Progressive Gaussian Filter Using Importance Sampling and Particle Flow

Christof Chlebek, **Jannik Steinbring**, and **Uwe D. Hanebeck**
Intelligent Sensor-Actuator-Systems Laboratory (ISAS)
Institute for Anthropomatics and Robotics (IAR)
Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany
christof.chlebek@kit.edu, jannik.steinbring@kit.edu, uwe.hanebeck@ieee.org

***Abstract*—We propose a novel progressive Gaussian filter for nonlinear stochastic systems. A Gaussian approximation of the posterior is computed without an explicit assumption of a linear relation between the system state and the measurement. This allows for better quality of the estimation compared to Kalman filters for nonlinear problems like the EKF or UKF. In this work, we use the progressive filter framework, which gradually incorporates information of a measurement into the state estimate by considering a flow of probability mass from the prior to the posterior state estimate. We propose a novel particle flow by utilizing a simple linear model. This model predicts the movement of single particles over the course of the filter progression. The predicted trajectory is corrected using importance sampling and moment matching. The proposed method is evaluated in comparison with other state-of-the-art nonlinear Bayesian filters.**

***Keywords*—Gaussian filter, nonlinear Bayesian state estimation, nonlinear filtering, importance sampling, progressive filtering, particle flow, homotopy continuation**

I. Introduction

In this work, we consider the problem of estimating the state of a discrete-time nonlinear dynamic system on the basis of disturbed sensor measurements. State estimation is essential in many applications, like the control of stochastic systems with imperfect state information, where the feedback is given by noisy measurements [1], [2]. Another challenging example is extended object tracking, where measurement information is used to estimate the trajectory and shape of moving objects [3], [4]. In general, we seek to condense obtained measurements and represent them in a compact manner, without losing information. Thereby we want a constant memory demand and prevent an increasing calculation complexity over time.

Using the Bayesian estimation framework, noisy information characterizing the system state can be represented in form of a conditional probability density function (PDF). The system state propagation over time and the relation between the system state and measurements are given by stochastic models [5]. In case of stochastic systems with linear dynamics disturbed by additive and state-independent Gaussian white noise, the Kalman filter yields the closed-form optimal solution in the sense of minimizing the mean squared error [6]. Unfortunately, in case of stochastic discrete-time nonlinear dynamic systems, as considered in this work, closed-form Bayesian state estimation cannot be performed. As a consequence, many approximate approaches have been developed.

Particle Filters (PF) employ a set of weighted particles in order to represent the state estimate [7], [8]. Whereas, this group of filters has the ability to handle strong nonlinearities, the main drawbacks are a high computational cost due to the curse of dimensionality and sample degeneracy as the result of particle reweighting subject to likelihood evaluation. One principle underlying the PF is importance sampling, which will also play a major role in this work. The general idea is to generate a set of particles from a density, which cannot be sampled directly [8]. Instead, we sample from a similar - let us say a well guessed density. Then we correct the sample set by adjusting the weights accordingly. This guess is the so-called *proposal density*. The obvious problem is how to decide on a good proposal density. It has been shown that the optimal proposal density is one that minimizes the variance of the sample weights. Unfortunately, this is the density we are actually looking for, the posterior [8].

Following the Kalman filter, many filter for nonlinear systems work with Gaussian densities [9], [10]. Whilst sharing the state representation being a Gaussian density, the characteristics and underlying ideas can be very distinct. For one, the Extended Kalman Filter (EKF) and Iterated Extended Kalman Filter (IEKF) work with explicit model linearization [11]. Also assuming a linear relation between system state and measurement, the Linear Regression Kalman Filters (LRKF) approximate the joint density of state and measurement by a Gaussian density and use implicit statistical linearization [12], [13]. Methods implementing the statistical linearization idea include the Unscented Kalman Filter (UKF) [14], the High-Degree Cubature Kalman Filter [15], and the Smart Sampling Kalman Filter (S^2KF) [13], [16]. These filters are generally fast and easy to implement. They do not require explicit evaluation of the likelihood function, but may lack in estimation quality, especially in the presence of strong nonlinearities.

In this work, we focus on a less restrictive approximation. We do not explicitly assume a linear relation, but we only assume all state estimates to be Gaussian distributed. The main challenge here is to find the best Gaussian distributed approximation of the true posterior after every filter step. In literature, there are several approaches to achieve this goal. It is possible to employ Monte-Carlo integration to approximate the true posterior and calculate its mean and covariance [17]. Instead of a random sample approach, we utilize a progressive framework. The idea is to interpret the calculation of the posterior as a continuous process, where the prior is transformed into the posterior by a homotopy [18]. Thus, we assume a continuous flow of probability mass. If particles are used to represent this flow of probability mass this method of progressive filtering is often called particle

flow. The state of the homotopy will be called progression time. There are two major concepts to progressive filtering previously published, namely physics-inspired and stochastic-inspired progressive filter. The first class of approaches uses ideas like diffusion to derive a model from an explicit solution of an ordinary or partial differential equation [19]. This has been implemented to applied to Gaussian filter in [20]–[22] and general parametric distributions in [23]. The second class of approaches reformulates the problem such that the posterior is step-wise recursively computed [24]–[26].

Contribution: We propose to take advantages of both ideas and implement a two-step approach. Instead of physical particle models, we estimate a simple motion model to represent the movement of each particle and exploit statistical importance sampling to derive a step-wise update of the posterior. We use a deterministic sampling strategy to generate particles from Gaussian densities (e.g., [27]). The continuity of particle flow is exploited particle-wise by a deterministic first-order linear model. The continuity of single sample trajectories over different progression steps is decided upon by solving an assignment problem. We utilize this relationship to calculate the model parameters, i.e., the constant motion speed of each particle. Using this model for each particle, we calculate the proposal density and update the posterior of the next progression time step. The preliminary posteriors will be called *interim posterior* throughout this work. This proposed procedure results in a predictor-corrector approach, predicting the interim posterior exploiting the motion model and correcting this proposal density.

Outline: Subsequently, this paper is structured as follows: in the next section, the considered problem is given formally. Sec. III elaborates on the key ideas of this paper and introduces some groundwork later employed. The novel approach is then derived in Sec. IV in detail and evaluated in Sec. V by a simulation. There, we compare the new method to other Bayesian state estimation methods in a target tracking example. Finally, we conclude this work in Sec. VI and give some outlook on future work.

Notation: Throughout this work, bold face lower case letters $\boldsymbol{x}$ denote random variables. We distinguish vectors $\underline{x}$ from scalar quantities x by underlining the corresponding variables. The notation $\underline{\boldsymbol{x}} \sim f(\underline{x})$ denotes that $\underline{\boldsymbol{x}}$ is characterized by the probability distribution $f(\underline{x})$. Finally, a matrix $\mathbf{A}$ is denoted by bold face capital letters.

II. Problem Formulation

We consider the problem of estimating the hidden state of a discrete-time stochastic nonlinear dynamic system based on noise-corrupted measurements. The relation between a measurement $\tilde{\underline{y}}_k$ and the system state $\underline{\boldsymbol{x}}_k$ is given by the time-variant measurement model

$$\underline{\boldsymbol{y}}_k = \underline{h}_k(\underline{\boldsymbol{x}}_k, \underline{\boldsymbol{v}}_k) \ , \tag{1}$$

where $\tilde{\underline{y}}_k$ is a realization of the random vector $\underline{\boldsymbol{y}}_k \sim f_k^y(\underline{y}_k)$ characterizing the probability of possible measurements. The random vector $\underline{\boldsymbol{v}}_k \sim f_k^v(\underline{v}_k)$ describes an arbitrary state independent measurement noise.

The goal of this work is to update a given prior state estimate $\underline{\boldsymbol{x}}_k^p$ characterized by

$$f_k^p(\underline{x}_k) = f_k^p(\underline{x}_k|\tilde{\underline{y}}_0, \dots \tilde{\underline{y}}_{k-1})$$

at every time k, by recursively including the newly received measurement $\tilde{\underline{y}}_k$ using the Bayes' rule. Thus, we seek to obtain the posterior state estimate $\underline{\boldsymbol{x}}_k^e$ characterized by

$$f_k^e(\underline{x}_k) = f_k^e(\underline{x}_k|\tilde{\underline{y}}_0, \dots \tilde{\underline{y}}_k)$$

Please note that we focus on the measurement update only. The temporal evolution of the system state, i.e., the update of the prior $\underline{\boldsymbol{x}}_k^p$, is not in the scope of this work.

Using the general Bayesian estimation formulation, a new measurement $\tilde{\underline{y}}_k$ is incorporated into the state estimate by

$$f_k^e(\underline{x}_k) = c_k \cdot f_k^L(\tilde{\underline{y}}_k|\underline{x}_k) \cdot f_k^p(\underline{x}_k) \ , \tag{2}$$

where $c_k = 1/\int f_k^L(\tilde{\underline{y}}_k|\underline{x}_k) f_k^p(\underline{x}_k) \mathrm{d}\underline{x}_k$ is a normalization constant and $f_k^L(\tilde{\underline{y}}_k|\underline{x}_k)$ is the likelihood function for the considered measurement $\tilde{\underline{y}}_k$. The likelihood can be derived by transforming the generative model (1) into a probabilistic model by

$$f_k^L(\tilde{\underline{y}}_k|\underline{x}_k) = \int \delta(\tilde{\underline{y}}_k - \underline{h}_k(\underline{x}_k, \underline{v}_k)) \cdot f_k^v(\underline{v}_k) \mathrm{d}\underline{v}_k \ , \tag{3}$$

where $\delta(\cdot)$ is the Dirac delta function. As already discussed in the introduction, it is not possible to perform the measurement update (2) analytically in general. This holds true, even if the likelihood (3) is given in closed-form. Actually, after only one filter step, the state estimate $f_k^e(\underline{x}_k)$ can be of arbitrary form. This means, it is not Gaussian, even if the prior $f_k^p(\underline{x}_k)$ is given as a Gaussian. Throughout this work, we assume that the true underlying probability density functions representing the prior and posterior states are of Gaussian form. Thus, the considered problem is to calculate the Gaussian approximation of $f_k^e(\underline{x}_k) \approx \mathcal{N}(\hat{\underline{x}}_k^e, \mathbf{C}_k^e)$, under the assumption that we also have a Gaussian prior of the form $f_k^p(\underline{x}_k) = \mathcal{N}(\hat{\underline{x}}_k^p, \mathbf{C}_k^p)$.

In the course of this work, we focus on the derivation of one filter step. Thus, we are not moving forward in time and, for the reason of better readability, we omit the time index k in Sec. III and Sec. IV.

III. Key Idea and Groundwork

We propose to systematically calculate mean $\hat{\underline{x}}^e$ and covariance matrix $\mathbf{C}^e$ of the posterior estimate $f^e(\underline{x})$, by moment matching of a sample set generated by exploiting importance sampling. Key challenge in importance sampling is to find a proposal density as similar as possible to the true posterior, from which particles can be generated and their weights adjusted accordingly. We solve this problem by utilizing progressive filtering, where a homotopy is employed to the likelihood function. This progressive likelihood defines a continuous transformation from the prior $f^p(\underline{x})$ to the posterior $f^e(\underline{x})$ over a progression time given by a parameter γ ranging from 0 to 1.

This continuous progression time allows for small progression steps $\Delta\gamma$, where the movement or flow of each particle can be approximated by a first-order linear model. Using this prediction model, we can systematically generate proposal densities within the progression time. We compensate for the prediction error by adjusting the particle weights and resample the resulting interim posterior. This predictor-corrector process is repeated until the end of the progression time is reached and therefore, the last correction of the proposal density at $\gamma = 1$ yields the posterior $f^e(\underline{x})$.

In the following, we introduce some of the fundamental work employed in the presented method.

A. Deterministic Sampling

For the utilization of importance sampling, we need a discrete approximation of the occurring Gaussian probability densities. Furthermore, we consider the continuous flow of probability mass by a homotopy. The product of likelihood and prior, i.e., $f^L(\tilde{\underline{y}}|\underline{x}) \cdot f^p(\underline{x})$, can be of arbitrary form. Therefore, we cannot consider the entire probability mass, but rather look for a representative set of particles.

The challenge of predicting the trajectory of each particle restricts us to deterministic sampling procedures. Therefore, we utilize the LCD approximation first introduced in [28]. An approximation of an arbitrary probability density can be calculated by solving an optimization problem, where the multivariate generalization of the Cramér-von Mises Distance

$$D = \int_{\mathbb{R}} \omega(b) \int_{\mathbb{R}^N} \left(F_1(\underline{m}, b) - F_2(\underline{m}, b)\right)^2 \, d\underline{m} \, db$$

is minimized. Here $\omega(b) : \mathbb{R}_+ \to [0, 1]$ denotes a weighting function and probability densities $f_1(\underline{x}) : \mathbb{R}^N \to \mathbb{R}$ and $f_2(\underline{x}) : \mathbb{R}^N \to \mathbb{R}$ are compared by their corresponding Localized Cumulative Distributions $F_1(\underline{m}, b)$ and $F_2(\underline{m}, b)$ given in the form

$$F(\underline{m}, b) = \int_{\mathbb{R}^N} f(\underline{x}) \cdot K(\underline{x} - \underline{m}, b) \, d\underline{x} \ ,$$

where $K(\cdot, \cdot)$ a Gaussian kernel at position $\underline{m}$ and size b.

Minimizing this distance, we can systematically maximize the similarity between two probability densities and approximate each occurring continuous probability density by a Dirac mixture density

$$\tilde{f}(\underline{x}) = \sum_{i=1}^{L} \omega^{(i)} \cdot \delta(\underline{x} - \underline{x}^{(i)}) \ , \tag{4}$$

with L particles. Every particle i is positioned at $\underline{x}^{(i)}$ and is weighted by $\omega^{(i)}$. The weights should always be normalized to $\sum \omega^{(i)} = 1$. Further details and a closed-form solution for the approximation of Gaussian densities can be found in [27]. The ingenious of this approach is that it approximates the shape of the underlying PDF. Therefore, we obtain an exhaustive approximation of the probability mass by a consistently distributed set of samples. For the approximation of Gaussian densities, we can work efficiently with pre-calculated approximations of standard normal distributions and adjust these by using the Cholesky decomposition. A non-exhaustive alternative using very few samples to approximate Gaussians is the unscented transformation, which can be found in [14].

B. Importance Sampling

Having established a deterministic sampling procedure, we want to employ it such that we can calculate the posterior $f^e(\underline{x})$ by updating the particle weights. Therefore, we quickly recapitulate importance sampling.

Let us consider the integral $\int g(\underline{x}) \, \mathrm{d}\underline{x}$ factorized by

$$\int g(\underline{x}) \, \mathrm{d}\underline{x} = \int \phi(\underline{x}) f(\underline{x}) \, \mathrm{d}\underline{x} \ ,$$

where $\phi(\underline{x})$ is an arbirtrary function and $f(\underline{x})$ a probability density function. Expanding the function by the proposal probability density function $q(\underline{x})$, the product can be rewritten by

$$\int g(\underline{x}) \, \mathrm{d}\underline{x} = \int \phi(\underline{x}) \cdot \frac{f(\underline{x})}{q(\underline{x})} \cdot q(\underline{x}) \, \mathrm{d}\underline{x} \ . \tag{5}$$

We want to evaluate the densities at discrete points, thus, we approximate the latter $q(\underline{x})$ by a Dirac mixture density $\tilde{q}(\underline{x})$ as given in (4), which leads to

$$\int g(\underline{x}) \, \mathrm{d}\underline{x} \approx \sum_{i=1}^{L} \omega^{(i)} \cdot \phi(\underline{x}^{(i)}) \cdot \frac{f(\underline{x}^{(i)})}{q(\underline{x}^{(i)})} \ . \tag{6}$$

This means, we can approximate the integral by exploiting the sifting property of Dirac mixture densities and only have to evaluate the function $\phi(\underline{x})$ and the probability densities $f(\underline{x})$ and $q(\underline{x})$ at positions $\underline{x}^{(i)}$. Further details on importance sampling in the context of Bayesian filtering can be found in [8].

C. Progressive Filtering

The main challenge using importance sampling is that the quality of the calculated approximation is strongly dependent on the choice of the proposal density $q(\underline{x})$. Therefore, we further introduce the procedure of progressive filtering [18], which is later employed to systematically compute $q(\underline{x})$.

The progressive filtering approach defines a homotopy over the likelihood $f^L(\tilde{\underline{y}}|\underline{x})$, in order to introduce a continuous incorporation of the measurement $\tilde{\underline{y}}$ over the so-called *progression time*. The progression time is a fictional time, which runs from zero to one over a single filtering step. The homotopy is called progressive likelihood function $f^L(\tilde{\underline{y}}, \gamma|\underline{x})$ and is given by

$$f^L(\tilde{\underline{y}}, \gamma \ |\underline{x}) = (f^L(\tilde{\underline{y}}|\underline{x}))^\gamma$$

where $\gamma = [0, 1]$ is the parameter defining the progression time.

Using the progressive likelihood, the interim posterior $f^e(\underline{x}, \gamma)$ is calculated by the progressive Bayesian update

$$f^e(\underline{x}, \gamma) = c(\gamma) \cdot (f^L(\tilde{\underline{y}}|\underline{x}))^\gamma \cdot f^p(\underline{x}) \ . \tag{7}$$

The normalization constant $c(\gamma)$ depends on the progressive likelihood analogous to the normalization constant in (2). The progressive update can be interpreted as adjustable formulation controlling how much information of the measurement $\tilde{\underline{y}}$ is incorporated into the state estimate. Moreover, the prior and posterior can be stated by the extremes of γ, i.e.,

$$\begin{aligned} f^e(\underline{x}) &= f^e(\underline{x}, 1) \ , \\ f^p(\underline{x}) &= f^e(\underline{x}, 0) \ . \end{aligned} \tag{8}$$

We can picture this, by thinking of the prior and the likelihood as two potentials pulling the posterior in their direction. We progressively vary the intensity of the likelihood in the calculation from completely irrelevant to its full contribution. The inherent movement of probability mass throughout the progression is used to systematically obtain the proposal density $q(\underline{x})$ and therefore, the posterior $f^e(\underline{x})$. This is done by portioning the estimation problem into several iteration steps $\Delta\gamma \ll 1$.

IV. Progressive Proposal Density Filtering

So far, we have established the foundation on which this work is built. In this section, we piece together the groundwork and give the final details on the combination of deterministic sampling, importance sampling, and progressive filtering.

Let us first recapture the problem as a result of the introduced approach. We are given a Gaussian prior state estimate $f^p(\underline{x}) \sim \mathcal{N}(\hat{\underline{x}}^p, \mathbf{C}^p)$ and a measurement $\underline{\tilde{y}}$, from which we can point-wise evaluate a progressive likelihood $f^L(\underline{\tilde{y}}, \gamma \mid \underline{x})$. We perform one filter step by a set of small progression steps $\Delta\gamma$, which leads the progression time from $\gamma = 0$ to $\gamma = 1$ and step-wise increases the influence of the likelihood. This results in an recursive procedure, where between two progression steps the resulting interim posteriors $f^e(\underline{x}, \gamma)$ are of high similarity. We also assume interim posteriors to be Gaussian. The main challenge is to calculate interim posteriors $f^e(\underline{x}, \gamma)$, which ultimately leads to the posterior $f^e(\underline{x}) \sim \mathcal{N}(\hat{\underline{x}}^e, \mathbf{C}^e)$.

The computation of the τ-th interim posterior $f^e(\underline{x}, \gamma_\tau)$ for every progression step γ_τ is calculated in two steps. First, we predict a proposal density $q_\tau(\underline{x}) \sim \mathcal{N}(\hat{\underline{x}}^{q\tau}, \mathbf{C}^{q\tau})$ using the continuity assumption resulting from the progressive filtering approach. Second, we calculate the interim posterior $f^e(\underline{x}, \gamma_\tau)$ utilizing importance sampling with the corresponding progressive likelihood $f^L(\underline{\tilde{y}}, \gamma_\tau \mid \underline{x})$, the prior $f^p(\underline{x})$, and the proposal density $q_\tau(\underline{x})$.

We can interpret this methodology as a predictor-corrector method. The optimal proposal density $q_\tau(\underline{x})$ would be the interim posterior $f^e(\underline{x}, \gamma_\tau)$ itself. Thus, we try to predict the result, i.e., the interim posterior, and then correct the prediction using importance sampling. Subsequently, we are describing the predictor and corrector steps in detail, and elaborate on the initialization and parametrization, and complexity.

A. Predictor

As previously described, the goal of the predictor is to systematically calculate a proposal density of the next progression time step, where we exploit the particle flow of the progression. In order to do that, we first need to approximate the previously calculated interim posterior $f^e(\underline{x}, \gamma_\tau)$ by a Dirac mixture density

$$\tilde{f}^e(\underline{x}, \gamma_\tau) = \sum_{i=1}^{L} \omega_\tau^{(i)} \cdot \delta(\underline{x} - \underline{x}_\tau^{(i)}) \ , \tag{9}$$

which gives us a fresh set of particles at the positions $\underline{x}_\tau^{(i)}$. The particles are equal weighted, i.e., $\omega_\tau^{(i)} = 1/L$ for $i = 1 \dots L$.

We assume a certain smoothness of the particle flow over the progression time. This means for small progression time increments $\Delta\gamma$ we assume a linear prediction model to be sufficient to represent the particle displacement. The displacement or velocity of the particles is directly calculated from a one-step history and recomputed in every progression step.

Let us first consider a single particle $\underline{x}_\tau^{(i)}$. We can predict its displacement from one progression time step τ to $\tau + 1$ over the progression time γ by

$$\underline{x}_{\tau+1}^{(i)} = \mathbf{F}_\tau \begin{bmatrix} \underline{x}_\tau^{(i)} \\ \underline{x}_{\tau-1}^{(i)} \end{bmatrix} \ , \tag{10}$$

where the transition matrix $\mathbf{F}_\tau$ consists of two concatenated diagonal matrices and is of the form

$$\mathbf{F}_\tau = \begin{bmatrix} 1{+}T & 0 & \dots & 0 & -T & 0 & \dots & 0 \\ 0 & 1{+}T & \ddots & \vdots & 0 & -T & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 & \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & 1{+}T & 0 & \dots & 0 & -T \end{bmatrix} .$$

The prediction is adjusted by coefficient $T = \Delta\gamma_\tau / \Delta\gamma_{\tau-1}$ for not equidistant time steps, i.e., if $\Delta\gamma_\tau \neq \Delta\gamma_{\tau-1}$.

This can now be easily extend to the prediction of all samples by arranging them column-wise. This means, we can predict all samples by

$$\begin{bmatrix} \underline{x}_{\tau+1}^{(1)} & \dots & \underline{x}_{\tau+1}^{(L)} \end{bmatrix} = \mathbf{F}_\tau \begin{bmatrix} \underline{x}_\tau^{(1)} & \dots & \underline{x}_\tau^{(L)} \\ \underline{x}_{\tau-1}^{(1)} & \dots & \underline{x}_{\tau-1}^{(L)} \end{bmatrix} . \tag{11}$$

We would like to emphasize that the sample positions on the right-hand side of the equation always refer to the corrected samples – not the predicted ones. This means they are taken directly from the approximation of $f^e(\underline{x}, \gamma_\tau)$. Using moment matching, we calculate the proposal density $q_\tau(\underline{x})$ from the predicted sample set.

The last missing piece of the predictor function is to find the corresponding samples $\underline{x}_{\tau-1}^{(i)}$ and $\underline{x}_\tau^{(i)}$ in two independently generated particle sets belonging to $\tilde{f}^e(\underline{x}, \gamma_{\tau-1})$ and $\tilde{f}^e(\underline{x}, \gamma_\tau)$, respectively. In general, we are looking for a sample assignment minimizing the so-called Wasserstein distance [29]. The second-order Wasserstein distance for two Dirac mixture densities $\tilde{f}_r(\underline{x}_r)$ and $\tilde{f}_s(\underline{x}_s)$, given by their particles sets $R = \{\underline{r}_1, \dots, \underline{r}_L\}$ and $S = \{\underline{s}_1, \dots, \underline{s}_L\}$ is calculated by

$$D(\underline{\boldsymbol{x}}_r, \underline{\boldsymbol{x}}_s) = \left(\frac{1}{L} \inf_{\lambda \in \Lambda_L} \left(\sum_{i=1}^{L} d\left(\underline{r}_i, \underline{s}_{\lambda(i)} \right)^2 \right) \right)^{1/2} , \tag{12}$$

where the minimum is computed over all permutations Λ_L of the set $\{1, \dots, L\}$. In this notation $\underline{s}_{\lambda(i)}$ is the i-th element of the permutation λ of S. In this work, the point-wise distance function $d(\cdot, \cdot)$ is the Euclidean distance.

In other words, in order to find the optimal assignment between two sets of particles, we minimize the sum of all single-point distances. The most performant implementation to do so is the so-called Hungarian algorithm [30]. Proficient implementation does not require solving the association problem in every progression step. More details are given in Sec. IV-C.

B. Corrector

The corrector step basically consists of one importance sampling step, where we exploit the proposal density $q_\tau(\underline{x})$ and update the weights of the predicted sample set. The updated sample set is then used to calculate the interim posterior $f^e(\underline{x}, \gamma_\tau)$ via moment matching.

Let us consider the importance sampling as presented in Sec. III-B in the context the given correction problem. We restate and normalize (5) to calculate the interim posterior $f^e(\underline{x}, \gamma_\tau)$ by

$$f^e(\underline{x}, \gamma_\tau) = c(\gamma_\tau) f^L(\underline{\tilde{y}}, \gamma_\tau \mid \underline{x}) \cdot \frac{f^p(\underline{x})}{q_\tau(\underline{x})} \cdot q_\tau(\underline{x}) \ .$$

Analogous to (6), we replace the latter proposal density $q_\tau(\underline{x})$ by its corresponding Dirac mixture, i.e., the particle set

resulting from the predictor. Please keep in mind that these particles have equal weight. We now calculate new weights by particle-wise evaluating the progressive likelihood, the prior, and the proposal density leading to the not yet normalized particle weights

$$\breve{\omega}^{(i)} = f^L(\tilde{\underline{y}}, \gamma_\tau \mid \underline{x}^{(i)}) \cdot \frac{f^p(\underline{x}^{(i)})}{q_\tau(\underline{x}^{(i)})} \ . \tag{13}$$

Finally, normalization of all $\breve{\omega}^{(i)}$, such that $\sum_{i=1}^{L} \tilde{\omega}^{(i)} = 1$, facilitates the calculation of the interim posterior Gaussian $f^e(\underline{x}, \gamma_\tau)$.

C. Initialization, Parametrization, and Complexity

In this section, we give some details on the presented method. First, we talk about some points to be considered during initialization. Then, we discuss important questions, like numerical stability and implementation issues, to keep in mind. Finally, we briefly analyze the complexity of the presented method.

Initialization: In the first step of the progression, we neither can perform a predictor step, since there is no previous step $\tau - 1$, nor can we perform a corrector step as introduced, due to the lack of a proposal density. The initialization is done by using the prior as proposal density. This assumption is often also used by particle filter implementations (e.g., [8]). For a complete filter step this is arguably not the best this choice, but in our case, due to the progression, the prior $f^p(\underline{x})$ (i.e., $f^e(\underline{x}, \gamma_0)$ with $\gamma_0 = 0$) and the first interim posterior $f^e(\underline{x}, \gamma_1)$ can be assumed to be very similar for small $\Delta\gamma_0 = \gamma_1 - \gamma_0$.

Numeric Stability: In extreme cases, the samples given by the proposal Dirac mixture $\tilde{q}_\tau(\underline{x})$ may be positioned, where only very little probability mass of the prior is positioned or the likelihood yields very small values. This happens primarily when an unexpected measurement occurs, i.e., the likelihood and the prior have little overlap. In these cases, the evaluation of (13) yields extremely small (intermediate) results, which can lead to numerical instability. Hence, we recommend taking the logarithm of (13).

Implementation: The computationally most expensive part of the calculation is to solve the association problem. Let us righteously assume, we use small progression steps, leading to similar probability densities over one progression step. Furthermore, we assume the deterministic sampling approximates similar densities without permuting the resulting vectors of samples (i.e., writing the set of samples in a vector, samples with similar positions occupy the same vector position). In this case, the association is already given. In order to check, if the assumption is true, we can evaluate the single point distances for this given association and only evaluate the Hungarian algorithm if the single point distance between at least two samples is surprisingly large.

Parametrization: The proposed method has basically two parameters which values have to be decided on: first, the number of particles used in the progression; second, the number of steps to be performed during the progression. Both parameters have an influence on estimation quality and computational cost. The particles have to be able to meaningfully represent a Gaussian. Although rarely yielding good results, the bare minimum is $n + 1$ particles, with n the dimensionality of the state. A more practical minimum is $2n + 1$, which is also the number of particles the unscented transformation generates [14]. In general, this algorithm works well with the minimum number, as the evaluation shows. If computational power is available, or the sampling time allows it, we propose to experiment with a larger set of samples, like $2n^2 + 1$, which attains a very good coverage of the probability mass of a Gaussian. The progression step size on the other hand strongly depends on the nonlinearity of the estimation problem. It can be compared to the sampling rate of a control system, where strong nonlinearities tend to generate higher frequencies. Thus, weakly nonlinear problems can be handled with very few steps, whereas strongly nonlinear problems need smaller and therefore more steps in order to perform well. As can be seen in the evaluation, we have achieved a good performance for the considered problem with $P = 14$ progression steps with an equidistant step length of $\Delta\gamma \approx 0.07$.

Complexity: Let us briefly summarize the computational complexity of the presented algorithm. We consider P progression steps with L samples in a N-dimensional state space. The deterministic sampling and the importance sampling have components, like the Cholesky decomposition and matrix inversion with the complexity $O(N^3)$. Reshaping the standard normal distributed samples to represent the Gaussian distribution accordingly, the deterministic sampling also performs a matrix multiplication with the complexity $O(N^2 \cdot L)$. The association can be solved in $O(L^3)$ using the Hungarian algorithm. If the association problem does not have to be solved explicitly, only the single point distances for a given association are computed, i.e., we are left with $O(L)$. The above stated calculations have to be performed for all P progression step. Since L has to be larger than N, we are left with $O(P \cdot N^2 \cdot L)$, if the Hungarian algorithm does not have to be evaluated, and with $O(P \cdot L^3)$, if we solve the association problem in every progression step explicitly.

V. Evaluation

We evaluate the proposed progressive Gaussian filter by means of a simulation, which was performed using the Nonlinear Estimation Toolbox for MATLAB [31]. We compare the novel filter to the Extended Kalman Filter (EKF), the Smart-Sampling Kalman Filter (S^{2}KF) [16], the Sampling Importance Resampling Particle Filter (SIR-PF) with resampling threshold using a normalized effective sample size of 0.5 [8], and finally, another progressive filter, namely the Progressive Gaussian Filter using explicit likelihoods (PGF) presented in [25].

This comparison covers the most common methodological groups of state-of-the-art nonlinear Bayesian filters, namely Kalman filter with explicit linearization, linear regression Kalman filter (LRKF) with statistical linearization, particle filter, and progressive filter. In the simulation we use a target tracking example, where we want to estimate five-dimensional hidden system state vector

$$\underline{x}_k = \begin{bmatrix} x_k & y_k & \phi_k & v_k & \varphi_k \end{bmatrix}^\top \ , \tag{14}$$

where x_k and y_k define the position on a plane, ϕ_k is the orientation, and v_k and φ_k are the positional and angular velocity, respectively. We consider a noisy directional distance sensor, generating polar coordinate measurements by the nonlinear sensor model

$$\underline{\boldsymbol{y}}_k = \begin{bmatrix} \sqrt{\boldsymbol{x}_k^2 + \boldsymbol{y}_k^2} \\ \boldsymbol{\psi}_k \end{bmatrix} + \underline{\boldsymbol{v}}_k, \tag{15}$$

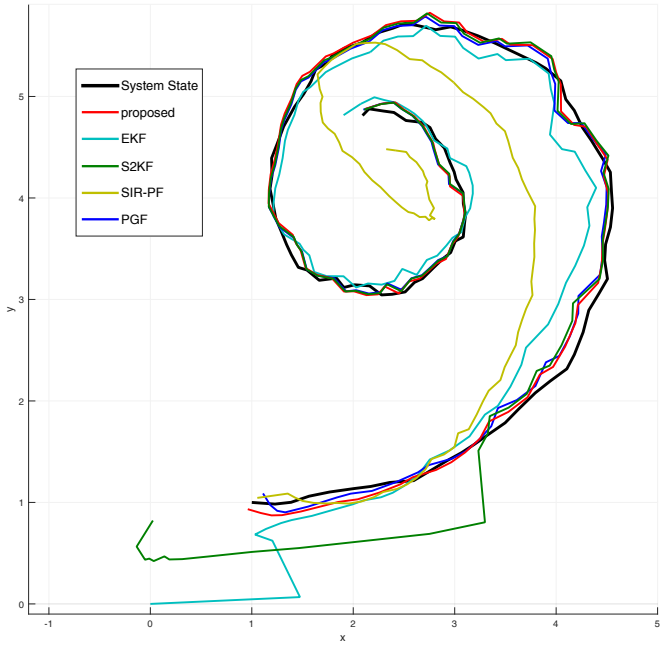

Fig. 1. Evaluation of one exemplary simulation run of system (16), with the start position $[x_0, y_0]^\top = [1, 1]^\top$. The trajectory of the actual system state is depicted as solid black line. The trajectories estimated by the filters in comparison are shown by thin colored lines, where we have depicted the proposed filter (red), the EKF (cyan), the S^{2}KF (green), the SIR-PF (yellow), and the PGF (blue). The filters are initialized with the mean position at $[\hat{x}_0, \hat{y}_0]^\top = [0, 0]^\top$. They are depicted after the incorporation of the first measurement to illustrate the difference in the initial convergence.

with $\boldsymbol{\psi}_k$ angle in direction of the target, which can be computed by atan2$(\boldsymbol{y}_k, \boldsymbol{x}_k)$. The measurement noise $\underline{\boldsymbol{v}}_k = [\boldsymbol{v}_k^{(d)}, \boldsymbol{v}_k^{(\psi)}]^\top$ is state-independent white noise characterized by the normal distribution $\mathcal{N}(\underline{0}, \text{diag}([0.01, 0.0001])$. System state changes over time are modelled by

$$\underline{\boldsymbol{x}}_{k+1} = \begin{bmatrix} \boldsymbol{x}_k + \boldsymbol{v}_k + \cos(\boldsymbol{\phi}_k) \\ \boldsymbol{y}_k + \boldsymbol{v}_k + \sin(\boldsymbol{\phi}_k) \\ \boldsymbol{\phi}_k + \boldsymbol{\varphi}_k \\ \boldsymbol{v}_k \\ \boldsymbol{\varphi}_k \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ u_k^{(v)} \\ u_k^{(\varphi)} \end{bmatrix} + \begin{bmatrix} \boldsymbol{w}_k^{(x)} \\ \boldsymbol{w}_k^{(y)} \\ \boldsymbol{w}_k^{(\phi)} \\ \boldsymbol{w}_k^{(v)} \\ \boldsymbol{w}_k^{(\varphi)} \end{bmatrix}, \quad (16)$$

where we can influence the positional and angular velocity by the input vector $u_k = [u_k^{(v)}, u_k^{(\varphi)}]^\top$. The system noise $\underline{\boldsymbol{w}}_k = [\boldsymbol{w}_k^{(x)}, \boldsymbol{w}_k^{(y)}, \boldsymbol{w}_k^{(\phi)}, \boldsymbol{w}_k^{(v)}, \boldsymbol{w}_k^{(\varphi)}]^\top$ is characterized by the normal distribution $\mathcal{N}(\underline{0}, \text{diag}([1, 1, 0.01, 1, 0.01] \cdot 10^{-3})$. The system state is initialized by $\underline{x}_k = [1, 1, 0, 20, 5]^\top$ and we constantly decrease the positional velocity and increase the rotational velocity by $\underline{u}_k = [-0.1, 0.1]^\top$. The simulation is run over 100 time steps and results in a spiraling trajectory as depicted in Fig. 1, where we have plotted one exemplary simulation outcome.

In the evaluation, we have used the following filter setup:

Proposed	#particles: 51, #progression steps: 14
EKF	No parameters to decide on
S^{2}KF	#particles: 201
SIR-PF	#particles: 1000
PGF	#particles: 51

The filters are initialized with a rough estimate $\underline{\boldsymbol{x}}_0^e$ with mean $\underline{\hat{x}}_0^e = [0, 0, 0, 10, 2]^\top$ and covariance matrix $\mathbf{C}_k^e = \text{diag}([10, 10, 0.1, 5, 5])$.

The outcome of 100 Monte-Carlo simulations is depicted in Fig. 2. The dashed lines indicate the maximum error (MAXE) and the solid lines show the root mean squared error (RMSE). Although due to different reasons, the EKF, as well as the SIR-PF have problems track the trajectory of the hidden state. The linearization around the mean is not a sufficient approximation of the presented problem, which lead to a poor performance of the EKF. On the other hand, the particle filter has problems to adjust for particle degeneration. The S^{2}KF and both progressive filters, the PGF and the proposed approach, are able to reconstruct the hidden state from the measurements. In the presented scenario the S^{2}KF shows a solid performance, but having some problems in the initialization phase. We can see a significant improvement after 15 steps in average and after 25 steps in the worst case. This is outperformed by both progressive filters, whereas the proposed approach has the best performance in the initialization phase. On the other hand the performance does not reach the same level in the long run. Where in average the S^{2}KF, PGF, and the proposed method show only little difference, the proposed method can be seen to be less performant in the worst case.

VI. Conclusions and Future Work

We presented a novel approach to Gaussian filtering using the progressive Bayesian filter framework. The combination of importance sampling and particle flow allows us for good estimation results in critical situations, i.e., if the likelihood and the prior estimate have very little overlap. Especially in these cases particle filters suffer from particle degeneration and show very poor performance.

The combination of importance sampling and particle flow is very general. Requirements are a consistent and systematic sampling procedure and re-approximation by a continuous probability density, where the samples can be evaluated. Therefore, a generalization of this work to multi-modal density representations, e.g, Gaussian mixture densities, will be considered in future work. Furthermore, other ways of modelling the particle flow should be investigated in upcoming extensions to this work.

References

[1] Christof Chlebek and Uwe D. Hanebeck, "Stochastic Nonlinear Model Predictive Control Based on Deterministic Scenario Generation", *in Proceedings of the 18th International Conference on Information Fusion (Fusion 2015)*, Washington D. C., USA, July 2015.

[2] Christof Chlebek and Uwe D. Hanebeck, "Approximation of Stochastic Nonlinear Closed-Loop Model Predictive Control with Application to Miniature Walking Robots", *in Proceedings of the 2016 European Control Conference (ECC 2016)*, Aalborg, Denmark, June 2016.

[3] Jannik Steinbring, Marcus Baum, Antonio Zea, Florian Faion, and Uwe D. Hanebeck, "A Closed-Form Likelihood for Particle Filters to Track Extended Objects with Star-Convex RHMs", *in Proceedings of the 2015 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2015)*, San Diego, California, USA, September 2015.

[4] Antonio Zea, Florian Faion, and Uwe D. Hanebeck, "Exploiting Clutter: Negative Information for Enhanced Extended Object Tracking", *in Proceedings of the 18th International Conference on Information Fusion (Fusion 2015)*, Washington D. C., USA, July 2015.

[5] Yaakov Bar-Shalom, X. Rong Li, and Thiagalingam Kirubarajan, *Estimation with Applications to Tracking and Navigation*, New York Chichester Weinheim Brisbane Singapore Toronto: Wiley-Interscience, 2001.

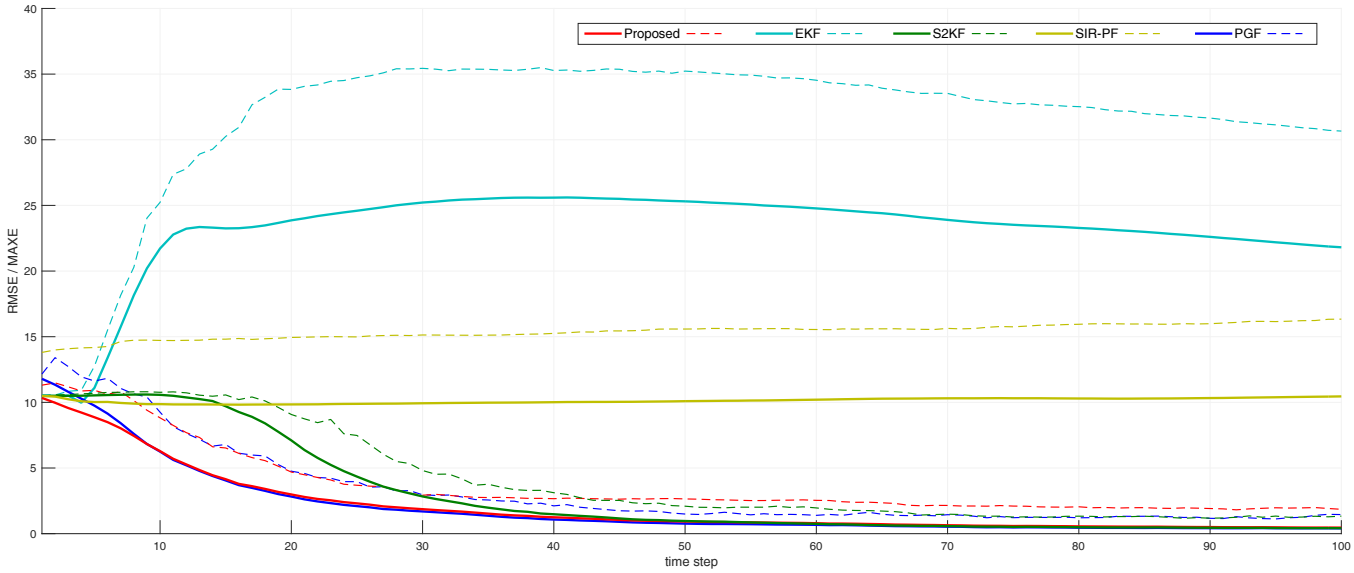

Fig. 2. This figure shows the RMSE (solid lines) and the MAXE (dashed lines) of 100 Monte-Carlo simulation runs of system (16). The performance of the proposed filter (red), the EKF (cyan), the S^2KF (green), the SIR-PF (yellow), and the PGF (blue) are depicted over 100 time steps.

[6] Rudolf E. Kalman, "A New Approach to Linear Filtering and Prediction Problems", *in Transaction of the ASME - Journal of Basic Engineering*, pp. 35–45, March 1960.

[7] Arnaud Doucet, and Adam M. Johansen, "A Tutorial on Particle Filtering and Smoothing: Fifteen Years Later", *in Oxford Handbook of Nonlinear Filtering*, pp. 656–704, 2011.

[8] Branko Ristic, Sanjeev Arulampalam, and Neil Gordon, *Beyond the Kalman Filter: Particle filters for Tracking Applications*, Artech House Publishers, 2004.

[9] K. Ito and K. Xiong, "Gaussian Filters for Nonlinear Filtering Problems", *in IEEE Transactions on Automatic Control*, vol. 45, no. 5, pp. 910–927, May 2000.

[10] Tine Lefebvre, Herman Bruyninckx, Joris De Schutter, "Kalman Filters for Non-Linear Systems: A Comparison of Performance", *in International Journal of Control*, vol. 77, no. 7, pp. 639–653, 2004.

[11] Dan Simon, *Optimal State Estimation*, 1st ed. Wiley & Sons, 2006.

[12] T. Lefebvre, H. Bruyninckx, and J. De Schutter, "A Linear Regression Kalman Filter", *in Nonlinear Kalman Filtering for Force-Controlled Robot Tasks*, ser. Springer Tracts in Advanced Robotics, vol. 19, 2005.

[13] Jannik Steinbring and Uwe D. Hanebeck, "LRKF Revisited: The Smart Sampling Kalman Filter (S2KF)", *in Journal of Advances in Information Fusion*, vol. 9(2), pp. 106–123, December 2014.

[14] Simon J. Julier and Jeffrey K. Uhlmann, "Unscented filtering and nonlinear estimation", *in Proceedings of the IEEE*, Vol. 92, No. 3, pp. 401–422, March 2004

[15] Bin Jia, Ming Xin and Yang Cheng, "High-Degree Cubature Kalman Filter", *in Automatica*, vol. 49, no. 2, pp. 510–518, February 2013.

[16] Jannik Steinbring and Uwe D. Hanebeck, "S2KF: The Smart Sampling Kalman Filter", *in Proceedings of the 16th International Conference on Information Fusion (Fusion 2013)*, Istanbul, Turkey, July 2013.

[17] J. H. Kotecha and P. M. Djuric, "Gaussian Particle Filtering", *in IEEE Transactions on Signal Processing*, vol. 51, no. 10, pp. 2592–2601, October 2003.

[18] Uwe D. Hanebeck, Kai Briechle, and Andreas Rauh, "Progressive Bayes: A New Framework for Nonlinear State Estimation", *in Proceedings of SPIE, AeroSense Symposium*, vol. 5099, pp.256–267, Orlando, Florida, USA, May 2003.

[19] F. Daum and J. Huang, "Nonlinear filters with log-homotopy", *in Proc. SPIE Conf. Signal and Data Processing of Small Targets*, San Diego, CA, USA, Sept. 2007.

[20] Fred Daum, Jim Huang, and Arjang Noushin, "Exact particle flow for nonlinear filters", *in in Proc. SPIE Conference Signal Processing, Sensor Fusion, Target Recognition*, Orlando, Florida, USA, April 2010.

[21] Tao Ding and Mark J. Coates, "Implementation of the Daum-Huang exact-flow particle filter", *in 2012 IEEE/SP Workshop on Statistical Signal Processing (SSP)*, pp. 257–260, Ann Arbor, Michigan, USA, 2012.

[22] Uwe D. Hanebeck, Jannik Steinbring, "Progressive Gaussian Filtering Based on Dirac Mixture Approximations", *in Proceedings of the 15th International Conference on Information Fusion (Fusion 2012)*, Singapore, July 2012.

[23] Jonas Hagmar, Mats Jirstrand, Lennart Svensson, and Mark Morelande, "Optimal parameterization of posterior densities using homotopy", *in Proceedings of the 14th International Conference on Information Fusion (Fusion 2011)*, Chicago, USA, July 2011.

[24] Uwe D. Hanebeck, "PGF 42: Progressive Gaussian Filtering with a Twist", *in Proceedings of the 16th International Conference on Information Fusion (Fusion 2013)*, Istanbul, Turkey, July 2013.

[25] Jannik Steinbring, Uwe D. Hanebeck, "Progressive Gaussian Filtering Using Explicit Likelihoods", *in Proceedings of the 17th International Conference on Information Fusion (Fusion 2014)*, Salamanca, Spain, July 2014.

[26] Jannik Steinbring and Uwe D. Hanebeck, "GPU-Accelerated Progressive Gaussian Filtering with Applications to Extended Object Tracking", *in Proceedings of the 18th International Conference on Information Fusion (Fusion 2015)*, Washington D. C., USA, July 2015.

[27] Igor Gilitschenski, Uwe D. Hanebeck, "Efficient Deterministic Dirac Mixture Approximation", *in Proceedings of the 2013 American Control Conference (ACC 2013)*, Washington D.C., USA, June 2013.

[28] U. D. Hanebeck and V. Klumpp, "Localized Cumulative Distributions and a Multivariate Generalization of the Cramér-von Mises Distance", *in Proceedings of the 2008 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2008)*, pp. 33-39, Seoul, Republic of Korea, Aug. 2008.

[29] Cédric Villani, "The Wasserstein Distances", *in in Optimal Transport: Old and New*, Chap. 6, pp. 93–11, Springer, 2008, ISBN 978-3-540-71049-3.

[30] Harold W. Kuhn, "The Hungarian Method for the Assignment Problem", *in Naval Research Logistics Quarterly*, pp. 83–97, 1955.

[31] Jannik Steinbring, "Nonlinear Estimation Toolbox", URL: https://bitbucket.org/nonlinearestimation/toolbox