Progressive Bayesian Particle Flows
Based on Optimal Transport Map Sequences

Uwe D. Hanebeck

Intelligent Sensor-Actuator-Systems Laboratory (ISAS)
Institute for Anthropomatics and Robotics
Karlsruhe Institute of Technology (KIT), Germany
Uwe.Hanebeck@kit.edu

Abstract—We propose a method for optimal Bayesian filtering
with deterministic particles. In order to avoid particle degene-
ration, the filter step is not performed at once. Instead, the
particles progressively flow from prior to posterior. This is
achieved by splitting the filter step into a series of sub-steps.
In each sub-step, optimal resampling is done by a map that
replaces non-equally weighted particles with equally weighted
ones. Inversions of the maps or monotonicity constraints are not
required, greatly simplifying the procedure. The parameters
of the mapping network are optimized w.r.t. a particle set distance.
This distance is differentiable, and compares non-equally and
equally weighted particles. Composition of the sequence of maps
provides a final mapping from prior to posterior particles. Radial
basis function neural networks are used as maps. It is important
that no intermediate continuous density representation is required.
The entire flow works directly with particle representations. This
avoids costly density estimation.

II. STATE OF THE ART

The state of the art in calculating transport maps or flows will
be investigated. The focus is on methods for converting the
Bayesian filter step. However, some methods are only derived
for the case of mapping between two given densities without
explicitly considering the Bayes update.

A. Continuous Density Flow Filters

We start with flows of continuous densities as they have
been developed earlier than particle flows. The first pro-
gressive Bayesian filter [1] employs a continuous density
for representing the posterior. Progressively introducing the
likelihood function leads to a homotopy continuation approach.
A Gaussian mixture flow is derived that minimizes the squared
integral deviation from the true posterior. This results in a
system of explicit ordinary first-order differential equations
solved over an artificial time from 0 to 1. The approach is
generalized in [2] for the Kullback-Leibler divergence and the
squared Hellinger distance.

B. Filters based on Transform Maps

Transformation of a random vector \( \mathbf{x} \) via a known nonlinear
map \( \mathbf{y} = g(\mathbf{x}) \) is a classic problem. The goal is to calculate
the output density \( f_y(y) \) given the input density \( f_x(x) \). For
that purpose, a different map is required that maps densities to
densities, i.e., \( f_y = G(f_x) \). It can be derived from the original
mapping \( g(\cdot) \), its roots, and its Jacobian. The derivation is
simplified for monotonic maps.

Here, we are faced with a more complex problem. Given
two densities \( f_x(x) \) and \( f_y(y) \), we want to find the map \( g(\cdot) \)
between their corresponding random vectors \( \mathbf{x} \) and \( \mathbf{y} \). When the
two densities are continuous, a mapping exists, is unique, and
monotonic [3]. Finding the map is challenging, especially under
monotonicity constraints. For univariate \( f_x(\cdot) \) and \( f_y(\cdot) \), the
map can be composed from the input cumulative distribution
function (CDF) \( F_x(\cdot) \) and the output quantile function \( F_y^{-1}(\cdot) \)
as

\[
g(x) = F_y^{-1}(F_x(x))
\]

[4, p. 102, eq. 5-20]. However, calculation of cumulative
distributions and quantile functions can be challenging with
analytic expression only available in special cases. This is
exacerbated in multivariate settings.
It gets even more challenging, when only samples of the two densities \(f_x(y)\) and \(f_y(y)\) are given. In that case, standard distance measures (such as the KL divergence) between densities cannot be employed. The most complicated case is the Bayes update, when only samples are given for \(f_x(y)\). Then \(f_y(y)\) is only given implicitly as the product of the samples and the likelihood and we cannot obtain values of the posterior density nor can we sample from it.

Filters based on Knothe-Rosenblatt Triangular Maps:
In [5], a continuous prior density is assumed to be given from which samples can easily be drawn. In addition, a given likelihood is assumed that can be evaluated up to a constant. A map is characterized by the minimum of either the Hellinger metric or the KL divergence between prior and posterior, see [5, pp. 7818–7819]. The map is computed by solving a nonlinear, nonconvex minimization problem, where the distance is evaluated by Monte Carlo simulations from the prior density. Knothe-Rosenblatt rearrangements [6] are used as these are triangular and monotonic and thus easy to invert\(^1\). It is important to note that these maps are not optimal with respect to a transportation distance. To reduce complexity, decomposability of transport maps is investigated in [8], leading to sparse triangular maps. The construction of maps is generalized in [9], where two types of maps are considered. The first map\(^2\) (direct transport) transforms a reference measure to a target measure [9, p. 6]. The reference density is known, the target density can be evaluated but is unnormalized. The second map (inverse transport) transforms the target measure to the given reference measure [9, p. 13]. This is useful when the target density is unknown and only samples of it are given. Finding the map is equivalent to maximum likelihood estimation and can be solved via convex optimization. These two maps are used in [10] to derive a nonlinear ensemble filter [10, p. 20] for the case when only samples of the prior density are available: An inverse map is constructed for transforming prior samples to a convenient reference measure. A direct map is used to transform the reference measure to the posterior.

Filters based on Normalizing Flows:
Popularized in the machine learning community, normalizing flows are used in variational inference problems [11] to perform density estimation in order to model complex data distributions. A normalizing flow is a sequence of invertible transformations mapping a reference measure to a set of samples from a desired target density [12, p. 3]. The flows correspond to the inverse transport in [9] and are determined by maximum likelihood estimation.

C. Particle Flow Filters: Continuous Derivation
The first breed of particle flow filters derive a PDE or ODE based on a suitably parametrized posterior while assuming continuity of the involved densities. In a second step, the required discretizations are performed.

Daum-Huang (DH) Particle Flows: Particle flows are derived in [13] from a log-homotopy relating prior and posterior.

\(^1\)For general triangular transformations, see [7].
\(^2\)This is the map used in [5].

The flow is represented by a partial differential equation (Fokker-Planck) assuming continuous densities. Hence, estimating the required gradients from the particles is a challenging problem, see [14]. Several versions of DH flows have been proposed [15], many for coping with stiffness in the flow. However, they usually rely on some sort of nonlinear Kalman filter running in parallel that compromises performance [16].

Particle Flows based on Liouville Equation: In [17], a homotopy continuation approach similar to [1] is used. Assuming an ODE for moving particles from prior to posterior, the corresponding Liouville PDE is derived\(^3\). (A similar approach is pursued in [18, p. 5].) The desired ODE velocity field is then obtained as the solution of the Liouville PDE. Tractable solutions are obtained in the univariate case. The multivariate case requires a Gibbs approximation. This requires the full conditional distributions, which are numerically approximated. Finally, the mapped particles are “just” used as proposal distributions inside of sequential Monte Carlo samplers.

Particle Flows with Repulsion Kernels: It would be possible to employ density estimation to find an intermediate continuous representation to calculate gradients. However, density estimation usually is not differentiable. In [19], repulsion kernels [20] are used that represent the spread of probability mass around the particles. This leads to an ODE for the particle locations over an artificial time from 0 to 1.

D. Particle Flow Filters: Direct Discrete Derivation
The second breed of particle flow filters acknowledges that a computer implementation requires discretization anyway and directly derives a sequence of discrete updates.

The Bayesian particle flow proposed in this paper belongs to this category. It follows the procedure in [21]: The likelihood is adaptively split into several sub-likelihoods, each of which is easier to process. The sub-likelihoods are used to sequentially update prior particles. After each update, the weighted particles are optimally resampled with equally weighted ones. Resampling is performed by minimizing a suitable particle set distance [22] and [23]. As a result, the prior particles are moved to regions with high posterior density. The resampling step in this paper differs from the one in [21]: Transport maps are used for mapping weighted samples to unweighted ones instead of using direct optimization. This is advantageous as the number of map parameters is typically smaller than the number of parameters of the samples. In addition, the transport map allows to more easily incorporate smoothness.

III. Problem Formulation
We consider a dynamic system with a state \(\mathbf{x} \in \mathbb{R}^N\) and state dimension \(N\). A transition density describing the state evolution provides a forecast in the form of a prior PDF.

Given a continuous prior density \(f_x(\mathbf{z})\) and a continuous likelihood function \(f_L(\mathbf{z})\), the Bayesian filter step for calculating

\(^3\)The Liouville PDE or continuity equation is a special case of the Fokker-Planck PDE for zero diffusivity.

\(^4\)The likelihood function is usually obtained by plugging a specific measurement, say \(\mathbf{y}_k\), into the conditional density \(f_y(\mathbf{y}|\mathbf{z})\) describing the relation between measurement \(\mathbf{y}\) and state \(\mathbf{z}\) such that \(f_L(\mathbf{z}) = f_y(\mathbf{y}|\mathbf{z})\).
the continuous posterior density $f_p^*(\mathbf{x})$ is

$$
    f_p^*(\mathbf{x}) \propto f_p^*(\mathbf{x}) \cdot f_L(\mathbf{x}) .
$$

(2)

The symbol $\propto$ indicates that a normalization is required after the multiplication of $f_p^*(\cdot)$ and $f_L(\cdot)$.

For the important case that the prior density $f_p^*(\mathbf{x})$ is given as a set of weighted samples (or particles) it can be formally written as a Dirac mixture density

$$
    f_p^*(\mathbf{x}) = \sum_{i=1}^{L} w_{p,i} \cdot \delta(\mathbf{x} - \mathbf{x}_{p,i}) .
$$

(3)

The weights $w_{p,i} > 0$, $\sum_{i=1}^{L} w_{p,i} = 1$, and sample locations $\mathbf{x}_{p,i}$ selected such that $f_p^*(\mathbf{x}) \approx f_p^*(\mathbf{x})$.

For a given Dirac mixture prior, the Bayesian filter step becomes

$$
    \tilde{f}_e(\mathbf{x}) \propto f_L(\mathbf{x}) \cdot \sum_{i=1}^{L} w_{p,i} \cdot \delta(\mathbf{x} - \mathbf{x}_{p,i})
$$

$$
    = \sum_{i=1}^{L} \frac{w_{p,i} \cdot f_L(\mathbf{x})}{w_{e,i}} \cdot \delta(\mathbf{x} - \mathbf{x}_{p,i}) .
$$

(4)

Upon normalization, we obtain the posterior weights $\tilde{w}_{e,i} = w_{e,i} / \sum_{i=1}^{L} w_{e,i}$. The posterior Dirac mixture is now given as

$$
    \tilde{f}_e(\mathbf{x}) = \sum_{i=1}^{L} \tilde{w}_{e,i} \cdot \delta(\mathbf{x} - \tilde{\mathbf{x}}_{e,i}) .
$$

(5)

The posterior sample locations do not change w.r.t. the prior samples, i.e., $\tilde{\mathbf{x}}_{e,i} = \mathbf{x}_{p,i}$ for $i = 1, \ldots, L$.

The posterior $\tilde{f}_e(\mathbf{x})$ in (5) is derived from the straightforward application of the Bayesian filter step to Dirac mixtures. This leads to a serious problem: The samples are not equally weighted anymore and do not equally contribute to the representation of the posterior. Often, some particle weights are (close to) zero, in fact dying out, leading to particle degeneracy mentioned above. A typical scenario is large system noise, which spreads the particles during the prediction step combined with low measurement noise leading to narrow likelihoods.

Many solutions, some systematic, many of heuristic nature, have been proposed to solve the degeneracy problem, which is a fundamental and difficult problem.

**Remark III.1.** Our goal is to derive a Bayesian filter that inherently avoids degeneracy without any heuristic approaches. It should be easy to understand, simple to implement, numerically stable, and robust.

Furthermore, we propose to use deterministic particles instead of random ones. This (i) reduces the required number of particles as the placement is more homogeneous and (2) ensures reproducibility. In this paper, we use the sampling method from [24] also used in [25].

### IV. Optimal Resampling

We now develop an optimal resampling step. It replaces the non-equally weighted Dirac mixture $f_e(\mathbf{x})$ with its equally weighted approximation $f_e(\mathbf{x})$. $f_e(\mathbf{x})$ is composed of non-equal weights $\tilde{w}_{e,i}$ and locations $\tilde{x}_{e,i}$. $f_e(\mathbf{x})$ has equal weights, i.e., $w_{e,i} = w_{p,i} = 1/L$ and new locations $\tilde{x}_{e,i}$.

We collect the weights and locations in sets for $f_e(\mathbf{x})$

$$
    W_e = \{w_{e,1}, \ldots, w_{e,L}\} , \quad X_e = \{x_{e,1}, \ldots, x_{e,L}\} ,
$$

(6)

and for $\tilde{f}_e(\mathbf{x})$

$$
    \tilde{W}_e = \{\tilde{w}_{e,1}, \ldots, \tilde{w}_{e,L}\} , \quad \tilde{X}_e = \{\tilde{x}_{e,1}, \ldots, \tilde{x}_{e,L}\} ,
$$

(7)

instead of vectors and matrices to underline that there is no inherent order.

**Remark IV.1.** We first consider the case that no degeneration of $f_e(\mathbf{x})$ occurred. This means that the weight variance in $\tilde{w}_{e,i}$ is small and all samples contribute to the density representation. Degeneration will be treated in Sec. V.

The key idea to finding $\tilde{X}_{e,i}$ is to use an optimal map $M(\cdot)$ that transforms the prior random vector $\mathbf{x}_p$ to the posterior random vector $\mathbf{x}_e$

$$
    \mathbf{x}_e = M(\mathbf{x}_p) .
$$

(8)

This map is used to map prior samples $\mathbf{x}_{p,i}$ to posterior samples $\tilde{\mathbf{x}}_{e,i}$. Mapping samples does not change their weights but their locations. This guarantees equally weighted posterior samples $\tilde{\mathbf{x}}_{e,i}$. We now have to find a map $M(\cdot)$ that leads to posterior samples that fulfill $f_e(\mathbf{x}) \approx \tilde{f}_e(\mathbf{x})$.

The map generation is shown in Fig. 1. $\tilde{f}_e(\mathbf{x})$ is obtained by a Bayes update, i.e., by multiplying $f_p(\mathbf{x})$ with the likelihood $f_L(\mathbf{x})$ (upper path). It serves as the reference density. Its locations $\tilde{x}_{e,i}$ are identical to those of $f_p(\mathbf{x})$, only its weights $\tilde{w}_{e,i}$ are changed. In the lower path, the map $M(\cdot)$ propagates $f_p(\mathbf{x})$ to $f_e(\mathbf{x})$. $f_e(\mathbf{x})$ has identical weights as $f_p(\mathbf{x})$. Its locations $\tilde{x}_{e,i}$ have changed due to the mapping though. We desire $f_e(\mathbf{x})$ to be close to $\tilde{f}_e(\mathbf{x})$ w.r.t. an appropriate distance measure $D$. The map $M(\cdot)$ is adjusted accordingly by minimizing $D(f_e(\mathbf{x}), \tilde{f}_e(\mathbf{x}))$.

**A. Properties of Map**

It would be sufficient to use a discrete map for mapping the $L$ samples individually. However, in that case the number of map parameters would be equal to the number of samples. As a result, we would not have a complexity gain compared to a direct reapproximation. In addition, there would be no smoothing.

Hence, we use a continuous map, preferably one with as few parameters as possible to reduce complexity. This automatically allows interpolation between samples. In summary, a few prior samples economically produce a smooth mapping that can be used to map many samples.

**Remark IV.2.** Interpolation can be used to increase the number of posterior samples $\tilde{x}_{e,i}$. This can be done after the map has been generated.
Prior $f_p(x)$  

\[
\text{Likelihood } f_L(x) \quad \text{Locations: unchanged}
\]
\[
\tilde{f}_e(x) \quad \text{Weights: changed}
\]

Bayes Update  

\[
f_e(x) = M(x) \quad \text{f.e. Resampled Posterior}
\]
\[
\tilde{f}_e(x) \quad \text{Locations: changed}
\]
\[
\tilde{f}_e(x) \quad \text{Weights: unchanged}
\]

Distance Measure  

\[
x_e = M(x_p) \quad \text{True Posterior}
\]
\[
f_e(x) \quad \text{Resampled Posterior}
\]

Fig. 1: Block diagram of generating a resampling map given prior $f_p(x)$ and likelihood $f_L(x)$. $f_e(x)$ is the non-equally weighted posterior resulting from the base update. $\tilde{f}_e(x)$ is the resampled equally weighted posterior with $f_e(x) \approx \tilde{f}_e(x)$.

The map $M(.)$ has the following properties:

1. **Property 1** Might be non-monotonic.
2. **Property 2** No inverse required.
3. **Property 3** Only used for mapping sample values.
4. **Property 4** Differentiability w.r.t. parameters required.
5. **Property 5** Differentiability not required w.r.t. $x$.

### B. Specific Map

According to Remark IV.1, in this section we assume that the change from $f_p(x)$ to $f_e(x)$ is small. Hence, the map $M(.)$ is close to the identity mapping.

Here, we propose a combination of an affine base map combined with a radial basis function nonlinearity. For a single output it is given by

\[
M_i(x) = a^T x + b_i + \sum_{r=1}^{R} w_{r,i} \cdot \text{RBF}_r(x - x_r),
\]

with weights $w_{r,i} \in \mathbb{R}$, locations $x_r$, and radial basis kernels $\text{RBF}_r(.)$. The complete vector-valued map $M(.) : \mathbb{R}^N \rightarrow \mathbb{R}^N$ is given by

\[
M(x) = [M_1(x), M_2(x), \ldots, M_N(x)]^T.
\]

For the optimization, initial values can simply be set to $a_0 = 1, b_0 = 0$, and $w_{r,i} = 0$ for $r = 1, \ldots, R, i = 1, \ldots, N$. The RBF locations $x_r$ could be set to fixed a priori locations for $r = 1, \ldots, R$.

### C. Distance Measure

We have to compare the weighted Dirac mixture $\tilde{f}_e(x)$ with its unweighted counterpart $f_e(x)$. A suitable distance measure should satisfy the following requirements:

1. **Requirement 1**: Handle Dirac mixture densities\(^5\).
2. **Requirement 2**: Handle non-equal weights.
3. **Requirement 3**: Handle non-equal supports.
4. **Requirement 4**: Be differentiable w.r.t. locations (and weights). The first three properties cannot be handled with standard distances that require continuous densities. This includes the KL-divergence and integral squared distances. On the other hand, Wasserstein distances could be used. However, they suffer from large complexity. In addition, property 4 is not fulfilled. Here, we propose the use of the Cramér-von Mises distance [22] based on Localized Cumulative Distributions [23].

For two Dirac mixture densities $f_e$ with $L$ components, weights $w_{x,1}, w_{x,2}, \ldots, w_{x,L}$, locations

\[
x_i = [x_{1,i}, x_{2,i}, \ldots, x_{N,i}]^T \in \mathbb{R}^N
\]

for $i = 1, \ldots, L$ and $f_y$ with $M$ components, weights $w_{y,1}, w_{y,2}, \ldots, w_{y,M}$, locations

\[
y_j = [y_{1,j}, y_{2,j}, \ldots, y_{N,j}]^T \in \mathbb{R}^N
\]

for $j = 1, \ldots, M$, the Cramér-von Mises distance $D$ is given by

\[
D = D_{yy} - 2D_{xy} + D_{xx} + c D_E,
\]

with

\[
D_{yy} = \sum_{i=1}^{M} \sum_{j=1}^{M} w_{y,i} \cdot w_{y,j} \cdot \log \left( \frac{D}{d=1} (y_{d,i} - y_{d,j})^2 \right),
\]

\[
D_{xy} = \sum_{i=1}^{L} \sum_{j=1}^{M} w_{x,i} \cdot w_{y,j} \cdot \log \left( \frac{D}{d=1} (x_{d,i} - y_{d,j})^2 \right),
\]

\[
D_{xx} = \sum_{i=1}^{L} \sum_{j=1}^{L} w_{x,i} \cdot w_{x,j} \cdot \log \left( \frac{D}{d=1} (x_{d,i} - x_{d,j})^2 \right),
\]

and $\log(z) = z \cdot \log(z)$. $D_E$ can be viewed as a penalty term (with weight $c$) that ensures equal means and is given by

\[
D_E = \sum_{d=1}^{N} \left( \sum_{i=1}^{L} w_{x,i} \cdot x_{d,i} - \sum_{i=1}^{M} w_{x,i} \cdot y_{d,i} \right)^2.
\]

When only the minimizer w.r.t. parameters of $f_e$ is desired, but not the corresponding value of $D$, $D_{yy}$ can be neglected.

### D. Map Optimization

In this paper, $L$ and $M$ are assumed to be equal, with

\[
w_{x,i} = w_{e,i}, \quad w_{y,i} = w_{e,i}, \quad x_i = \tilde{x}_e, \quad y_i = \tilde{x}_e,
\]

for $i = 1, \ldots, L$. This leads to the following dependency of $D$

\[
D = D(W_e, \tilde{X}_e, \tilde{W}_e, \tilde{X}_e).
\]

As $\tilde{X}_e$ is obtained from $\tilde{X}_e$ via the map $\tilde{X}_e = M(\tilde{X}_e)$, this can be rewritten as

\[
D = D(W_e, M(\tilde{X}_e), \tilde{W}_e, \tilde{X}_e).
\]

The optimal map $M^*$ is now found by minimization

\[
M^* = \arg \min_{M \in \mathcal{M}} D(W_e, M(\tilde{X}_e), \tilde{W}_e, \tilde{X}_e),
\]

where $\mathcal{M}$ is the set of viable maps.

The gradient of $D$ in (13) is available in closed form [22]. When the likelihood is given in analytic form and differentiable, the gradient with respect to the map parameters can be derived. A BFGS quasi-Newton method is used for optimization.
In the previous section, we assumed that reweighting of particles with the likelihood \( f_L(x) \) kept all particles ‘alive’. Usually, however, performing the Bayes update in one step leads to particle degeneration. Only a few samples stay ‘alive’, the remaining ones are close to zero. In that case, resolution is lost as not all particles contribute to the density representation.

A proven remedy for keeping particles “alive” is to perform progressive processing [21]. The likelihood is decomposed into a product of sub-likelihoods, each of which is carefully selected to avoid degeneration

\[
f_L(x) = f_L^{(1)}(x) \cdot f_L^{(2)}(x) \cdots f_L^{(K)}(x) = \prod_{k=1}^{K} f_L^{(k)}(x). \tag{22}
\]

As we use a product decomposition, each sub-likelihood is intuitively “wider” than the original one. Sequential Bayes sub-updates with the sub-likelihoods then provide the desired posterior:

\[
f_e(x) \propto f_p(x) \cdot f_L(x) = f_p(x) \cdot f_L^{(1)}(x)
\]

non-equally weighted \( \overline{f}_L^{(1)}(x) \)

Resampling RS

equally weighted \( f_p(x) \cdot f_L^{(2)}(x) \)

non-equally weighted \( \overline{f}_L^{(1)}(x) \cdot f_L^{(2)}(x) \)

Resampling RS

equally weighted \( f_p(x) \cdot f_L^{(3)}(x) \)

\[ \cdots \]

non-equally weighted \( \overline{f}_L^{(K)}(x) \)

equally weighted \( f_p(x) \cdot f_L^{(K)}(x) \approx f_e(x) \)

After every sub-update, we obtain an non-equally weighted sub-posterior \( f_e^{(k)}(x) \). In order to prepare for the next sub-update, the optimal resampling method from Sec. IV is used. This involves mapping samples \( \overline{x}_{e,i}^{(k)} \) to \( \tilde{x}_{e,i}^{(k)} \) with sub-mapping \( M^{(k)}(.) \). This results in an equally weighted sub-posterior \( f_e^{(k)}(x) \approx f_e^{(k)}(x) \). After \( K \) sub-update steps, the result is an equally weighted sub-posterior \( f_e^{(K)}(x) \), which is equal to \( f_e(x) \).

The total map from prior samples \( \overline{x}_{p,i} \) to posterior samples \( \overline{x}_{e,i} \) is given by composition of the sequence of individual mappings as

\[
M(x) = M^{(K)} \left( M^{(K-1)} \left( \cdots M^{(2)} \left( M^{(1)}(x) \right) \right) \right) \tag{23}
\]

or

\[
M = M^{(K)} \circ M^{(K-1)} \circ \cdots \circ M^{(2)} \circ M^{(1)}. \tag{24}
\]
Fig. 2: Results of a linear update with $L = 10$ samples. The true continuous posterior is shown in green. The non-equally weighted Dirac mixture $\tilde{f}_e(x)$ after multiplication of $f_p(x)$ with the likelihood $f_L(x)$ is shown in black. The equally weighted Dirac mixture $f_e(x)$ produced by the proposed Bayesian particle flow is shown in red. Please note the different x-axes scales.

Fig. 3: Results of a linear update with $L = 30$ samples. The true continuous posterior $f_t(x)$ is shown in green. The equally weighted Dirac mixture $f_e(x)$ produced by the proposed Bayesian particle flow is shown in red. The reference Dirac mixture obtained by directly sampling from the true continuous posterior $f_t(x)$ is shown in black. Please note the different x-axes scales.

$$f_L(x) = f_v(\hat{y} - x^3) = N(\hat{y}; x^3, \sigma_v).$$ The true posterior is given by

$$\tilde{f}_e(x) \propto f_p(x) \cdot f_L(x) = N(0, 1) \cdot N(\hat{y}; x^3, \sigma_v).$$  \hspace{1cm} (25)

Remark VI.1. Sampling from $\tilde{f}_e(x)$ is difficult. Calculating the required CDF and its inverse can only be done by numerical integration.

We are given only samples $x_{p,i}$ of the prior density $f_p(x)$ and the analytic likelihood $f_L(x)$. Samples $x_{e,i}$ of the posterior $f_e(x)$ are calculated with the proposed Bayesian flow. Fig. 4 (4, 5) show the prior PDF $f_p(x)$, the prior CDF $F_p(x)$, and its samples $x_{p,i}$. Fig. 4 (3) shows the flow of the particles. Fig. 4 (1, 2) show the posterior CDF $F_e(x)$, the posterior PDF $f_e(x)$, and its samples $x_{e,i}$. The total mapping from $x_p$ to $x_e$ is shown in Fig. 7 (in red) compared to a reference map $M'(\cdot)$ (in green). $M'(\cdot)$ is calculated from $F_p'(\cdot)$ and a numerically calculated $Q'(\cdot) = (F_e')^{-1}(\cdot)$ with (1). Again, the maps are almost identical in the relevant region.

C. Comparison with Particle Filter

The proposed Bayesian particle flow will now be compared with the standard particle filter. The underlying continuous prior is given by $f_p(x) = N(x; 0, 1)$ and shown in blue in Fig. 8. We only have a Dirac mixture approximation $f_p(x) \approx f_p'(x)$ as in (3) available. The likelihood

$$f_L(x) = \exp\left(\frac{1}{2}((x - 1.2)(x - 1.5)(x + 1.2)(x + 1.5))^2\right)$$  \hspace{1cm} (26)

is shown in green. The true continuous posterior $f_t'(x)$ (in red) is unknown to the estimator.

The results are shown in Fig. 5. The posterior estimate of the proposed Bayesian particle flow for $L = 50$ particles in Fig. 5 (1) is very close to the true posterior. Results of the standard particle filter are shown in Fig. 5 (2)–(5) for different $L$ with ten runs each. For $L = 50$, the results are of low resolution at the peaks of the posterior. This is due to the fact that the filter step of the particle filter produces weighted samples. In addition, there is large variability between different
runs. For increasing $L$, the resolution degradation becomes less pronounced and the variability between runs decreases. However, even for $L = 500$ the variability is still clearly present.

VII. CONCLUSIONS

A new Bayesian particle flow has been derived that deterministically guides particles from prior to posterior. It does not require any continuous density representations, neither in its derivation nor in its implementation. It is composed of a finite sequence of potentially non-monotonic maps for propagating particles. The filter works with arbitrary nonlinear measurement equations. However, it is assumed that it has already been converted to a likelihood function that can be evaluated. The method is easy to understand and its implementation is straightforward.

Several distance measures could be employed for map optimization, see Subsec. IV-A. However, we found the distance derived in [22] most useful as it has low complexity and is differentiable. Its complexity is quadratic in the number of particles $L$ and linear in the number of dimensions $D$.

Several aspects have been omitted in this paper due to space restrictions. (i) We did not give details on how to find appropriate sub-likelihoods that keep particles “alive” and their number. For that purpose, we use the method described in [26]. (ii) We assumed that the number of particles $L$ is constant. This is not necessary. The number of samples can be adapted to the complexity of the underlying density. Methods for an efficient adaptation will be developed. (iii) We did not discuss the prediction step, i.e., propagating particles through the system model. For deterministic particles, the prediction step is significantly different from the random case in terms of combining state and noise samples. In order to avoid a full Cartesian product, the method from [27] can be used.

REFERENCES

[1] Uwe D. Hanebeck, Kai Briechle, and Andreas Rauh. “Progressive Bayesian Particle Flow for Single Measurement Update with Likelihood Shown in Fig. 8. (1) Result of proposed filter for $L = 50$. (2)–(5) Results of particle filter for different $L$ with 10 runs each.

Several aspects have been omitted in this paper due to space restrictions. (i) We did not give details on how to find appropriate sub-likelihoods that keep particles “alive” and their number. For that purpose, we use the method described in [26]. (ii) We assumed that the number of particles $L$ is constant. This is not necessary. The number of samples can be adapted to the complexity of the underlying density. Methods for an efficient adaptation will be developed. (iii) We did not discuss the prediction step, i.e., propagating particles through the system model. For deterministic particles, the prediction step is significantly different from the random case in terms of combining state and noise samples. In order to avoid a full Cartesian product, the method from [27] can be used.
Fig. 6: Map $M(\cdot)$ for linear filter step.  

Fig. 7: Map $M(\cdot)$ for cubic sensor problem.  

Fig. 8: Green: Likelihood $f_t(z|x)$ in (26). Blue: $f_t^p(x)$. Red: $f_t^e(x)$.


