Progressive Gaussian Filtering Using Explicit Likelihoods

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

***Abstract*—In this paper, we introduce a new sample-based Gaussian filter. In contrast to the popular Nonlinear Kalman Filters, e.g., the UKF, we do not rely on linearizing the measurement model. Instead, we take up the Gaussian progressive filtering approach introduced by the PGF 42 but explicitly rely on likelihood functions. Progression means, we incorporate the information of a new measurement gradually into the state estimate. The advantages of this filtering method are on the one hand the avoidance of sample degeneration and on the other hand an adaptive determination of the number of likelihood evaluations required for each measurement update. By this means, less informative measurements can be processed quickly, whereas measurements containing much information automatically receive more emphasis by the filter. These properties allow the new filter to cope with the demanding problem of very narrow likelihood functions in an efficient way.**

***Index Terms*—Progressive Filtering, Bayesian Inference, Deterministic Gaussian Sampling, Extended Object Tracking**

I. Introduction

Estimating the hidden state of a stochastic dynamic system based on noisy measurements is an important and demanding process required in many fields of engineering and science such as navigation, (extended) object and group tracking [1]–[5], and robotics [6]. Due to the fact that we have to rely on imperfect measurements and utilized models are a simplification of the real world, only a probability distribution of the current system state can be obtained.

For linear systems corrupted by additive, white, and state-independent Gaussian noise, the well-known Kalman Filter yields the optimal estimator in the sense of a Minimum Mean Square Error (MMSE) [7], [8]. However, almost all systems of interest are nonlinear and may suffer from non-additive noise. Hence, other, preferably nonlinear estimators are required to obtain adequate results. Moreover, maintaining the true, in general multimodal, state probability distribution is intractable, and thus, only suboptimal solutions can be obtained [2].

A widespread class of nonlinear estimators are the Particle Filters (PFs) [2], [9], [10]. They represent their state estimate as a set of weighted particles (or samples) and update it by reweighting the particles using the measurement model in form of the likelihood function and subsequently performing random importance (re)sampling. The advantages of such filters are their easy implementation as well as the ability to capture arbitrary state distributions and multimodalities. However, they become computational intractable when it comes to higher state dimensions.

A reasonable simplification of the estimation problem is the reduction to (or assumption of) a Gaussian distributed state. Multimodal estimators, i.e., Gaussian mixture estimators, can then be built on top of these filters [11]. The advantage of Gaussian estimators is the compact and constant amount of information describing their state estimates [12]. A special Particle Filter that realizes a Gaussian estimator is the so-called Gaussian Particle Filter (GPF) [9].

Another class of filters are the Nonlinear Kalman Filters. These are Gaussian estimators that approximate the nonlinear measurement relationships as a linear one. In so doing, they do not need explicit likelihood functions, are easy to use, and possess good runtime performance. Nevertheless, the linearization can lead to a diminished estimation performance. Explicit linearization techniques are used by the Extended Kalman Filter (EKF) and its iterated versions (IEKF) [8]. In contrast, Linear Regression Kalman Filters (LRKFs) rely on an implicit statistical linearization. Examples are the Unscented Kalman Filter (UKF) [12], the Cubature Kalman Filter (CKF) [13], the Gaussian Filter (GF) [14], or the Smart Sampling Kalman Filter (S^{2}KF) [15].

In [16], a Gaussian estimator, called Progressive Gaussian Filter 42 (PGF 42), was presented that avoids such linearization and still does not need any explicit likelihood function. Hence, this filter can directly be used as an replacement for an LRKF. The key idea of the PGF 42 is to incorporate new measurements gradually into the state estimate by using progressive filtering in combination with deterministic Gaussian sampling. Furthermore, an explicit likelihood is avoided by additionally estimating the actual noise *realizations* for each measurement update besides the actual system state, i.e., performing a state augmentation with the noise variables. However, in case of many noise variables, e.g., when processing many measurements at once, this state augmentation can be too demanding for the estimator as the augmented state becomes very large.

In order to overcome this issue and achieve better estimation results, in this paper we take up the Gaussian progressive filtering approach introduced by the PGF 42 but explicitly rely on likelihood functions instead. That is, we give up the approach of directly working with the generative measurement model, and hence, avoid estimating the actual noise realizations. In case of nonlinear measurement models corrupted by additive Gaussian noise, the new filter is very similar to PGF 42 as in such cases the PGF 42 does not have to estimate any noise variables, too.

Moreover, the PGF 42 has three parameters: forced sample weight ratio, maximum allowed deviation between two successive intermediate Gaussians, and number of samples per recursion step. Based on a study of the progression characteristics, we reduce the parameters required by the new PGF to the number of samples per recursion step.

Other filters relying on the progressive approach are presented in [17], [18]. Here, the filters rely on solving partial or ordinary differential equations in order to perform a measurement update. In [19], an EKF using a progressive update is presented. The authors of [20] also take up the PGF 42 approach and adapt it to the estimation of angular systems using nonlinear measurement models.

This paper is structured as follows. First, we formulate the general problem of Bayesian inference and the difficulties of sample degeneration when relying on sample-based approaches. In Sec. III, we describe the approach of progressive likelihood functions. Based on this, in Sec. IV, we introduce a new Progressive Gaussian Filter that directly works with likelihoods. An evaluation of the new filter against PFs is performed in Sec. V. Finally, the conclusions are given in Sec. VI.

II. Problem Formulation

We consider estimating the hidden state $\underline{\boldsymbol{x}}_k$ of a discrete-time stochastic dynamic system based on noisy measurements[1]. The relationship between $\underline{\boldsymbol{x}}_k$ and a received measurement $\tilde{\underline{y}}_k$ is described according to the nonlinear measurement model

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

where the subscript k denotes the discrete time step, $\underline{\boldsymbol{y}}_k$ the measurement random vector from which $\tilde{\underline{y}}_k$ originates, and $\underline{\boldsymbol{v}}_k$ an arbitrary state-independent measurement noise process.

Our goal is to obtain a state estimate at time step k after incorporating k received measurements $\tilde{\underline{y}}_1, \tilde{\underline{y}}_2, \ldots, \tilde{\underline{y}}_k$ approximated as a conditional Gaussian distribution according to

$$\begin{aligned} f_k^e(\underline{x}_k) &= f(\underline{x}_k \,|\, \tilde{\underline{y}}_k, \tilde{\underline{y}}_{k-1}, \ldots, \tilde{\underline{y}}_1) \\ &\approx \mathcal{N}(\underline{x}_k; \hat{\underline{x}}_k^e, \mathbf{C}_k^e) \ . \end{aligned} \tag{2}$$

As we receive new measurements over time, a recursive determination of (2) by exploiting Bayes' rule is desired, i.e., a recursive estimator. That is, given a prior Gaussian state estimate based on the last $k-1$ measurements

$$\begin{aligned} f_k^p(\underline{x}_k) &= f(\underline{x}_k \,|\, \tilde{\underline{y}}_{k-1}, \ldots, \tilde{\underline{y}}_1) \\ &\approx \mathcal{N}(\underline{x}_k; \hat{\underline{x}}_k^p, \mathbf{C}_k^p) \ , \end{aligned}$$

and assuming that a new measurement $\tilde{\underline{y}}_k$ is conditionally independent of these previously received measurements given this prior state estimate, the updated (or corrected) Gaussian state estimate is

$$\begin{aligned} f_k^e(\underline{x}_k) &= c_k \cdot f(\tilde{\underline{y}}_k \,|\, \underline{x}_k) \cdot f_k^p(\underline{x}_k) \\ &\approx \mathcal{N}(\underline{x}_k; \hat{\underline{x}}_k^e, \mathbf{C}_k^e) \ , \end{aligned} \tag{3}$$

where $f(\tilde{\underline{y}}_k \,|\, \underline{x}_k)$ is the likelihood function and c_k only a normalization constant.

Usually, only the generative measurement model (1) is at hand and the likelihood function has to be obtained from it according to

$$\begin{aligned} f_k^L(\underline{x}_k) &:= f(\tilde{\underline{y}}_k \,|\, \underline{x}_k) \\ &= \int \delta(\tilde{\underline{y}}_k - \underline{h}_k(\underline{x}_k, \underline{v}_k)) \cdot f_k^v(\underline{v}_k) \, \mathrm{d}\underline{v}_k \ , \end{aligned} \tag{4}$$

where $\delta(\cdot)$ denotes the Dirac delta function and $f_k^v(\cdot)$ the measurement noise probability density function.

However, even if the likelihood (4) is available in closed form, it is almost always impossible to solve the Bayesian update (3) analytically except for special cases such as linear measurement models corrupted by additive Gaussian noise. A naïve approximative solution to this would be as follows. First, sample the prior Gaussian in some way (randomly or deterministically), that is, create a Dirac mixture approximation

$$f_k^p(\underline{x}_k) \approx \sum_{i=1}^{M} w_i \cdot \delta(\underline{x}_k - \underline{x}_{k,i}) \tag{5}$$

with sample positions $\underline{x}_{k,i}$ and corresponding weights w_i which sum up to one. Second, plug this into (3) in order to obtain a Dirac mixture approximation of the posterior state estimate

$$f_k^e(\underline{x}_k) \approx c_k \cdot \sum_{i=1}^{M} w_i \cdot f_k^L(\underline{x}_{k,i}) \cdot \delta(\underline{x}_k - \underline{x}_{k,i}) \ . \tag{6}$$

Finally, a subsequent moment matching is used to obtain a posterior Gaussian distribution. As can be seen, this straightforward approach leaves the sample positions $\underline{x}_{k,i}$ unchanged and only reweights the samples according to the respective likelihood values $f_k^L(\underline{x}_{k,i})$.

This approach is simple to implement and in theory solves the Bayesian update problem. However, in practice it suffers from the serious problem of sample degeneracy. If the intersection of the likelihood support and a significant amount of the probability mass of the prior state density is small, only a few samples will remain with substantial weights contributing to the posterior state density. In extreme cases, none or only a single non-zero sample are left over, and thus, no valid posterior Gaussian state density can be obtained. The solution of increasing the number of samples is intractable in larger state spaces due to the curse of dimensionality. Moreover, this would worsen the problem that samples placed in irrelevant regions of state space waste computational power and time as they will not contribute to the posterior density. Particle Filters try to improve this situation by using proper proposal densities. Nevertheless, these are not easy to find and require an individual, problem specific treatment [21], [22]. Consequently, a more elaborate measurement update technique is required to solve the Bayesian update problem satisfactorily.

[1] Random variables are printed in bold face and vectors are underlined.

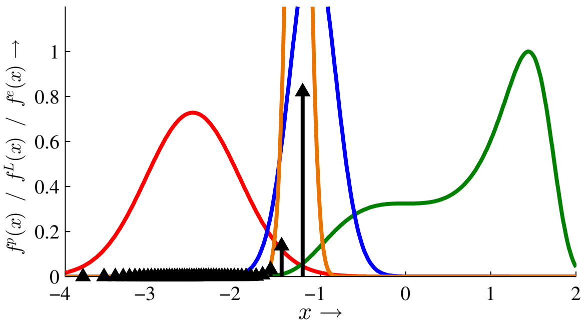


Figure 1: Naïve Bayesian inference. Prior Gaussian state estimate (red), likelihood (green), true posterior Gaussian state estimate (blue), 50 reweighted samples (black arrows), and sample-based posterior Gaussian (orange) with a too low variance.

Fig. 1 illustrates these well-known problems in Bayesian filtering in case of a one-dimensional system state and 50 samples. It should be noted that almost all samples are down-weighted to (nearly) zero and only a couple of samples with larger weights remain. Hence, only the rightmost samples contribute to the posterior state estimate which, in turn, is of too low variance.

III. Progressive Filtering

In this Section, we recapitulate the idea of the progressive filtering approach [16], [23]. Here, the progressive likelihood provides the basis for this filtering technique.

Definition III.1 (Progressive Likelihood) *Let $f^L(\underline{x})$ be a likelihood function*[2]*. Then, a function $f^L(\underline{x}, \gamma)$ with progression parameter $\gamma \in [0, 1]$ is a progressive likelihood if it satisfies for every $\underline{x}$*

$$f^L(\underline{x}, \gamma) = \begin{cases} 1 & , \textit{ if } \gamma = 0 \\ f^L(\underline{x}) & , \textit{ if } \gamma = 1 \end{cases} .$$

Using this definition, we can formulate a progressive Bayesian inference

$$f^e(\underline{x}, \gamma) = c(\gamma) \cdot f^L(\underline{x}, \gamma) \cdot f^p(\underline{x}) \ ,$$

with normalization constant $c(\gamma)$ and progression parameter $\gamma \in [0, 1]$. This parameter controls how much information of the received measurement $\underline{\tilde{y}}$ will be used to correct the prior state estimate $f^p(\underline{x})$. For the extreme case $\gamma = 0$

$$f^e(\underline{x}, 0) = f^p(\underline{x})$$

holds, that is, no information of the received measurement is processed yet, and thus, the posterior estimate equals the prior. The other extreme is $\gamma = 1$. Here, due to the fact that $f^L(\underline{x}, 1) = f^L(\underline{x})$, the entire information of the measurement $\underline{\tilde{y}}$ is processed and we have

$$f^e(\underline{x}, 1) = f^e(\underline{x}) \ ,$$

which is equal to the unmodified Bayesian update (3).

Progressive Bayesian inference gives us the possibility to include the information of a given measurement $\underline{\tilde{y}}$ gradually into the prior state estimate $f^p(\underline{x})$ by using a recursive algorithm. Suppose we have a Bayesian progression with parameter γ

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

and one with parameter $\gamma + \Delta$

$$f^e(\underline{x}, \gamma + \Delta) = c(\gamma + \Delta) \cdot f^L(\underline{x}, \gamma + \Delta) \cdot f^p(\underline{x}) \ , \tag{8}$$

where $\Delta \geq 0$ and $\gamma + \Delta \leq 1$. Based on (7) and (8), we can derive the recursion

$$f^e(\underline{x}, \gamma + \Delta) = \frac{c(\gamma + \Delta) \cdot f^L(\underline{x}, \gamma + \Delta)}{c(\gamma) \cdot f^L(\underline{x}, \gamma)} \cdot f^e(\underline{x}, \gamma) \ , \tag{9}$$

which transfers the Bayesian progression $f^e(\underline{x}, \gamma)$ with step size Δ to the Bayesian progression $f^e(\underline{x}, \gamma + \Delta)$. Starting with $f^e(\underline{x}, 0)$, i.e., the prior state estimate $f^p(\underline{x})$, and proper (not necessarily equal) step sizes Δ, we can recursively obtain the desired final posterior state estimate $f^e(\underline{x})$ by recursively exploiting Eq. (9). If each recursion step could be performed analytically, we could solve the actual desired Bayesian update (3) in closed-form, too. Unfortunately, this is in general not the case, and hence, we have to rely on sample-based methods again.

For that reason, we start the recursion by sampling the prior Gaussian state estimate $f^p(\underline{x})$, i.e., computing the Dirac mixture approximation (5). Together with an initial step size Δ_0, we can approximate the first recursion step according to

$$\begin{aligned} f^e(\underline{x}, \Delta_0) &= \frac{c(\Delta_0) \cdot f^L(\underline{x}, \Delta_0)}{f^L(\underline{x}, 0)} \cdot f^p(\underline{x}) \\ &\approx c(\Delta_0) \cdot \sum_{i=1}^{M} w_i \cdot f^L(\underline{x}_i, \Delta_0) \cdot \delta(\underline{x} - \underline{x}_i) \ . \end{aligned}$$

Using this Dirac mixture approximation as basis for an approximation of the second recursion step (with step size Δ_1) we obtain

$$\begin{aligned} f^e(\underline{x}, \Delta_1 + \Delta_0) &= \frac{c(\Delta_1 + \Delta_0) \cdot f^L(\underline{x}, \Delta_1 + \Delta_0)}{c(\Delta_0) \cdot f^L(\underline{x}, \Delta_0)} \cdot f^e(\underline{x}, \Delta_0) \\ &\approx c(\Delta_1 + \Delta_0) \cdot \sum_{i=1}^{M} w_i \cdot f^L(\underline{x}_i, \Delta_1 + \Delta_0) \cdot \delta(\underline{x} - \underline{x}_i) \ . \end{aligned}$$

After performing $n + 1$ recursion steps in this way, where $\Delta_0 + \ldots + \Delta_n = 1$, the final Dirac mixture approximation of the posterior state estimate becomes

$$f^e(\underline{x}) \approx c \cdot \sum_{i=1}^{M} w_i \cdot f^L(\underline{x}_i) \cdot \delta(\underline{x} - \underline{x}_i) \ .$$

As one can see, this recursion procedure is equivalent to directly reweighting the initial Dirac mixture weights w_i with the final likelihood values $f^L(\underline{x}_i)$ (see (6)), and hence, suffers from the same problems described in Sec. II. Consequently, further modifications are required as this recursion alone does not yield any benefits for the Bayesian update problem.

[2] For readability, we omit the time index k if its not needed.

IV. Progressive Gaussian Filtering Using Explicit Likelihoods

The presented recursive measurement update from Sec. III still suffers from the problem of sample degeneration. This is caused by the static sample positions $\underline{x}_i$. In order to avoid sample degeneration, samples have to move in some way in the state space during the recursion. Such sample movement during a measurement update is also referred to as *particle flow* [18]. A particle flow moves the utilized samples into the proper regions of state space, i.e., to the overlapping supports of prior state estimate and (progressive) likelihood.

For the introduced progressive recursion (9), the solution is to perform some sort of resampling after each recursion step in order to obtain a new set of equally weighted samples for the next recursion step. That is, the sample reweighting from one recursion step will be compensated by a movement of the samples. This technique is the basis of all Particle Filters [2] but, in contrast to these, we perform resampling multiple times during *each* measurement update.

We take up the resampling approach from the PGF 42 introduced in [16], which only considers measurement models suffering from additive Gaussian noise, and extend it to the case of explicit likelihoods (4). As we already force the posterior state estimate to be Gaussian and sampling therefore is relatively simple, the idea of the proposed approach is to approximate each intermediate state estimate $f^e(\underline{x}, \gamma)$ as Gaussian, too. Consequently, the basis of each recursion step is a set of new, equally weighted, samples representing the intermediate Gaussian $f^e(\underline{x}, \gamma)$.

A. Gaussian-Based Recursion

From now on, we utilize the specific progressive likelihood defined as

$$f^L(\underline{x}, \gamma) := [f^L(\underline{x})]^\gamma \enspace .$$

Based on this, we can simplify the recursion (9) to

$$f^e(\underline{x}, \gamma + \Delta) = \frac{c(\gamma + \Delta)}{c(\gamma)} \cdot [f^L(\underline{x})]^\Delta \cdot f^e(\underline{x}, \gamma) \enspace . \tag{10}$$

Performing one recursion step based on intermediate Gaussians is split into three parts. First, we compute a Dirac mixture approximation with M equally weighted samples of the current intermediate Gaussian

$$f^e(\underline{x}, \gamma) \approx \mathcal{N}(\underline{x}; \hat{\underline{x}}^{(\gamma)}, \mathbf{C}^{(\gamma)}) \approx \frac{1}{M} \sum_{i=1}^{M} \delta(\underline{x} - \underline{x}_i^{(\gamma)}) \enspace . \tag{11}$$

Second, by plugging this into (10), we obtain a Dirac mixture approximation of the next intermediate posterior state estimate

$$\begin{aligned} f^e(\underline{x}, \gamma + \Delta) &\approx \sum_{i=1}^{M} \underbrace{\frac{c(\gamma + \Delta)}{c(\gamma) \cdot M} \cdot [f^L(\underline{x}_i^{(\gamma)})]^\Delta}_{:= \tilde{w_i}^{(\gamma+\Delta)}} \cdot \delta(\underline{x} - \underline{x}_i^{(\gamma)}) \\ &= \sum_{i=1}^{M} \tilde{w_i}^{(\gamma+\Delta)} \cdot \delta(\underline{x} - \underline{x}_i^{(\gamma)}) \enspace . \end{aligned}$$

After normalizing the weights $\tilde{w_i}^{(\gamma+\Delta)}$ according to

$$w_i^{(\gamma+\Delta)} := \frac{\tilde{w_i}^{(\gamma+\Delta)}}{\sum_{j=1}^{M} \tilde{w_j}^{(\gamma+\Delta)}} = \frac{[f^L(\underline{x}_i^{(\gamma)})]^\Delta}{\sum_{j=1}^{M} [f^L(\underline{x}_j^{(\gamma)})]^\Delta} \enspace , \tag{12}$$

we can compute the sample mean

$$\hat{\underline{x}}^{(\gamma+\Delta)} = \sum_{i=1}^{M} w_i^{(\gamma+\Delta)} \cdot \underline{x}_i^{(\gamma)}$$

as well as the sample covariance

$$\mathbf{C}^{(\gamma+\Delta)} = \sum_{i=1}^{M} w_i^{(\gamma+\Delta)} \cdot (\underline{x}_i^{(\gamma)} - \hat{\underline{x}}^{(\gamma+\Delta)})(\underline{x}_i^{(\gamma)} - \hat{\underline{x}}^{(\gamma+\Delta)})^{\mathrm{T}}$$

of the next intermediate posterior $f^e(\underline{x}, \gamma + \Delta)$. Finally, we approximate the next intermediate posterior with the Gaussian

$$f^e(\underline{x}, \gamma + \Delta) :\approx \mathcal{N}(\underline{x}; \hat{\underline{x}}^{(\gamma+\Delta)}, \mathbf{C}^{(\gamma+\Delta)}) \enspace . \tag{13}$$

By starting the recursion with the prior Gaussian state estimate $f^p(\underline{x})$, i.e., setting $f^e(\underline{x}, 0) = f^p(\underline{x})$, we can recursively compute several intermediate Gaussians (13) until the desired Gaussian approximation of the posterior $f^e(\underline{x})$ is reached.

On the one hand, this approach introduces further density approximations as each, in general, non-Gaussian intermediate state estimate $f^e(\underline{x}, \gamma)$ gets approximated as such. That is, the final posterior state estimate is the result of consecutive Gaussian approximations. Hence, these approximation errors will accumulate to the final posterior state estimate. On the other hand, when using an adequate Gaussian sampling technique and selecting proper step sizes, this approach solves the problem of sample degeneration in an elegant and efficient way.

B. Gaussian Sampling

Performing a recursion step requires sampling of a multivariate normal distribution. Besides simple random sampling, there exist several ways to compute deterministic Gaussian Dirac mixture approximations, such as the sampling methods used by the UKF, the GF, or the CKF. However, we rely on a deterministic sampling approach based on the so-called Localized Cumulative Distribution (LCD) proposed in [24], [25], which is also successfully used in the S^{2}KF. It allows computing a Dirac mixture approximation of a normal distribution with an arbitrary number of samples placed in the entire state space. This is achieved by turning the approximation problem into an optimization problem. Fig. 2 shows such a Dirac mixture approximation with 15 samples using the LCD approach. We use the LCD approach to compute standard normal approximations offline and only transform them online, using the Mahalanobis transformation [26], to any normal distribution. By doing so, no significant overhead is caused by the Gaussian sampling (11) during filter execution.

C. Step Size Control

After solving the problem of sample degeneration by using deterministic Gaussian resampling, the remaining issue is to determine the number of recursion steps and their respective step sizes Δ for one measurement update. On the one hand, the larger the step sizes, the smaller the total number of intermediate Gaussian approximations is, and thus, less approximation errors accumulate until the final posterior Gaussian is reached. On the other hand, a larger step size will result in a larger error caused by one intermediate Gaussian approximation, which in turn negatively affects the overall approximation error of the measurement update. Consequently, we have to make a tradeoff such that each recursion step is as large as possible but as small as necessary to keep the overall approximation error of a measurement update small.

In theory, this boils down to finding an individual and optimal sequence of recursion steps for each measurement update that minimizes the deviation between true posterior state estimate (3) and the estimate obtained by the recursion introduced in Sec. IV-A. Unfortunately, such approach is intractable, as the true posterior state estimate is not at hand (this is what we actually pursue). As a consequence, we have to rely on a suboptimal approach. One solution would be to rely on many small recursion steps. However, this may lead to superfluous intermediate Gaussian approximations for less informative measurements, and as a result, to an unnecessarily large accumulated error and high computation time. Another option would be to use few large steps with the result of a too slight emphasis for measurements containing much information, which can lead to a very diminished estimation quality. Hence, none of these extreme cases should be used.

In order to put more emphasis on measurements containing much information and process less informative measurements quickly, a variable amount of recursion steps with individual step sizes for each recursion step is required. As with the PGF 42, we achieve this by using an automatic step size determination for each recursion step based on the most down-weighted sample weight

$$w_s^{(\gamma+\Delta)} := \min\left\{w_i^{(\gamma+\Delta)} \mid \forall\, 1 \le i \le M \wedge w_i^{(\gamma+\Delta)} > 0\right\}$$

and most up-weighted sample weight

$$w_l^{(\gamma+\Delta)} := \max\left\{w_i^{(\gamma+\Delta)} \mid \forall\, 1 \le i \le M \wedge w_i^{(\gamma+\Delta)} > 0\right\} \ .$$

By *forcing* the ratio between them to be

$$R \overset{!}{=} \frac{w_s^{(\gamma+\Delta)}}{w_l^{(\gamma+\Delta)}} = \frac{[f^L(\underline{x}_s^{(\gamma)})]^\Delta}{[f^L(\underline{x}_l^{(\gamma)})]^\Delta} \ ,$$

with ratio $R \in (0, 1)$, the step size Δ is determined by

$$\Delta = \frac{\log(R)}{\log(f^L(\underline{x}_s^{(\gamma)})) - \log(f^L(\underline{x}_l^{(\gamma)}))} \ . \tag{14}$$

The smaller the forced ratio R, the large will be the step size Δ. A value of $R = 1$ would imply that all samples are equally weighted, and hence, the recursion step would not continue the progression. The other extreme, a value of $R = 0$ which implies $w_s^{(\gamma+\Delta)} = 0$, would prohibit any useful value for Δ, and thus, we have to ensure a positive $w_s^{(\gamma+\Delta)}$. However, unlike the PGF 42, we have to deal with a general likelihood, and hence, zero sample weights might occur as well. We solve this issue by excluding all zero sample weights from the set of possible weight extremes.

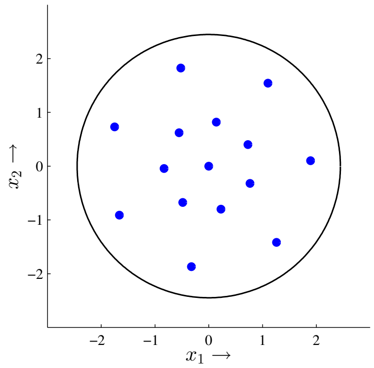


Figure 2: Sampling of a 2D standard normal distribution using the LCD approach with 15 samples (blue dots) [15].

Solving Eq. (14) only requires the smallest and largest log-likelihood value. For reasons of numerical stability, working with log-likelihoods is preferred anyway, and thus, does not result in any problem here.

After determining Δ, the actual progressive likelihood evaluations can be obtained according to

$$\left[f^L(\underline{x}_i^{(\gamma)})\right]^\Delta = \exp(\log(f^L(\underline{x}_i^{(\gamma)})) \cdot \Delta) \quad \forall\, 1 \le i \le M$$

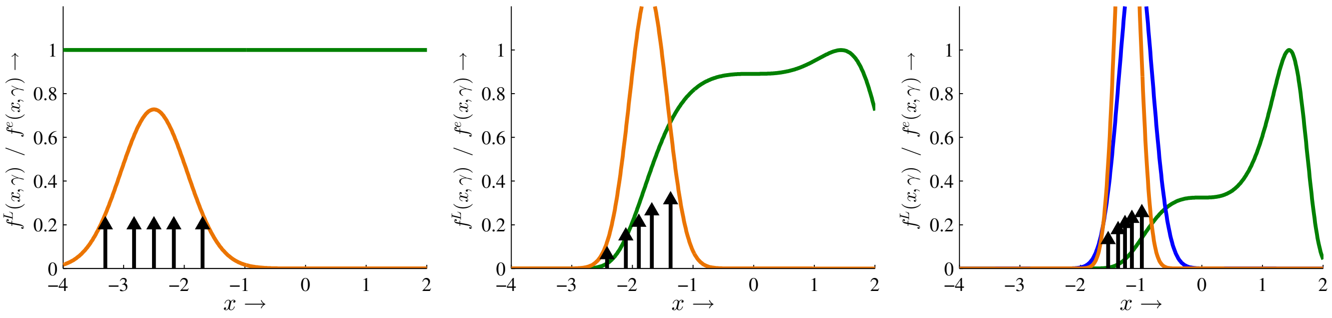
in order to compute the required normalized sample weights using (12).

The PGF 42 treated the ratio R as a freely configurable filter parameter. However, it was pointed out that increasing the used number of samples while leaving the forced ratio R constant leads to worse estimation results. This is an unintuitive behavior, as one should expect that using more samples would lead to better estimation results. This phenomenon is caused by the fact that more samples (with constant R) let the recursion require more steps, and thus, induces more intermediate approximation errors. This fact implies that a ratio has to be selected in such a way that it works well with the employed number of samples. Evaluations showed that a heuristic, where the forced sample weight ratio is set to

$$R := \frac{1}{M} \ ,$$

works very well. A positive side effect of setting R in this way is that the number of samples M used per recursion step is the only remaining filter parameter, which simplifies filter usage.

Besides the approach forcing a sample weight ratio, the PGF 42 additionally relies on a so-called forward-backward mechanism to determine the step size Δ. Here, the idea is that the deviation between two successive intermediate Gaussian approximations has to be small [16]. Unfortunately, this may result in many recursion steps with small step sizes, and thus, exacerbates the problem that many small errors can lead to a large approximation error in total. Consequently, we dropped this approach here and only rely on a step size determination based on the forced sample weight ratio R.



(a) Recursion start ($\gamma = 0$).

(b) After 4 recursion steps ($\gamma = 0.1$).

(c) After 9 recursion steps ($\gamma = 1$).

Figure 3: Example measurement update conducted by the new filter. Progressive likelihoods (green), true posterior Gaussian state estimate (blue), intermediate Gaussian approximations (orange), and reweighted Gaussian samples (black arrows).

D. The New Progressive Gaussian Filter

Algorithm 1 summarizes the measurement update procedure of the new Progressive Gaussian Filter (PGF) that works with explicit likelihood functions. It encompasses the introduced recursion based on progressive likelihoods with intermediate Gaussian approximations, Gaussian resampling using the LCD approach, and the proposed step size control with automatic selection of the forced weight ratio R. As a result, the new PGF only has one easy to use tuning parameter, namely the number of samples M used per recursion step.

Finally, we reconsider the measurement update example from Sec. II (see Fig. 1). The estimation result produced by the new PGF, when using only 5 samples per recursion step, is depicted in Fig. 3. One should note that the final posterior state estimate (Fig. 3c) is much closer to the true posterior state estimate than the result obtained by the naïve sample-based approach, while using only $9 \times 5 = 45$ samples.

Algorithm 1 Progressive Gaussian Filter (PGF)

```
Set $\hat{\underline{x}} = \hat{\underline{x}}_k^p, \mathbf{C} = \mathbf{C}_k^p, \gamma = 0$
while $\gamma < 1$ do
    Compute samples $\{\underline{x}_i\}_{i=1,\ldots,M}$ using LCD($\hat{\underline{x}}, \mathbf{C}$)
    $\underline{l}_i = \log(f_k^L(\underline{x}_i)) \quad \forall\, 1 \leq i \leq M$
    $\underline{l}_{\text{min}} = \min\,\{\underline{l}_i \mid \forall\, 1 \leq i \leq M \;\wedge\; \underline{l}_i > -\infty\}$
    $\underline{l}_{\text{max}} = \max\,\{\underline{l}_i \mid \forall\, 1 \leq i \leq M \;\wedge\; \underline{l}_i > -\infty\}$
    $\Delta = -\log(M) \;/\; (\underline{l}_{\text{min}} - \underline{l}_{\text{max}})$
    if $\gamma + \Delta > 1$ then
        $\Delta = 1 - \gamma$
    end if
    $f_k^L(\underline{x}_i)^\Delta = \exp(\underline{l}_i \cdot \Delta) \quad \forall\, 1 \leq i \leq M$
    $w_i = f_k^L(\underline{x}_i)^\Delta \;/\; \sum_{j=1}^M f_k^L(\underline{x}_j)^\Delta \quad \forall\, 1 \leq i \leq M$
    $\hat{\underline{x}} = \sum_{i=1}^M w_i \cdot \underline{x}_i$
    $\mathbf{C} = \sum_{i=1}^M w_i \cdot (\underline{x}_i - \hat{\underline{x}}) \cdot (\underline{x}_i - \hat{\underline{x}})^\mathrm{T}$
    $\gamma = \gamma + \Delta$
end while
Set $\hat{\underline{x}}_k^e = \hat{\underline{x}}, \mathbf{C}_k^e = \mathbf{C}$
```

V. Evaluation

We evaluate the new PGF against the PGF 42 and two Particle Filters using extended object tracking. For that purpose, we consider estimating length and position of a stick target over time. The 2D system state is modeled as $\underline{\boldsymbol{x}}_k = [\boldsymbol{l}_k, \boldsymbol{p}_k]^\mathrm{T}$, where $\boldsymbol{l}_k$ denotes the stick length and $\boldsymbol{p}_k$ the stick position (see Fig. 4). We model the relation between state and noisy measurements originating from the stick as a spatial distribution according to

$$\boldsymbol{y}_k = h(\underline{\boldsymbol{x}}_k, \boldsymbol{v}, \boldsymbol{r}) = \boldsymbol{l}_k \cdot \boldsymbol{v} + \boldsymbol{p}_k + \boldsymbol{r} \ ,$$

with state-independent additive noise $\boldsymbol{r} \sim \mathcal{N}(0, 0.15^2)$ and multiplicative noise $\boldsymbol{v} \sim \mathcal{U}(-1, 1)$ [27], [28].

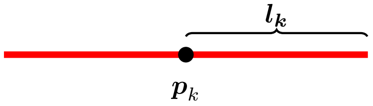


Figure 4: Stick target with extent $\boldsymbol{l_k}$ and position $\boldsymbol{p}_k$.

It is important to note that Nonlinear Kalman Filters are not capable of handling this measurement model without modifications [27], and hence, nonlinear estimators, such as the new PGF, are required for estimating the state $\underline{\boldsymbol{x}}_k$ correctly. The corresponding likelihood function is given as

$$f_k^L(\underline{x}_k) = \int f^r(\tilde{y}_k - (l_k \cdot v + p_k)) \cdot f^v(v)\,\mathrm{d}v \ ,$$

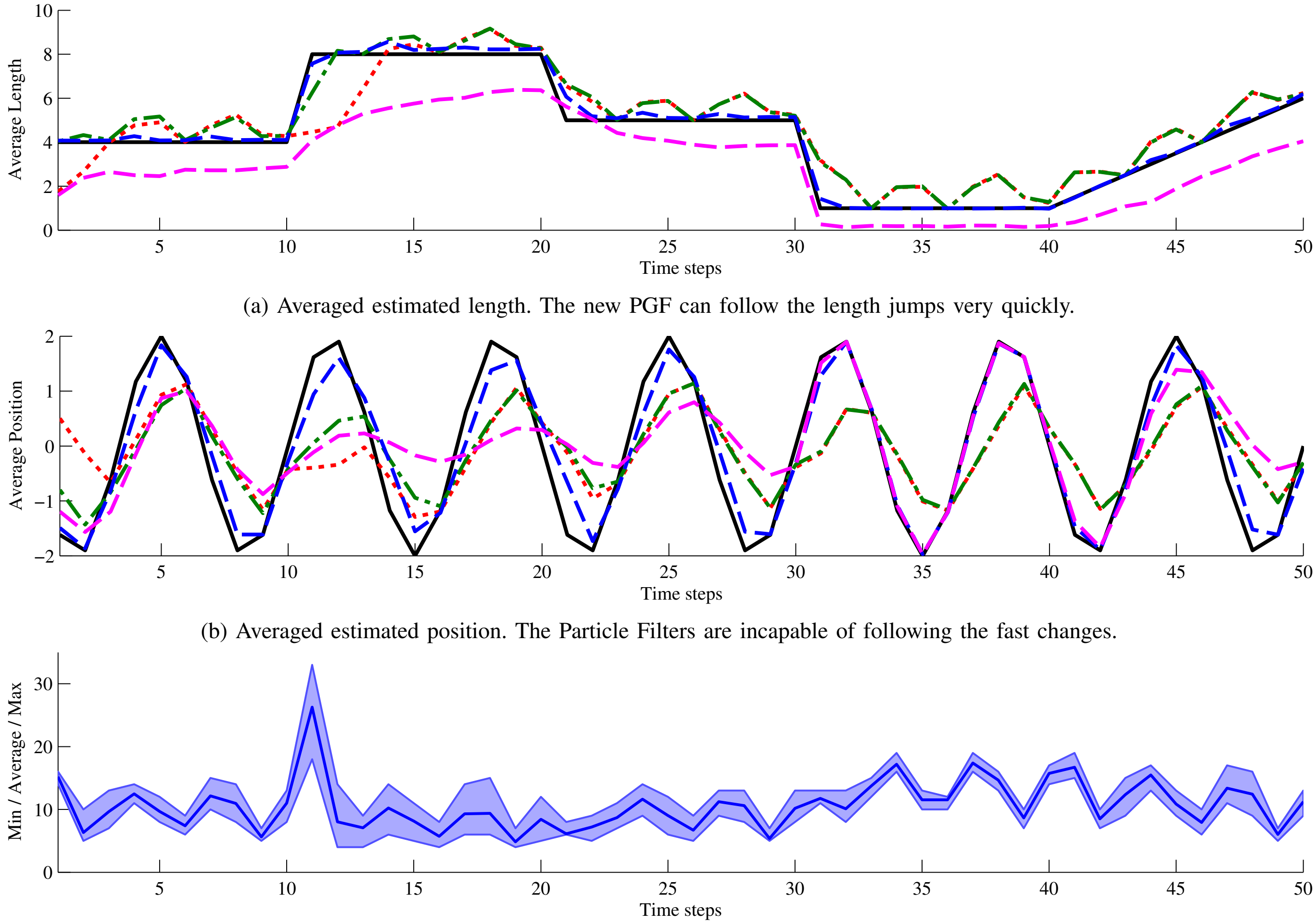
where $f^r(\cdot)$ and $f^v(\cdot)$ denote the probability density functions of $\boldsymbol{r}$ and $\boldsymbol{v}$, respectively. In order to allow system changes over time, we employ a simple identity system model, i.e., a random walk

$$\underline{\boldsymbol{x}}_k = \underline{a}(\underline{\boldsymbol{x}}_{k-1}, \underline{\boldsymbol{w}}) = \underline{\boldsymbol{x}}_{k-1} + \underline{\boldsymbol{w}} \ ,$$

with state-independent white noise $\underline{\boldsymbol{w}} \sim \mathcal{N}(\underline{0}, \mathrm{diag}(0.5, 0.1))$.

We compare the following estimators:

- The new PGF using 10 samples per progression step.
- The PGF 42 using 250 samples per progression step, a ratio $R = 0.1$, and a maximum allowed deviation of 0.5.
- The Gaussian Particle Filter (GPF) [9] using 500 particles with prior state estimate as proposal density.



(a) Averaged estimated length. The new PGF can follow the length jumps very quickly.

(b) Averaged estimated position. The Particle Filters are incapable of following the fast changes.

(c) Minimum, maximum, and average number of recursion steps required by the new PGF.

Figure 5: Stick tracking evaluation results using 50 measurements per update. Stick target ground truth (black solid lines), the new PGF (blue dashed lines), the PGF 42 (magenta dashed lines), SIR-PF (green dash-dotted lines), and GPF (red dotted lines).

- The Sequential Importance Resampling Particle Filter (SIR-PF) [2] also using 500 particles.

All filters are initialized with mean $\hat{\underline{x}}_0^e = [1, 1]^{\mathrm{T}}$ and covariance $\mathbf{C}_0^e = \mathbf{I}_2$. We simulate the true target state over 50 time steps, where the target moves back and forth in a sinusoidal way and changes its length several times (see black lines in Fig. 5a and 5b). On each time step, we process 50 noisy measurements *at once* resulting, together with the relatively small additive noise $\boldsymbol{r}$, in a very narrow likelihood. As a consequence, the need for a progressive filtering method becomes more evident as sample degeneration is even more present in such a case.

The evaluation results of 100 Monte Carlo runs are depicted in Fig. 5. Regarding the target length, the new PGF estimates the length over all time steps very well. It can follow the straight jumps at time steps 11, 21, and 31 quickly. In contrast, the PGF 42 is not capable of estimating the target length as its estimate is always too small. The GPF needs much time to converge to the correct lengths, especially at the beginning and at time step 11. The reason is sample degeneration. In these cases, there is only a small intersection of the likelihood support and a significant amount of the probability mass of the prior state estimate, and hence, not enough samples remain to compute a proper posterior state covariance matrix. Consequently, the state estimate has to be left unchanged. The SIR-PF copes better with this problem, as one non-zero reweighted sample is enough for a valid posterior state estimate. Nevertheless, both Particle Filters have problems estimating the length correctly and drift away from the true length regularly.

The stick position estimates are very similar to the length estimates. After the first time step, the new PGF is very close to true target position, whereas both Particle Filters, and especially the GPF, need some more time steps for convergence. In general, the new PGF can follow the quick target position changes very well. The Particle Filters behave much differently. They are incapable of following the quick changes, in particular on time steps 10-13 and 30-33. Also the PGF 42 has problems estimating the position apart from time steps 30 to 43.

When looking at the number of progression steps required by the new PGF, one should note that the number highly depends on the changes of true target state, that is, how much new information is given by the measurements received from the target. The most progression steps are required at time step 11. Here, the target length changes rapidly which causes a likelihood support far away from the prior state mean (very informative measurements). Averaged over all 50 time steps, the new PGF only needs 11 progression steps. As 10 samples are used per step, only $11 \times 10 = 110$ likelihood evaluations are required on average per measurement update. In contrast, both Particle Filters constantly need 500 evaluations and their estimation results are much worse than that of the new PGF. The PGF 42 requires on average over 300 progression steps, which causes very long runtimes and emphasizes the advantages of the new PGF.

VI. Conclusions

In this paper, we presented a new sample-based Gaussian filter as a variant of the PGF 42. Instead of linearizing the measurement model, the new filter relies on a progressive filtering technique using explicit likelihood functions. After describing the problem of nonlinear state estimation and sample degeneration as main motivation for the new filter, we recapitulated the idea of progressive filtering. Based on this, we took up the approach of the PGF 42 to formulate a recursive progression with intermediate Gaussian approximations that works with explicit likelihood functions. Together with an automatic step size determination, we define the measurement update algorithm of the new PGF. Compared to the PGF 42, the number of utilized samples is the only filter parameter, which allows quick and intuitive adjustments in order to achieve high-quality estimation results or short runtimes.

The evaluation using extended object tracking emphasizes the advantages of the new PGF over the PGF 42 in case of non-additive noise and when processing multiple measurements per update. Additionally, the new PGF also outperformed established Particle Filters. As the new PGF only requires a log-likelihood, it can directly replace PFs whenever unimodal state estimates are sufficient. Moreover, there is no need for demanding optimizations such as finding good proposal densities.

References

[1] 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.

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

[3] Marcus Baum and Uwe D. Hanebeck, "Shape tracking of extended objects and group targets with star-convex RHMs," in *Proceedings of the 14th International Conference on Information Fusion (Fusion 2011)*, Chicago, USA, Jul. 2011, pp. 1–8.

[4] Johann Wolfgang Koch, "Bayesian approach to extended object and cluster tracking using random matrices," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 44, no. 3, pp. 1042–1059, Jul. 2008.

[5] Antonio Zea, Florian Faion, Marcus Baum, and Uwe D. Hanebeck, "Level-set random hypersurface models for tracking non-convex extended objects," in *Proceedings of the 16th International Conference on Information Fusion (Fusion 2013)*, Istanbul, Turkey, Jul. 2013.

[6] Sebastian Thrun, Wolfram Burgard, and Dieter Fox, *Probabilistic Robotics*. Cambridge, London: MIT Press, 2005.

[7] Rudolf E. Kalman, "A new approach to linear filtering and prediction problems," in *Transaction of the ASME - Journal of Basic Engineering*, Mar. 1960, pp. 35–45.

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

[9] Jayesh H. Kotecha and Petar M. Djuric, "Gaussian particle filtering," *IEEE Transactions on Signal Processing*, vol. 51, no. 10, pp. 2592–2601, Oct. 2003.

[10] Arnaud Doucet and Adam M. Johansen, "A tutorial on particle filtering and smoothing: Fifteen years later," in *Oxford Handbook of Nonlinear Filtering*, 2011, pp. 656 – 704.

[11] Daniel L. Alspach and Harold W. Sorenson, "Nonlinear bayesian estimation using gaussian sum approximations," *IEEE Transactions on Automatic Control*, vol. 17, no. 4, pp. 439 – 448, Aug. 1972.

[12] Simon J. Julier and Jeffrey K. Uhlmann, "Unscented filtering and nonlinear estimation," in *Proceedings of the IEEE*, vol. 92, Mar. 2004, pp. 401–422.

[13] Ienkaran Arasaratnam and Simon Haykin, "Cubature kalman filters," *IEEE Transactions on Automatic Control*, vol. 54, no. 6, pp. 1254 – 1269, Jun. 2009.

[14] Marco F. Huber and Uwe D. Hanebeck, "Gaussian filter based on deterministic sampling for high quality nonlinear estimation," in *Proceedings of the 17th IFAC World Congress (IFAC 2008)*, vol. 17, Seoul, Republic of Korea, Jul. 2008.

[15] 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, Jul. 2013, pp. 2089 – 2096.

[16] 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, Jul. 2013.

[17] 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, Jul. 2011, pp. 1 – 8.

[18] Fred Daum and Jim Huang, "Particle flow for nonlinear filters, bayesian decisions and transport," in *Proceedings of the 16th International Conference on Information Fusion (Fusion 2013)*, Istanbul, Turkey, Jul. 2013, pp. 1072 – 1079.

[19] Renato Zanetti, "Recursive update filtering for nonlinear estimation," *IEEE Transactions on Automatic Control*, vol. 57, no. 6, pp. 1481–1490, Jun. 2012.

[20] Gerhard Kurz, Igor Gilitschenski, and Uwe D. Hanebeck, "Nonlinear measurement update for estimation of angular systems based on circular distributions (to appear)," in *Proceedings of the 2014 American Control Conference (ACC 2014)*, Portland, USA, Jun. 2014.

[21] Fred Daum, "Nonlinear filters: Beyond the kalman filter," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 20, no. 8, pp. 57 – 69, Aug. 2005.

[22] Simon Maskell and Simon Julier, "Optimised proposals for improved propagation of multi-modal distributions in particle filters," in *Proceedings of the 16th International Conference on Information Fusion (Fusion 2013)*, Istanbul, Turkey, Jul. 2013, pp. 296 – 303.

[23] Uwe D. Hanebeck and Jannik Steinbring, "Progressive gaussian filtering based on dirac mixture approximations," in *Proceedings of the 15th International Conference on Information Fusion (Fusion 2012)*, Singapore, Jul. 2012, pp. 1697–1704.

[24] Uwe D. Hanebeck, Marco F. Huber, and Vesa Klumpp, "Dirac mixture approximation of multivariate gaussian densities," in *Proceedings of the 2009 IEEE Conference on Decision and Control (CDC 2009)*, Shanghai, China, Dec. 2009.

[25] Igor Gilitschenski and Uwe D. Hanebeck, "Efficient deterministic dirac mixture approximation of gaussian distributions," in *Proceedings of the 2013 American Control Conference (ACC 2013)*, Washington D. C., USA, Jun. 2013.

[26] Wolfgang Härdle and Léopold Simar, *Applied Multivariate Statistical Analysis*, 2nd ed. Berlin Heidelberg: Springer, 2008.

[27] Marcus Baum, Florian Faion, and Uwe D. Hanebeck, "Modeling the target extent with multiplicative noise," in *Proceedings of the 15th International Conference on Information Fusion (Fusion 2012)*, Singapore, Jul. 2012, pp. 2406–2412.

[28] K. Gilholm and D. Salmond, "Spatial distribution model for tracking extended objects," *IEE Proceedings Radar, Sonar and Navigation*, vol. 152, no. 5, pp. 364 – 371, Oct. 2005.