵ, which is used in Algorithm 2, is set to $\epsilon = 0.0001$.

From Figure 1 it can be observed that most of the time the two nonlinear filters behave similarly. However, in some cases the IEKF tracks the actual state of the system more closely than the EKF does.

Let us now compute the PCRB for the system (18)–(19). It can be shown [67], [92] that in the case of additive Gaussian noises the matrices (11) are given by:

$$\mathbf{D}^{11}_k = \mathbb{E}_{\mathbf{x}_{1:k}}\left(\left(\nabla_{\mathbf{x}_k}\mathbf{f}^T_k\left(\mathbf{x}_k\right)\right)\left(\mathbf{Q}_k\right)^{-1}\left(\nabla_{\mathbf{x}_k}\mathbf{f}^T_k\left(\mathbf{x}_k\right)\right)^T\right), \tag{20a}$$
$$\mathbf{D}^{12}_k = -\mathbb{E}_{\mathbf{x}_{1:k}}\left(\nabla_{\mathbf{x}_k}\mathbf{f}^T_k\left(\mathbf{x}_k\right)\right)\left(\mathbf{Q}_k\right)^{-1}, \tag{20b}$$
$$\mathbf{D}^{22}_k = \left(\mathbf{Q}_k\right)^{-1} + \mathbb{E}_{\mathbf{x}_{1:k+1}}\left(\left(\nabla_{\mathbf{x}_{k+1}}\mathbf{h}^T_{k+1}\left(\mathbf{x}_{k+1}\right)\right)\left(\mathbf{R}_{k+1}\right)^{-1}\left(\nabla_{\mathbf{x}_{k+1}}\mathbf{h}^T_{k+1}\left(\mathbf{x}_{k+1}\right)\right)^T\right). \tag{20c}$$

The derivatives in (20) are evaluated in the true states of the system and the expectations are obtained by Monte Carlo [93] averaging over $10,000$ realizations of the independent trajectories of the system, with the initial distribution $p_0 = \mathcal{N}\left(\mathbf{x}_0, \mathbf{P}_0\right)$. The initial information matrix is given by

$$\mathbf{J}_0 = \left(\mathbf{P}_0\right)^{-1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

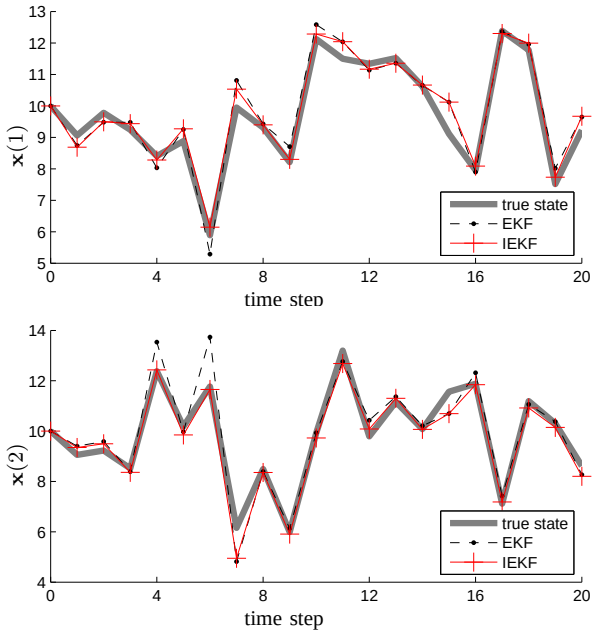

Fig. 1: The EKF estimate (thin solid line) and the IEKF estimate (dashed line) vs the sample trajectory $\mathbf{x}_{0:20}$ generated from the system (18)–(19) (thick solid line).

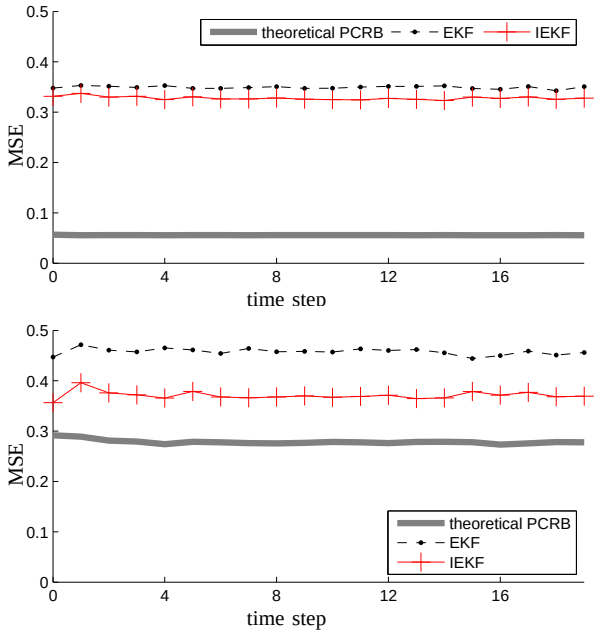

Fig. 2: RMSE of the estimators given by the EKF (dashed line) and the IEKF (thin solid line) compared with the squared roots of the theoretical PCRB (thick solid line) for the states $\mathbf{x}(1)$ (above) and $\mathbf{x}(2)$ (below).

Figure 2 shows the square roots of the theoretical PCRB for states $\mathbf{x}(1)$ and $\mathbf{x}(2)$. Furthermore, the theoretical lower bounds are compared with the *Root Mean Squared Errors* (RMSE) obtained from $10,000$ Monte Carlo runs of the system (18)–(19) with the same initial distribution and the same noise levels. From the figure it can be observed that the IEKF has a lower RMSE than the EKF.

IV. STATISTICAL APPROXIMATIONS: UKF, GHF, CDF

In this section we discuss an alternative approach to the nonlinear approximation problem, namely the statistical approach. Contrary to the methods presented in Section III, the filters described in the current section do not use the Taylor series expansion. Instead, we are interested in statistical information that can be extracted from the system (1)–(3) and used afterwards to estimate (4)–(5). All the filters discussed within this section can be considered as a part of a general class of *linear regression Kalman Filters* (LRKFs).

LRKFs have been proposed by several authors [17], [81], [87], [94], [95]. Similarly to the EKF, these filters approximate the prediction and the posterior density as Gaussian densities, hence the formulas (17a) and (17b) still hold. However, the approximations of $p(\mathbf{x}_k|\mathcal{Y}_{k-1})$ and $p(\mathbf{x}_k|\mathcal{Y}_k)$ are obtained by means of statistical regression rather than through analytical approximations of the nonlinear functions $\mathbf{f}_k$ and $\mathbf{h}_k$ as in the EKF setting. The motivation for this approach can be intuitively expressed as follows: *With a fixed number of parameters it is easier to approximate a Gaussian distribution than it is to approximate an arbitrary nonlinear function* [94]. The general idea is to represent the a priori distributions by a set of deterministically chosen representative points and weights that completely capture the mean and the covariance of the Gaussian distribution and then use those points in the prediction and the update steps of the filter. This resembles a Monte Carlo approach. However, unlike Monte Carlo methods, the LRKFs are, in general, numerically less expensive since the samples are not drawn at random and the number of the required points is relatively small when compared with the number of samples that are generated by the Monte Carlo algorithms. LRKFs achieve better accuracy than the EKF since the representative points propagated through the nonlinear transformation capture the mean and covariance of the actual distribution up to the second order of nonlinearity [94].

There are many methods for the choice of the representative points and their weights, and the three most popular ones are discussed in Sections [IV-A–IV-C]. However, for a moment, let us focus on the general framework of the LRKF.

The estimation proceeds as follows. At time step $k-1$ the approximation of the posterior distribution is given by a Gaussian variable $\mathcal{N}(\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{P}_{k-1|k-1})$ and the noise $\mathbf{v}_{k-1}$ is assumed to be distributed according to $\mathcal{N}(\mathbf{0}, \mathbf{Q}_{k-1})$. Both $\mathbf{x}_{k-1}$ and $\mathbf{v}_k$ are assumed to be uncorrelated and Gaussian. Therefore, the augmented state variable defined as $\begin{bmatrix}\mathbf{x}_{k-1} & \mathbf{v}_{k-1}\end{bmatrix}^T$ is also Gaussian with the mean $\boldsymbol{\mu}^a_{k-1}$ and the covariance $\mathbf{P}^a_{k-1}$ given by:

$$\boldsymbol{\mu}^a_{k-1} = \begin{bmatrix} \hat{\mathbf{x}}_{k-1|k-1} \\ \mathbf{0} \end{bmatrix}, \quad \mathbf{P}^a_{k-1} = \begin{bmatrix} \mathbf{P}_{k-1|k-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_{k-1} \end{bmatrix}.$$

The probability distribution $\mathcal{N}\left(\boldsymbol{\mu}^a_{k-1}, \mathbf{P}^a_{k-1}\right)$ is encoded in the sequence $\left\{\left(\boldsymbol{x}^i_{k-1}, \omega^i_{k-1}\right)\right\}_{i=1}^N$ that pairs each representative point $\boldsymbol{x}^i_{k-1}$ with its weight ω^i_{k-1}. The predicted state density $p\left(\mathbf{x}_k|\mathcal{Y}_{k-1}\right)$ is approximated by $\mathcal{N}\left(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k-1}, \mathbf{P}_{k|k-1}\right)$,

where the mean and covariance are computed as follows:

$$\hat{\boldsymbol{x}}_{k|k-1} = \sum_{i=1}^{N} \omega_{k-1}^{i} \mathbf{f}_{k-1}(\boldsymbol{x}_{k-1}^{i}), \tag{21}$$

$$\mathbf{P}_{k|k-1} = \sum_{i=1}^{N} \omega_{k-1}^{i} \left(\mathbf{f}_{k-1}(\boldsymbol{x}_{k-1}^{i}) - \hat{\boldsymbol{x}}_{k|k-1}\right) \left(\mathbf{f}_{k-1}(\boldsymbol{x}_{k-1}^{i}) - \hat{\boldsymbol{x}}_{k|k-1}\right)^{T} \tag{22}$$

The distribution of the predicted measurement is obtained in a similar manner as the distribution of the predicted state. Namely, the noise $\mathbf{w}_k$ is assumed to be zero-mean Gaussian with the covariance matrix $\mathbf{R}_k$, and independent of the state $\mathbf{x}_k$. Therefore, the variable $[\mathbf{x}_k \ \mathbf{w}_k]^T$ is also Gaussian with the mean $\boldsymbol{\mu}_k^a$ and the covariance $\mathbf{P}_k^a$ given by:

$$\boldsymbol{\mu}_k^a = \begin{bmatrix} \hat{\boldsymbol{x}}_{k|k-1} \\ \mathbf{0} \end{bmatrix}, \qquad \mathbf{P}_k^a = \begin{bmatrix} \mathbf{P}_{k|k-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_k \end{bmatrix}.$$

Next, the set of representative points and weights $\left\{\left(\boldsymbol{x}_{k|k-1}^{i}, \omega_{k|k-1}^{i}\right)\right\}_{i=1}^{N}$ that approximate the distribution $\mathcal{N}\left(\boldsymbol{\mu}_k^a, \mathbf{P}_k^a\right)$ is derived. The estimate of the measurement is then given by:

$$\hat{\boldsymbol{y}}_{k|k-1} = \sum_{i=1}^{N} \omega_{k|k-1}^{i} \mathbf{h}_k(\boldsymbol{x}_{k|k-1}^{i}). \tag{23}$$

Finally the mean and covariance for the normal density that approximates the posterior $p\left(\mathbf{x}_k|\mathcal{Y}_k\right)$ are computed as follows:

$$\hat{\boldsymbol{x}}_{k|k} = \hat{\boldsymbol{x}}_{k|k-1} + \mathbf{K}_k \left(\boldsymbol{y}_k - \hat{\boldsymbol{y}}_{k|k-1}\right), \tag{24}$$

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{P}_{xy}\mathbf{K}_k^T, \tag{25}$$

where the Kalman gain $\mathbf{K}_k$ and the covariances $\mathbf{P}_{xy}$, and $\mathbf{P}_{yy}$ are computed as:

$$\mathbf{P}_{xy} = \sum_{i=1}^{N} \omega_{k|k-1}^{i} \left(\boldsymbol{x}_{k|k-1}^{i} - \hat{\boldsymbol{x}}_{k|k-1}\right) \left(\mathbf{h}_k(\boldsymbol{x}_{k|k-1}^{i}) - \hat{\boldsymbol{y}}_{k|k-1}\right)^{T}, \tag{26}$$

$$\mathbf{P}_{yy} = \sum_{i=1}^{N} \omega_{k|k-1}^{i} \left(\mathbf{h}_k(\boldsymbol{x}_{k|k-1}^{i}) - \hat{\boldsymbol{y}}_{k|k-1}\right) \left(\mathbf{h}_k(\boldsymbol{x}_{k|k-1}^{i}) - \hat{\boldsymbol{y}}_{k|k-1}\right)^{T}, \tag{27}$$

$$\mathbf{K}_k = \mathbf{P}_{xy}\left(\mathbf{P}_{yy}\right)^{-1}. \tag{28}$$

The LRKF algorithm can be simplified for systems with additive noises, i.e., for systems where the functions $\mathbf{f}_k$ and $\mathbf{h}_k$ have the form:

$$\mathbf{f}_k(\mathbf{x}_k, \mathbf{v}_k) = \mathbf{f}_k(\mathbf{x}_k) + \mathbf{F}_k\mathbf{v}_k,$$
$$\mathbf{h}_k(\mathbf{x}_k, \mathbf{w}_k) = \mathbf{h}_k(\mathbf{x}_k) + \mathbf{H}_k\mathbf{w}_k,$$

where both $\mathbf{F}_k$ and $\mathbf{H}_k$ are linear matrices. For such system one starts from computing the representative points and weights that approximate the distribution $\mathcal{N}\left(\hat{\boldsymbol{x}}_{k-1|k-1}, \mathbf{P}_{k-1|k-1}\right)$. Next, the predicted state $\hat{\boldsymbol{x}}_{k|k-1}$ is computed according to (21). In order to compute the covariance of the predicted state $\mathbf{P}_{k|k-1}$ the right-hand side of (22) is modified by adding the term $\mathbf{F}_{k-1}\mathbf{Q}_{k-1}\mathbf{F}_{k-1}^T$ which corresponds to the influence of the noise $\mathbf{v}_{k-1}$ [8]:

$$\mathbf{P}_{k|k-1} = \sum_{i=1}^{N} \omega_{k-1}^{i} \left(\mathbf{f}_{k-1}(\boldsymbol{x}_{k-1}^{i}) - \hat{\boldsymbol{x}}_{k|k-1}\right) \left(\mathbf{f}_{k-1}(\boldsymbol{x}_{k-1}^{i}) - \hat{\boldsymbol{x}}_{k|k-1}\right)^{T} + \mathbf{F}_{k-1}\mathbf{Q}_{k-1}\mathbf{F}_{k-1}^{T}. \tag{29}$$

The next step is to approximate the distribution $\mathcal{N}\left(\hat{\boldsymbol{x}}_{k|k-1}, \mathbf{P}_{k|k-1}\right)$ by the set of representative points and weights. The procedure of obtaining the final estimates of $\hat{\boldsymbol{x}}_{k|k}$ and $\mathbf{P}_{k|k}$ is similar to the one described by equations (23)–(27). The only difference is that the transformed covariance of the observation noise $\mathbf{w}_k$, i.e., $\mathbf{H}_k\mathbf{R}_k\mathbf{H}_k^T$ has to be added to the right-hand side of (27). Therefore, $\mathbf{P}_{yy}$ is given by [8]:

$$\mathbf{P}_{yy} = \sum_{i=1}^{N} \omega_{k-1}^{i} \left(\mathbf{h}_k(\boldsymbol{x}_{k|k-1}^{i}) - \hat{\boldsymbol{y}}_{k|k-1}\right) \left(\mathbf{h}_k(\boldsymbol{x}_{k|k-1}^{i}) - \hat{\boldsymbol{y}}_{k|k-1}\right)^{T} + \mathbf{H}_k\mathbf{R}_k\mathbf{H}_k^T \tag{30}$$

Note that the dimensions of the Gaussian variables $\mathcal{N}\left(\hat{\boldsymbol{x}}_{k-1|k-1}, \mathbf{P}_{k-1|k-1}\right)$ and $\mathcal{N}\left(\hat{\boldsymbol{x}}_{k|k-1}, \mathbf{P}_{k|k-1}\right)$ are lower than the dimensions of the variables $\mathcal{N}\left(\boldsymbol{\mu}_{k-1}^a, \mathbf{P}_{k-1}^a\right)$ and $\mathcal{N}\left(\boldsymbol{\mu}_k^a, \mathbf{P}_k^a\right)$ approximated within the general algorithm. Therefore, a smaller number of representative points is required, and consequently fewer nonlinear transformations have to be performed. Instead, they are replaced by linear operations: $\mathbf{F}_{k-1}\mathbf{Q}_{k-1}\mathbf{F}_{k-1}^T$ and $\mathbf{H}_k\mathbf{R}_k\mathbf{H}_k^T$.

It has been observed [17] that the performance of a filter given by (21)–(28) strongly depends on the choice of the representative points. In what follows we review methods that have been proposed in the recent years. In order to keep the algorithms simple, we focus on filters designed for dynamical systems with additive noises. We motivate this choice by the fact that LRKF equations for systems with non-additive noises are conceptually identical. We start by describing the most popular LRKF, i.e., the *Unscented Kalman Filter* (UKF) and its variations. Next, other types of LRKF are discussed, namely the *Gauss-Hermite Filter* (GHF), and the *Central Difference Filter* (CDF). Finally, all the aforementioned filters are illustrated with an example.

A. Unscented Kalman Filter

Before we proceed to detailed description of the UKF framework, we start with explaining the *Unscented Transformation* (UT) [81]. This is a method of selecting representative points and weights that approximate a variable after a nonlinear transformation. The UKF uses UT in a dynamic framework to obtain the approximations of the predicted state density and the predicted update density.

The UT is a general method for approximating the distribution of a Gaussian random variable after a nonlinear transformation. Let $\mathbf{x}$ be such a variable, with the mean $\bar{\boldsymbol{x}}$ and covariance $\mathbf{P}_{\mathbf{x}}$, and let $\mathbf{g} : \mathbb{R}^n \to \mathbb{R}^p$ be an arbitrary nonlinear function. The objective is to compute the statistics of a random variable $\mathbf{y}$ defined as:

$$\mathbf{y} = \mathbf{g}(\mathbf{x}). \tag{31}$$

In order to do that, first one has to generate a set $\boldsymbol{\Sigma} = \{\sigma_i\}$ of the *sigma points*, i.e., a set that is of zero sample mean

and the points of this set have sample covariance equal to $\mathbf{P_x}$. For the n-dimensional variable $\mathbf{x}$, $2n$ sigma points are computed as rows (or columns) of the matrix $\pm\sqrt{(n+\lambda)\mathbf{P_x}}$, where $\lambda = \alpha^2(n+\kappa) - n$ with a spread parameter α and a scaling factor κ. The common choice for the spread parameter is $\alpha = 1$ [96] in which case $\lambda = \kappa$ [81], [94].

The set $\mathbf{\Sigma}$ has the same mean and covariance as a zero mean Gaussian variable with a covariance matrix $\mathbf{P_x}$. Furthermore, since it is symmetric, all the odd central moments are equal to zero as is the case with every zero mean Gaussian distribution. Therefore, the first three sample moments of $\mathbf{\Sigma}$ are equal to the theoretical moments of the variable $\mathbf{x}$. Hence, the approximation errors can occur only in fourth and higher moments. The representative points of a distribution of the variable $\mathbf{x}$ are generated by a translation of each sigma point by $\bar{\boldsymbol{x}}$ and an assignment of appropriate weights [81], [96]:

$$\begin{aligned} \boldsymbol{x}_0 &= \bar{\boldsymbol{x}} & \omega_0 &= \frac{\lambda}{n+\lambda}, \\ \boldsymbol{x}_i &= \bar{\boldsymbol{x}} + \sigma_i & \omega_i &= \frac{1-\omega_0}{2n}. \end{aligned}$$

The distribution of the transformed random variable $\mathbf{y} = \mathbf{g}(\mathbf{x})$ is then represented by the set $\{(\mathbf{g}(\boldsymbol{x}_i), \omega_i)\}_{i=0}^{2n}$.

The errors in the calculation of the mean and covariance of $\mathbf{y}$ are of fourth order in case of Gaussian inputs [81] and of third order in case of non-Gaussian inputs [96]. The approximation accuracy can be further improved by an appropriate choice of a scaling factor κ [81]. The popular choice is $\kappa = 3 - n$ [94], [96]. Setting $\kappa = 0$ leads to the Cubature Kalman Filter introduced in [29], [97]. Further improvements on quality of estimation can be achieved with the adaptive selection of κ which is done by the Adaptive UKF [98]. If available, the information on the higher order moments of the estimated variable can be used to modify the weight ω_0, which reduces the higher-order errors of the UT [19], [20], [96]. It is also possible to capture higher moments of the true distribution by augmenting the number of sigma points used in the approximation [19], [20], [99], [100]. For instance, $2n^2 + 1$ sigma points are required to match the first four moments of a Gaussian distribution [19], [20]. The accuracy of the approximation might further increase if the sigma points are scaled, so that all the sigma points lay in an appropriate ellipsoid centered at the mean [19], [20], [99], or on the 1σ, 2σ and 3σ contours [101]. The $k\sigma$ contour is the boundary of the ellipsoid defined by $k\sqrt{\mathbf{P_x}}$. The latter method requires $6n+1$ sigma points. The purpose of the scaling is to concentrate all the sigma points in the area of the highest probability.

Note that there are infinitely many square roots of the matrix $\sqrt{\mathbf{P_x}}$ that can be chosen to compute $k\sigma$ contour [81]. Therefore, the improvement of the computational properties of UT is possible by the choice of an efficient numerical method for matrix square root computation. The most popular algorithm is the Cholesky decomposition, but other techniques, such as the more robust, but more computationally involved, singular value decomposition can also be used [102]. The computational efficiency of the UT can be further increased by reducing the number of sigma points that need to be generated in order to capture the desired properties of a distribution of the investigated random variable [19], [20], [103]. The minimal number of sigma points that is required to capture the mean and covariance is $n + 1$. The computational complexity of UT grows linearly with the number of dimensions. However, with increasing dimensions of the state the accuracy of the UT approximation decreases [104].

The UKF employs the UT at each filtering step following the procedure described in Algorithm 3.

Algorithm 3 Unscented Kalman Filter

Require: λ, $\mathbf{P}_{k-1|k-1}$, $\hat{\boldsymbol{x}}_{k-1|k-1}$
Compute the sigma points σ_i as the columns of the matrix: $\sqrt{(n+\lambda)\,\mathbf{P}_{k-1|k-1}}$
Prediction step:
Set the central point: $\boldsymbol{x}^0_{k-1} = \hat{\boldsymbol{x}}_{k-1|k-1}$
Set the central weight: $\omega^0_{k-1} = \frac{\lambda}{n+\lambda}$
for $i = 1, \ldots, 2n$ **do**
  Compute the representative points: $\boldsymbol{x}^i_{k-1} = \hat{\boldsymbol{x}}_{k-1|k-1} \pm \sigma_i$
  Assign weights: $\omega^i_{k-1} = \frac{1-\omega^0_{k-1}}{2n}$
end for
Compute the predicted mean $\hat{\boldsymbol{x}}_{k|k-1}$ using (21)
Compute the predicted covariance $\mathbf{P}_{k|k-1}$ using (29)
Update step:
for $i = 0, \ldots, 2n$ **do**
  Compute the representative points: $\boldsymbol{x}^i_{k|k-1} = \mathbf{f}_k(\boldsymbol{x}^i_{k-1})$
  Assign weights: $\omega^i_{k|k-1} = \omega^i_{k-1}$
end for
Compute the estimated measurement $\hat{\boldsymbol{y}}_{k|k-1}$ using (23)
Compute the covariance of the predicted observation $\mathbf{P}_{yy}$ using (30)
Compute the cross-covariance of the predicted observation and the predicted state $\mathbf{P}_{xy}$ using (26)
Compute the Kalman gain $\mathbf{K}_k$ using (28)
Compute the estimated mean $\hat{\boldsymbol{x}}_{k|k}$ using (24)
Compute the estimated covariance $\mathbf{P}_{k|k}$ using (25)

B. Gauss-Hermite Filter

An alternative method of determining the representative points with their weights is employed in the Gauss-Hermite Filter (GHF). The GHF is a Gaussian filter that utilizes the Gaussian-Hermite quadrature rule. This is an approximation technique used for evaluating an integral I of the form:

$$I = \int_{\mathbb{R}^n} f(\boldsymbol{x}) \frac{1}{(2\pi)^{n/2}} \exp\left(-\frac{\|\boldsymbol{x}\|^2}{2}\right) d\boldsymbol{x}, \qquad (32)$$

where f is a given nonlinear function. In other words, I is the expectation of a standard normal variable propagated through the nonlinear function f. The integral above is approximated by the *m-th* order quadrature rule I_m:

$$I_m = \sum_{i_1=1}^{m} \dots \sum_{i_n=1}^{m} \omega_{i_1} \dots \omega_{i_n} f(x_{i_1}, \dots, x_{i_n}) = \sum_{i=1}^{m^n} \omega_i f(\boldsymbol{x}_i), \quad (33)$$

where for each $1 \leq i \leq m^n$ the following holds: $\boldsymbol{x}_i = (x_{i_1}, \dots, x_{i_n})^T$ and $\omega_i = \prod_{j=1}^{n} \omega_{i_j}$. For each $1 \leq j \leq n$ the one dimensional *m-th* order quadrature rule $\{(x_l, \omega_l)\}_{l=1}^{m}$ is derived by the following method [105]:

Suppose that $\mathbf{J}$ is a symmetric tridiagonal matrix with zeros on the diagonal and the other entries defined by:

$$\mathbf{J}_{i,j} = \begin{cases} \sqrt{i/2}, & j = i+1 \\ 0, & otherwise \end{cases}$$

The quadrature point x_l is defined as the *l-th* eigenvalue ϵ_l of the matrix $\mathbf{J}$, multiplied by $\sqrt{2}$. The corresponding weight ω_l is set to be equal to the square of the first element of the normalized *l-th* eigenvector $\boldsymbol{v}_l$ of $\mathbf{J}$. To summarize:

$$x_l = \sqrt{2}\epsilon_l, \quad (34)$$

$$\omega_l = ((\boldsymbol{v}_l)_1)^2 \quad (35)$$

The approximation holds for all polynomials of the form $s_1^{i_1} \dots s_n^{i_n}$, with $1 \leq i_k \leq 2m-1$. It is well known that the precision of the estimate increases with the order of the quadrature [106]. However, at the same time the computational burden grows with the rate m^n. Indeed, by (33), the computation of I_m requires m^n function evaluations, i.e., m^n representative points need to be computed. Therefore, even for moderate state dimensions n, higher-order GHF ($m > 5$) requires significant computational load, which makes it impractical for online applications. Furthermore, for $m > 1$ and large n the Gauss-Hermite quadrature rule is numerically more involved than the UT. In the special case of $\lambda = 2$ and $n = 1$ the UT matches the I_3.

Note that this algorithm can easily be generalized for Gaussian variables with arbitrary mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$, simply by replacing f with $\tilde{f}(\boldsymbol{x}) = f\left(\sqrt{\boldsymbol{\Sigma}}^T \boldsymbol{x} + \boldsymbol{\mu}\right)$.

The GHF utilizing the *m-th* order quadrature rule is presented in Algorithm 4.

Algorithm 4 Gauss-Hermite Filter

Require: $\mathbf{P}_{k-1|k-1}$, $\hat{\boldsymbol{x}}_{k-1|k-1}$, m
Compute the one dimensional quadrature rule $\{(x_i, \omega_i)\}_{i=1}^{m}$ using (34)–(35).
for $1 \leq i_1, ..., i_n \leq m$ **do**
Compute the representative points $\boldsymbol{x}_i = (x_{i_1}, \dots, x_{i_n})^T$,
Compute the corresponding weights $\omega_i = \prod_{j=1}^{n} \omega_{i_j}$
end for
Prediction step:
Factorize the posterior covariance: $\mathbf{P}_{k-1|k-1} = \mathbf{S}^T\mathbf{S}$
for $i = 1, \dots, m$ **do**
Compute the representative points:
$\boldsymbol{x}_{k-1}^i = \mathbf{S}^T\boldsymbol{x}_i + \hat{\boldsymbol{x}}_{k-1|k-1}$
Assign weights: $\omega_{k-1}^i = \omega_i$
end for
Compute the predicted mean $\hat{\boldsymbol{x}}_{k|k-1}$ according to (21)
Compute the predicted covariance $\mathbf{P}_{k|k-1}$ according to (22)
Update step:
Factorize the predicted covariance: $\mathbf{P}_{k|k-1} = \tilde{\mathbf{S}}^T\tilde{\mathbf{S}}$
for $i = 1, \dots, m$ **do**
Compute the representative points:
$\boldsymbol{x}_{k|k-1}^i = \tilde{\mathbf{S}}^T\boldsymbol{x}_i + \hat{\boldsymbol{x}}_{k|k-1}$
Assign weights: $\omega_{k|k-1}^i = \omega_i$
end for
Compute the estimated measurement $\hat{\boldsymbol{y}}_{k|k-1}$ using (23)
Compute the covariance of the predicted observation $\mathbf{P}_{yy}$ using (30)
Compute the cross-covariance of the predicted observation and the predicted state $\mathbf{P}_{xy}$ using (26)
Compute the Kalman gain $\mathbf{K}_k$ using (28)
Compute the estimated mean $\hat{\boldsymbol{x}}_{k|k}$ using (24)
Compute the estimated covariance $\mathbf{P}_{k|k}$ using (25)

C. Central Difference Filter

To choose the representative points the Central Difference Filter (CDF) or Divided Difference Filter [18], [87] uses a different method than the previously discussed UKF and GHF. The CDF algorithm is based on the central difference approximation of the integral (32). The basic feature of this method is to approximate the nonlinear function f with a quadratic function P_2 defined by:

$$P_2(\boldsymbol{x}) = f(\mathbf{0}) + \boldsymbol{a}\boldsymbol{x} + \frac{1}{2}\boldsymbol{x}^T\mathbf{H}\boldsymbol{x},$$

where the vector $\boldsymbol{a} = (a_i)$ and the symmetric matrix $\mathbf{H} = (\mathbf{H}_{i,j})$ are given by [17]:

$$a_i = \frac{f(h\mathbf{e}_i) - f(-h\mathbf{e}_i)}{2h}, \quad 1 \leq i \leq n$$

$$\mathbf{H}_{i,i} = \frac{f(h\mathbf{e}_i) - 2f(\mathbf{0}) + f(-h\mathbf{e}_i)}{h^2}, \quad 1 \leq i \leq n$$

$$\mathbf{H}_{i,j} = \frac{f(h\mathbf{e}_i + h\mathbf{e}_j) - f(-h\mathbf{e}_i) - f(-h\mathbf{e}_j) + f(\mathbf{0})}{h^2}, \quad 1 \leq i < j \leq n.$$

Here $h > 0$ is a chosen step size and $(\mathbf{e}_i)$ is a canonical basis for $\mathbb{R}^n$. Note that the exact value of h is not specified a priori, hence an additional degree of freedom is added to the filter. More details concerning the filtering applications of central difference approximations can be found in [17], [18], [107].

The central difference approximation of a Gaussian variable with mean $\bar{\boldsymbol{x}}$ and covariance $\mathbf{P} = \mathbf{S}^T\mathbf{S}$ is given by $2n+1$ representative points with the corresponding weights:

$$\boldsymbol{x}_0 = \bar{\boldsymbol{x}} \qquad \omega_0 = \frac{h^2 - n}{h^2},$$

$$\boldsymbol{x}_i = \bar{\boldsymbol{x}} \pm \mathbf{S}^T h\mathbf{e}_i \qquad \omega_i = \frac{1}{2h^2}.$$

Note that by such a definition the weight of the central point ω_{k-1}^0 can be negative.

The CDF employs the central difference approximation in both the prediction and the update steps of the filtering algorithm. The complete CDF is presented in Algorithm 5.

Algorithm 5 Central Difference Filter

Require: h, $\mathbf{P}_{k-1|k-1}$, $\hat{\boldsymbol{x}}_{k-1|k-1}$
Prediction step:
Factorize the posterior covariance: $\mathbf{P}_{k-1|k-1} = \mathbf{S}^T\mathbf{S}$
Set the central point: $\boldsymbol{x}^0_{k-1} = \hat{\boldsymbol{x}}_{k-1|k-1}$
Set the central weight: $\omega^0_{k-1} = \frac{h^2-n}{h^2}$
for $i = 1, \ldots, 2n$ **do**
Compute the representative points:
$\boldsymbol{x}^i_{k-1} = \hat{\boldsymbol{x}}_{k-1|k-1} \pm \mathbf{S}^T h\mathbf{e}_i$
Assign weights: $\omega^i_{k-1} = \frac{1}{2h^2}$
end for
Compute the predicted mean $\hat{\boldsymbol{x}}_{k|k-1}$ according to (21)
Compute the predicted covariance $\mathbf{P}_{k|k-1}$ according to (22)
Update step:
Factorize the predicted covariance: $\mathbf{P}_{k|k-1} = \tilde{\mathbf{S}}^T\tilde{\mathbf{S}}$
Set the central point: $\boldsymbol{x}^0_{k|k-1} = \hat{\boldsymbol{x}}_{k|k-1}$
Set the central weight: $\omega^0_{k|k-1} = \frac{h^2-n}{h^2}$
for $i = 1, \ldots, 2n$ **do**
Compute the representative points:
$\boldsymbol{x}^i_{k|k-1} = \hat{\boldsymbol{x}}_{k|k-1} \pm \tilde{\mathbf{S}}^T h\mathbf{e}_i$
Assign weights: $\omega^i_{k|k-1} = \frac{1}{2h^2}$
end for
Compute the estimated measurement $\hat{\boldsymbol{y}}_{k|k-1}$ using (23)
Compute the covariance of the predicted observation $\mathbf{P}_{yy}$ using (30)
Compute the cross-covariance of the predicted observation and the predicted state $\mathbf{P}_{xy}$ using (26)
Compute the Kalman gain $\mathbf{K}_k$ using (28)
Compute the estimated mean $\hat{\boldsymbol{x}}_{k|k}$ using (24)
Compute the estimated covariance $\mathbf{P}_{k|k}$ using (25)

When the parameter h is chosen to be small the central-difference approximation is based on points that are close to the center (mean). When h is large the approximation accounts for the points located at the tails of the Gaussian distribution.

D. Example: Prediction step

To illustrate the advantages that the LRKF filters have over the EKF we will use an example of the nonlinear noise-free process given by [87]:

$$\mathbf{x}_{k+1}(1) = (x_k(1))^2, \tag{36a}$$
$$\mathbf{x}_{k+1}(2) = x_k(1) + 3x_k(2). \tag{36b}$$

To better see the differences between the EKF, the UKF, the CDF and the GHF we focus only on a one step ahead prediction problem. The analysis of the update step follows the same steps [87].

Assume that at time step k the state $\mathbf{x}_k$ is normally distributed with mean $\mathbf{x}_{k|k} = [10\ 15]^T$ and covariance $\mathbf{P}_{k|k} = \begin{bmatrix} 36 & 0 \\ 0 & 3,600 \end{bmatrix}$. We want to predict the distribution of the state $\mathbf{x}_{k+1|k}$. The linearization method that is employed by the EKF yields:

$$\hat{\mathbf{x}}^{\text{EKF}}_{k+1|k} = \begin{bmatrix} 100 \\ 55 \end{bmatrix}, \mathbf{P}^{\text{EKF}}_{k+1|k} = \begin{bmatrix} 14,400 & 720 \\ 720 & 32,436 \end{bmatrix}.$$

The UKF with parameter $\lambda = 1$ approximates the distribution of $\mathbf{x}_{k+1|k}$ with five points:

$$\begin{bmatrix} 100 \\ 55 \end{bmatrix}, \begin{bmatrix} 416 \\ 65 \end{bmatrix}, \begin{bmatrix} 0 \\ 45 \end{bmatrix}, \begin{bmatrix} 100 \\ 367 \end{bmatrix}, \begin{bmatrix} 100 \\ -257 \end{bmatrix}$$

weighted $1/3, 1/6, 1/6, 1/6, 1/6$ respectively. Thus, the mean and covariance of the UKF estimate of the state $\mathbf{x}_{k+1|k}$ are given by

$$\hat{\mathbf{x}}^{\text{UKF}}_{k+1|k} = \begin{bmatrix} 136 \\ 55 \end{bmatrix}, \mathbf{P}^{\text{UKF}}_{k+1|k} = \begin{bmatrix} 16,992 & 720 \\ 720 & 32,436 \end{bmatrix}.$$

For the CDF there are also five representative points, e.g., for $h = 2$ we have

$$\begin{bmatrix} 100 \\ 55 \end{bmatrix}, \begin{bmatrix} 484 \\ 67 \end{bmatrix}, \begin{bmatrix} 4 \\ 43 \end{bmatrix}, \begin{bmatrix} 100 \\ 415 \end{bmatrix}, \begin{bmatrix} 100 \\ -305 \end{bmatrix}$$

weighted $1/2, 1/4, 1/4, 1/4, 1/4$ respectively. From these the mean and variance are computed

$$\hat{\mathbf{x}}^{\text{CDF}}_{k+1|k} = \begin{bmatrix} 136 \\ 55 \end{bmatrix}, \mathbf{P}^{\text{CDF}}_{k+1|k} = \begin{bmatrix} 18,288 & 720 \\ 720 & 32,436 \end{bmatrix}.$$

The number of representative points utilized by the GHF depends on the quadrature order m. The smallest feasible order is $m = 2$ which yields the quadrature rule $\{(x_l, \omega_l)\} = \{(1, 1/2), (-1, 1/2)\}$ from which the representative points are computed:

$$\begin{bmatrix} 16 \\ -131 \end{bmatrix}, \begin{bmatrix} 16 \\ 229 \end{bmatrix}, \begin{bmatrix} 256 \\ -119 \end{bmatrix}, \begin{bmatrix} 256 \\ 241 \end{bmatrix}$$

weighted $1/4, 1/4, 1/4, 1/4$ respectively.

$$\hat{\mathbf{x}}^{\text{GHF}}_{k+1|k} = \begin{bmatrix} 136 \\ 55 \end{bmatrix}, \mathbf{P}^{\text{GHF}}_{k+1|k} = \begin{bmatrix} 14,400 & 720 \\ 720 & 32,436 \end{bmatrix}.$$

Instead of analyzing algebraic properties of all the covariance matrices $\mathbf{P}_{k+1|k}$ obtained by each filter it is convenient to look at the error ellipses that they yield:

$$\left(\mathbf{x} - \hat{\mathbf{x}}_{k+1|k}\right)^T \left(\mathbf{P}_{k+1|k}\right)^{-1} \left(\mathbf{x} - \hat{\mathbf{x}}_{k+1|k}\right) = 1.$$

The estimated means and error ellipses obtained by the EKF, the UKF with $\lambda = 1$, the CDF with $h = 1/2$ and the GHF with $m = 2$ are all compared in Figure 3. As a reference, which is labelled as "true" error set, we use the error ellipse defined by the sample mean and sample covariance obtained from 10^6 Monte Carlo experiments. These are given by

$$\hat{\mathbf{x}}^{\text{MC}}_{k+1|k} = \begin{bmatrix} 136 \\ 55 \end{bmatrix}, \mathbf{P}^{\text{MC}}_{k+1|k} = \begin{bmatrix} 17039 & 757 \\ 757 & 32436 \end{bmatrix}.$$

As can be observed, only the EKF yields an estimated mean that does not coincide with the "true mean". The UKF, the CDF and the GHF all provide accurate estimates of the "true" mean but the error ellipses that they produce are different. With such a choice of the parameters ($\lambda = 1$ for the UKF, $h = 1/2$ for the CDF, $m = 2$ for the GHF) the error ellipse obtained by the UKF is the closest to the true one. However, with different parameter setting we can tune the error ellipses

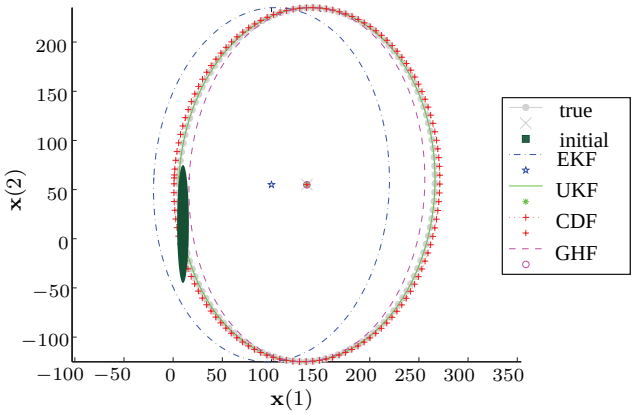

Fig. 3: Error ellipses of the EKF (dot-dashed line), the UKF (solid line), the CDF (crosses) and the GHF (dashed line) compared with the "true" error ellipse (filled circles) obtained by 10^6 Monte Carlo experiments. The shaded area denotes the initial covariance. The means of the UKF (asterisk), the CDF (cross) and the GHF (circle) coincide with the true mean (large x).

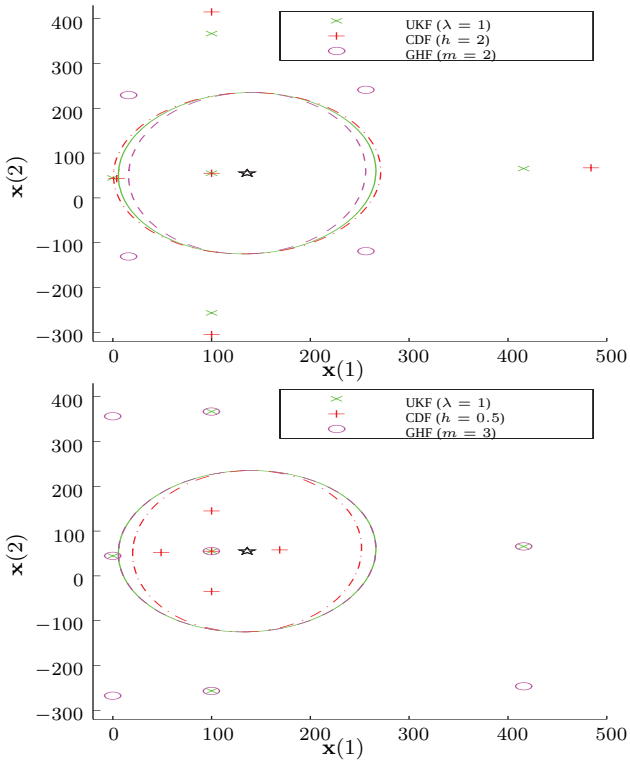

Fig. 4: Representative points of the UKF (x), the CDF (crosses) and the GHF (circles) and the error ellipses that correspond to covariance matrices obtained by the UKF (solid line), the CDF (dashed-dotted line) and the GHF (dashed line) for $\lambda = 1, h = 2, m = 2$ (above) and $\lambda = 1, h = 1/2, m = 3$ (below). Regardless of the parameters' setting all filters yield the same mean (pentagram).

to more desirable shapes. In general, by decreasing the value of the parameter h we will shrink the ellipses $\mathbf{P}^{\mathrm{CDF}}_{k+1|k}$, and by increasing the quadrature order m we are able to expand the error ellipses $\mathbf{P}^{\mathrm{GHF}}_{k+1|k}$. Representative points produced by the UKF, the CDF and the GHF together with the corresponding error ellipses for two parameter settings are presented in Figure 4.

V. GAUSSIAN SUM APPROXIMATIONS: GSF

So far the discussion has been restricted to systems with Gaussian process and measurement noises. Although this type of stochasticity is most commonly used in modeling real-life processes, in a number of situations one has to deal with non-Gaussian random variables that influence the process or the measurement model [108]. The filters discussed in Sections III–IV assume Gaussian noises, hence, when this assumption is violated they no longer perform as expected. Furthermore, even if the noises are Gaussian, the nonlinearities of the state model $\mathbf{f}_k$ and the observation model $\mathbf{h}_k$ might produce densities $p(\mathbf{x}_k|\mathcal{Y}_{k-1})$ or $p(\mathbf{x}_k|\mathcal{Y}_k)$ that cannot be accurately approximated by a single normal variable [109]. A possible solution to these problems is the *Gaussian Sum Filter* (GSF) that is described in this section.

A. GSF Algorithm

The GSF is based on the theoretical result that an arbitrary probability distribution $p(\boldsymbol{x})$ can be approximated by a density $p^N_A(\boldsymbol{x})$ of a form:

$$p^N_A(\boldsymbol{x}) = \sum_{i=1}^{N} a^i \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}^i, \boldsymbol{\Sigma}^i),$$

where for each $1 \leq i \leq N$, $\mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}^i, \boldsymbol{\Sigma}^i)$ is a probability density of a normal distribution, with the mean $\boldsymbol{\mu}^i$ and the covariance $\boldsymbol{\Sigma}^i$, evaluated at $\mathbf{x}$, and a^i are nonnegative weights that sum up to one. The density $p^N_A(\boldsymbol{x})$ uniformly converges to the original density $p(\boldsymbol{x})$ as the number of terms N increases and each covariance matrix $\boldsymbol{\Sigma}_i$ approaches the zero matrix (see [102], [110], and the references therein). Before the *Gaussian Sum* (GS) approximation can be used, one has to specify the parameters $a^i, \boldsymbol{\mu}^i, \boldsymbol{\Sigma}^i$. These are usually given as solutions of a certain optimization algorithm. The choice of the optimization method is not trivial, and in general depends on the particular estimation problem. Different methods for parameter selection are discussed in [111]. Another approach to the problem, which is based on expectation-maximization algorithms is derived in [112].

As always, there is a tradeoff between computational complexity and the accuracy of the approximation. If one uses too many terms in the summations, the computational time will increase and the filter will no longer be feasible for online applications. On the other hand, if there are too few terms in the GS, the algorithm will produce a poor approximation of the true densities.

The main feature of the GSF is the use of the GS approximation of both the predicted state density $p(\mathbf{x}_k|\mathcal{Y}_{k-1})$ and the posterior density $p(\mathbf{x}_k|\mathcal{Y}_k)$. At each time step k the

aforementioned densities are assumed to be given by:

$$p(\mathbf{x}_k|\mathcal{Y}_{k-1}) = \sum_{i=1}^{N_1} a^i_{k|k-1}\mathcal{N}\left(\mathbf{x}_k; \boldsymbol{\mu}^i_{k|k-1}, \boldsymbol{\Sigma}^i_{k|k-1}\right), \quad (37)$$

$$p(\mathbf{x}_k|\mathcal{Y}_k) = \sum_{i=1}^{N_2} a^i_{k|k}\mathcal{N}\left(\mathbf{x}_k; \boldsymbol{\mu}^i_{k|k}, \boldsymbol{\Sigma}^i_{k|k}\right). \quad (38)$$

As it was stated before, there is much flexibility in choosing the weights a^i and the Gaussian parameters $\boldsymbol{\mu}^i$ and $\boldsymbol{\Sigma}^i$. Note that in general the number of terms N_1 in (37) does not have to be equal to the number of terms N_2 in (38).

One might consider the GSF as a collection of nonlinear Kalman Filters, such as the ones described in the previous sections, working in parallel. Indeed, in the original formulation of Alspach and Sorensen [111], the GSF that they derived is composed of the parallel EKFs. A GSF that exploits UKFs is presented in [113], whereas a GHF-based GSF can be found in [102].

The filtering proceeds as follows. Let us assume that at time step $k-1$ the posterior density $p(\mathbf{x}_{k-1}|\mathcal{Y}_{k-1})$ is represented as a sum of Gaussian densities, i.e.,

$$p(\mathbf{x}_{k-1}|\mathcal{Y}_{k-1}) = \sum_{i=1}^{K} \alpha^i_{k-1|k-1}\mathcal{N}\left(\mathbf{x}_{k-1}; \hat{\boldsymbol{x}}^i_{k-1|k-1}, \mathbf{P}^i_{k-1|k-1}\right),$$

where $\alpha^i_{k-1|k-1}$ are weights that sum up to one, and $\hat{\boldsymbol{x}}^i_{k-1|k-1}$ and $\mathbf{P}^i_{k-1|k-1}$ are the *i*-th estimate of the mean and the covariance, respectively. Furthermore, let us also approximate the state noise $\mathbf{v}_k$ by a GS:

$$p(\mathbf{v}_k) = \sum_{j=1}^{L} \alpha^j_{v,k}\mathcal{N}\left(\mathbf{v}_k; \hat{\mathbf{v}}^j_k, \mathbf{P}^j_{v,k}\right),$$

with weights $\alpha^j_{v,k}$, means $\hat{\mathbf{v}}^j_k$, and covariances $\mathbf{P}^j_{v,k}$ chosen to match the non-Gaussian random variable $\mathbf{v}_k$. Then for each pair $\{(i,j), i = 1,...,K, j = 1,...,L\}$ the (i,j)-th component of the predicted state density is computed by the nonlinear KF of one's choice. The predicted state density is thus given by:

$$p\left(\mathbf{x}_k|\mathcal{Y}_{k-1}\right) = \sum_{i=1}^{K}\sum_{j=1}^{L} \alpha^{i,j}_{k|k-1}\mathcal{N}\left(\mathbf{x}_k; \hat{\boldsymbol{x}}^{i,j}_{k|k-1}, \mathbf{P}^{i,j}_{k|k-1}\right),$$

where the weights $\alpha^{i,j}_{k|k-1}$ are computed as:

$$\alpha^{i,j}_{k|k-1} = \alpha^i_{k-1|k-1}\alpha^j_{v,k},$$

and $\hat{\boldsymbol{x}}^{i,j}_{k|k-1}$ and $\mathbf{P}^{i,j}_{k|k-1}$ are estimates of the mean and the covariance, respectively, that are obtained by the application of one of the filters described in Sections III–IV to the model with index (i,j). To perform the update step, we again use the GS to approximate the observation noise $\mathbf{w}_k$:

$$p(\mathbf{w}_k) = \sum_{l=1}^{M} \alpha^l_{w,k}\mathcal{N}\left(\mathbf{w}_k; \hat{\mathbf{w}}^l_k, \mathbf{P}^l_{w,k}\right).$$

Next, for each tuple $\{(i,j,l), i = 1,...,K, j = 1,...,L, l = 1,...,M\}$ the update step is performed by a nonlinear KF of one's choice (EKF, UKF, GHF, etc.). Finally, the separate steps are combined, resulting in the posterior density:

$$p\left(\mathbf{x}_k|\mathcal{Y}_k\right) = \sum_{i=1}^{K}\sum_{j=1}^{L}\sum_{l=1}^{M} \alpha^{i,j,l}_{k|k}\mathcal{N}\left(\mathbf{x}_k; \hat{\boldsymbol{x}}^{i,j,l}_{k|k}, \mathbf{P}^{i,j,l}_{k|k}\right), \quad (39)$$

where the weights $\alpha^{i,j,l}_{k|k}$ are given by:

$$\alpha^{i,j,l}_{k|k} = \frac{\alpha^{i,j}_{k|k-1}\alpha^l_{w,k}p_{i,j,l}\left(\boldsymbol{y}_k|\mathcal{Y}_{k-1}\right)}{\sum_{i=1}^{K}\sum_{j=1}^{L}\sum_{l=1}^{M}\alpha^{i,j}_{k|k-1}\alpha^l_{w,k}p_{i,j,l}\left(\boldsymbol{y}_k|\mathcal{Y}_{k-1}\right)},$$

and the mean $\hat{\boldsymbol{x}}^{i,j,l}_{k|k}$ and the covariance $\mathbf{P}^{i,j,l}_{k|k}$ are obtained from the chosen nonlinear KF applied separately to each triple (i,j,l). In the above formula the term $p_{i,j,l}\left(\boldsymbol{y}_k|\mathcal{Y}_{k-1}\right)$ denotes the (i,j,l)-th component of a PDF of observing $\boldsymbol{y}_k$ at time step *k* given the past observations $\mathcal{Y}_{k-1}$, which can be approximated by the Gaussian:

$$p_{i,j,l}\left(\boldsymbol{y}_k|\mathcal{Y}_{k-1}\right) = \mathcal{N}\left(\boldsymbol{y}_k; \hat{\boldsymbol{y}}^{i,j,l}_{k|k-1}, \mathbf{P}^{i,j,l}_{yy}\right).$$

Algorithm 6 summarizes the GSF that applies the UKF ($\boldsymbol{x}_\sigma$ and ω_σ are computed according to Algorithm 3) to each component of the GS approximation in both the prediction and in the update stage. Note that if one replaces the UKF with another nonlinear filter, e.g., the EKF, the general structure of Algorithm 6 remains intact. Indeed, the two algorithms are different only in the formulas for the means: $\hat{\boldsymbol{x}}_{k|k-1}, \hat{\boldsymbol{x}}_{k|k}, \hat{\boldsymbol{y}}_{k|k-1}$ and the covariances: $\mathbf{P}_{k|k-1}, \mathbf{P}_{k|k}, \mathbf{P}_{yy}$.

B. Reduction methods

In the general framework presented in Section V-A, at the beginning of the algorithm there are K components in the summation, whereas the final number of terms to sum up is KLM. At the next filtering step the algorithm starts with KLM initial expressions, and hence it finishes with KL^2M^2. After *k* steps there are KL^kM^k terms to sum up. This means that as the filtering proceeds, the number of the expressions in the summation grows exponentially. Therefore, in its basic form, the GSF has a very limited practical use.

To overcome this potential drawback of the GSF, several techniques have been developed to reduce the number of terms in the GS approximations [17], [102], [114]–[117]. Among the popular methods are:

1. Pruning: In this approach the mixture components with negligible weights are discarded from the GS, whereas the remaining terms have the weights uniformly rescaled so that the GS forms a probability density function. Depending on the problem, one might discard every component which has the weight smaller than a fixed threshold ϵ or terms that have the cumulative weight smaller than ϵ [102].

2. Merging: When using this method one joins the Gaussian densities that are close to each other with respect to a certain distance, namely the Mahalonobis distance [102], [114], [115], [118]:

$$d^2_{ij} = \frac{\alpha^i\alpha^j}{\alpha^i+\alpha^j}\left(\hat{\boldsymbol{x}}^i - \hat{\boldsymbol{x}}^j\right)^T\left(\mathbf{P}^i + \mathbf{P}^j\right)^{-1}\left(\hat{\boldsymbol{x}}^i - \hat{\boldsymbol{x}}^j\right).$$

Algorithm 6 Gaussian Sum Filter as a collection of UKFs

Require: $\left\{\left(\alpha^i_{k-1|k-1}, \hat{\boldsymbol{x}}^i_{k-1|k-1}, \mathbf{P}^i_{k-1|k-1}\right)\right\}_{i=1}^{K}$, $\left\{\left(\alpha^j_{v,k}, \hat{\mathbf{v}}^j_k, \mathbf{P}^j_{v,k}\right)\right\}_{j=1}^{L}$, $\left\{\left(\alpha^l_{w,k}, \hat{\mathbf{w}}^l_k, \mathbf{P}^l_{w,k}\right)\right\}_{l=1}^{M}$,

Prediction step:

for $i = 1, \ldots, K, j = 1, \ldots, L$ **do**

Compute the predicted mean:

$\hat{\boldsymbol{x}}^{i,j}_{k|k-1} = \sum_\sigma \omega^{i,j}_\sigma \mathbf{f}_k\left(\boldsymbol{x}^{i,j}_\sigma\right)$

Compute the predicted covariance:

$$\mathbf{P}^{i,j}_{k|k-1} = \sum_\sigma \omega^{i,j}_\sigma \left(\mathbf{f}_k\left(\boldsymbol{x}^{i,j}_\sigma\right) - \hat{\boldsymbol{x}}^{i,j}_{k|k-1}\right)\left(\mathbf{f}_k\left(\boldsymbol{x}^{i,j}_\sigma\right) - \hat{\boldsymbol{x}}^{i,j}_{k|k-1}\right)^T$$

Compute the associated weight:

$\alpha^{i,j}_{k|k-1} = \alpha^i_{k-1|k-1}\alpha^j_{v,k}$

end for

Approximate the predicted state density with the Gaussian Sum:

$$p\left(\mathbf{x}_k|\mathcal{Y}_{k-1}\right) = \sum_{i=1}^{K}\sum_{j=1}^{L} \alpha^{i,j}_{k|k-1}\mathcal{N}\left(\mathbf{x}_k; \hat{\mathbf{x}}^{i,j}_{k|k-1}, \mathbf{P}^{i,j}_{k|k-1}\right)$$

Update step:

for $i = 1, \ldots, K, j = 1, \ldots, L, l = 1, \ldots, M$ **do**

Compute the mean of the predicted observation:

$\hat{\boldsymbol{y}}^{i,j,l}_{k|k-1} = \sum_\sigma \omega^{i,j,l}_\sigma \mathbf{h}_k\left(\boldsymbol{x}^{i,j,l}_\sigma\right)$

Compute the covariance of the predicted observation:

$$\mathbf{P}^{i,j,l}_{yy} = \sum_\sigma \omega^{i,j,l}_\sigma \left(\mathbf{h}_k\left(\boldsymbol{x}^{i,j,l}_\sigma\right) - \hat{\boldsymbol{y}}^{i,j,l}_{k|k-1}\right)\left(\mathbf{h}_k\left(\boldsymbol{x}^{i,j,l}_\sigma\right) - \hat{\boldsymbol{y}}^{i,j,l}_{k|k-1}\right)^T$$

Compute the cross-covariance of the predicted observation and the predicted state:

$$\mathbf{P}^{i,j,l}_{xy} = \sum_\sigma \omega^{i,j,l}_\sigma \left(\boldsymbol{x}^{i,j,l}_\sigma - \hat{\boldsymbol{x}}^{i,j}_{k|k-1}\right)\left(\mathbf{h}_k(\boldsymbol{x}^{i,j,l}_\sigma) - \hat{\boldsymbol{y}}^{i,j,l}_{k|k-1}\right)^T$$

Use (24)-(28) to compute the updated mean:

$\hat{\boldsymbol{x}}^{i,j,l}_{k|k} = \hat{\boldsymbol{x}}^{i,j}_{k|k-1} + \mathbf{K}^{i,j,l}_k(\mathbf{y}_k - \hat{\boldsymbol{y}}^{i,j,l}_{k|k-1})$

Use (25)-(28) to compute the updated covariance:

$\mathbf{P}^{i,j,l}_{k|k} = \mathbf{P}^{i,j}_{k|k-1} - \mathbf{P}^{i,j,l}_{xy}\left(\mathbf{K}^{i,j,l}_k\right)^T$

Compute the associated weight:

$\alpha^{i,j,l}_{k|k} = \frac{\alpha^{i,j}_{k|k-1}\alpha^l_{w,k}\mathcal{N}\left(\mathbf{y}_k; \hat{\boldsymbol{y}}^{i,j,l}_{k|k-1}, \mathbf{P}^{i,j,l}_{yy}\right)}{\sum_{i=1}^{K}\sum_{j=1}^{L}\sum_{l=1}^{M}\alpha^{i,j}_{k|k-1}\alpha^l_{w,k}\mathcal{N}\left(\mathbf{y}_k; \hat{\boldsymbol{y}}^{i,j,l}_{k|k-1}, \mathbf{P}^{i,j,l}_{yy}\right)}$

end for

Approximate the posterior density with the Gaussian Sum:

$$p\left(\mathbf{x}_k|\mathcal{Y}_k\right) = \sum_{i=1}^{K}\sum_{j=1}^{L}\sum_{l=1}^{M} \alpha^{i,j,l}_{k|k}\mathcal{N}\left(\mathbf{x}_k; \hat{\boldsymbol{x}}^{i,j,l}_{k|k}, \mathbf{P}^{i,j,l}_{k|k}\right)$$

This algorithm in general merges mixture terms that have lower weights rather than those that are associated to higher weights [102].

The GS approximations obtained by pruning or merging procedure converge weakly to the exact posterior distribution [117].

3. Integral Squared Error-Based Gaussian Mixture Reduction: In this method one obtains the reduced Gaussian mixture expressions by minimizing the L^2 distance between the original and the reduced densities [17], [102]:

$$\underset{\alpha,\boldsymbol{\mu},\boldsymbol{\Sigma},N}{\operatorname{argmin}} \int \left(p\left(\mathbf{x}_k|\mathcal{Y}_k\right) - \sum_{i=1}^{N}\alpha^i\mathcal{N}\left(\mathbf{x}_k; \boldsymbol{\mu}^i, \boldsymbol{\Sigma}^i\right)\right)^2 d\mathbf{x}_k$$

where $p\left(\mathbf{x}_k|\mathcal{Y}_k\right)$ is the original Gaussian Sum approximation defined by (39), N is the desired number of components in the Gaussian mixture that is usually much smaller than the number of terms in original GS, and $\alpha, \boldsymbol{\mu}, \boldsymbol{\Sigma}$ are the parameters with respect to which the optimization is performed. In some cases instead of the L^2 distance other metrics are used as the optimization criterion [118].

Using one of the aforementioned techniques one has control over the number of terms in the GS, and hence the growing memory requirement ceases to be a problem. However, the reduction procedure, which can be very computationally expensive, has to be preformed at each filtering step. Therefore, depending on the problem, an appropriate choice of the reduction method is crucial to make the GSF an effective online filter. Note that if both the process and the observation noises can be accurately approximated by single Gaussians, no reduction method is necessary because the number of expressions in the GS is constant over the time.

C. Example: Kinematic Model

Let us consider a second order kinematic model in two-dimensional space [65], [119], [120]. The model is described by four states:

$$\mathbf{x}_k = \left[\begin{array}{cccc} p_x(k) & \dot{p}_x(k) & p_y(k) & \dot{p}_y(k) \end{array}\right]^T,$$

where $(p_x(k), p_y(k))$ is the position of the object at time k in XY plane and $\dot{p}_x(k)$ and $\dot{p}_y(k)$ denote the velocity of the object at time k in X-direction and Y-direction respectively. The evolution of the object in discrete-time is modeled by:

$$\mathbf{x}_{k+1} = \left[\begin{array}{cccc} 1 & T_s & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T_s \\ 0 & 0 & 0 & 1 \end{array}\right]\mathbf{x}_k \tag{40a}$$

$$+ \left[\begin{array}{cc} \frac{T_s^2}{2} & 0 \\ T_s & 0 \\ 0 & \frac{T_s^2}{2} \\ 0 & T_s \end{array}\right]\left(\mathbf{f}\left(\mathbf{x}_k\right) - \left[\begin{array}{c} 0 \\ g \end{array}\right]\right) + \mathbf{v}_k, \tag{40b}$$

where the function $\mathbf{f}\left(\mathbf{x}_k\right)$ is given by

$$\mathbf{f}\left(\mathbf{x}_k\right) = -0.5\frac{g}{\beta}\rho\left(\mathbf{x}_k(3)\right)\sqrt{\left(\mathbf{x}_k(2)\right)^2 + \left(\mathbf{x}_k(4)\right)^2}\left[\begin{array}{c} \mathbf{x}_k(2) \\ \mathbf{x}_k(4) \end{array}\right], \tag{41}$$

with parameters being: $T_s = 1s$ the sampling time, $g = 9.81\text{m/s}^2$ the gravitational acceleration, $\beta = 100\text{kg/m}^2$ the ballistic coefficient, $\rho\left(\mathbf{x}_k(3)\right) = 1.754\exp(-1.491\mathbf{x}_k(3))$ the air density (typically modeled as an exponentially decaying function of height [65]). Furthermore, the variable $\mathbf{v}_k$ models random process noise, which is zero-mean Gaussian with the

covariance matrix $\mathbf{Q}_k$ equal to

$$\mathbf{Q}_k = \begin{bmatrix} 33\frac{1}{3} & 50 & 0 & 0 \\ 50 & 100 & 0 & 0 \\ 0 & 0 & 33\frac{1}{3} & 50 \\ 0 & 0 & 50 & 100 \end{bmatrix}. \tag{42}$$

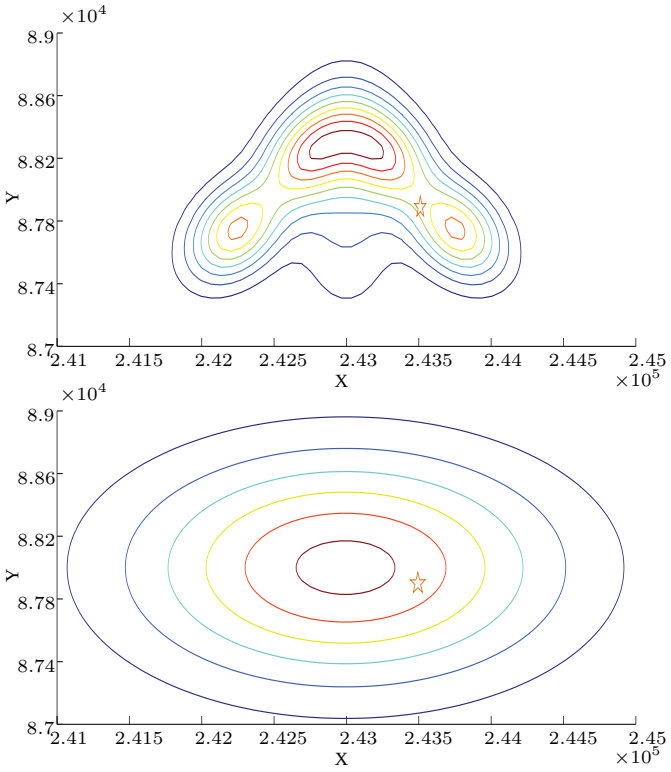

Fig. 5: Contour of the initial pdf of the GSF (above) vs contour of the initial pdf of the UKF (below) both in the XY position plane. The pentagram denotes the true initial state of the system.

For the observation model we assume that at each time step k the range $\mathbf{y}_k(1)$ and bearing $\mathbf{y}_k(2)$ measurements are available [65], [120]. Thus, in the cartesian coordinates the measurement model is given by:

$$\mathbf{y}_k(1) = \sqrt{(\mathbf{x}_k(1))^2 + (\mathbf{x}_k(3))^2} + \mathbf{w}_k(1), \tag{43a}$$

$$\mathbf{y}_k(2) = \arctan\left(\frac{\mathbf{x}_k(3)}{\mathbf{x}_k(1)}\right) + \mathbf{w}_k(2), \tag{43b}$$

where the zero-mean Gaussian variable $\mathbf{w}_k$ models the random measurement noise with covariance matrix $\mathbf{R}_k = \begin{bmatrix} 10^4 & 0 \\ 0 & 0.01 \end{bmatrix}$. With such a choice of $\mathbf{R}_k$ the standard deviation of the range errors is equal to $\sigma_r = 100$m and the standard deviation in bearing errors is given by $\sigma_\theta = 0.1$rad.

We have simulated a trajectory of the ballistic object for $T = 90$s, starting from the initial state $\mathbf{x}_0 = [243.5\text{km}, 1000\text{m/s}, 87.9\text{km}, 0\text{m/s}]^T$. The simulation was repeated 1000 times.

We use the Monte Carlo experiment described above to compare the performance of a five-term GSF with a UKF. The initial condition $\mathbf{x}_0^{\text{UKF}}$ and initial covariance $\mathbf{P}_0^{\text{UKF}}$ for the UKF are given by:

$$\mathbf{x}_0^{\text{UKF}} = \begin{bmatrix} 243 \cdot 10^3 \\ 10^3 \\ 88 \cdot 10^3 \\ 0 \end{bmatrix}, \quad \mathbf{P}_0^{\text{UKF}} = \begin{bmatrix} 10^6 & 0 & 0 & 0 \\ 0 & 100 & 0 & 0 \\ 0 & 0 & 25 \cdot 10^4 & 0 \\ 0 & 0 & 0 & 100 \end{bmatrix}.$$

The initial condition for the GSF is given by five equally weighted Gaussians with means $\boldsymbol{\mu}_i$

$$\boldsymbol{\mu}_1 = \begin{bmatrix} 242,250 \\ 1,000 \\ 87,750 \\ 0 \end{bmatrix}, \boldsymbol{\mu}_2 = \begin{bmatrix} 242,750 \\ 1,000 \\ 88,250 \\ 0 \end{bmatrix}, \boldsymbol{\mu}_3 = \begin{bmatrix} 243,250 \\ 1,000 \\ 88,250 \\ 0 \end{bmatrix},$$

$$\boldsymbol{\mu}_4 = \begin{bmatrix} 243,750 \\ 1,000 \\ 87,750 \\ 0 \end{bmatrix}, \boldsymbol{\mu}_5 = \begin{bmatrix} 243,000 \\ 1,000 \\ 88,000 \\ 0 \end{bmatrix},$$

respectively, and the covariances

$$\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = \begin{bmatrix} \frac{3\cdot 250^2}{4} & 0 & \frac{250^2}{4} & 0 \\ 0 & 100 & 0 & 0 \\ \frac{250^2}{4} & 0 & \frac{3\cdot 250^2}{4} & 0 \\ 0 & 0 & 0 & 100 \end{bmatrix},$$

$$\boldsymbol{\Sigma}_3 = \boldsymbol{\Sigma}_4 = \begin{bmatrix} \frac{3\cdot 250^2}{4} & 0 & -\frac{250^2}{4} & 0 \\ 0 & 100 & 0 & 0 \\ -\frac{250^2}{4} & 0 & \frac{3\cdot 250^2}{4} & 0 \\ 0 & 0 & 0 & 100 \end{bmatrix},$$

$$\boldsymbol{\Sigma}_5 = \begin{bmatrix} 250^2 & 0 & 0 & 0 \\ 0 & 100 & 0 & 0 \\ 0 & 0 & 500^2 & 0 \\ 0 & 0 & 0 & 100 \end{bmatrix}.$$

The contour of the initial distribution of the GSF is visualized in Figure 5. The initial pdf has been chosen to resemble the parabolic shape of the trajectory of the ballistic object. Such a shape cannot be achieved by a single Gaussian distribution.

Figure 6 presents the simulated XY-trajectory of the ballistic object together with the estimates obtained by the GSF and the UKF. In the figure it can be easily observed that the GSF outperforms a single UKF. This is confirmed by the analysis of the RMSE of each filter obtained from 1000 Monte Carlo runs of the system (40)–(43) with the same initial condition and the same noise levels. In Figure 7 the RMSE of the GSF and the UKF are compared with the squared root of the theoretical PCRB that is computed using (20).

VI. CONCLUSIONS AND DISCUSSION

The main objective of this section is to analyze the properties of the Parametric Filters presented in the previous sections. For general nonlinear non-Gaussian systems, there exists no optimal solution to the filtering problem (in the MMSE sense). This means that there are no results stating that a particular filter has the lowest possible MMSE error [8].

We have presented three types of nonlinear parametric filtering methods:

I. Filters based on analytical approximations: EKF, IEKF.
II. Filters based on statistical approximations: UKF, GHF, CDF.
III. Filters based on Gaussian Sum approximations: GSF.

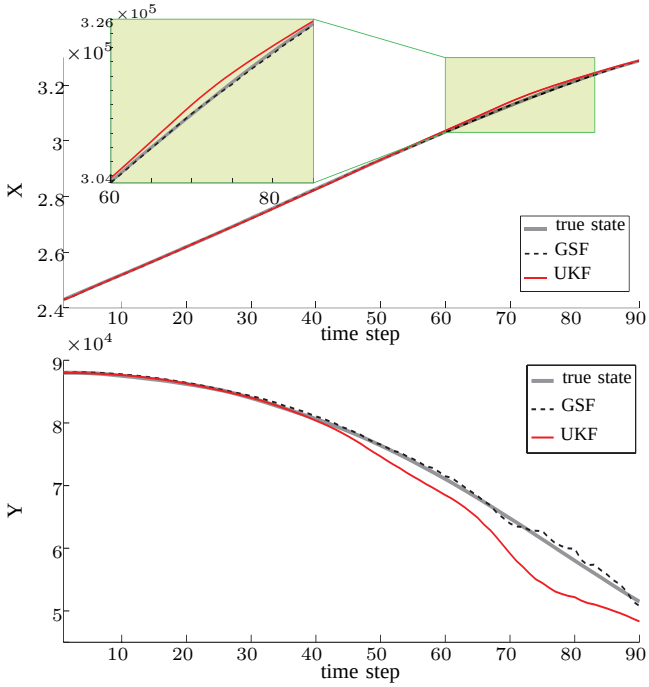

Fig. 6: Tracking of the X-position (above) and Y-position (below) of the ballistic object (thick solid line) by the GSF (dashed line) and the UKF (thin solid line).

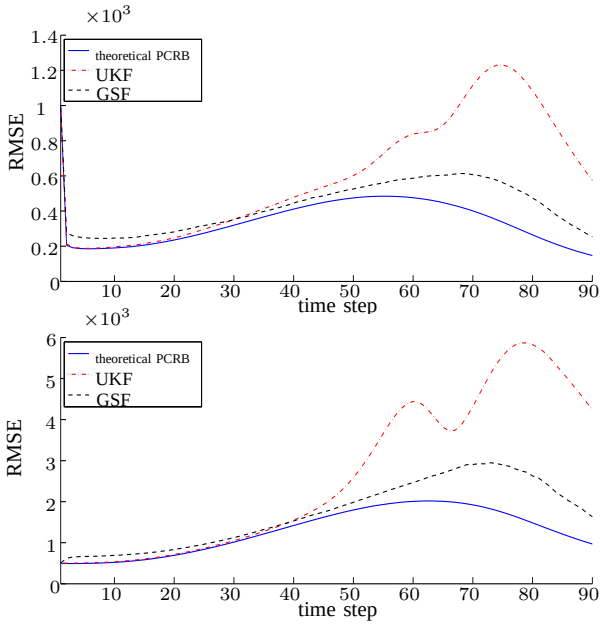

Fig. 7: RMSE of the GSF (dashed line) and the UKF (dashed-dotted line) compared with the square root of the theoretical PCRB (solid line) for X-position (above) and Y-position (below).

The EKF, the IEKF, the UKF, the GHF, and the CDF approximate the predicted (4) and the posterior (5) densities as Gaussians. The EKF and the IEKF utilize the Taylor series expansion to exploit the analytical structure of nonlinear functions $\mathbf{f}_k$ and $\mathbf{h}_k$. The UKF, the GHF and the CDF exploit statistical properties of Gaussian variables that undergo nonlinear transformations. In contrast to the aforementioned methods the GSF approximates the densities (4) and (5) with the sum of Gaussian densities, which are no longer Gaussian.

The Taylor approximation, which is a basic principle of the EKF and the IEKF, requires functions $\mathbf{f}_k$ and $\mathbf{h}_k$ to be differentiable. The UKF, the GHF and the CDF are derivative-free filters, i.e., they can be applied to systems with non-differentiable dynamics. The same applies to the GSF if it uses one of the derivative-free methods.

The numerical complexity of the UKF and the CDF grows linearly with the dimension of the state n, the numerical complexity of the EKF and the IEKF grows quadratically with n, and the complexity of the GHF grows exponentially with n. In the case of the GSF there is no straightforward relation between the dimension of the state space n and the computational complexity of the filter. The latter depends on the number of terms K in the GS that are required for an accurate approximation of the densities (4) and (5). In general, a larger K is necessary for higher dimensions n [81], but the exact relation always depends on the particular structure of the approximated densities.

The Taylor series approximation truncates the higher moments of nonlinear function. Therefore, filters derived from this principle, such as the EKF, are better suited for the systems where functions $\mathbf{f}_k$ and $\mathbf{h}_k$ are mildly nonlinear. From this perspective the strong advantage of the UKF, the GHF and the CDF over the EKF is that these filters match higher-order moments and thus, can handle stronger nonlinearities in the system equations. Among these three filters the UKF has the simplest form and while being more accurate than the EKF, it retains its low computational complexity. The CDF, though similar to the UKF, is able to estimate the state covariance more precisely. This, however, comes with the price of increased computational complexity. The GHF, using sufficiently large quadrature rule, is able to accurately approximate heavy tailed distributions. The disadvantage of the GHF over the EKF, the CDF and the UKF is its large numerical complexity which often yields the GHF impractical for high-frequency online applications.

The performance of the EKF can be improved by using the measurement to minimize linearization errors. This is achieved by the IEKF the trade-off being rise of numerical complexity.

The performance of the EKF, the IEKF, the UKF, the GHF or the CDF can deteriorate if the predicted and the posterior densities cannot be accurately approximated by a single Gaussian. If the system exhibits severely non-Gaussian characteristics, the GSF offers a neat alternative to the aforementioned filters.

The GHF, and to a lesser degree the EKF and the IEKF, suffers from the curse of dimensionality. Therefore, from the computational perspective and the GHF, the EKF, and the IEKF are better suited for small-scale systems, whereas the UKFs and the CDFs are more suitable for large-scale applications. Whenever the nonlinear functions $\mathbf{f}_k$ or $\mathbf{h}_k$ have complicated analytical structures, which make it difficult to compute the Jacobians $\partial\mathbf{f}_k$ or $\partial\mathbf{h}_k$, the derivative-free filters (UKF, CDF, GHF) are numerically preferable over the EKF.

We would like to conclude the article with a brief overview

of freely available implementations of the algorithms discussed throughout the paper. Mathworks provides the MatLab codes for the EKF [121] and the UKF [122]. A very useful overview of open source MatLab and C++ toolboxes used for nonlinear filtering, including KF, EKF and UKF, is provided by Greg Welch and Gary Bishop [123]. A comprehensive collection of MatLab toolboxes suited for nonlinear filtering, among others EKF, UKF, CDF, GSF, is provided by the Identification and Decision Making Research Group at the University of West Bohemia [124].

ACKNOWLEDGEMENT

This research is funded by the dredging company IHC Systems B. V. P. O. Box 41, 3360 AA Sliedrecht, the Netherlands.

REFERENCES

[1] S. Haykin, *Adaptive Filter Theory*. Prentice-Hall, 1991.

[2] J. Weare, "Particle Filtering with Path Sampling and an Application to a Bimodal Ocean Current Model," *Journal of Computational Physics*, vol. 228, pp. 4312–4331, 2009.

[3] M. Murshed, B. Huang, and K. Nandakumar, "Estimation and Control of Solid Oxide Fuel Cell System," *Computers and Chemical Engineering*, vol. 34, pp. 96–111, 2010.

[4] R. Barbieri, L. M. Frank, D. P. Nguyen, M. C. Quirk, V. Solo, M. A. Wilson, and E. N. Brown, "Dynamic Analyses of Information Encoding in Neural Ensembles," *Neural Computation*, vol. 16, pp. 277–307, 2004.

[5] J. Danielsson, "Stochastic Volatility in Asset Prices Estimation with Simulated Maximum Likelihood," *Journal of Econometrics*, vol. 64, pp. 375–400, 1994.

[6] J. Rodríguez-Millán and C. González, "Three Mathematica Supported Proposals for the Discretization of Nonlinear Dynamical Control Systems," *Nonlinear Analysis*, vol. 63, pp. 617–628, 2005.

[7] D. Talay, "Simulation of Stochastic Differential Systems," *Lecture Notes in Physics: Probabilistic Methods in Applied Physics*, vol. 451, pp. 54–96, 1995.

[8] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman Filter: Particle Filters for Tracking Application*. Artech House, 2004.

[9] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A Tutorial on Particle Filters for Online Nonlinear/Non- Gaussian Bayesian Tracking," *IEEE Transactions on Signal Processing*, vol. 50, pp. 174–188, 2002.

[10] T. R. Bewley and A. S. Sharma, "Efficient Grid-Based Bayesian Estimation of Nonlinear Low-Dimensional Systems with Sparse Non-Gaussian PDFs," *Automatica*, vol. 48, pp. 1286–1290, 2012.

[11] S. Kramer and H. Sorenson, "Recursive Bayesian Estimation Using Piece-wise Constant Approximations," *Automatica*, vol. 24, pp. 789–801, 1988.

[12] M. Šimandl, J. Královec, and T. S′oderstr′om, "Advanced Point-Mass Method for Nonlinear State Estimation," *Automatica*, vol. 42, pp. 1133–1145, 2006.

[13] M. Šimandl, J. Královec, and T. S′oderstr′om, "Anticipative Grid Design in Point-Mass Approach to Nonlinear State Estimation," *IEEE Transactions on Automatic Control*, vol. 47, pp. 699–702, 2002.

[14] V. E. Beneš, "Exact Finite-Dimensional Filters for Certain Diffusions with Nonlinear Drift," *Stochastics*, vol. 5, pp. 65–92, 1981.

[15] F. E. Daum, "Exact Finite-Dimensional Nonlinear Filters," *IEEE Transactions on Automatic Control*, vol. 31, pp. 616–622, 1986.

[16] F. E. Daum, "Nonlinear Filters: Beyond the Kalman Filter," *IEEE A& E Systems Magazine*, vol. 20, no. 8, pp. 57–69, 2005.

[17] K. Ito and K. Xiong, "Gaussian Filters for Nonlinear Filtering Problems," *IEEE Transactions on Automatic Control*, vol. 45, pp. 910–927, 2000.

[18] M. Nørgaard, N. Poulsen, and O. Ravn, "New Developments in State Estimation for Nonlinear Systems," *Automatica*, vol. 36, pp. 1627–1638, 2000.

[19] S. Julier and J. Uhlmann, "Unscented Filtering and Nonlinear Estimation," *Proceedings of the IEEE*, vol. 92, pp. 401–422, 2004.

[20] S. Julier and J. Uhlmann, "Corrections to Unscented Filtering and Nonlinear Estimation," *Proceedings of the IEEE*, vol. 92, p. 1958, 2004.

[21] A. Doucet, S. Godsill, and C. Andrieu, "On Sequential Monte Carlo Sampling Methods for Bayesian Filtering," *Statistics and Computing*, vol. 10, pp. 197–208, 2000.

[22] D. Cristian and A. Doucet, "A Survey of Convergence Results on Particle Filtering Methods for Practitioners," *IEEE Transactions on Signal Processing*, vol. 50, pp. 736–746, 2002.

[23] H. A. Blom and E. A. Bloem, "Optimal Decomposed Particle Filtering of Two Closely Spaced Gaussian Targets," in *Proceedings of the IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*, (Orlando, Florida), pp. 7895–7901, 2011.

[24] F. L. Gland, F. Monbet, and V. D. Tran, "Large Sample Asymptotics for the Ensemble Kalman Filter," in *The Oxford Handbook of Nonlinear Filtering* (D. Crisan and B. Rozovskii, eds.), ch. 22, pp. 598–631, Oxford University Press, 2011.

[25] G. Burgers, P. J. van Leeuwen, and G. Evensen, "Analysis Scheme in the Ensemble Kalman Filter," *Monthly Weather Review*, vol. 126, pp. 1719–1724, 1998.

[26] G. Evensen, "The Ensemble Kalman Filter: Theoretical Formulation and Practical Implementation," *Ocean Dynamics*, vol. 53, pp. 343–367, 2003.

[27] G. Evensen, *Data Assimilation: The Ensemble Kalman Filter*. Springer-Verlag, Berlin, 2006.

[28] S. Särkkä, "Bayesian Estimation of Time-Varying Systems: Discrete-Time Systems." Written material for the course held in Spring 2012, 2012.

[29] I. Arasaratnam and S. Haykin, "Cubature Kalman Filters," *IEEE Transactions on Automatic Control*, vol. 54, pp. 1254–1269, 2009.

[30] T. Yang, P. Mehta, and S. Meyn, "A Mean-field Control-oriented Approach to Particle Filtering," in *Proceedings of the American Control Conference*, (San Francisco, California), pp. 2037–2043, 2011.

[31] S. Roweis and Z. Ghahramani, "Learning Nonlinear Dynamical Systems using the Expectation-Maximization Algorithms," in *Kalman Filtering and Neural Networks* (S. Haykin, ed.), ch. 6, pp. 175–220, Wiley, 2001.

[32] O. Cappé, E. Moulines, and T. Rydén, *Inference in Hidden Markov Models*. Springer, 2009.

[33] O. Cappé and E. Moulines, "On-line Expectation-Maximization Algorithm for Latent Data Models," *Journal of the Royal Statistical Society Series B*, vol. 71, pp. 593–613, 2009.

[34] D. Simon, *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*. Wiley-Interscience, 2006.

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

[36] G. Kitagawa, "Non-Gaussian State-Space Modeling of Nonstationary Time Series," *Journal of the American Statistical Association*, vol. 82, pp. 1032–1041, 1987.

[37] M. Šimandl and J. Duník, "Derivative-Free Estimation Methods: New Results and Performance Analysis," *Automatica*, vol. 45, pp. 1749–1757, 2009.

[38] S. J. Godsill, A. Doucet, and M. West, "Monte Carlo Smoothing for Nonlinear Time Series," *Journal of the American Statistical Association*, vol. 99, no. 465, pp. 156–168, 2004.

[39] H. E. Rauch, F. Tung, and C. T. Striebel, "Maximum Likelihood Estimates of Linear Dynamic Systems," *AIAA Journal*, vol. 3, no. 8, pp. 1445–1450, 1965.

[40] M. Šimandl, J. Královec, and P. Tichavsky, "Filtering, Predictive, and Smoothing Cramér-Rao Bounds for Discrete-Time Nonlinear Dynamic Systems," *Automatica*, vol. 37, pp. 1703–1716, 2001.

[41] S. Särkkä and J. Hartikainen, "On Gaussian Optimal Smoothing of Non-Linear State Space Models," *IEEE Transaction on Automatic Control*, vol. 55, no. 8, pp. 1938–1941, 2010.

[42] H. A. Blom and E. A. Bloem, "Decomposed Particle Filtering and Track Swap Estimation in Tracking Two Closely Spaced Targets," in *Proceedings of the 14th International Conference on Information Fusion (FUSION)*, (Chicago, Illinois), pp. 1–8, 2011.

[43] E. H. Aoki, A. Bagchi, P. Mandal, and Y. Boers, "A Theoretical Analysis of Bayes-optimal Multi-target Tracking and Labelling," tech. rep., University of Twente, the Netherlands, 2011.

[44] S. Rangan, A. Fletcher, and V. Goyal, "Asymptotic Analysis of MAP Estimation via the Replica Method and Compressed Sensing," in *Proceedings of the 23rd Annual Conference on Neural Information Processing Systems*, (Vancouver, Canada), pp. 1545–1553, 2009.

[45] J.-L. Gauvain and C.-H. Lee, "Maximum a Posteriori Estimation for Multivariate Gaussian Mixture Observations of Markov Chains," *IEEE Transactions on Speech and Audio Processing*, vol. 2, pp. 291–298, 1994.

[46] L. Ljung and S. Gunnarsson, "Adaptation and Tracking in System Identification - A Survey," *Automatica*, vol. 26, pp. 7–21, 1990.

[47] L. H. Matthies, R. Szeliski, and T. Kanade, "Kalman Filter-based Algorithms for Estimating Depth from Image Sequences," *International Journal of Computer Vision*, vol. 3, pp. 209–236, 1989.

[48] G. Welch and G. Bishop, "An Introduction to the Kalman Filter," Tech. Rep. 95-041, Department of Computer Science, University of North Carolina at Chapel Hill., 1995.

[49] B. Øksendal, *Stochastic Differential Equations: An Introduction with Applications*, ch. The Filtering Problem, pp. 81–108. Springer Verlag, 2003.

[50] Y. C. Ho and R. C. Lee, "Bayesian Approach to Problems in Stochastic Estimation and Control," *IEEE Transactions on Automatic Control*, vol. 9, pp. 333–339, 1964.

[51] R. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Transctions of the ASME - Journal of Basic Engineering*, vol. 82, pp. 35–45, 1960.

[52] B. D. O. Anderson and J. B. Moore, *Optimal Filtering*. NJ: Prentice-Hall, 1979.

[53] R. Mehra, "On the Identification of Variances and Adaptive Kalman Filtering," *IEEE Transactions on Automatic Control*, vol. 15, pp. 175–184, 1970.

[54] D. Alspach, "Comments on "On the Identification of Variances and Adaptive Kalman Filtering"," *IEEE Transactions on Automatic Control*, vol. 17, pp. 843–845, 1972.

[55] S. Sangsuk-Iam and T. Bullock, "Analysis of Discrete-Time Kalman Filtering Under Incorrect Noise Covariances," *IEEE Transactions on Automatic Control*, vol. 35, pp. 1304–1308, 1990.

[56] B. M. Akesson, J. B. Jorgensen, N. K. Poulsen, and S. B. Jorgensen, "A Generalized Autocovariance Least-Squares Method for Kalman Filter Tuning," *Journal of Process Control*, vol. 18, pp. 769–779, 2008.

[57] M. Rajamani and J. Rawlings, "Estimation of the Disturbance Structure from Data Using Semidefinite Programming and Optimal Weighting," *Automatica*, vol. 45, pp. 142–148, 2009.

[58] R. Reynolds, "Robust Estimation of Covariance Matrices," *IEEE Transactions on Automatic Control*, vol. 35, pp. 1047–1051, 1990.

[59] J. M. Mendel, *Lessons in Estimation Theory for Signal Processing, Communications, and Control*, ch. State Estimation: Filtering Examples. Prentice Hall, 1995.

[60] V. Pugachev and I. Sinitsyn, *Stochastic Systems: Theory and Applications*. Singapore: World Scientific, 2001.

[61] H. J. Kushner and P. Dupuis, *Numerical Methods for Stochastic Control Problems In Continuous Time*. Applications of Mathematics, New York: Springer-Verlag, 2001.

[62] H. van Trees, *Detection, Estimation, and Modulation Theory: Part I*. Wiley, New York, 1968.

[63] H. Cramer, *Mathematical Methods of Statistics*. Princeton University Press, Princeton, New Yersey, 1946.

[64] C. R. Rao, "Information and the Accuracy Attainable in the Estimation of Statistical Parameters," *Bulletin of Calcutta Mathematical Society*, vol. 37, pp. 81–91, 1945.

[65] A. Farina, B. Ristic, and D. Benvenuti, "Tracking a Ballistic Target: Comparison of Several Nonlinear Filters," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 38, pp. 854–867, 2002.

[66] T. Kerr, "Status of CR-Like Lower Bounds for Nonlinear Filtering," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 25, pp. 590–600, 1989.

[67] P. Tichavsky, C. Muravchik, and A. Nehorai, "Posterior Cramer-Rao Bounds for Discrete-Time Nonlinear Filtering," *IEEE Transactions on Signal Processing*, vol. 46, pp. 1386–1396, 1998.

[68] M. Lei, B. van Wyk, and Y. Qi, "Online Estimation of the Approximate Posterior Cramer-Rao Lower Bound for Discrete-Time Nonlinear Filtering," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 47, pp. 37–57, 2011.

[69] L. Zuo, R. Niu, and P. K. Varshney, "Conditional Posterior Cramer-Rao Lower Bounds for Nonlinear Sequential Bayesian Estimation," *IEEE Transactions on Signal Processing*, vol. 59, pp. 1–14, 2011.

[70] M. Hernandez, B. Ristic, A. Farina, and L. Timmoneri, "A Comparison of Two Cramer-Rao Bounds for Nonlinear Filtering with $p_d < 1$," *IEEE Transactions on Signal Processing*, vol. 52, pp. 2361–2370, 2004.

[71] R. Niu, P. Willett, and Y. Bar-Shalom, "Matrix CRLB Scaling due to Measurements of Uncertain Origin," *IEEE Transactions on Signal Processing*, vol. 49, pp. 1325–1335, 2001.

[72] X. Zhang, P. Willett, and Y. Bar-Shalom, "Dynamic Cramer-Rao Bound for Target Tracking in Clutter," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 41, pp. 1154–1167, 2005.

[73] D. Crisan and B. Rozovskii, *The Oxford Handbook of Nonlinear Filtering*. Oxford University Press, 2011.

[74] R. van Handel, "Nonlinear Filtering and Systems Theory," in *Proceedings of the 19th International Symposium on Mathematical Theory of Networks and Systems*, (Budapest, Hungary), 2010.

[75] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman Filter: Particle Filters for Tracking Application*, ch. Suboptimal Nonlinear Filters, pp. 19–34. Artech House, 2004.

[76] H. Tanizaki, *Nonlinear Filters: Estimation and Applications*, ch. Approximations of Nonlinear Filters, pp. 175–204. Springer Berlin, 1996.

[77] M. Verlaan and A. W. Heemink, "Nonlinearity in Data Assimilation Applications: A Practical Method for Analysis," *Monthly Weather Review*, vol. 129, pp. 1578–1589, 2001.

[78] S. Ali-Löytty, "Efficient Gaussian Mixture Filter for Hybrid Positioning," in *IEEE/ION Position Location and Navigation Symposium*, (Monterey, California), pp. 60–66, 2008.

[79] H. Tanizaki and R. S. Mariano, "Nonlinear Filters based on Taylor Series Expansions," *Communications in Statistics: Theory and Methods*, vol. 25, pp. 1261–1282, 1996.

[80] K. Xiong, H. Zhang, and L. Liu, "Adaptive Robust Extended Kalman Filter for Nonlinear Stochastic Systems," *IET Control Theory Applications*, vol. 3, pp. 239–250, 2008.

[81] S. Julier and J. Uhlmann, "A General Method for Approximating Nonlinear Transformations of Probability Distributions," tech. rep., Robotics Research Group, Department of Engineering Science, University of Oxford, 1996.

[82] S. Bolognani, L. Tubiana, and M. Zigliotto, "Extended Kalman Filter Tuning in Sensorless PMSM Drives," *IEEE Transactions on Industry Applications*, vol. 39, pp. 1741–1747, 2003.

[83] C.-B. Chang and J. Tabaczynski, "Application of State Estimation to Target Tracking," *IEEE Transactions on Automatic Control*, vol. 29, pp. 98–109, 1984.

[84] H. Tanizaki, *Nonlinear Filters: Estimation and Applications*. Springer Berlin, 1996.

[85] A. Jazwinski, *Stochastic Processing and Filtering Theory*. Academic Press, 1970.

[86] B. M. Bell and F. W. Cathey, "The Iterated Kalman Filter Update as a Gauss-Newton Method," *IEEE Transactions on Automatic Control*, vol. 38, pp. 294–297, 1993.

[87] T. Lefebvre, H. Bruyninckx, and J. Schutter, "Kalman Filters for Nonlinear Systems: A Comparison of Performance," *International Journal of Control*, vol. 77, pp. 639–653, 2004.

[88] K. Spingarn, "Passive Position Location Estimation Using the Extended Kalman Filter," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 23, pp. 558–567, 1987.

[89] R. Wishner, J. Tabaczynski, and M. Athans, "A Comparison of Three Non-Linear Filters," *Automatica*, vol. 5, pp. 487–496, 1969.

[90] S. J. Gosh, D. Roy, and C. S. Manohar, "New Forms of Extended Kalman Filter via Transversal Linearization and Applications to Structural System Identification," *Computer Methods in Applied Mechanics and Engineering*, vol. 196, pp. 5063–5083, 2007.

[91] N. Saha and D. Roy, "Extended Kalman Filters Using Explicit and Derivative-Free Local Linearizations," *Applied Mathematical Modelling*, vol. 33, pp. 2545–2563, 2009.

[92] X. Song, P. Willett, and S. Zhou, "Posterior Cramer-Rao Bounds for Doppler Biased Multistatic Range-only Tracking," in *Proceedings of the 14th International Conference on Information Fusion (FUSION)*, (Chicago, Illinois), pp. 1–8, 2011.

[93] B. Ripley, *Stochastic Simulation*. New York: Wiley, 1987.

[94] S. Julier, J. Uhlmann, and H. F. Durrant-Whyte, "A New Method for the Nonlinear Transformation of Means and Covariances in Filters and Estimators," *IEEE Transactions on Automatic Control*, vol. 45, pp. 477–482, 2000.

[95] E. A. Wan and R. van der Merwe, "The Unscented Kalman Filter for Nonlinear Estimation," in *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium*, (Lake Louise, Alberta, Canada), pp. 153–158, 2000.

[96] E. A. Wan and R. V. D. Merwe, "The Unscented Kalman Filter," in *Kalman Filtering and Neural Networks* (S. Haykin, ed.), ch. 7, pp. 221–280, Wiley, 2001.

[97] I. Arasaratnam, S. Haykin, and T. R. Hurd, "Cubature Kalman Filtering for Continuous-Discrete Systems: Theory and Simulations," *IEEE Transactions on Signal Processing*, vol. 58, no. 10, pp. 4977–4993, 2010.

[98] J. Duník, M. Šimandl, and O. Straka, "Adaptive Choice of Scaling Parameter in Derivative-Free Local Filters," in *Proceedings of the 13th*

International Conference on Information Fusion (FUSION), (Edinburgh), pp. 1–8, 2010.

[99] S. Julier, "The Scaled Unscented Transform," in *Proceedings of the American Control Conference*, vol. 6, (Anchorage, Alaska), pp. 4555–4559, 2002.

[100] D. Tenne and T. Singh, "The Higher Order Unscented Filter," in *Proceedings of the American Control Conference*, (Denver, Colorado), pp. 2441–2446, June 4-6 2003.

[101] Y. Wu, M. Wu, D. Hu, and X. Hu, "An Improvement to Unscented Transformation," in *Proceedings of the 17th Australian Joint Conference on Artificial Intelligence*, vol. 3339, (Cairns, Australia), pp. 1024–1029, 2004.

[102] I. Arasaratnam, S. Haykin, and R. Elliott, "Discrete-Time Nonlinear Filtering Algorithms Using Gauss-Hermite Quadrature," *Proceedings of the IEEE*, vol. 95, pp. 953–977, 2007.

[103] S. Julier and J. Uhlmann, "Reduced Sigma Point Filters for the Propagation of Means and Covariances Through Nonlinear Transformations," in *Proceedings of the American Control Conference*, vol. 2, (Anchorage, Alaska), pp. 887–892, 2002.

[104] A. Honkela, "Approximating Nonlinear Transformations of Probability Distributions for Nonlinear Independent Component Analysis," in *Proceedings of the International Joint Conference on Neural Networks*, vol. 3, (Budapest, Hungary), pp. 2169–2174, 2004.

[105] G. H. Golub and J. H. Welsch, "Calculation of Gauss Quadrature Rules," *Mathematics of Computation*, vol. 23, pp. 221–230, 1969.

[106] K. Pomorski, "Gauss-Hermite Approximation Formula," *Computer Physics Communications*, vol. 174, pp. 181–186, 2006.

[107] T. S. Schei, "A Finite-Difference Method for Linearization in Nonlinear Estimation Algorithms," *Automatica*, vol. 33, pp. 2053–2058, 1997.

[108] I. Bilik and J. Tabrikian, "Target Tracking in Glint Noise Environment Using Nonlinear Non-Gaussian Kalman Filter," in *Proceedings of IEEE International Radar Conference*, (Verona, New York), pp. 282–286, 2006.

[109] N. M. Kwok, Q. M. Ha, S. Huang, G. Dissanayake, and G. Fang, "Mobile Robot Localization and Mapping Using a Gaussian Sum Filter," *International Journal of Control, Automation, and Systems*, vol. 5, pp. 251–268, 2007.

[110] D. L. Alspach and H. W. Sorenson, "Nonlinear Bayesian Estimation Using Gaussian Sum Approximations," *IEEE Transactions on Automatic Control*, vol. 17, pp. 439–448, 1972.

[111] D. Alspach and H. Sorenson, "Recursive Bayesian Estimation Using Gaussian Sums," *Automatica*, vol. 7, pp. 465–479, 1971.

[112] J. J. Verbeek, J. R. Nunnik, and N. Vlassis, "Accelerated EM-based Clustering of Large Data Sets," *Data Mining Knowledge Discovery*, vol. 13, pp. 291–307, 2006.

[113] J. Vermaak, S. Maskell, and M. Briers, "Online Sensor Registration," in *Proceedings of the IEEE Aerospace Conference*, (Big Sky, Montana), pp. 2117–2125, 2005.

[114] W. I. Tam, K. N. Plataniotis, and D. Hatzinakos, "An Adaptive Gaussian Sum Algorithm for Radar Tracking," *Signal Processing*, vol. 77, pp. 85–104, 1999.

[115] J. T. Horwood and A. B. Poore, "Adaptive Gaussian Sum Filters for Space Surveillance," *IEEE Transactions on Automatic Control*, vol. 56, pp. 1777–1790, 2011.

[116] G. Terejanu, P. Singla, T. Singh, and P. D. Scott, "A Novel Gaussian Sum Filter Method for Accurate Solution to the Nonlinear Filtering Problem," in *Proceedings of the 11th International Conference on Information Fusion,*, (Cologne, Germany), pp. 1–8, 2008.

[117] S. Ali-Löytty, "On the Convergence of the Gaussian Mixture Filter," Tech. Rep. 89, Tampere University of Technology, Department of Mathematics, 2008.

[118] J. L. Williams, "Gaussian mixture reduction for tracking multiple maneuvering targets in clutter," Master's thesis, Air Force Institute of Technology, Wright-Patterson Air Force Base, OH., 2003. Available: http://ssg.mit. edu/jlwil/.

[119] P. Zarchan, *Tactical and Strategic Missile Guidance*, vol. 199 of *Progress in astronautics and aeronautics*. American Institute of Aeronautics and Astronautics, 2002.

[120] X. R. Li and V. P. Jilkov, "Survey of Maneuvering Target Tracking. Part i: Dynamic Models," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 39, pp. 1333–1364, 2003.

[121] Mathworks, *www.mathworks.com/matlabcentral/fileexchange/18189*.

[122] Mathworks, *www.mathworks.com/matlabcentral/fileexchange/18217*.

[123] G. Welch and G. Bishop, *www.cs.unc.edu/ welch/kalman*. The University of Northern Carolina at Chapel Hill, Department of Computer Science.

[124] Identification and Decision Making Research Group at the University of West Bohemia, *http://nft.kky.zcu.cz/tools*.