Gaussian Flow Sigma Point Filter for Nonlinear Gaussian State-Space Models

Henri Nurminen, Robert Piché
Laboratory of Automation and Hydraulics
Tampere University of Technology
Tampere, Finland
Emails: {henri.nurminen, robert.piche}@tut.fi

Simon Godsill
Department of Engineering
University of Cambridge
Cambridge, UK
Email: sjg@eng.cam.ac.uk

***Abstract*—We propose a deterministic recursive algorithm for approximate Bayesian filtering. The proposed filter uses a function referred to as the approximate Gaussian flow transformation that transforms a Gaussian prior random variable into an approximate posterior random variable. Given a Gaussian filter prediction distribution, the succeeding filter prediction is approximated as Gaussian by applying sigma point moment-matching to the composition of the Gaussian flow transformation and the state transition function. This requires linearising the measurement model at each sigma point, solving the linearised models analytically, and introducing the measurement information gradually to improve the linearisation points progressively. Computer simulations show that the proposed method can provide higher accuracy and better posterior covariance matrix approximation than some state-of-the art computationally light approximative filters when the measurement model function is nonlinear but differentiable and the noises are additive and Gaussian. We also present a highly nonlinear scenario where the proposed filter occasionally diverges. In the accuracy–computational complexity axis the proposed algorithm is between Kalman filter extensions and Monte Carlo methods.**

I. Introduction

The Kalman filter (KF) [1] is the optimal linear filter in mean square error sense for linear state-space models. However, many practical systems include nonlinear state transition and measurement equations. Therefore, several extensions of KF have been developed to approximate the Bayesian filtering distribution for nonlinear state-space models. These include the extended Kalman filter (EKF) [2, Ch. 8.3] and unscented Kalman filter (UKF) [3]. EKF has difficulties with highly nonlinear models, and UKF as well as other conventional sigma point filters [4] can have problems with measurement model functions whose details are too fine-grained for the resolution of the prior-distribution based sigma point representation. Both filters can suffer if the filter prediction distribution is highly diffuse compared to the measurement accuracy [5]. If the filtering distribution is multimodal, EKF and UKF tend to follow one mode only.

Fig. 1 shows an example of a measurement update with a diffuse Gaussian prior and measurements of distances to three anchors with Gaussian measurement noises. The model is highly nonlinear, because one anchor is close to the target. The EKF is inaccurate because it uses only the prior mean as the linearisation point, and the UKF cannot capture the measurement model function's derivatives properly because one anchor is inside the sigma point cloud.

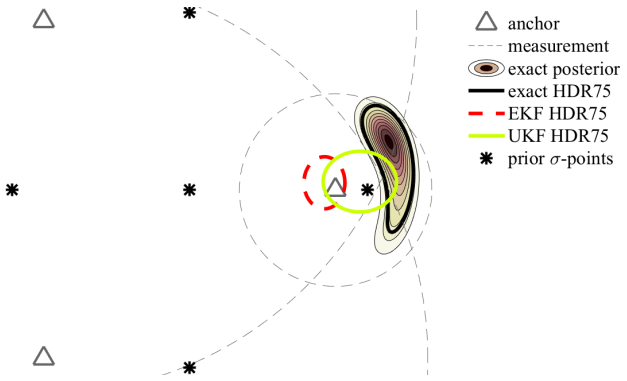

Fig. 1. An example of a highly nonlinear measurement update using three distance measurements with Gaussian noises from the model (18b). EKF and UKF perform poorly because the measurement anchors are near. HDR75 is the highest-density region that contains 75 % of the posterior probability.

Particle flow particle filters (PFPFs) [6]–[8] are Monte Carlo based filtering algorithms and related to the well-known particle filters. PFPFs implement the Bayes rule by moving each sample (a.k.a. particle) using a function that transforms the samples' distribution in the desired way. Computation of this function involves solving an ordinary differential equation (ODE) that in this context is referred to as the particle flow. That is, the PFPFs replace or complement the particle weighting step in the measurement assimilation of the conventional particle filters with the particle flow step. PFPFs mitigate particle degeneracy by including information about the current measurement in the particle propagation step of the particle filter [8], [9]. This can be considered as approximating the optimal importance distribution of the particle locations [8], [10]. Intuitively, the idea is to concentrate most approximation accuracy in the areas where the posterior density is highest. In this paper this idea is applied to deterministic sigma point filtering.

In this paper we propose an approximative nonlinear filter

Henri Nurminen receives funding from Tampere University of Technology Graduate School, Nokia Technologies Oy, Tekniikan edistämissäätiö, the Foundation of Nokia Corporation, and Emil Aaltosen säätiö.

Citation: H. Nurminen, R. Piché, and S. Godsill, Gaussian Flow Sigma Point Filter for Nonlinear Gaussian State-Space Models, *20th International Conference on Information Fusion (FUSION)*, July 2017.

where deterministic sigma points are propagated using a particle flow. The proposed method approximates the filter prediction distribution as a Gaussian distribution which it represents using sigma points. The sigma points are propagated by numerically solving the approximate Gaussian flow equations introduced in [8]. This implies linearising the measurement model at each sigma point and gradually introducing more measurement information to move the linearisation points towards the high-density region of the true filtering distribution. The sigma points are then propagated directly through the state transition function of the next time instant. This procedure gives a sigma point based moment-matching approximation for the composition of the approximate Gaussian flow transformation and the state transition function. The measurement model function is assumed to be differentiable.

A strength of the proposed filter compared to EKF is that the proposed filter is not solely based on the local derivative information in a single point. Instead, the proposed filter averages the linearisation over several points through the sigma point distribution. The gradual introduction of the measurement information aims at improving the linearisations through the particle flow propagation; sigma points are moved only slightly when the linearisation points are still mainly prior-based, and when approaching the complete measurement update, the linearisation is expected to become more accurate in areas of high posterior density. A strength compared to UKF is that the proposed method automatically moves the sigma points to areas where posterior density is high, thus enabling the sigma points to capture more fine-grained details of the measurement model function even when the measurements have high precision. Furthermore, the proposed filter does not require Gaussian approximation of the filtering distribution: the Gaussian approximation is made only once per recursion, after the prediction step. A strength compared to PFPFs is that the number of systematically placed sigma points can be smaller than the number of randomly placed particles typically needed by particle filters. Further, the proposed method does not use computationally costly weighting procedures. Weaknesses of the proposed method are higher computational complexity compared to EKF and UKF, and flexibility inferior to that of particle filters in approximating non-Gaussian filtering prior distributions. The proposed filter also requires analytical differentiation unlike UKF, although numerical differentiation could also be used.

In this paper we present computer simulations where the proposed filter gives more accurate estimates than state-of-the-art comparison methods. We also show another simulation with a highly nonlinear measurement model where the proposed filter occasionally diverges, in spite of showing good overall performance. Providing an adaptive procedure that damps the sigma point movement in case of a bad linearisation is a topic for future research.

The structure of this paper is the following: In Section II we explain the nonlinear Bayesian filtering problem. Then in Section III we give an overview of the existing particle flow methods and give a definition of the approximate Gaussian flow. In Section IV we derive the Gaussian flow sigma point filter. In Section V we show computer simulations where the proposed filter gives more accurate estimates than some state-of-the-art comparison methods. Section VI summarizes the conclusions.

II. Nonlinear Gaussian filtering problem

We consider the nonlinear additive Gaussian state-space model

$$x_0 \sim \mathrm{N}(x_{0|0}, P_{0|0}), \tag{1a}$$

$$x_k = a_k(x_{k-1}) + w_{k-1}, \quad w_{k-1} \sim \mathrm{N}(0, Q_{k-1}), \tag{1b}$$

$$y_k = c_k(x_k) + v_k, \quad v_k \sim \mathrm{N}(0, R_k), \tag{1c}$$

where $x_k \in \mathbb{R}^{n_x}$ is the unknown state at time index k, $a_k : \mathbb{R}^{n_x} \to \mathbb{R}^{n_x}$ is the possibly nonlinear state transition function, $Q_k \in \mathbb{R}^{n_x \times n_x}$ is the process noise covariance matrix, $y_k \in \mathbb{R}^{n_y}$ is the measurement vector, $c_k : \mathbb{R}^{n_x} \to \mathbb{R}^{n_y}$ is the possibly nonlinear but differentiable measurement model function, $R_k \in \mathbb{R}^{n_y \times n_y}$ is the measurement noise covariance matrix, and $w_k \in \mathbb{R}^{n_x}$ and $v_k \in \mathbb{R}^{n_y}$ are white and mutually independent Gaussian noise processes referred to as the process noise and the measurement noise. $\mathrm{N}(\mu, \Sigma)$ denotes the multivariate normal distribution with mean μ and covariance matrix Σ, and $x_{0|0}$ and $P_{0|0}$ are the initial state's mean and covariance matrix.

At the kth time instant, we want to compute the filtering distribution $p(x_k|y_{1:k})$. When the functions a_k and/or c_k are nonlinear, the filtering distribution does not generally belong to any known family of probability distributions, so we approximate it with a distribution defined by a small number of parameters. These parameters are updated with every new measurement. The procedure should be recursive in the sense that a filter update should be based only on the previous filtering distribution approximation and the new measurement, not on earlier measurements, whose information is already contained in the previous filtering distribution.

III. Particle flow in nonlinear filtering

A. Existing particle flow methods

PFPFs are Monte Carlo methods, where a pseudo-random sample (a particle) of the previous filtering distribution is propagated into the current time instant by generating a pseudo-random realisation from the state propagation model. After this, the particle is interpreted as the initial value x_0 of the particle flow ODE, and the filter update step takes the ODE final value as a sample of the filtering distribution. Approximation errors arise because the exact particle flow is not known or because the flow must be solved using numerical methods. Some schemes include also particle weighting and stochastic resampling steps.

PFPFs were originally proposed by Daum and Huang in [6] and are also related to the ensemble transform filters of Reich [7], [11]. Particles are generated using the state transition distribution and are then moved to a new position by numerically solving the particle flow ODE. This ODE includes

the analytic expression of the filter prediction distribution which is typically not available but needs to be approximated with a Gaussian [7] or a Gaussian mixture [11] distribution. The PFPF of [6] does not include particle weighting or resampling.

Since formulating and solving the particle flow requires approximations, a particle weighting scheme based on the concept of approximate Gaussian flow is proposed in [8]. The weighting scheme of [8] also involves some approximations, but if their effect is neglected, the weighting will reduce the theoretical discrepancy between the particle location distribution and the true filtering distribution.

In some versions, the particle flow itself is a stochastic differential equation, and the solution methods use random number generation [8], [12]. Our method could also be extended to this direction, but we have not explored this yet.

The iterated posterior linearisation filter (IPLF) of [13] also seeks to move deterministic sigma points towards the areas with high posterior density. However, this filter does not introduce the measurement information gradually, which can be problematic if the initial sigma point update overshoots or is otherwise inaccurate. Furthermore, the algorithm makes a Gaussian approximation after each iteration, so the sigma point constellation always follows a Gaussian sigma point rule, and the filtering distribution is always approximated as Gaussian.

The progressive Bayesian filter of [14] is similar to ours in the sense that it uses a deterministic sigma point representation of the filtering distributions, and introduces measurement information gradually to move the sigma points closer to the areas with high filtering density. However, in [14], the sigma points are moved differently from our filter: at each progression step each sigma point is copied, and the child sigma points are slightly spread and weighted using part of the measurement-based likelihood. The representation is then downsampled and the weights equalised using a deterministic procedure, where a nonlinear optimisation problem is solved to attain an optimal Dirac delta mixture approximation with a certain number of components. No concept of continuous flow equations is used. This approach requires implementation and tuning of rather complicated details such as the upsampling algorithm, and the filter does not become a KF for a linear–Gaussian model. This approach can also become computationally heavy due to the nonlinear optimisation at the downsampling procedure that lacks a closed-form solution.

The proposed algorithm also shares some similarity with the recursive update sigma point filter [15], [16]. However, in this approach the partial measurement information is included in the state distribution and the remaining part of the measurement information is then treated as correlated with the state distribution. This may result in filter divergence due to accumulation of approximation errors as reported in [17].

B. Approximate Gaussian flow

Consider now a nonlinear Gaussian model with prior $p(x) = \mathrm{N}(x; m_0, P_0)$ and likelihood $p(y|x) = \mathrm{N}(y; c(x), R)$. We denote the "pseudo-time" with $\lambda \in [0, 1]$, i.e. the integration variable of the flow. For any λ, the distribution

$$\hat{p}_\lambda(x|y) = \frac{p(x)\, p(y|x)^\lambda}{\int p(x)\, p(y|x)^\lambda \,\mathrm{d}x} \tag{2}$$

is conventionally approximated by a Gaussian distribution using the information form of the first-order linearised Kalman filter update

$$P_\lambda^{[x^*]} = \big(P_0^{-1} + \lambda (C^{[x^*]})^\mathsf{T} R^{-1} C^{[x^*]}\big)^{-1} \tag{3a}$$

$$m_\lambda^{[x^*]} = P_\lambda^{[x^*]} \big(P_0^{-1} m_0 + \lambda (C^{[x_*]})^\mathsf{T} R^{-1} \widehat{y}^{[x^*]}\big), \tag{3b}$$

where the subscript λ indicates direct dependence on λ, the superscript $[x^*]$ indicates that the quantity is with respect to the first-order Taylor series approximation of c at the point x^*, $C^{[x^*]}$ is the Jacobian matrix of c evaluated at x^*, and $\widehat{y}^{[x^*]} = y - c(x^*) + C^{[x^*]} x^*$.

Following [8], the approximate Gaussian flow is defined to be the solution x_λ, $\lambda \in [0, 1]$, of the ODE

$$\frac{\mathrm{d}x_\lambda}{\mathrm{d}\lambda} = A_\lambda^{[x_\lambda]} x_\lambda + b_\lambda^{[x_\lambda]}, \tag{4}$$

where

$$A_\lambda^{[x_\lambda]} = -\tfrac{1}{2} P_\lambda^{[x_\lambda]} (C^{[x_\lambda]})^\mathsf{T} R^{-1} C^{[x_\lambda]}, \tag{5}$$

$$b_\lambda^{[x_\lambda]} = P_\lambda^{[x_\lambda]} (C^{[x_\lambda]})^\mathsf{T} R^{-1} \big(\widehat{y}^{[x_\lambda]} - \tfrac{1}{2} C^{[x_\lambda]} m_\lambda^{[x_\lambda]}\big), \tag{6}$$

where the matrix $P_\lambda^{[x^*]}$ and the vector $m_\lambda^{[x^*]}$ are as given in (3).

The solution of the ODE (4) can be approximated using the special structure of the problem. We use a step-wise method by solving exactly the ODEs

$$\frac{\mathrm{d}x_\lambda}{\mathrm{d}\lambda} = A_\lambda^{[x_{\lambda_i}]} x_\lambda + b_\lambda^{[x_{\lambda_i}]}, \quad \lambda \in [\lambda_i, \lambda_{i+1}] \tag{7}$$

between the pseudo-time discretisation points $\lambda_0 = 0, \lambda_1, \lambda_2, \ldots, \lambda_i, \ldots, \lambda_{n_\lambda} = 1$, where n_λ is the number of pseudo-time discretisation steps. Each step is an exact Gaussian flow [8], so the analytical solution of the ith step is given by

$$x_{\lambda_{i+1}} = m_{\lambda_{i+1}}^{[x_{\lambda_i}]} + \big(P_{\lambda_{i+1}}^{[x_{\lambda_i}]} (P_{\lambda_i}^{[x_{\lambda_{i-1}}]})^{-1}\big)^{\frac{1}{2}} (x_{\lambda_i} - m_{\lambda_i}^{[x_{\lambda_{i-1}}]}), \tag{8}$$

where the matrix square root is the principal square root defined by $A^{\frac{1}{2}} A^{\frac{1}{2}} = A$, as shown in [8]. Intuitively speaking, the vector $x_{\lambda_{i+1}}$ is now a deterministic sample of the distribution $\mathrm{N}(m_{\lambda_{i+1}}^{[x_{\lambda_i}]}, P_{\lambda_{i+1}}^{[x_{\lambda_i}]})$.

The ODE (4) can also be solved directly with standard numerical ODE solvers.

IV. Gaussian flow sigma point filter

A. Algorithm derivation

Let us denote the approximate Gaussian flow transformation with the function $g(x_0) = x_1$, where the vector x_1 is the final value of the solution x_λ of the approximate Gaussian flow (4) with the initial value x_0.

The idea of our proposed Gaussian flow sigma point filter (GFSPF) is the following: Assume first that we have a Gaussian filter prediction distribution

$$p(x_k|y_{1:k-1}) = \mathrm{N}(x_k; x_{k|k-1}, P_{k|k-1}), \tag{9}$$

where $x_{k|k-1}$ and $P_{k|k-1}$ are known or given by the previous recursions of the algorithm. We use the third order spherical cubature rule [18], [19] to compute the approximate mean $x_{k+1|k}$ and covariance matrix $P_{k+1|k}$ of the transformed random variable $a_k(g_k(x_k))|y_{1:k-1}$, where g_k is the approximate Gaussian flow transformation for the measurement at time k and a_k is the state transition function from (1b). Then, we make the Gaussian moment-matching approximation

$$p(x_{k+1}|y_{1:k}) \approx \mathrm{N}(x_{k+1}|x_{k+1|k}, P_{k+1|k}). \tag{10}$$

This approximation completes the recursion.

In the spherical cubature approximation, a sigma point rule is used to choose a set of n_σ points $\chi^{(i)}_{k|k}$ representing the distribution $\mathrm{N}(x_k; x_{k|k-1}, P_{k|k-1})$, and the composite function $a_k \circ g_k$ is then evaluated in the sigma points. The mean and covariance matrix are approximated using the weighted sample statistics

$$x_{k+1|k} = \sum_{i=0}^{n_\sigma - 1} \omega_i \cdot a_k(g_k(\chi^{(i)}_{k|k})) \tag{11}$$

$$\approx \int a_k(g_k(x_k))\, \mathrm{N}(x_k; x_{k|k-1}, P_{k|k-1})\, \mathrm{d}x_k, \tag{12}$$

and

$$P_{k+1|k} = \sum_{i=0}^{n_\sigma - 1} \omega_i \cdot \big(a_k(g_k(\chi^{(i)}_{k|k})) - x_{k+1|k}\big) \times \big(a_k(g_k(\chi^{(i)}_{k|k})) - x_{k+1|k}\big)^{\mathsf{T}} + Q_k \tag{13}$$

$$\approx \int \big(a_k(g_k(x_k)) - x_{k+1|k}\big)\big(a_k(g_k(x_k)) - x_{k+1|k}\big)^{\mathsf{T}} \times \mathrm{N}(x_k; x_{k|k-1}, P_{k|k-1})\, \mathrm{d}x_k + Q_k. \tag{14}$$

In the third order spherical cubature rule the number of the sigma points is $n_\sigma = 2n_x + 1$, the sigma point locations are [19]

$$\chi^{(i)}_{k|k} = \begin{cases} x_{k|k-1} & , \ i = 0 \\ x_{k|k-1} + \sqrt{n_x + \kappa} \cdot [P^{\frac{1}{2}}_{k|k-1}]_{:,i} & , \ 0 < i \leq n_x \\ x_{k|k-1} - \sqrt{n_x + \kappa} \cdot [P^{\frac{1}{2}}_{k|k-1}]_{:,i} & , \ n_x < i \leq 2n_x \end{cases} \tag{15}$$

where $[P^{\frac{1}{2}}]_{:,i}$ denotes the ith column of any matrix square root for which $P^{\frac{1}{2}}(P^{\frac{1}{2}})^{\mathsf{T}} = P$, and the sigma point weights are [19]

$$\omega^{(i)} = \begin{cases} \frac{\kappa}{n_x + \kappa} & , \ i = 0 \\ \frac{1}{2(n_x + \kappa)} & , \ i \neq 0 \end{cases}. \tag{16}$$

The matrix square root can be the principal square root, and $\kappa \in (-n_x, \infty)$ is a parameter that determines the spread of the sigma points and affects the sigma point weighting. Higher-order cubature rules are also available, but they require a superlinear number of sigma points [18]. Other possibilities include Gauss–Hermite rules [20], [21, Ch. 6.3] and optimal Dirac mixture approximations [22]. The details of the proposed filtering algorithm are given in Algorithm 1.

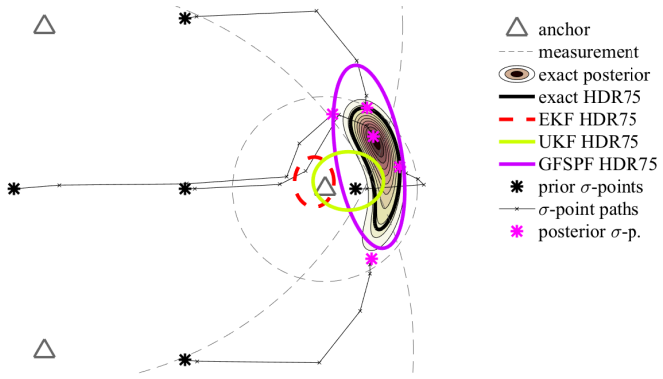

Fig. 2. An example of a highly nonlinear measurement update using three distance measurements with Gaussian noises from the model (18b). The proposed GFSPF outperforms EKF and UKF. HDR75 is the highest-density region that contains 75 % of the posterior probability.

Fig. 2 shows an example of a measurement update of the GFSPF. It depicts the same scenario as Fig. 1. The GFSPF places the sigma points close to the posterior distribution, and thus achieves highest approximation accuracy in this area, outperforming EKF and UKF. Furthermore, the resulting sigma point set is not symmetric, thus capturing some non-Gaussian properties of the posterior. In our proposed algorithm the sigma point configurations are more flexible than in standard sigma point methods. The proposed filter does not require Gaussian approximation of the filtering distribution; only the filter prediction distribution is approximated as Gaussian. This can be an advantage especially if the Gaussian process noise has a relatively large variance, as this makes the filter prediction distribution closer to Gaussian.

The proposed algorithm has some important special cases. With a linear model, it becomes the KF because the ODEs can be solved exactly and the moment-matching becomes exact. If there is only one step in the λ-discretisation, the algorithm becomes a form of approximate statistical linearisation, although different from the form presented in [21, Sec. 5.3]: it approximates the expectation of the whole KF update step of the linearised model, not only the expectation of the linearisation. Algorithm 1 uses a sigma point representation based on the third order spherical cubature rule, but any other valid sigma point representation is also possible. In Algorithm 1 the λ-discretisation is fixed, but the discretisation can be determined separately for each time instant and each sigma point, if necessary.

V. Simulations

We demonstrate the performance of the proposed Gaussian flow sigma point filter (GFSPF) using simulated data. The tests were carried out using Matlab. The compared algorithms are GFSPF, the extended Kalman filter (EKF), the unscented Kalman filter (UKF), the iterated posterior linearisation filter

Algorithm 1 Approximate Gaussian Flow Sigma Point Filter

Inputs: initial prior $x_{0|0}$, $P_{0|0}$; process model a_k, Q_k; measurement model $c_k(x)$, $C_k^{[x]}$, R_k; measurements $y_{1:K}$; filter parameter κ; pseudo-time discretisation $\lambda_0 = 0, \lambda_1, \cdots, \lambda_{n_\lambda} = 1$.
Outputs: $x_{k|k}$, $P_{k|k}$ for $k = 0, \ldots, K$.

Initial prior sigma points
$\omega_0 \leftarrow \frac{\kappa}{n_x+\kappa}$, $\omega_i \leftarrow \frac{1}{2(n_x+\kappa)}$ for $i = 1, 2, \ldots, 2n_x$
$\chi_{0|0}^{(0)} \leftarrow x_{0|0}$
for i = 1 to n_x **do**
  $\chi_{0|0}^{(i)} \leftarrow x_{0|0} + \sqrt{n_x+\kappa}\,[P_{0|0}^{\frac{1}{2}}]_{:,i}$
  $\chi_{0|0}^{(n_x+i)} \leftarrow x_{0|0} - \sqrt{n_x+\kappa}\,[P_{0|0}^{\frac{1}{2}}]_{:,i}$
end for
for k = 1 to K **do**
  Prediction step
  for i = 0 to $2n_x$ **do**
    $\chi_{k|k-1}^{(i)} \leftarrow a_k(\chi_{k-1|k-1}^{(i)})$
  end for
  $x_{k|k-1} \leftarrow \sum_{i=0}^{2n_x} \omega_i \chi_{k|k-1}^{(i)}$
  $P_{k|k-1} \leftarrow \sum_{i=0}^{2n_x} \omega_i (\chi_{k|k-1}^{(i)} - x_{k|k-1})(\chi_{k|k-1}^{(i)} - x_{k|k-1})^\mathsf{T} + Q_k$
  Update step
  $\chi_{k|k}^{(0)} \leftarrow x_{k|k-1}$ ▷ Prediction sigma points
  for i = 1 to n_x **do**
    $\chi_{k|k}^{(i)} \leftarrow x_{k|k-1} + \sqrt{n_x+\kappa}\,[P_{k|k-1}^{\frac{1}{2}}]_{:,i}$
    $\chi_{k|k}^{(n_x+i)} \leftarrow x_{k|k-1} - \sqrt{n_x+\kappa}\,[P_{k|k-1}^{\frac{1}{2}}]_{:,i}$
  end for
  for i = 0 to $2n_x$ **do**
    $m_- \leftarrow x_{k|k-1}$, $S_- \leftarrow P_{k|k-1}$
    for j = 1 to n_λ **do** ▷ Flow transformation
      $J \leftarrow C_k^{[\chi_{k|k}^{(i)}]}$
      $S \leftarrow \left(P_{k|k-1}^{-1} + \lambda_j J^\mathsf{T} R_k^{-1} J\right)^{-1}$
      $m \leftarrow S\big(P_{k|k-1}^{-1} x_{k|k-1}$
        $+\lambda_j J^\mathsf{T} R_k^{-1}(y_k - c_k(\chi_{k|k}^{(i)}) + J\chi_{k|k}^{(i)})\big)$
      $\chi_{k|k}^{(i)} \leftarrow m + (SS_-^{-1})^{\frac{1}{2}}(\chi_{k|k}^{(i)} - m_-)$ ▷ $A^{\frac{1}{2}}A^{\frac{1}{2}} = A$
      $m_- \leftarrow m$, $S_- \leftarrow S$
    end for
  end for
  $x_{k|k} \leftarrow \sum_{i=0}^{2n_x} \omega_i \chi_{k|k}^{(i)}$
  $P_{k|k} \leftarrow \sum_{i=0}^{2n_x} \omega_i (\chi_{k|k}^{(i)} - x_{k|k})(\chi_{k|k}^{(i)} - x_{k|k})^\mathsf{T}$
end for

(IPLF) of [13], and the Gaussian flow based particle flow particle filter (PFPF) of [8].

In the tests, GFSPF and PFPF used a fixed 8-step discretisation grid for pseudo-time λ, $\lambda_{1:8} = 2^{[-20,-15,-10,-5,-3,-1,-0.5,0]}$. The PFPF used only $2n_x + 1$ particles, which equals the number of the sigma points in GFSPF. The parameter κ was set to 0.5 for UKF, IPLF, and GFSPF to give uniformly weighted sigma points. The UKF used the third order spherical cubature rule, i.e. the parameter values $\alpha_{\text{UKF}} = 1$, $\kappa_{\text{UKF}} = \kappa$, and $\beta_{\text{UKF}} = 0$ in the conventional UKF parametrisation [21, Ch. 5.5], which matches the cubature Kalman filter (CKF) [19]. The IPLF's iterations were terminated when the Kullback–Leibler divergence between two consecutive posterior approximations was less than or equal to 0.005 or after 50 iterations. The matrix square roots in UKF, PFPF, IPLF, and GFSPF were computed as lower triangular Cholesky decompositions, except for the principal square root in GFSPF in line 27 of Algorithm 1.

A. Nonlinear benchmark model

We generated a realisation of the classical nonlinear benchmark model [23], [24]

$$x_0 \sim \mathrm{N}(0, 10^2), \tag{17a}$$

$$x_k = 0.5x_{k-1} + \frac{25x_{k-1}}{1+x_{k-1}^2} + 8\cos(1.2(k-1)) + w_{k-1}, \tag{17b}$$

$$y_k = \frac{x_k^2}{20} + v_k, \tag{17c}$$

where $w_{k-1} \overset{\text{iid}}{\sim} \mathrm{N}(0, 3^2)$ and $v_k \overset{\text{iid}}{\sim} \mathrm{N}(0, 1^2)$. This measurement-based likelihood is unimodal when the measurement is close to zero or negative, but with large positive measurements the likelihood is bimodal [24].

Fig. 3 presents 51 first time instants of the realisation as well as the estimates and 95 % posterior credibility intervals of each compared algorithm. The credibility intervals were computed by assuming that each posterior approximation is Gaussian with the mean and variance given by the respective algorithm. The figure shows that the proposed GFSPF gives a mean value between the modes and a large variance, while the other methods are prone to choose one peak and omit the rest. Between time instants 26 and 31, for example, EKF, UKF, and IPLF estimate the state as being positive with a very small 95 % interval, while the true state is negative with about the same absolute value. The GFSPF's estimate is close to zero with the 95 % interval including the true value. Near $k = 32$ the state value is close to zero, which pulls all the estimates closer to the true value and diminishes the GFSPF's uncertainty. In some cases, such as between time instants 10 and 16, EKF, UKF, and IPLF give more precise estimates than GFSPF because they happen to track the correct posterior mode.

During the first 1000 time instants of the generated realisation, the true state was within the 95 % posterior credibility interval at 40 % of the time instants for EKF, 71 % for UKF, 57 % for IPLF, and 92 % for GFSPF. That is, all the algorithms underestimate the uncertainty of the posterior estimate, but the proposed GFSPF gives the least underestimated posterior variance approximation. The root mean square error (RMSE) was 23.2 for EKF, 11.9 for UKF, 14.1 for IPLF, and 9.1 for GFSPF. That is, the GFSPF estimate is also the most accurate in RMSE in this test.

B. Navigation with distance measurements

In this example the state consists of position $p_k \in \mathbb{R}^2$ and velocity $v_k \in \mathbb{R}^2$. The dynamical model is a damped constant velocity model and the state is observed through distance

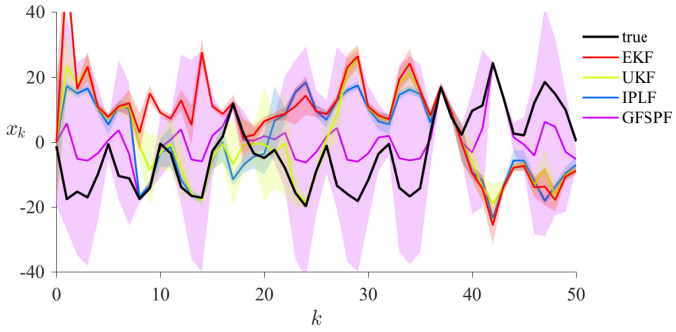

Fig. 3. Filter estimates and 95 % posterior credibility intervals for the nonlinear benchmark model (17). When the posterior is multimodal, the proposed GFSPF gives a mean value between the modes and a large variance, while the other methods follow one peak and underestimate the uncertainty.

measurements to anchors at known locations $s_{j,k} \in \mathbb{R}^2$. The state-space model is thus

$$\begin{bmatrix} p_k \\ v_k \end{bmatrix} = \begin{bmatrix} I_2 & \frac{1-e^{-0.1}}{0.1} I_2 \\ \mathrm{O}_2 & e^{-0.1} I_2 \end{bmatrix} \begin{bmatrix} p_{k-1} \\ v_{k-1} \end{bmatrix} + w_{k-1}, \tag{18a}$$

$$[y_k]_j = \|x_k - s_{j,k}\| + [v_k]_j, \tag{18b}$$

where

$$w_{k-1} \overset{\text{iid}}{\sim} \begin{bmatrix} \frac{0.2-3+4e^{-0.1}-e^{-0.2}}{2\cdot 0.1^3} I_2 & \frac{1-2e^{-0.1}+e^{-0.2}}{2\cdot 0.1^2} I_2 \\ \frac{1-2e^{-0.1}+e^{-0.2}}{2\cdot 0.1^2} I_2 & \frac{1-e^{-0.2}}{0.2} I_2 \end{bmatrix}, \tag{19}$$

$$[v_k]_j \overset{\text{iid}}{\sim} \mathrm{N}(0, r^2), \tag{20}$$

and the measurement noise standard deviation r is a parameter. The anchor positions $s_{j,k} \in \mathbb{R}^2$ were generated at every fifth time instant from the distribution $s_{k,j} \sim \mathrm{N}(\mu_k, \rho^2 I_2)$, where μ_k is the average of the ground truth positions $p_{k:k+4}$ and ρ is a parameter. The anchors were located relatively close to the true positions to generate high nonlinearity, and ρ was varied to compare performances with different levels of nonlinearity; roughly speaking, the smaller ρ is, the closer the anchors are to the true position and the higher the nonlinearity. This example is therefore tailored to favour filters that work well even in highly nonlinear situations. If GFSPF or EKF required derivative at any $s_{j,k}$, the linearisation point was perturbed slightly to make the measurement function differentiable. We generated trajectories of 300 time instants and measurement sets from the state-space model (18). Our criteria for evaluating the filters' accuracy are the position RMSE

$$\mathrm{RMSE} = \sqrt{\frac{1}{300} \sum_{k=1}^{300} \left\| [p_{k|k}]_{\text{estimate}} - [p_k]_{\text{true}} \right\|^2} \tag{21}$$

and the average normalised estimation error squared (NEES) of the final time instant's position estimate

$$\begin{aligned} \mathrm{NEES}_{300} = \frac{1}{N_{\mathrm{MC}}} \sum_{i=1}^{N_{\mathrm{MC}}} & ([p_{300|300}]_{\text{estimate}} - [p_{300}]_{\text{true}})^{\mathsf{T}} \\ & \times [P_{300|300}]^{-1}_{1:2,1:2} ([p_{300|300}]_{\text{estimate}} - [p_{300}]_{\text{true}}), \end{aligned} \tag{22}$$

TABLE I

NEES_{300} VALUES (22) AVERAGED OVER 1000 MONTE CARLO REPLICATIONSFOR NAVIGATION WITH TWO SIMULTANEOUS MEASUREMENT ANCHORS. ON THE LEFT $\rho = 5$, AND ON THE RIGHT $r = 0.5$. THE NOMINAL NEES IS 2, SO GFSPF UNDERESTIMATES THE POSTERIOR COVARIANCE MATRIX LESS THAN THE OTHER ALGORITHMS.

r	EKF	UKF	IPLF	GFSPF	ρ	EKF	UKF	IPLF	GFSPF
0.1	1900	600	230	2.1	0.5	49	1.1	11	2.5
0.25	210	47	38	2.2	1	56	5.6	11	2.5
0.5	72	14	29	2.2	2	69	12	15	2.7
0.75	55	8.0	20	2.2	3	73	22	22	2.6
1	35	4.7	15	2.0	4	70	23	23	2.3
2	21	3.0	7.2	2.1	5	76	23	25	2.1

where N_{MC} is the number of Monte Carlo replications. Under the hypothesis that the filter gives a realistic Gaussian posterior, the NEES is chi-squared-distributed with 2 degrees of freedom [25, Ch. 5.4.2]. Thus, the expected value of the NEES should be 2.

Fig. 4 shows the RMSE distributions as a function of the measurement noise standard deviation r and with $\rho = 5$. The results are shown for scenarios with three and two simultaneous measurement anchors. The box levels are 5 %, 25 %, 50 %, 75 %, and 95 % quantiles of RMSE, and the asterisks show the minimum and maximum values. The results are based on 1000 Monte Carlo replications per each value of r. With large r values the proposed GFSPF outperforms the other methods in accuracy. With small r and three simultaneous measurements the proposed method shows slightly higher median RMSE than UKF and IPLF but is reliable, having a low maximal RMSE. When there are only two distance measurements per time instant, the proposed algorithm is the most accurate with any r also in median RMSE, although with small r UKF and IPLF have occasional runs with very low RMSE. In almost all scenarios, PFPF has accuracy similar or worse than that of GFSPF. PFPF's accuracy could be improved by adding particles, but this would come with the cost of increased computational burden.

The good performance of the proposed GFSPF in the two-anchor case can be explained by the fact that this measurement model is underdetermined and thus prone to multimodal posterior distributions. When the exact posterior distribution is multimodal, the proposed GFSPF tends to move some sigma points to each mode, so the posterior approximation will have the mean in between the modes and a covariance matrix that covers all the modes. This difference is illustrated by Fig. 5. In conclusion, the low-RMSE runs of UKF and IPLF are cases where UKF and IPLF choose the correct mode, while they are also likely to diverge when a wrong mode is chosen, which makes GFSPF more reliable.

Fig. 6 shows the RMSE distributions as a function of different anchor distances ρ and with measurement noise level $r = 0.5$, for the cases with three and two simultaneous measuring anchors. The proposed GFSPF is the most accurate especially with small ρ and in two-measurements-only cases, i.e. when the system is most nonlinear and underdetermined.

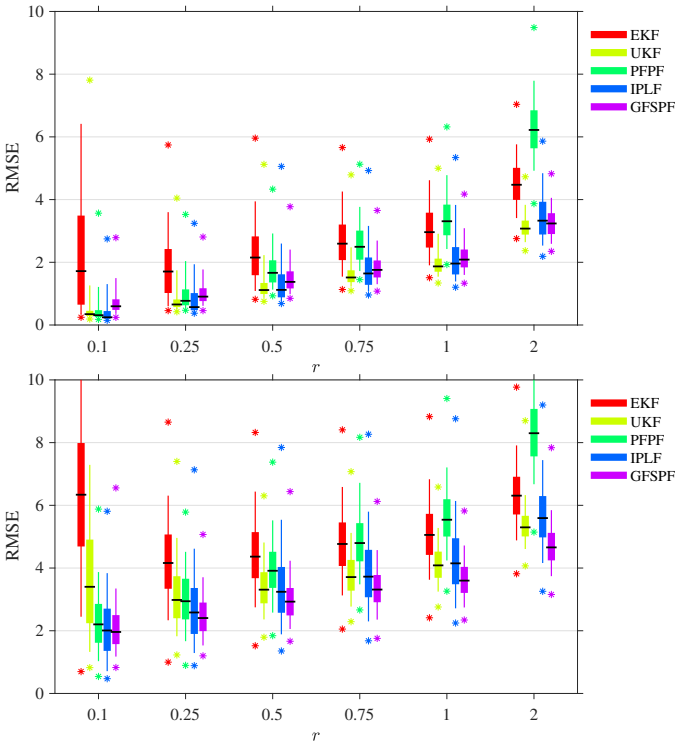

Fig. 4. RMSE distributions for the distance measurement model as a function of the measurement noise standard deviation r for three (upper) and two (lower) measurement anchors.

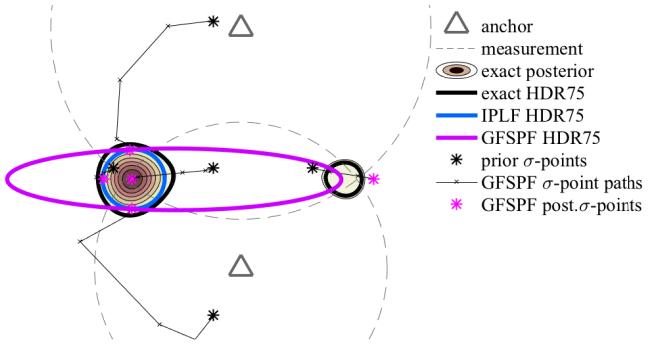

Fig. 5. Gaussian prior and two distance measurements resulting in a bimodal posterior. The posterior sigma point set of the proposed GFSPF tends to cover all posterior modes, while IPLF typically converges to one mode. HDR75 is the highest-density region that contains 75 % of the posterior probability.

With large ρ and three simultaneous measurements, the proposed algorithm again shows a larger median RMSE but lower maximal RMSE than UKF and IPLF.

Table I shows the average NEES_{300} values (22) for the case with two simultaneous measurement anchors. The average NEES of GFSPF is slightly larger than 2, which indicates a slight underestimation of the posterior covariance matrix. However, the other compared algorithms show drastically larger NEES for most parameter values, indicating that they underestimate the posterior covariance matrix more seriously than GFSPF.

The computational burden of GFSPF with the eight-step λ-discretisation is roughly $8 \cdot (2n_x + 1)$ times that of the EKF plus

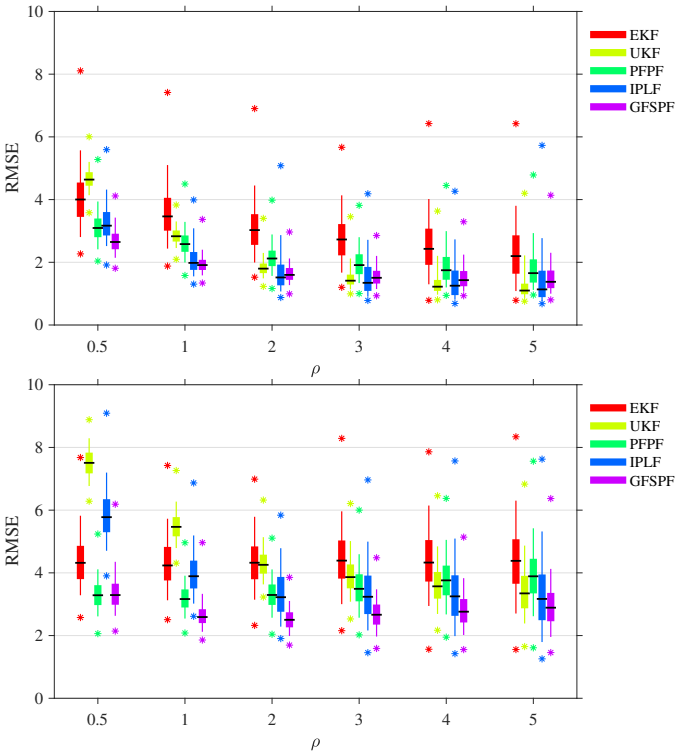

Fig. 6. RMSE distributions for the distance measurement model as a function of the anchor distance parameter ρ for three (upper) and two (lower) measurement anchors. The proposed GFSPF outperforms EKF, and UKF when ρ is small or when there are only two measurements per time instant, i.e. when the model is most nonlinear.

8 matrix square roots of $n_x \times n_x$ matrices. The runtime of our GFSPF implementation is typically between one to five times the IPLF's runtime. IPLF becomes faster with large r and ρ because the adaptive scheme sets the number of IPLF iterations low. Because the number of particles in PFPF equals the number of sigma points in GFSPF, the PFPF's computational complexity is at least as high as that of the GFSPF, in practice considerably higher because PFPF involves extensive weight computations.

C. Navigation with logarithmic-distance measurements

In this example we replace the measurement model (18b) with logarithmic-distance model that is typically used with received signal strength measurement measurements. The measurement model used in this simulation is

$$[y_k]_j = -45 - 17 \log_{10} \left(\| x_k - s_{j,k} \| \right) + [v_k]_j, \qquad (23)$$

where $[v_k]_j \overset{\text{iid}}{\sim} \text{N}(0, r^2)$. We used $r = 5$, and the anchor positions $s_{j,k}$ were generated similarly to the previous section. The RMSE distributions for the case with three simltaneously measured anchors are shown in Fig. 7. The results show that with $\rho \geq 1$ the proposed GFSPF outperforms the other algorithms in median RMSE but has large maximal RMSEs. This implies that the proposed filter can diverge when the model is highly nonlinear. This seems to happen when a strong signal strength is received and the filter covariance is large.

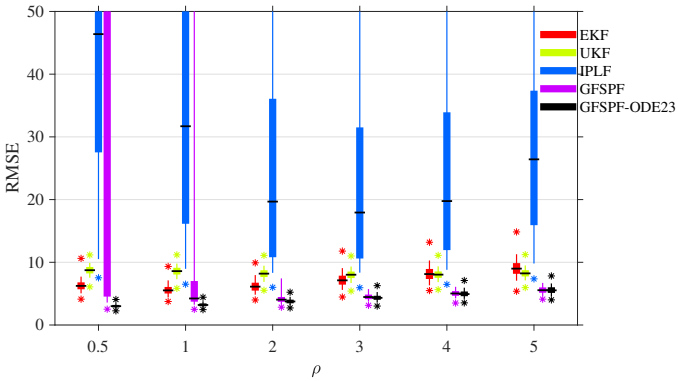

Fig. 7. RMSE distributions for the logarithmic-distance measurement model (23). The proposed GFSPF has low median RMSEs, but large maximal RMSEs indicate that with a non-adaptive λ-discretisation the proposed filter can diverge in highly nonlinear situations. The use of ode23 solver fixes the divergence problem but increases computational burden.

Some linearisations are then made far from the anchor, which can result in a bad approximation of the logarithmic model.

In order to better cope with the aforementioned situation, the fixed ODE integration step could be replaced by an adaptive stepping procedure. This algorithm variant is GFSPF-ODE23 in Fig. 7, which solves the ODE (4) using Matlab's standard ODE solver `ode23` with the relative error tolerance 0.01. Using the standard solver can increase the computational burden by an order of magnitude, but the improvement in position accuracy is impressive with small ρ. This indicates that the divergence problem of the GFSPF is mainly only due to insufficient ODE solver accuracy.

VI. Conclusions

In this paper we propose a novel deterministic approximative Bayesian filter that is based on the concept of flow transformation, which is a function that transforms a prior-distributed random variable into a posterior-distributed one. The proposed filter uses three approximations:

1) Gaussian cubature based moment matching is applied to the composition of the flow transformation and the time update to approximate the filter prediction distribution.
2) The optimal flow transformation is approximated by the approximate Gaussian flow transformation.
3) The approximate Gaussian flow ODE is integrated numerically.

The filter is applicable to nonlinear and Gaussian state-space models where the measurement function is differentiable. Our simulations showed that the proposed filtering algorithm can outperform some state-of-the-art low-complexity algorithms in accuracy when the model is nonlinear. However, the proposed filter occasionally diverges in some highly nonlinear models, but this problem is rectified adaptively adjusting the step lengths in the numerical integration of the flow. Future work is however needed to speed up the adaptive-step solver.

Acknowledgment

We thank Dr. Tohid Ardeshiri for his helpful comments on this manuscript.

References

[1] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME – Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.
[2] A. H. Jazwinski, *Stochastic Processes and Filtering Theory*, ser. Mathematics in Science and Engineering. Academic Press, 1970, vol. 64.
[3] S. J. Julier, J. K. Uhlmann, and H. F. Durrant-Whyte, "A new approach for filtering nonlinear systems," in *American Control Conference*, vol. 3, 1995, pp. 1628–1632.
[4] R. Van Der Merwe and E. Wan, "Sigma-point Kalman filters for probabilistic inference in dynamic state-space models," in *Workshop on Advances in Machine Learning*, 2003.
[5] M. R. Morelande and A. F. García-Fernández, "Analysis of Kalman filter approximations for nonlinear measurements," *IEEE Transactions on Signal Processing*, vol. 61, no. 22, pp. 5477–5484, November 2013.
[6] F. Daum and J. Huang, "Particle flow for nonlinear filters with log-homotopy," in *Proceedings of SPIE, Signal and Data Processing of Small Targets*, vol. 6969, 2008.
[7] S. Reich, "A dynamical systems framework for intermittent data assimilation," *BIT Numerical Mathematics*, vol. 51, pp. 235–249, 2011.
[8] P. Bunch and S. Godsill, "Approximations of the optimal importance density using Gaussian particle flow importance sampling," *Journal of the Americal Statistical Association*, vol. 111, no. 514, pp. 748–762, 2016.
[9] F. Daum and J. Huang, "Particle degeneracy: root cause and solution," in *Proceedings of SPIE, Signal Processing, Sensor Fusion, and Target Recognition XX*, vol. 8050, 2011.
[10] 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.
[11] S. Reich, "A Gaussian-mixture ensemble transform filter," *Quarterly Journal of the Royal Meteorological Society*, vol. 138, pp. 222–233, January 2012.
[12] F. Daum and J. Huang, "Particle flow with non-zero diffusion for nonlinear filters," in *Proceedings of SPIE, Signal Processing, Sensor Fusion, and Target Recognition XXII*, vol. 8745, 2013.
[13] A. F. García-Fernández, L. Svensson, M. R. Morelande, and S. Särkkä, "Posterior linearization filter: Principles and implementation using sigma points," *IEEE Transactions on Signal Processing*, vol. 63, no. 20, pp. 5561–5573, October 2015.
[14] U. D. Hanebeck and M. Pander, "Progressive Bayesian estimation with deterministic particles," in *19th International Conference on Information Fusion (FUSION)*, July 2016.
[15] R. Zanetti, "Recursive update filtering for nonlinear estimation," *IEEE Transactions on Automatic Control*, vol. 57, no. 6, pp. 1481–1490, 2012.
[16] Y. Huang, Y. Zhang, N. Li, and L. Zhao, "Design of sigma-point Kalman filter with recursive updated measurement," *Circuits, Systems and Signal Processing*, vol. 35, no. 5, pp. 1767–1782, May 2016.
[17] M. Raitoharju, R. Piché, and H. Nurminen, "A systematic approach for Kalman-type filtering with non-Gaussian noises," in *19th International Conference on Information Fusion (FUSION)*, July 2016.
[18] Y. Wu, D. Hu, M. Wu, and X. Hu, "A numerical-integration perspective on Gaussian filters," *IEEE Transactions on Signal Processing*, vol. 54, no. 8, pp. 2910–2921, August 2006.
[19] I. Arasaratnam and S. Haykin, "Cubature Kalman filters," *IEEE Transactions on Automatic Control*, vol. 54, no. 6, pp. 1254–1269, June 2009.
[20] K. Ito and K. Xiong, "Gaussian filters for nonlinear filtering problems," *IEEE Transactions on Automatic Control*, vol. 45, no. 5, pp. 910–927, May 2000.
[21] S. Särkkä, *Bayesian Filtering and Smoothing*. Cambridge, UK: Cambridge University Press, 2013.
[22] U. D. Hanebeck, M. F. Huber, and V. Klumpp, "Dirac mixture approximation of multivariate Gaussian densities," in *48th IEEE Conference on Decision and Control (CDC)*, December 2009.
[23] M. Andrade Netto, L. Gimeno, and M. Mendes, "On the optimal and suboptimal nonlinear filtering problem for discrete-time systems," *IEEE Transactions on Automatic Control*, vol. 23, pp. 1062–1067, 1978.
[24] N. J. Gordon, D. J. Salmond, and A. F. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," *IEE Proceedings F*, vol. 140, no. 2, pp. 107–113, April 1993.
[25] Y. Bar-Shalom, R. X. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation, Theory Algorithms and Software*. John Wiley & Sons, 2001.