Progressive Bayesian Filtering with Coupled Gaussian and Dirac Mixtures

Daniel Frisch and **Uwe D. Hanebeck**
Intelligent Sensor-Actuator-Systems Laboratory (ISAS)
Institute for Anthropomatics and Robotics
Karlsruhe Institute of Technology (KIT), Germany
daniel.frisch@kit.edu, uwe.hanebeck@kit.edu

***Abstract*—Nonlinear filtering is the most important aspect in state estimation with real-world systems. While the Kalman filter provides a simple though optimal estimate for linear systems, feasible filters for general systems are still subject of intensive research. The previously proposed Progressive Gaussian Filter PGF42 marked a new milestone, as it was able to efficiently compute an optimal Gaussian approximation of the posterior density in nonlinear systems [1]. However, for highly nonlinear systems where true posteriors are "banana-shaped" (e.g., cubic sensor problem) or multimodal (e.g., extended object tracking), even an optimal Gaussian approximation is an inadequate representation. Therefore, we generalize the established framework around the PGF42 from Gaussian to Gaussian mixture densities that are better able to approximate arbitrary density functions. Our filter simultaneously holds approximate Gaussian mixture and Dirac mixture representations of the same density, what we call coupled discrete and continuous densities (CoDiCo). For conversion between discrete and continuous representation, we employ deterministic sampling and the expectation–maximization (EM) algorithm, which we extend to deal with weighted particles.**

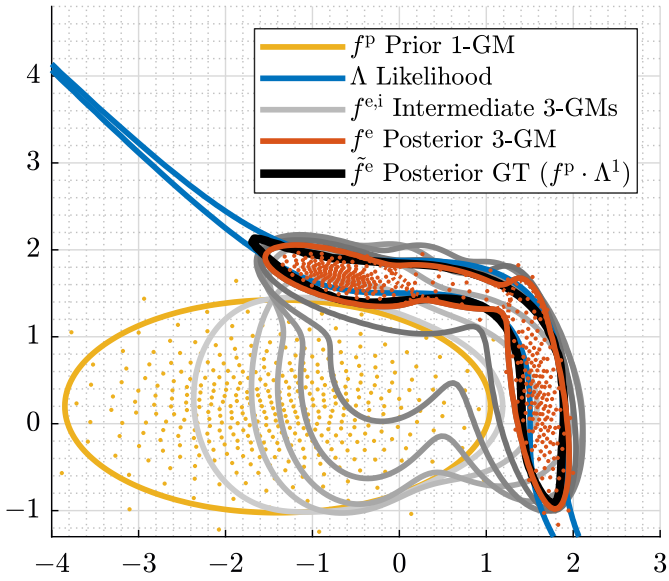


Fig. 1: Example of a "banana-shaped" true posterior (black) resulting from Bayesian fusion of a Gaussian prior (yellow) and the likelihood function from a cubic sensor problem (blue). Grey lines show intermediate results during progression. Red line is the posterior approximation obtained with the proposed nonlinear filtering scheme. Shown are contour lines enclosing 95 % of the probability mass.

I. Introduction

Context: We consider the filter step in general state estimation. It fuses given prior knowledge with additional information provided by measurements. In concrete terms, the prior state density as obtained from the last prediction step is fused with the likelihood from the measurement equation. Ideally, this is done according to Bayes' rule, i.e., by multiplying the prior state density with the likelihood function. Then the resulting normalized density is the true posterior state density.

Considered Problem: However, for most nonlinear systems, recursive application of said ideal filter step and the ideal prediction step is not feasible. The resulting true state density representations may require more and more memory with every step or the necessary computing power may increase as the algorithm proceeds. In most cases, the true state densities do not even exist in closed form. To avoid this problem, we aim at *approximating* the true posterior density in such a way that computational demands remain within certain bounds over time. By doing so, the approximated density should still contain the essential information from the true density, of course.

State-of-the-art: Due to the importance and difficulty of nonlinear state estimation, the available literature on this topic is vast [2]. Only for linear systems, the first two moments can be exactly propagated, which corresponds to the well-known Kalman filter [3]. Real systems are always nonlinear, though. The Extended Kalman Filter (EKF) therefore linearizes the models around the current estimated state using Jacobian matrices [4]. This works well for nonlinearities that are "mild" within the current range of uncertainty. Instead of tangents, the Unscented Kalman Filter (UKF) uses secants for linearization, where the intersection points are selected according to the current state covariance [5], [6]. To improve accuracy, such a stochastic linearization can be performed also with larger numbers of reference points carefully selected based on the Localized Cumulative Distribution (LCD) [7], [8], [9], resulting in the Smart Sampling Kalman Filter (S^2KF) [10], [11]. However, the best result that any of these methods can yield is merely an estimate based on a *linearized* measurement model,

i.e., a joint Gaussian approximation of the joint distribution of state and measurement.

For truly nonlinear filtering, early approaches used Gaussian mixture (GM) approximations of the prior [12] that were then processed in separate EKF filter steps [13], [14] with individual local linearizations. Conversely, a GM approximation of the likelihood can be used to directly perform a nonlinear filter step on GM priors [15]. Of course, the number of GM components would grow exponentially when repeating this procedure, which makes a reduction [16] indispensable.

An alternating procedure involving Gaussian mixtures and particles has been proposed with the Gaussian Sum Particle Filter (GSPF) [17]. In the default setup, it keeps the associations between particles and Gaussian mixture components constant. An optional use of the expectation–maximization (EM) algorithm is briefly discussed as a means to prevent the mixture from collapsing.

Most of the advanced state-of-the-art nonlinear filtering methods are implemented with some kind of iteration and/or recursion inside each update step. Hence, they can be seen as *numerical solvers* to obtain the parameters of a "good approximation" of the true posterior state density.

First of all, we will discuss filter steps based on a homotopy continuation. That is, the measurement information is introduced gradually, in infinitesimally small sub-update steps. In concrete terms, an ordinary differential equation (ODE) is formulated that smoothly transforms the parameters of the prior (such as mean and variance, or sample locations) into parameters that represent the posterior. These ODEs require some assumptions to ensure a unique solution. For example, they can be based on incompressible particle flow [18], [19], or on an additional Gaussian assumption that allows to compute an exact particle flow [20], [21], [22]. This concept has also been transferred to the world of particle filters, where special particle flow filters provide reliable proposal densities for the particle filter [23], [24].

The first publication introducing continuously executed filter steps via ODEs [25] instead employed distance measures such as the L^2-norm between true density and desired GM approximation to define and track the optimal solution. This can similarly be done also for Dirac mixture (DM) approximations [26]. The mathematical formulations of the ODEs though include some integral operations that are available in closed form only for special cases like separable nonlinear measurement models [27]. When no closed form expression is available, a sampled version of the current intermediate density can serve as proposal density, enabling efficient numerical integration in this sub-step [28].

Instead of formulating the problem as an ODE and employing an off-the-shelf solver, the continuous progressive update procedures can be stated in as recursive algorithm with discrete progression steps as well. This also allows for more flexibility in the design and can help to reduce computational cost. For example, purely sample-based algorithms of this class would progressively re-weight a prior unweighted DM with suitable "flattened" versions of the likelihood. The resulting

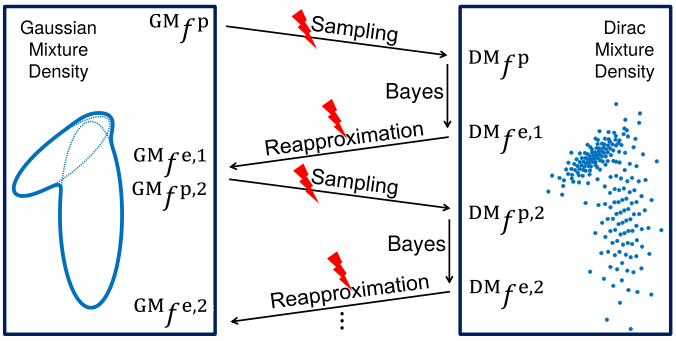


Fig. 2: Visualization of the proposed progressive filtering strategy. In each progression step, we switch back and forth between GM and DM. This produces approximation errors, indicated by red arrows. The first two progression steps are shown here, where R progression steps yield one complete filter step (e.g. in Fig. 1, $R = 7$). Note that two approximate representations of the same density are held simultaneously during the update, hence the name CoDiCo.

weighted DM must then be re-approximated by an unweighted DM before the next progression step. This can be done through minimization of the Cramér–von Mises distance of the corresponding LCDs [29], but that works well only if the reference DM has more samples than the approximating one. Therefore, an adaptive up-sampling should be performed before the re-approximation step [30]. Alternatively, an "energy" penalty term can provide proper regularization [31]. It exploits the assumption of an underlying smooth density. A similar idea has been pursued in the regularized particle filter (RPF), where identical Epanechnikov kernels are placed on each particle for a stochastic resampling step [32], [33].

Once more, a Gaussian assumption of the posterior allows for the fastest progressive filters – for arbitrary measurement equations [1] or arbitrary likelihoods [34], [35]. Note that even though the posterior is assumed Gaussian, it does not require the additional joint Gaussian assumption of the measurement model like UKF or S^{2}KF do. Instead, this filter provides a faithful estimate of the true posterior's first two moments, no more, no less. One of lots of possible application scenarios is extended object tracking [36], [37].

However, especially in extended object tracking, densities can become multimodal due to symmetries [38], [39]. The same holds for multi-target tracking [40] or estimation with so-called negative information [41].

Therefore, we propose an efficient "assumed Gaussian *mixture* density" progressive filter that can provide faithful state estimations for problems with multimodal state densities as well.

II. Key Idea

We use GM densities as a universal approximator for arbitrary density functions. To perform the Bayesian update, we draw unweighted samples from the prior GM and subsequently

re-weight these samples with the likelihood function. Afterwards, we re-approximate the now weighted samples with a new GM. These steps thus transform a prior GM into a posterior GM using an arbitrary continuous likelihood function.

In order to avoid sample degeneration, we apply said update procedure progressively with "flattened" versions of the original likelihood function. We choose the exponents (between zero and one) in such a way that the loss of information arising from the re-weighting is bounded.

For re-approximation of weighted samples with GMs, we use the EM algorithm that is known to be reliable and fast. By doing so, we can conveniently initialize the EM with the GM from the respective previous progression step.

We repeatedly switch back and forth between GMs and DMs by 1) deterministic sampling and 2) EM. After such a transformation, both densities contain approximately the same information, even though they belong to different classes of functions. Therefore, we call them coupled discrete and continuous densities (CoDiCo). We make use of their individual properties, i.e., 1) it is easy to re-weight a DM with a continuous function, and 2) it is easy to obtain unweighted samples from a Gaussian density. See Fig. 2 for an overview of the method.

III. Method Derivation

Our method provides nonlinear Bayesian filtering under quite general assumptions. Required inputs are

- I1 prior density f^{p}, given as GM or DM, and
- I2 likelihood Λ, given as a continuous function.

Based on these, we compute another GM density f^{e} that approximates the true posterior density $\tilde{f}^{\mathrm{e}}$. State space is the D^x-dimensional Euclidean, with state vector $\underline{x} \in \mathbb{R}^{D^x}$.

A. Prior Density

The prior density is given as a GM ${}^{\mathrm{GM}}f^{\mathrm{p}}(\underline{x})$ with M^{p} components

$$ {}^{\mathrm{GM}}f^{\mathrm{p}}(\underline{x}) = \sum_{m=1}^{M^{\mathrm{p}}} {}^{\mathrm{GM}}w_m^{\mathrm{p}}\, \mathcal{N}(\underline{x} - {}^{\mathrm{GM}}\underline{\mu}_m^{\mathrm{p}},\, {}^{\mathrm{GM}}\mathbf{C}_m^{\mathrm{p}}) \ , \quad (1) $$

$$ \mathcal{N}(\underline{x}-\underline{\mu},\, \mathbf{C}) = \frac{\exp\left\{-\frac{1}{2}\left(\underline{x}-\underline{\mu}\right)^{\top}\left(\mathbf{C}\right)^{-1}\left(\underline{x}-\underline{\mu}\right)\right\}}{\sqrt{(2\pi)^{D^x}\det(\mathbf{C})}} \ , \quad (2) $$

with positive weights ${}^{\mathrm{GM}}w_m^{\mathrm{p}} > 0$ that sum up to one, $\sum_{m=1}^{M^{\mathrm{p}}} {}^{\mathrm{GM}}w_m^{\mathrm{p}} = 1$, mean vectors ${}^{\mathrm{GM}}\underline{\mu}_m^{\mathrm{p}} \in \mathbb{R}^{D^x}$, and potentially non-diagonal positive definite covariance matrices ${}^{\mathrm{GM}}\mathbf{C}_m^{\mathrm{p}} \in \mathbb{R}^{D^x \times D^x}$ for the individual GM components $m \in 1, 2, \ldots, M^{\mathrm{p}}$, respectively.

In a first step, L^{p} deterministic samples are drawn from the prior, yielding a DM

$$ {}^{\mathrm{DM}}f^{\mathrm{p}}(\underline{x}) = \sum_{i=1}^{L^{\mathrm{p}}} {}^{\mathrm{DM}}w_i^{\mathrm{p}}\, \delta(\underline{x} - {}^{\mathrm{DM}}\underline{x}_i^{\mathrm{p}}) \ , \quad (3) $$

$$ \int_{\mathbb{R}^{D^x}} \delta(\underline{x}) \cdot f(\underline{x})\, \mathrm{d}\underline{x} = f(\underline{0}) \ , \quad (4) $$

again with positive sample weights ${}^{\mathrm{DM}}\underline{x}_i^{\mathrm{p}}$ that sum up to one, and sample locations ${}^{\mathrm{DM}}\underline{x}_i^{\mathrm{p}} \in \mathbb{R}^{D^x}$ for the individual samples $i \in 1, 2, \ldots, L^{\mathrm{p}}$, respectively.

The prior may alternatively be given as a DM instead of a GM from the last prediction step, then we can skip the sampling procedure and directly continue with ${}^{\mathrm{DM}}f^{\mathrm{p}}(\underline{x})$.

B. Likelihood

The likelihood $\Lambda(\underline{x}) = f(\hat{\underline{y}} \,|\, \underline{x})$ maps the information from measurement $\hat{\underline{y}} \in \mathbb{R}^{D^y}$ to the state space $\underline{x} \in \mathbb{R}^{D^x}$. It is usually derived from a generative measurement model. A general formulation for an arbitrary measurement model with arbitrary measurement noise $\underline{\boldsymbol{v}}$ is

$$ \underline{\boldsymbol{y}} = h(\underline{\boldsymbol{x}}, \underline{\boldsymbol{v}}) \ , \qquad \underline{\boldsymbol{v}} \sim f^v(\underline{v}) \ . \quad (5) $$

Hereby $\underline{\boldsymbol{y}}$ denotes the measurement's random variable, whereas $\hat{\underline{y}}$, the actual measurement, is a realization of (5). With the definition of conditional probability density functions as well as marginalization, we can derive the likelihood function Λ for this measurement model

$$ f(\hat{\underline{y}} \,|\, \underline{x}, \underline{v}) = \delta(\hat{\underline{y}} - h(\underline{x}, \underline{v})) \quad (6) $$

$$ f(\hat{\underline{y}}, \underline{v} \,|\, \underline{x}) = \delta(\hat{\underline{y}} - h(\underline{x}, \underline{v}))\, f^v(\underline{v}) \quad (7) $$

$$ \Lambda(\underline{x}) = f(\hat{\underline{y}} \,|\, x) = \int_{\mathbb{R}^{D^y}} \delta(\hat{\underline{y}} - h(\underline{x}, \underline{v}))\, f^v(\underline{v})\, \mathrm{d}\underline{v} \ . \quad (8) $$

The integral can be solved for arbitrary noise densities $f^v(\underline{v})$ when the equation $\hat{\underline{y}} = h(\underline{\boldsymbol{x}}, \underline{\boldsymbol{v}})$ can be solved for $\underline{\boldsymbol{v}}$. This includes, for example, additive arbitrary noise

$$ \hat{\underline{y}} = h(\underline{\boldsymbol{x}}, \underline{\boldsymbol{v}}) = h^x(\underline{\boldsymbol{x}}) + \underline{\boldsymbol{v}} \quad (9) $$

$$ \Rightarrow \ \Lambda(\underline{x}) = f^v(\hat{\underline{y}} - h(\underline{x})) \ , \quad (10) $$

or multiplicative arbitrary noise, here for scalar measurements ($D^y = 1$)

$$ \hat{y} = h(\underline{\boldsymbol{x}}, \boldsymbol{v}) = h^x(\underline{\boldsymbol{x}}) \cdot \boldsymbol{v} \quad (11) $$

$$ \Rightarrow \ \Lambda(\underline{x}) = \frac{1}{|h^x(\underline{x})|} \cdot f^v\!\left(\frac{\hat{y}}{h^x(\underline{x})}\right) \ . \quad (12) $$

C. Bayes' Theorem

In the end, we are interested in the posterior density $f^{\mathrm{e}}(\underline{x})$, that is, the density of $\underline{\boldsymbol{x}}$ given $\hat{\underline{y}}$ as well as all information from previous measurements $\hat{\underline{y}}_{\text{past}}$ that is included in the prior $f^{\mathrm{p}}(\underline{x})$

$$ f^{\mathrm{p}}(\underline{x}) = f(\underline{x} \,|\, \hat{\underline{y}}_{\text{past}}) \ , \quad (13) $$

$$ f^{\mathrm{e}}(\underline{x}) = f(\underline{x} \,|\, \hat{\underline{y}}, \hat{\underline{y}}_{\text{past}}) \ . \quad (14) $$

Again with the definition of conditional density functions, we obtain

$$ f^{\mathrm{e}}(\underline{x}) = f(\underline{x} \,|\, \hat{\underline{y}}, \hat{\underline{y}}_{\text{past}}) \quad (15) $$

$$ = \frac{f(\underline{x}, \hat{\underline{y}} \,|\, \hat{\underline{y}}_{\text{past}})}{f(\hat{\underline{y}} \,|\, \hat{\underline{y}}_{\text{past}})} \quad (16) $$

$$ = \frac{f(\hat{\underline{y}} \,|\, \underline{x}, \hat{\underline{y}}_{\text{past}}) \cdot f(\underline{x} \,|\, \hat{\underline{y}}_{\text{past}})}{f(\hat{\underline{y}} \,|\, \hat{\underline{y}}_{\text{past}})} \ . \quad (17) $$

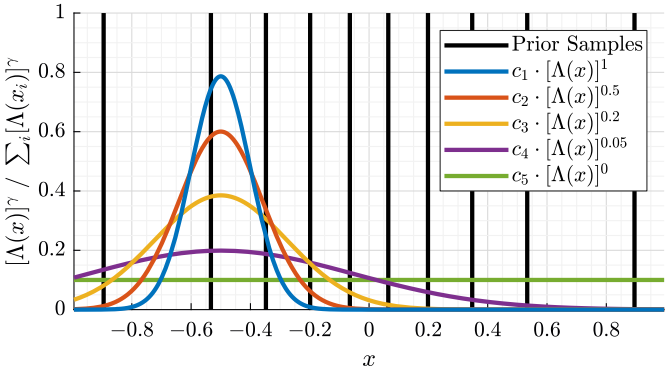


Fig. 3: Ten prior samples (black vertical lines) are to be re-weighted by the likelihood (blue line) or some "flattened" version of it (other colors). The likelihoods are normalized in such a way that intersections with the black vertical lines (sample locations) directly indicate the sample weights. Note that the smaller the likelihood exponent γ gets, the more equally-weighted the samples become. For $\gamma = 0$ (green line), samples are actually equally weighted.

According to the measurement equation (5), the density of $\underline{\boldsymbol{y}}$ is fully described when $\underline{\boldsymbol{x}}$ is given,

$$f(\hat{\underline{y}} \,|\, \underline{x},\, \hat{\underline{y}}_{\text{past}}) = f(\hat{\underline{y}} \,|\, \underline{x}) = \Lambda(\underline{x}) \quad . \tag{18}$$

Therefore, the measurement update step is given by

$$f^{\text{e}}(\underline{x}) = \frac{1}{c} \cdot \Lambda(\underline{x}) \cdot f^{\text{p}}(\underline{x}) \quad , \tag{19}$$

where c is a scalar normalization constant that ensures that $\int_{\mathbb{R}^{D^x}} f^{\text{e}}(\underline{x})\, \mathrm{d}\underline{x} = 1$.

Inserting the prior (3) into (19), we have

$$f^{\text{e}}(\underline{x}) \tag{20}$$

$$= \frac{1}{c} \cdot \Lambda(\underline{x}) \cdot \left(\sum_{i=1}^{L^{\text{p}}} {}^{\text{DM}}w_i^{\text{p}} \cdot \delta(\underline{x} - {}^{\text{DM}}\underline{x}_i^{\text{p}}) \right) \tag{21}$$

$$= \frac{1}{c} \cdot \left(\sum_{i=1}^{L^{\text{p}}} \underbrace{{}^{\text{DM}}w_i^{\text{p}} \cdot \Lambda({}^{\text{DM}}\underline{x}_i^{\text{p}})}_{\text{new sample weights}} \cdot \delta(\underline{x} - {}^{\text{DM}}\underline{x}_i^{\text{p}}) \right) \quad . \tag{22}$$

This means that the posterior sample weights ${}^{\text{DM}}w_i^{\text{e}}$ are obtained by multiplication of the prior sample weights ${}^{\text{DM}}w_i^{\text{p}}$ with the value of the likelihood function Λ at the respective sample locations ${}^{\text{DM}}\underline{x}_i^{\text{p}}$

$${}^{\text{DM}}w_i^{\text{e}} = \frac{1}{c} \cdot {}^{\text{DM}}w_i^{\text{p}} \cdot \Lambda({}^{\text{DM}}\underline{x}_i^{\text{p}}) \quad , \tag{23}$$

$$c = \sum_{i=1}^{L^{\text{p}}} {}^{\text{DM}}w_i^{\text{p}} \cdot \Lambda({}^{\text{DM}}\underline{x}_i^{\text{p}}) \quad , \tag{24}$$

where the posterior sample locations ${}^{\text{DM}}\underline{x}_i^{\text{e}}$ are just the same as the prior ones,

$${}^{\text{DM}}\underline{x}_i^{\text{e}} = {}^{\text{DM}}\underline{x}_i^{\text{p}} \quad . \tag{25}$$

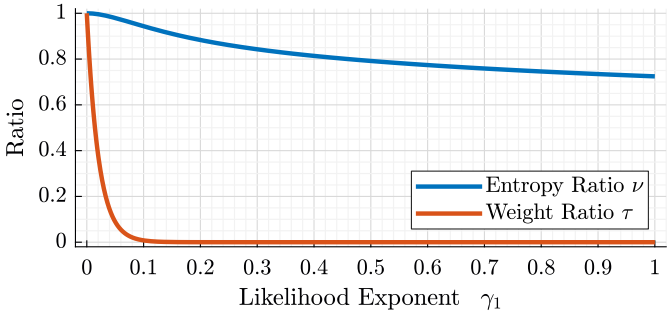


Fig. 4: Blue: Entropy, relative to entropy of equally weighted samples. Red: ratio of smallest and largest sample weight. Abscissa γ_1 is the likelihood exponent for the first progression step. $\gamma_1 = 1$ would be the full Bayesian update in a single step, i.e., without any progression steps.

This sounds simple, but there is a problem. Consider for example the 400 prior samples (yellow dots) in Fig. 1. Simply re-weighting them by the likelihood function (blue) would, also after normalization, assign a negligible weight to most of the samples, thus effectively reducing the number of samples drastically. The few remaining samples would not contain enough information (locations and weights) to give an adequate approximation to the true posterior. This effect is called sample degeneration.

D. Progressive Update

We realize that re-weighting of samples can come with information loss. Therefore, we propose to split up the measurement update step (19) into two steps

$$f^{\text{e}}(\underline{x}) = \frac{1}{c} \cdot \sqrt{\Lambda(\underline{x})} \cdot \sqrt{\Lambda(\underline{x})} \cdot f^{\text{p}}(\underline{x}) \tag{26}$$

or an arbitrary number R of so-called *progression steps*

$$f^{\text{e}}(\underline{x}) = \frac{1}{c} \cdot \prod_{r=1}^{R} [\Lambda(\underline{x})]^{\gamma_r} \cdot f^{\text{p}}(\underline{x}) \quad , \tag{27}$$

where $\gamma_r > 0$ and $\sum_{r=1}^{R} \gamma_r = 1$. That is, we do not directly use the full likelihood $\Lambda(\underline{x})$ for re-weighting, but rather $[\Lambda(\underline{x})]^{\gamma_r}$, where $0 < \gamma_r < 1$. In the example of Fig. 3, only two samples would "survive" a weighting with the full likelihood Λ^1, whereas after a re-weighting with $\Lambda^{0.05}$, about eight out of ten samples can still contribute information.

But how should γ_r be chosen in each progression step? On the one hand, it must be small enough to prevent the majority of samples from starvation. On the other hand, too many progression steps for a single measurement step should be avoided as each progression step introduces approximation errors from re-approximating a GM with a DM and vice versa, see also Fig. 2. The easiest choice would be predefined exponents like $\{\gamma_1, \gamma_2\} = \{0.5, 0.5\}$ or $\{\gamma_1, \gamma_2, \gamma_3\} = \{0.05, 0.15, 0.8\}$. An adaptive and still computationally cheap choice is to limit the ratio τ between the largest and the smallest sample weight to, say, $\tau = 10\,\%$. The corresponding γ_r can be calculated in

Algorithm 1: Fast LCD-based deterministic sampling of a GM.

Function ${}^{\mathrm{DM}}f^{\mathrm{p}} \leftarrow \texttt{samplePrior}\,({}^{\mathrm{GM}}f^{\mathrm{p}},\ \tilde{L}^{\mathrm{p}})$
Input: ${}^{\mathrm{GM}}f^{\mathrm{p}}$: GM weights, means, and covariances $\left\{{}^{\mathrm{GM}}w_m^{\mathrm{p}},\ {}^{\mathrm{GM}}\underline{\mu}_m^{\mathrm{p}},\ {}^{\mathrm{GM}}\mathbf{C}_m^{\mathrm{p}}\right\}_{m=1}^{M^{\mathrm{p}}}$,
$\tilde{L}^{\mathrm{p}}$: target number of samples
Output: ${}^{\mathrm{DM}}f^{\mathrm{p}}$: DM samples and weights $\left\{{}^{\mathrm{DM}}w_i^{\mathrm{p}},\ {}^{\mathrm{DM}}\underline{x}_i^{\mathrm{p}}\right\}_{i=1}^{L^{\mathrm{p}}}$, where $L^{\mathrm{p}} \approx \tilde{L}^{\mathrm{p}}$

$i \leftarrow 0$;
for $m \leftarrow 1$ **to** M^{p} **do**
  $N \leftarrow \texttt{round}\,(\tilde{L}^{\mathrm{p}} \cdot {}^{\mathrm{GM}}w_m^{\mathrm{p}})$;
  $\left\{{}^{\mathrm{DM}}\underline{x}_j^{\mathrm{std}}\right\}_{j=1}^{N} \leftarrow \texttt{stdNormalSamples}\,(D^x, N)$;
  ${}^{\mathrm{GM}}\mathbf{L}_m^{\mathrm{p}} \leftarrow \texttt{chol}\,({}^{\mathrm{GM}}\mathbf{C}_m^{\mathrm{p}})$;
  for $j \leftarrow 1$ **to** N **do**
    $i \leftarrow i + 1$;
    ${}^{\mathrm{DM}}\underline{x}_i^{\mathrm{p}} \leftarrow {}^{\mathrm{GM}}\mathbf{L}_m^{\mathrm{p}}\ {}^{\mathrm{DM}}\underline{x}_j^{\mathrm{std}} + {}^{\mathrm{GM}}\underline{\mu}_m^{\mathrm{p}}$;
    ${}^{\mathrm{DM}}w_i^{\mathrm{p}} \leftarrow N \big/ {}^{\mathrm{GM}}w_m^{\mathrm{p}}$;
  end
end
$\left\{{}^{\mathrm{DM}}w_i^{\mathrm{p}}\right\}_{i=1}^{L^{\mathrm{p}}} \leftarrow \texttt{normalizeL1}\,(\left\{{}^{\mathrm{DM}}w_i^{\mathrm{p}}\right\}_{i=1}^{L^{\mathrm{p}}})$;

closed form from the maximum and minimum of $\Lambda({}^{\mathrm{DM}}\underline{x}_i^{\mathrm{p}})$ [34]

$$\tau = \frac{\min_i \left[\Lambda({}^{\mathrm{DM}}\underline{x}_i^{\mathrm{p},r})\right]^{\gamma_r}}{\max_i \left[\Lambda({}^{\mathrm{DM}}\underline{x}_i^{\mathrm{p},r})\right]^{\gamma_r}} \ , \tag{28}$$

$$\Rightarrow\ \gamma_r = \ln(\tau) \Big/ \ln\left(\frac{\min_i \Lambda({}^{\mathrm{DM}}\underline{x}_i^{\mathrm{p},r})}{\max_i \Lambda({}^{\mathrm{DM}}\underline{x}_i^{\mathrm{p},r})}\right) \ . \tag{29}$$

A different way to quantify the loss of information introduced by re-weighting is based on the information entropy of Claude Shannon,

$$\nu(\gamma_r) = -\frac{1}{\log_2(L^{\mathrm{p}})} \sum_{i=1}^{L^{\mathrm{p}}} {}^{\mathrm{DM}}w_i^{\mathrm{e},r} \cdot \log_2\left({}^{\mathrm{DM}}w_i^{\mathrm{e},r}\right) \ , \tag{30}$$

$${}^{\mathrm{DM}}w_i^{\mathrm{e},r} = \frac{\left[\Lambda({}^{\mathrm{DM}}\underline{x}_i^{\mathrm{p},r})\right]^{\gamma_r}}{\sum_{i=1}^{L^{\mathrm{p}}} \left[\Lambda({}^{\mathrm{DM}}\underline{x}_i^{\mathrm{p},r})\right]^{\gamma_r}} \ . \tag{31}$$

The maximum of $\nu(\gamma)$ is normalized to one and occurs for equal weighting ($\gamma = 0$). The function decreases the more different the weights become. An optimal γ can be obtained with a simple bisection algorithm that solves for example $\nu(\gamma) = 0.97$. Other than (29), this approach includes not only two but all available samples into the consideration. See Fig. 4 for a plot of the two measures.

E. Expectation–Maximization

After re-weighting ${}^{\mathrm{DM}}f^{\mathrm{p},r}$ with Λ^{γ_r} yielding the sub-posterior ${}^{\mathrm{DM}}f^{\mathrm{e},r}$, a re-approximation with unweighted samples

Algorithm 2: Conduct one progression step for a progressive Bayesian update scheme.

Function $\left[{}^{\mathrm{GM}}f^{\mathrm{e}},\ \gamma\right] \leftarrow \texttt{progrStep}\,({}^{\mathrm{GM}}f^{\mathrm{p}},\ \Lambda,\ \gamma^{\mathrm{p}})$
Input: ${}^{\mathrm{GM}}f^{\mathrm{p}}$: GM weights, means, and covariances $\left\{{}^{\mathrm{GM}}w_m^{\mathrm{p}},\ {}^{\mathrm{GM}}\underline{\mu}_m^{\mathrm{p}},\ {}^{\mathrm{GM}}\mathbf{C}_m^{\mathrm{p}}\right\}_{m=1}^{M^{\mathrm{p}}}$ of prior density,
Λ: likelihood function,
γ^{p}: likelihood exponent accumulated in previous progression steps, or 0
Output: ${}^{\mathrm{GM}}f^{\mathrm{e}}$: GM weights, means, and covariances $\left\{{}^{\mathrm{GM}}w_m^{\mathrm{e}},\ {}^{\mathrm{GM}}\underline{\mu}_m^{\mathrm{e}},\ {}^{\mathrm{GM}}\mathbf{C}_m^{\mathrm{e}}\right\}_{m=1}^{M^{\mathrm{e}}}$ of estimated posterior density

$\left\{{}^{\mathrm{DM}}w_i^{\mathrm{p}},\ {}^{\mathrm{DM}}\underline{x}_i^{\mathrm{p}}\right\}_{i=1}^{L^{\mathrm{p}}} \leftarrow \texttt{samplePrior}\,({}^{\mathrm{GM}}f^{\mathrm{p}},\ \tilde{L}^{\mathrm{p}})$;
$\left\{{}^{\mathrm{DM}}w_i^{\mathrm{e,full}}\right\}_{i=1}^{L^{\mathrm{p}}} \leftarrow \left\{\Lambda({}^{\mathrm{DM}}\underline{x}_i^{\mathrm{p}})\right\}_{i=1}^{L^{\mathrm{p}}}$;
$\gamma \leftarrow \texttt{getLikelihoodExponent}\,({}^{\mathrm{DM}}w_i^{\mathrm{e,full}})$;
$\gamma \leftarrow \min\,(\gamma,\ 1 - \gamma^{\mathrm{p}})$;
$\left\{{}^{\mathrm{DM}}w_i^{\mathrm{e}}\right\}_{i=1}^{L^{\mathrm{p}}} \leftarrow \left\{{}^{\mathrm{DM}}w_i^{\mathrm{p}} \cdot \left[{}^{\mathrm{DM}}w_i^{\mathrm{e,full}}\right]^{\gamma}\right\}_{i=1}^{L^{\mathrm{p}}}$;
$\left\{{}^{\mathrm{DM}}w_i^{\mathrm{e}}\right\}_{i=1}^{L^{\mathrm{p}}} \leftarrow \texttt{normalizeL1}\,(\left\{{}^{\mathrm{DM}}w_i^{\mathrm{e}}\right\}_{i=1}^{L^{\mathrm{p}}})$;
if $M^{\mathrm{e}} \neq M^{\mathrm{p}}$ **then**
  ${}^{\mathrm{GM}}f^{\mathrm{e}} \leftarrow \texttt{emRand}\,(\left\{{}^{\mathrm{DM}}w_i^{\mathrm{e}},\ {}^{\mathrm{DM}}\underline{x}_i^{\mathrm{p}}\right\}_{i=1}^{L^{\mathrm{p}}})$;
else
  ${}^{\mathrm{GM}}f^{\mathrm{e}} \leftarrow \texttt{emInit}\,(\left\{{}^{\mathrm{DM}}w_i^{\mathrm{e}},\ {}^{\mathrm{DM}}\underline{x}_i^{\mathrm{p}}\right\}_{i=1}^{L^{\mathrm{p}}},\ {}^{\mathrm{GM}}f^{\mathrm{p}})$;
end

${}^{\mathrm{DM}}f^{\mathrm{p},r+1}$ must be performed before the samples can also be re-weighted with $\Lambda^{\gamma_{r+1}}$ and so on (until $\sum_r \gamma_r = 1$). Therefore, we re-approximate the weighted DM ${}^{\mathrm{DM}}f^{\mathrm{e},r}$ with a GM ${}^{\mathrm{GM}}f^{\mathrm{p},r+1}$, and after that, we can proceed from Sec. III-A again.

For re-approximating a DM with a GM we propose the EM algorithm [42], [43]. It requires some GM as initial guess and iteratively improves it by executing expectation step and maximization step alternately. In the expectation step, likelihoods of associations between DM samples and GM components are determined. The maximization step calculates new GM weights, mean vectors, and covariance matrices from the DM samples. For doing so, the association likelihoods are used as weights.

In general, even though the EM algorithm has better global convergence properties than Newton-based optimization methods, it may, depending on the initial value, converge to any local or global maximum or stationary point of the underlying likelihood [44], [45]. Therefore, in case the initial guess is randomly chosen, it should be repeated several times, and the result with maximum likelihood should be selected in the end. Fortunately, this is rarely the case. As a rule, we can fall back on the last sub-posterior and use it as initial guess.

Algorithm 3: Performs a complete progressive update/filter step involving several progression steps.

Function ${}^{\mathrm{GM}}f^{\mathrm{e}} \leftarrow \texttt{progrFilter}\,({}^{\mathrm{GM}}f^{\mathrm{p}},\ \Lambda)$
Input: ${}^{\mathrm{GM}}f^{\mathrm{p}}$: GM weights, means, and covariances $\left\{{}^{\mathrm{GM}}w_m^{\mathrm{p}},\ {}^{\mathrm{GM}}\underline{\mu}_m^{\mathrm{p}},\ {}^{\mathrm{GM}}\mathbf{C}_m^{\mathrm{p}}\right\}_{m=1}^{M^{\mathrm{p}}}$ of prior density, Λ: likelihood function
Output: ${}^{\mathrm{GM}}f^{\mathrm{e}}$: GM weights, means, and covariances $\left\{{}^{\mathrm{GM}}w_m^{\mathrm{e}},\ {}^{\mathrm{GM}}\underline{\mu}_m^{\mathrm{e}},\ {}^{\mathrm{GM}}\mathbf{C}_m^{\mathrm{e}}\right\}_{m=1}^{M^{\mathrm{e}}}$ of approximated posterior density

```
GM f^{e,0} ← GM f^p ;
γ^p ← 0 ;
r ← 0 ;
while γ^p < 1 do
    r ← r + 1 ;
    GM f^{p,r} ← GM f^{e,r-1} ;
    [GM f^{e,r}, γ_r] ← progrStep(GM f^{p,r}, Λ, γ^p) ;
    γ^p ← γ^p + γ_r ;
end
GM f^e ← GM f^{e,r} ;
```

IV. Implementation

In this section, we will detail how the proposed progressive filtering algorithm is implemented.

A. Deterministic Gaussian Samples

To make sampling from the sub-prior GM ${}^{\mathrm{GM}}f^{\mathrm{p},r}$ in Sec. III-A as efficient as possible, we draw samples from the individual Gaussian components, where the number of samples is derived from the respective component weight ${}^{\mathrm{GM}}w_m^{\mathrm{p}}$. For that purpose, we use deterministic Gaussian sampling that is obtained by minimizing the Cramér–von Mises distance of the LCDs of a single Gaussian component and a DM [8], [46], [11]. The implementation in [47] maintains a "sample cache" (folder of `.sample` files) with deterministic samples from standard normal distributions for each dimension and each number of samples, respectively. To avoid delays during filtering, this sample cache should be populated with the necessary sample files for all possible numbers of samples $\{1, 2, \ldots, L^{\mathrm{p}}\}$ in D^x dimensions beforehand.

As samples are drawn for the GM components individually with the numbers of samples determined by the mixture component weight, a rounding error is involved when translating the component weight into the number of samples. This can be optionally compensated by introducing a slightly different weighting for the individual groups of samples, see Alg. 1.

B. Progressive Update

Alg. 2 details the actions necessary for a single progression step. The complete progressive filter step (Alg. 3) finally consists of several of these progression steps.

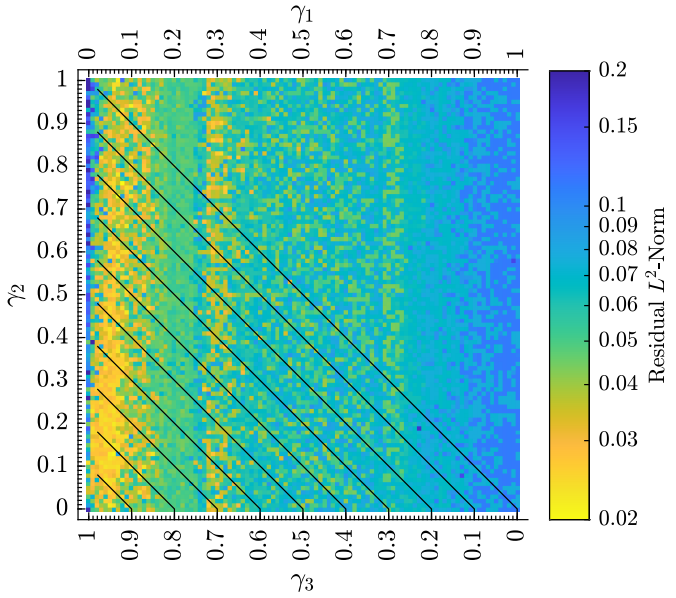


Fig. 5: Filtering with up to three progression steps. γ_1, γ_2, γ_3 are the likelihood exponents, where $\gamma_1 + \gamma_2 + \gamma_3 = 1$. Thus, each point in the area corresponds to a specific choice of likelihood exponents during a progressive filter step. Points are colored by the L^2-norm of the difference between ground truth posterior density and its approximating GM. In this example, it seems to be important that γ_1 is chosen close to 0.05 for good results, compare also Fig. 4. The rightmost column shows the results obtained without any progression, i.e., $\gamma_1 = 1$. Setup and filter parameters are detailed in Sec. V.

V. Evaluation

For evaluation, illustration, and to highlight the strengths of the proposed nonlinear filtering method, we define two difficult filtering problems. The first one is shown in Fig. 1, featuring a unimodal but "banana-shaped" true posterior. The second example, depicted in Fig. 5 and Fig. 6, will be described in full detail now.

We have a Gaussian prior ($M^{\mathrm{p}} = 1$) with

$$ {}^{\mathrm{GM}}\underline{\mu}_1^{\mathrm{p}} = \begin{bmatrix} -0.7 \\ 0.1 \end{bmatrix} , \qquad {}^{\mathrm{GM}}\mathbf{C}_1^{\mathrm{p}} = \begin{bmatrix} 3 & 0 \\ 0 & 1.2 \end{bmatrix} . \tag{32} $$

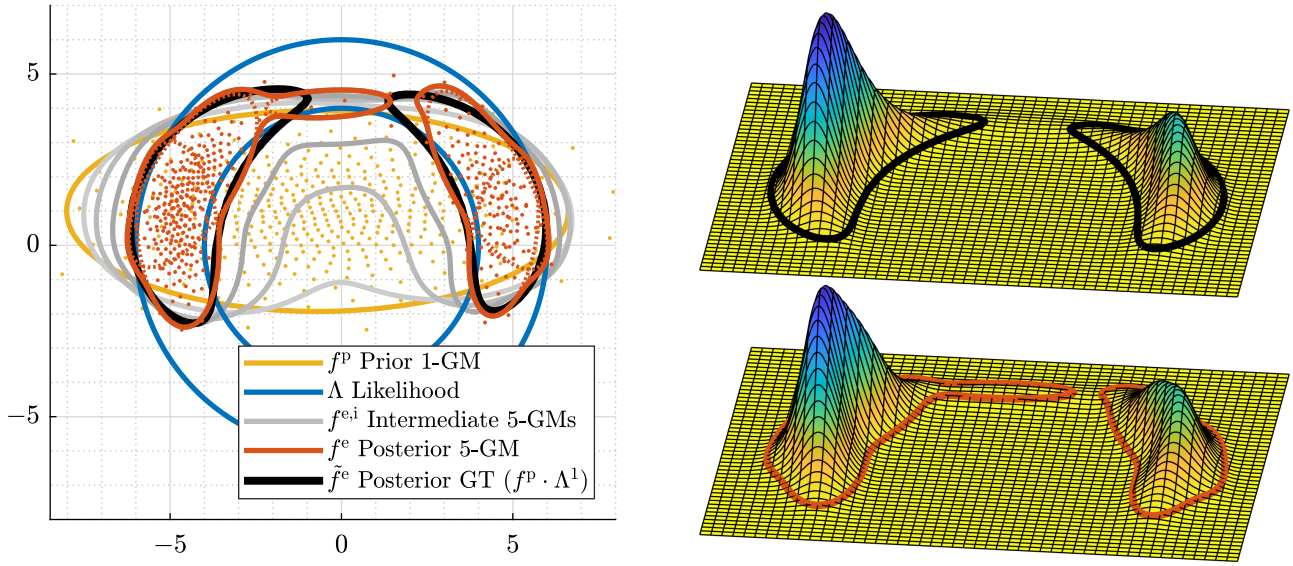
Furthermore, we define an uncertain distance measurement with additive Gaussian noise

$$ 5 = \hat{y} = \|\underline{\boldsymbol{x}}\|_2 + \boldsymbol{v} , \qquad f^v(v) = \mathcal{N}(v - 0,\ 0.5^2) , \tag{33} $$

and hence the likelihood function (10)

$$ \Lambda(\underline{x}) = \mathcal{N}(\|\underline{x}\|_2 - 5,\ 0.5^2) . \tag{34} $$

We choose $L^{\mathrm{p}} = 400$ samples for deterministic sampling from the posterior, see Alg. 1 and (3). The re-weighted posteriors are re-approximated with GMs with always $M^{\mathrm{e}} = 5$ components. For this purpose, 50 expectation and maximization steps are performed in the EM approximation at the end of each progression step. Thereby the EM routine usually converged



(a) contour lines enclosing 95 % probability mass

(b) full density functions

Fig. 6: Example with multimodal posterior true density. (a) contour lines of prior density (yellow), intermediate densities during progression (grey), and the proposed posterior approximation (red); for the latter also individual mixture components in dashed lines. Likelihood is blue, ground truth is black. (b) top: ground truth density, contour line from (a) is indicated. (b) bottom: proposed approximate posterior density, contour line from (a) is indicated. L^2-norm of the difference between both density functions is 0.025, compare also Fig. 5.

such that the component means changed by less than 0.01 in the 50$^{\text{th}}$ EM step. Because the initial prior is a single Gaussian ($1 = M^{\mathrm{p}} \neq M^{\mathrm{e}} = 5$), following Alg. 2 we use random initialization of the expecation-maximization in the first progression step ($r = 1$). For this random initialization, we draw 30 random choices for ${}^{\mathrm{GM}}f^{\mathrm{e},1}$, perform 50 EM iterations on each, and continue with the result yielding the maximum likelihood. In the next progression steps, we always have ($5 = M^{\mathrm{p}} = M^{\mathrm{e}} = 5$), so we simply use ${}^{\mathrm{GM}}f^{\mathrm{p},r}$ as initial guess for ${}^{\mathrm{GM}}f^{\mathrm{e},r}$ and perform 50 EM iterations on that.

In Fig. 5, a grid search is performed to study the effect of choosing various combinations of the γ_i. The number of progressions is limited to a maximum of three, such that the results can easily be plotted. Then in Fig. 6, the progression step sizes γ_i are automatically determined based on (30) with the constraint $\nu(\gamma_r) \geq 0.97$. This results in four progression steps: $\gamma_1 \approx 0.06$, $\gamma_2 \approx 0.11$, $\gamma_3 \approx 0.24$, and $\gamma_4 \approx 0.59$. Interestingly, the first two of these step sizes are somehow connected to Fig. 5, as they lie in the "yellow area" there.

VI. Conclusion

We propose a novel nonlinear filtering algorithm. It takes nonlinear GMs or DMs as prior densities, gradually introduces an arbitrary likelihood function through a few progression steps, and returns a nonlinear GM approximation of the posterior. The relatively high error at the right border ($\gamma_1 = 1$) in Fig. 5 demonstrates that the conventional full update performs inferior to the progressive update (yellow area).

Especially the EM part has been kept very simple yet, and we plan to include an adaption of the number of GM components that facilitates automatic merging [48], [49] or splitting [50], [51] where necessary. We also plan to integrate this filter into the nonlinear estimation toolbox [47] in order to allow for a comparison with other nonlinear filters in terms of accuracy and computational complexity.

References

[1] U. 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.

[2] Z. Chen *et al.*, "Bayesian filtering: From Kalman filters to particle filters, and beyond," *Statistics*, vol. 182, no. 1, pp. 1–69, 2003.

[3] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 03 1960.

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

[5] S. J. Julier and J. K. Uhlmann, "New extension of the Kalman filter to nonlinear systems," in *Signal processing, sensor fusion, and target recognition VI*, vol. 3068. International Society for Optics and Photonics, 1997, pp. 182–193.

[6] S. J. Julier, "The scaled unscented transformation," in *Proceedings of the 2002 American Control Conference (IEEE Cat. No.CH37301)*, vol. 6, 2002, pp. 4555–4559 vol.6.

[7] 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)*, Seoul, Republic of Korea, Aug. 2008, pp. 33–39.

[8] U. D. Hanebeck, M. F. Huber, and V. 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.

[9] I. Gilitschenski and U. D. Hanebeck, "Efficient Deterministic Dirac Mixture Approximation," in *Proceedings of the 2013 American Control Conference (ACC 2013)*, Washington D.C., USA, Jun. 2013.

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

[11] J. Steinbring, M. Pander, and U. D. Hanebeck, "The Smart Sampling Kalman Filter with Symmetric Samples," *Journal of Advances in Information Fusion*, vol. 11, no. 1, pp. 71–90, Jun. 2016.

[12] H. W. Sorenson and D. L. Alspach, "Gaussian sum approximations for nonlinear filtering," in *1970 IEEE Symposium on Adaptive Processes (9th) Decision and Control*, 1970, pp. 193–193.

[13] H. Sorenson and D. Alspach, "Recursive Bayesian Estimation Using Gaussian Sums," *Automatica*, vol. 7, no. 4, pp. 465 – 479, 1971.

[14] D. Alspach and H. Sorenson, "Nonlinear Bayesian estimation using Gaussian sum approximations," *IEEE Transactions on Automatic Control*, vol. 17, no. 4, pp. 439–448, 1972.

[15] M. F. Huber, D. Brunn, and U. D. Hanebeck, "Efficient Nonlinear Measurement Updating based on Gaussian Mixture Approximation of Conditional Densities," in *Proceedings of the 2007 American Control Conference (ACC 2007)*, New York, New York, USA, Jul. 2007, pp. 4425–4430.

[16] M. F. Huber and U. D. Hanebeck, "Progressive Gaussian Mixture Reduction," in *Proceedings of the 11th International Conference on Information Fusion (Fusion 2008)*, Cologne, Germany, Jul. 2008.

[17] J. H. Kotecha and P. M. Djuric, "Gaussian Sum Particle Filtering," *IEEE Transactions on Signal Processing*, vol. 51, no. 10, pp. 2602–2612, 2003.

[18] F. Daum and J. Huang, "Particle flow for nonlinear filters with log-homotopy," in *Signal and Data Processing of Small Targets 2008*, O. E. Drummond, Ed., vol. 6969, International Society for Optics and Photonics. SPIE, 2008, pp. 414 – 425.

[19] ——, "Nonlinear filters with particle flow induced by log-homotopy," in *Signal Processing, Sensor Fusion, and Target Recognition XVIII*, I. Kadar, Ed., vol. 7336, International Society for Optics and Photonics. SPIE, 2009, pp. 76 – 87.

[20] F. Daum, J. Huang, and A. Noushin, "Exact particle flow for nonlinear filters," in *Signal Processing, Sensor Fusion, and Target Recognition XIX*, I. Kadar, Ed., vol. 7697, International Society for Optics and Photonics. SPIE, 2010, pp. 92 – 110.

[21] S. Choi, P. Willett, F. Daum, and J. Huang, "Discussion and application of the homotopy filter," in *Signal Processing, Sensor Fusion, and Target Recognition XX*, I. Kadar, Ed., vol. 8050, International Society for Optics and Photonics. SPIE, 2011, pp. 734 – 745.

[22] S. Datta Gupta, J. Y. Yu, M. Mallick, M. Coates, and M. Morelande, "Comparison of angle-only filtering algorithms in 3D using EKF, UKF, PF, PFF, and ensemble KF," in *2015 18th International Conference on Information Fusion (Fusion)*, 2015, pp. 1649–1656.

[23] S. Pal and M. Coates, "Gaussian sum particle flow filter," in *2017 IEEE 7th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, 2017, pp. 1–5.

[24] ——, "Particle Flow Particle Filter for Gaussian Mixture Noise Models," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 4249–4253.

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

[26] U. D. Hanebeck, "FLUX: Progressive State Estimation Based on Zakai-type Distributed Ordinary Differential Equations," 2018.

[27] U. D. Hanebeck and O. Feiermann, "Progressive Bayesian Estimation for Nonlinear Discrete-Time Systems: The Filter Step for Scalar Measurements and Multidimensional States," in *Proceedings of the 2003 IEEE Conference on Decision and Control (CDC 2003)*, Maui, Hawaii, USA, Dec. 2003, pp. 5366–5371.

[28] U. D. Hanebeck and J. Steinbring, "Progressive Gaussian Filtering Based on Dirac Mixture Approximations," in *Proceedings of the 15th International Conference on Information Fusion (Fusion 2012)*, Singapore, Jul. 2012.

[29] U. D. Hanebeck, "Optimal Reduction of Multivariate Dirac Mixture Densities," *at – Automatisierungstechnik, Oldenbourg Verlag*, vol. 63, no. 4, pp. 265–278, Apr. 2015.

[30] U. D. Hanebeck and M. Pander, "Progressive Bayesian Estimation with Deterministic Particles," in *Proceedings of the 19th International Conference on Information Fusion (Fusion 2016)*, Heidelberg, Germany, Jul. 2016.

[31] P. Ruoff, P. Krauthausen, and U. D. Hanebeck, "Progressive Correction for Deterministic Dirac Mixture Approximations," in *Proceedings of the 14th International Conference on Information Fusion (Fusion 2011)*, Chicago, Illinois, USA, Jul. 2011.

[32] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174–188, 2002.

[33] C. Musso, N. Oudjane, and F. Le Gland, *Improving Regularised Particle Filters*. New York, NY: Springer New York, 2001, pp. 247–271.

[34] J. Steinbring and U. D. Hanebeck, "Progressive Gaussian Filtering Using Explicit Likelihoods," in *Proceedings of the 17th International Conference on Information Fusion (Fusion 2014)*, Salamanca, Spain, Jul. 2014.

[35] J. Steinbring, A. Zea, and U. D. Hanebeck, "Semi-Analytic Progressive Gaussian Filtering," in *Proceedings of the 2016 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2016)*, Baden-Baden, Germany, Sep. 2016.

[36] J. Steinbring, M. Baum, A. Zea, F. Faion, and U. 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, Sep. 2015.

[37] J. Steinbring and U. 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, Jul. 2015.

[38] F. Faion, A. Zea, M. Baum, and U. D. Hanebeck, "Symmetries in Bayesian Extended Object Tracking," *Journal of Advances in Information Fusion*, vol. 10, no. 1, pp. 13–30, Jun. 2015.

[39] A. Zea, F. Faion, and U. D. Hanebeck, "Tracking Elongated Extended Objects Using Splines," in *Proceedings of the 19th International Conference on Information Fusion (Fusion 2016)*, Heidelberg, Germany, Jul. 2016.

[40] M. Baum and U. D. Hanebeck, "Association-free Tracking of Two Closely Spaced Targets," in *Proceedings of the 2010 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2010)*, Salt Lake City, Utah, USA, Sep. 2010.

[41] B. Noack, F. Pfaff, M. Baum, and U. D. Hanebeck, "State Estimation Considering Negative Information with Switching Kalman and Ellipsoidal Filtering," in *Proceedings of the 19th International Conference on Information Fusion (Fusion 2016)*, Heidelberg, Germany, Jul. 2016.

[42] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum Likelihood from Incomplete Data Via the EM Algorithm," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 39, no. 1, pp. 1–22, 1977.

[43] I. D. Gebru, X. Alameda-Pineda, F. Forbes, and R. Horaud, "EM Algorithms for Weighted-Data Clustering with Application to Audio-Visual Scene Analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 12, pp. 2402–2415, 2016.

[44] C. F. J. Wu, "On the Convergence Properties of the EM Algorithm," *The Annals of Statistics*, vol. 11, no. 1, pp. 95–103, 1983.

[45] R. A. Redner and H. F. Walker, "Mixture Densities, Maximum Likelihood and the Em Algorithm," *SIAM Review*, vol. 26, no. 2, pp. 195–239, 1984.

[46] J. Steinbring and U. D. Hanebeck, "LRKF Revisited: The Smart Sampling Kalman Filter (S2KF)," *Journal of Advances in Information Fusion*, vol. 9, no. 2, pp. 106–123, Dec. 2014.

[47] J. Steinbring, "Nonlinear Estimation Toolbox." [Online]. Available: https://bitbucket.org/nonlinearestimation/toolbox

[48] M. A. T. Figueiredo and A. K. Jain, "Unsupervised learning of finite mixture models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 3, pp. 381–396, 2002.

[49] D. F. Crouse, P. Willett, K. Pattipati, and L. Svensson, "A look at Gaussian mixture reduction algorithms," in *14th International Conference on Information Fusion*, 2011, pp. 1–8.

[50] J. J. Verbeek, N. Vlassis, and B. Kröse, "Efficient Greedy Learning of Gaussian Mixture Models," *Neural Computation*, vol. 15, no. 2, pp. 469–485, 2003.

[51] P. Paalanen, J.-K. Kamarainen, J. Ilonen, and H. Kälviäinen, "Feature representation and discrimination based on Gaussian mixture model probability densities—Practices and algorithms," *Pattern Recognition*, vol. 39, no. 7, pp. 1346 – 1358, 2006.