Progressive Particle Filtering Using Projected Cumulative Distributions

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

***Abstract*—We propose a progressive particle filter that inherently avoids sample degeneracy by splitting the likelihood into a product of wider functions applied step by step. Between steps, the particle distribution is resampled using projected cumulative distributions (PCDs). To be able to handle weighted Dirac mixture distributions, their corresponding one-dimensional densities are interpolated with a piecewise constant function. Both Cramér-von-Mises and 2-Wasserstein distance are used as base objective functions for PCDs to deterministically and optimally resample such distributions. The proposed filter is compared with a standard SIR particle filter on a simulated tracking problem.**

***Index Terms*—Nonlinear filtering, particle filter, progressive filtering, resampling, probability metrics.**

I. Introduction

Bayesian state estimation is an important topic for many different fields of application. It requires a prior distribution that encodes the current knowledge about the state and a likelihood function that encodes how likely each measurement is given a state. With these two building blocks, the posterior distribution can be calculated with Bayes rule by multiplying the prior distribution with the likelihood function for a given measurement and normalizing the result. Systems with a linear relation between the state and measurements, and only additive Gaussian noise can be optimally handled with a Kalman filter. In real systems this relation often is nonlinear and may not even be analytically tractable. There are various linearization techniques for filtering these systems. This can be the straightforward linearization of the measurement function as in an Extended Kalman Filter or stochastic linearization as in an Unscented Kalman Filter [1] or Smart Sampling Kalman Filter [2].

A different approach is to represent the prior distribution with samples, also called particles. This is the idea behind particle filters, that can handle arbitrary nonlinear state transition and measurement functions. There are no assumptions made about the Gaussianity of any of the involved distributions.

In the prediction step, the particles are moved to new locations by propagating them through the transition function. In the filter step, the particles are then weighted according to the measurement likelihood, but their position is not changed. Many particles are typically necessary to get sufficiently good approximations of the state distribution after each prediction step and filter step. A common challenge to be dealt with in particle filters is so-called sample degeneracy. Particles at locations where the likelihood function is close to zero will get assigned a weight that is almost zero. Without additional measures, more particles will get assigned almost zero weight until only very few of the original particles are left to represent the current state distribution. Many different approaches have been proposed to avoid sample degeneration and keep a good sample-based approximation of the posterior density. Nevertheless, this remains a demanding challenge and an open topic of research.

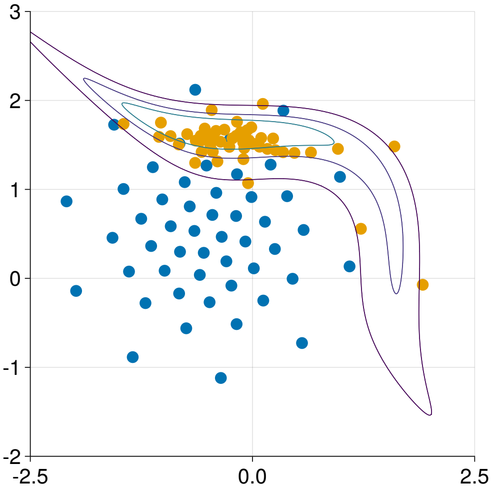


Fig. 1: Prior distribution (blue) and estimated posterior distribution (yellow) with 50 particles for a Gaussian prior and a cubic measurement function using the proposed progressive particle filter. The outlines of the 50%, 80% and 99% percentiles of the true posterior are shown in purple and blue.

We introduce a new type of progressive particle filter that inherently avoids sample degeneracy. Like in other progressive filters, the likelihood is applied to the prior distribution in multiple progression steps. After each of these steps, the current distribution is resampled to an equally weighted sample distribution. For the resampling, existing methods based on Projected Cumulative Distributions (PCDs) [3] were extended to be able to handle weighted distributions. Lastly, two different cost functions for PCD-based resampling are compared.

II. State of the Art

There exist many different strategies to handle sample degeneracy in a particle filter. Some of them will now be reviewed, focusing on methods similar to our proposed algorithm.

A. Resampling

Resampling is one very common way to avoid degeneration of particles. It works by approximating the set of posterior particles with a different set of particles that all have the same weight. As a result, particles with small weights are removed and particles with a large weight are replaced by multiple particles with smaller weights. A very well-known algorithm to achieve this is called importance sampling. It randomly samples particles from the posterior density with replacement based on their weight until the desired amount of particles is reached.

An overview of many resampling strategies for particle filters can be found in [4]. One distinction to be made between them is, if they use some kind of randomness like importance sampling does or if they deterministically compute a new set of particles. Some examples for deterministic resampling by minimizing an objective function are methods based on Localized Cumulative Distributions (LCDs) [5] and PCDs [6].

B. Progressive Filtering

In cases where only very few particles of the posterior have nonzero weight, resampling might not be enough to give an accurate representation of the true posterior. In these cases the multiplication of the prior with the likelihood can be split into multiple steps and applied progressively. The goal is to slowly transform the prior distribution into the posterior without loosing too much information.

A Gaussian progressive filter that uses samples as an intermediate density representation was introduced in [7]. This can similarly be done for filtering based on Gaussian mixture densities [8]. A progressive particle filter was described in [9], where the position and weight of each particle is corrected after every progression step. Another type of progressive particle filter is better known as annealing particle filter [10]. It uses simulated annealing for reapproximation of the posterior density between progression steps. Alternatively, sequences of optimal transport maps can also be used for doing this reapproximation [11].

C. Particle Flow Filters

Particle flow filters go one step further than progressive filters and model the transformation from prior to posterior distribution as a PDE or ODE. Daum-Huang particle flow filters use the Fokker-Planck equation to model this and solve it for the posterior sample positions [12]. Other approaches use the Liouville equation to model the flow [13] or derive an ODE based on repulsion kernels around particles [14].

III. Problem Formulation

We consider a general dynamic system with state vector $\underline{x} \in \mathbb{R}^D$ of dimension D. This system can be observed through measurements $\hat{\underline{z}}$ that are connected to the state through a likelihood function $\mathcal{L}_{\hat{\underline{z}}}(\underline{x})$.

Given a prior distribution $f^p(\underline{x})$ of the state, the posterior distribution $f^e(\underline{x})$ can be calculated using Bayes rule as

$$f^e(\underline{x}) = \eta \cdot \mathcal{L}_{\hat{\underline{z}}}(\underline{x}) \cdot f^p(\underline{x}) \ . \tag{1}$$

The normalization factor η makes sure that the resulting function integrates to one and therefore is a valid probability distribution.

In a particle filter, the prior distribution is approximated through samples of the underlying continuous distribution. This discrete sample distribution can be written as a Dirac mixture distribution

$$f^p(\underline{x}) = \sum_{n=1}^{N} w_n^p \cdot \delta(\underline{x} - \underline{x}_n^p) \tag{2}$$

with sample positions $\underline{x}_n^p$ and sample weights w_n^p. In the filter step of the particle filter, Bayes rule (1) is applied directly to the Dirac mixture approximation. Each of the particles is weighted according to its likelihood to obtain the posterior distribution

$$f^e(\underline{x}) = \eta \cdot \mathcal{L}_{\hat{\underline{z}}}(\underline{x}) \cdot \sum_{n=1}^{N} w_n^p \cdot \delta(\underline{x} - \underline{x}_n) \tag{3}$$

$$= \eta \cdot \sum_{n=1}^{N} w_n^p \cdot \mathcal{L}_{\hat{\underline{z}}}(\underline{x}_n) \cdot \delta(\underline{x} - \underline{x}_n) \tag{4}$$

$$= \eta \cdot \sum_{n=1}^{N} w_n^e \cdot \delta(\underline{x} - \underline{x}_n) \ . \tag{5}$$

This is also a Dirac mixture density with $w_n^e = w_n^p \cdot \mathcal{L}_{\hat{\underline{z}}}(\underline{x}_n)$ and $\eta = 1/\sum_{n=1}^{N} w_n^e$. In a simple particle filter, this density would be propagated through the transition function and the result used as the prior for the next filter step. It is important to note, that in the filter step only the weights of the particles change, but not their position. During the prediction step on the other hand, only the positions change and not the weights.

In systems with low measurement noise and therefore a narrow likelihood function, often only very few particles lie in areas, where the likelihood is nonzero. In these cases, many of the new particle weights are set close to zero during the filter step, effectively leading to dying out of particles in regions with low likelihood. Over time, this reduces the number of particles that effectively contribute weight to the Dirac mixture and therefore reduces the quality of the density approximation.

Importance resampling can be an effective measure to mitigate these effects. It randomly removes samples with small weights and clones samples with large weights. After the next prediction step, which spreads out the cloned samples, most of the particles lie in areas with high posterior weights. The weights of this new prior distribution are then set to be all equal to $\frac{1}{N}$. This means that all the information about the distribution is now in the position of the particles and their weights are "reset" for the next filter step. One drawback of this method is the poor approximation of the posterior density after resampling and before spreading the cloned particles out again. The spreading is commonly done in the prediction step of the filter, introducing an additional dependence of the filter step on the prediction step and its properties.

Our goal is to develop a progressive particle filter that can handle narrow likelihoods and large system noise while only using a small number of particles. An important part of its design is the introduction of a new deterministic resampling strategy that can approximate a weighted Dirac mixture density with another one where all particles have the same weight.

IV. Resampling

After the filter step of the particle filter, we would like to resample the posterior distribution in such a way that the new particles all have the same weight. We also keep the number of particles before and after resampling the same. This means that we would like to find the positions of N samples with weight $1/N$ that optimally represent an original set of N samples with unequal weights w_j.

With new sample positions x_i and given some original sample positions y_j we want to find

$$\sum_{i=1}^{N} \frac{1}{N} \delta(\underline{x} - \underline{x}_i) \approx \sum_{j=1}^{N} w_j \delta(\underline{x} - \underline{y}_j), \tag{6}$$

where both Dirac mixtures should represent the same underlying density.

The new sample positions should be found in a deterministic and optimal way. Therefore, a metric is needed to compare the two distributions. The optimal new sample positions can then be found by minimizing this metric.

Because Dirac mixture densities are zero almost everywhere, most classical statistical distances, like the KL-divergence, cannot be used to compare them. In fact all metrics that use the probability density functions (PDFs) directly, will run into this problem and might not give meaningful results for Dirac mixtures.

In one dimension this can be circumvented by using metrics that utilize the cumulative density functions (CDFs) of the involved distributions as the CDF of a Dirac mixture is just a staircase function. Examples for such distances are the Cramér-von-Mises distance and the 2-Wasserstein distance. Given two CDFs $F(x)$ and $G(x)$ of two distributions μ and ν the Cramér-von-Mises distance

$$d_C(\mu, \nu) = \int_{-\infty}^{\infty} (F(x) - G(x))^2 \, \mathrm{d}x \tag{7}$$

straight up integrates the squared difference between them. The 2-Wasserstein distance on the other hand can be written as

$$d_W(\mu, \nu) = \int_{0}^{1} (F^{-1}(x) - G^{-1}(x))^2 \, \mathrm{d}x, \tag{8}$$

using the quantile functions $F^{-1}(x)$ and $G^{-1}(x)$ [15]. Alternatively, it can be written in terms of an optimal transport plan $\pi(x, y)$ as

$$d_W(\mu, \nu) = \min_{\pi(x,y)} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \pi(x, y)(x - y)^2 \, \mathrm{d}x \, \mathrm{d}y. \tag{9}$$

In this interpretation, the 2-Wasserstein distance can be seen as the minimal cost needed to transform one of the distributions into the other one and is the result of the minimization problem to find the optimal transport plan, that does this.

For probability distributions in two or more dimensions a CDF cannot be uniquely defined [16]. Instead, the results depend on the order of integration. Alternatives to the CDF have been proposed that circumvent this problem and yield unambiguous results.

One of them are Localized Cumulative Distributions [16]. They work by integrating over Gaussian kernels of all different sizes and creating a unique and smooth representation of the local probability mass at each point. While this method can be used with any density, it is computationally quite expensive.

PCDs are another surrogate for a multi-dimensional CDF. Here, the density is projected onto each possible one-dimensional subspace and the CDFs of these projected distributions are compared. The concept of PCDs was first introduced to sample from probability distributions [6] and was later also extended to distributions on the circle [17]. They have also been used for sample reduction and deterministic resampling of sets of samples [3].

This concept is very similar to the so-called sliced Wasserstein distance [18] and other sliced measures for probability distributions. They approximate the true distance between two distributions with the average distance between projections onto one-dimensional slices through the distributions.

In fact, it can be shown that any distance metric between one-dimensional probability distributions can be used in this manner and induces a corresponding sliced metric [19].

With a general distance metric $d_G(\alpha, \beta)$ for one-dimensional distributions α and β, the corresponding sliced metric $D_G(\mu, \nu)$ between distributions μ and ν can be written as

$$D_G(\mu, \nu) = \frac{1}{V} \sum_{v=1}^{V} d_G(\underline{p}_v \# \mu, \underline{p}_v \# \nu), \tag{10}$$

where $\underline{p}_v \# \mu$ is the push-forward measure of μ projected onto $\underline{p}_v$. The vectors $\underline{p}_v$ are unit vectors uniformly sampled from the unit hypershpere and serve as projection directions or slicing directions. A fixed number of V of these directions are sampled and the distributions are projected onto each of them, resulting in V one-dimensional distributions. The final distance is then calculated as the average distance between the two distributions along all of the directions. The term "sliced" is a little misleading for these metrics, because the distributions are actually projected rather than just sliced.

We will use two of these sliced metrics with the 2-Wasserstein distance and Cramér-von-Mises distance for resampling.

A. Resampling of Weighted Dirac Mixtures

In previous research, PCDs were only used for sampling from continuous densities and reducing the number of samples for discrete densities. To be able to use them for resampling a weighted Dirac mixture some modifications have to be made.

The core challenge to be solved is that the resampling algorithm does not know about the underlying density, that the samples are drawn from. It only takes probability mass at the sample positions into account. This is fine when representing multiple samples with small weights through one new samples, that will lie somewhere in between them. On the other hand, if one sample with a large weight should be represented by several new samples, they will all lie at the original sample location, while ideally they would be spread around the original sample position. This effect leads to clumping and other undesired effects when trying to directly resample a weighted Dirac mixture with the proposed PCD-based algorithm.

To get around this limitation, some kind of interpolation or reconstruction of the underlying density from which the new samples can be generated is needed. We propose to interpolate each of the one-dimensional projected densities with a piecewise constant function leading to piecewise linear CDFs. The mass of a sample is then spread over an interval around the sample and not concentrated at its position. This approach raises the question of how to choose the size of this interval. A very simple method would be to draw the interval borders in the middle of each pair of neighboring samples. For the first and last sample, where only one neighbor is present, the outer borders can either be set as a fixed distance from the sample or the inner border can be mirrored.

This strategy can lead to problems in some situations, especially when there are only few original samples available as in the example in Fig. 2. In these cases, the density resulting from the projection and interpolation is still far from the projection of the ground truth underlying density, see Fig. 2b One reason for this is that distances between samples are generally compressed through projections onto subspaces in Euclidean spaces. However, this is not a uniform compression, but depends on the projection direction and positions of the samples. Samples might lie much closer together in the projections than they are in the original space, reducing the area, that their probability mass is be spread over.

To alleviate these negative effects, we propose to fix the interval width for each sample for all projection directions, see Fig. 2c. As a general approach, we propose to use the average distance of a sample to its k nearest neighbors as interval width. Some other option would be to use a predefined interval width or derive the optimal width from a k-dist graph, as is done in clustering, where similar challenges exist [20].

B. Sampling Using PCDs

After choosing an interpolation strategy for the original weighted Dirac mixture density, we now want to find new sample positions that minimize a sliced distance (10) to it.

Mathematically this means solving the minimization problem

$$\underline{\mathcal{X}}^* = \arg\min_{\underline{\mathcal{X}}} D_G(\mu(\underline{\mathcal{X}}), \nu) \quad . \tag{11}$$

In this notation, the sample positions $\underline{x}_1, \ldots, \underline{x}_N$ are stacked into a column vector $\underline{\mathcal{X}}$ in an arbitrary but fixed order. This optimization problem can be solved with standard gradient-based optimization methods such as gradient descent or Newton's method. The gradient of (11) with a general distance function can be written as

$$\nabla D_G(\mu, \nu) = \left[\frac{\partial D_G}{\partial \underline{x}_1}, \ldots, \frac{\partial D_G}{\partial \underline{x}_N}\right]^\top \quad . \tag{12}$$

To calculate the required partial derivatives for the 2-Wasserstein and Cramér-von-Mises distance, we can use the fact that the resampled density is a Dirac mixture density. This means, that its projected cumulative distribution $F_v(r; \underline{r}_v)$ for direction $\underline{p}_v$ is a staircase function that only depends on the projected sample positions $\underline{r}_v = [\underline{p}_v^\top \underline{x}_1, \ldots, \underline{p}_v^\top \underline{x}_N]^\top$. We further denote the PCD of the density to be resampled as $G_v(r)$. By substituting these definitions into (10) and using that $\frac{\mathrm{d} r_{v,i}}{\mathrm{d} \underline{x}_i} = \underline{p}_v$ we get

$$\frac{\partial D_G}{\partial \underline{x}_i} = \frac{1}{V} \sum_{v=1}^{V} \frac{\partial}{\partial r_{v,i}} d_G(F_v(r; \underline{r}_v), G_v(r)) \, \underline{p}_v \quad . \tag{13}$$

Replacing the general distance with the definitions of the Cramér-von-Mises distance (7) yields the final partial derivatives

$$\frac{\partial D_C}{\partial \underline{x}_i} = \frac{2}{V} \sum_{v=1}^{V} (F(r_{v,i}, \underline{r}_v) - G_v(r_{v,i})) \, \underline{p}_v \quad . \tag{14}$$

Note that due to the sifting property of the Dirac impulse the PCDs only need to be evaluated at the projected sample positions $r_{v,i}$.

Regarding the Hessian matrix required in the Newton step only the diagonal terms are of interest, as all cross terms are zero. With the probability density function $g_v(r)$ corresponding to $G_v(r)$ these can be derived to be

$$\frac{\partial^2 D_C}{\partial \underline{x}_i^2} = -\frac{2}{V} \sum_{v=1}^{V} g_v(r_{v,i}). \tag{15}$$

To calculate the gradient of (10) when the 2-Wasserstein distance is used, the optimal transport representation (9) is inserted into (12).

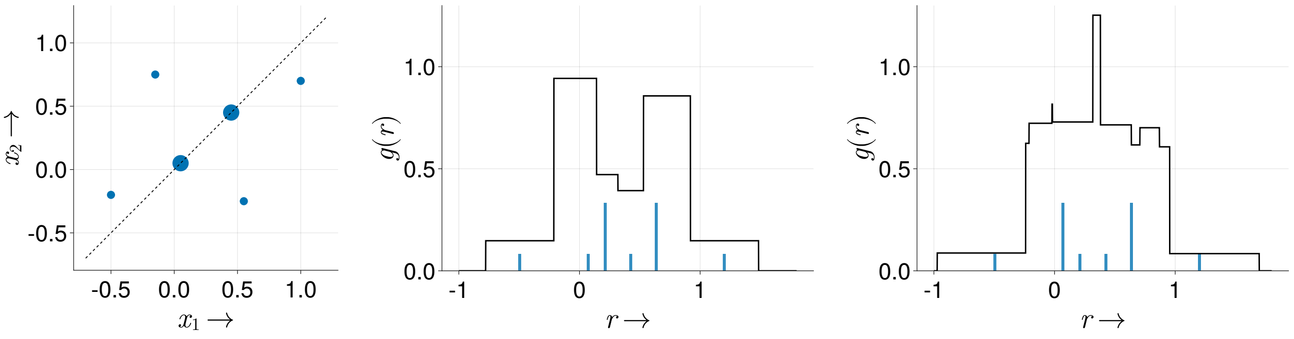
This makes the gradient of the original optimization problem dependent on another minimization problem to find the optimal transport plan. The EM-style algorithm from [3] is adopted to solve this nested optimization problem. It optimizes the transport plans and sample positions alternately until convergence, doing an optimization step for one of them, while keeping the other one fixed.

The calculation of the optimal transport plan between a Dirac mixture and another density can be done quite easily in one dimension. First, the samples of the Dirac mixture are sorted by position. The PDF of the second density is then split into intervals matching the probability masses of the Dirac mixture samples. The optimal transport plan for the i-th sample is then identical to the PDF of this second density in the i-th interval and zero everywhere else [21].

Gradient-based optimization methods are then used to optimize the sample positions. Starting with the general gradient (12) and inserting the 2-Wasserstein distance the partial derivatives are

$$\frac{\partial D_W}{\partial \underline{x}_i} = \frac{2}{V} \sum_{v=1}^{V} \int_{-\infty}^{\infty} \pi_{v,i}(r) \cdot (r_{v,i} - r) \, \mathrm{d}r \cdot \underline{p}_v \quad . \tag{16}$$

The fact that the resampled density is a Dirac mixture was used to convert one of the integrals in (9) into a sum over all sample positions. Therefore, in the partial derivative with respect to $\underline{x}_i$ all terms except the i-th one vanish. The transport plan was also simplified, so we have one continuous transport plan for each of the new sample positions $r_{v,i}$, that encodes



(a) (b) (c)

Fig. 2: (a) Example Dirac mixture density and projection direction with (b) interpolation based on interval centers and (c) interpolation with a fixed interval width, determined by the average distance of each sample to its 4 nearest neighbours.

which part of the complete probability mass is moved to this position.

By splitting the integral and using that the total probability mass moved to a sample position needs to equal its weight the derivatives can be simplified further to

$$\frac{\partial D_W}{\partial \underline{x}_i} = \frac{2}{V} \sum_{v=1}^{V} \underline{p}_v \left(\frac{1}{N} r_{v,i} - \int_{-\infty}^{\infty} \pi_{v,i}(r) \cdot r \,\mathrm{d}r \right) \quad . \tag{17}$$

Substituting back in $\underline{p}_v^\top \underline{x}_i$ for $r_{v,i}$ and setting $r_{v,i}^* = N \int \pi(r, r_{v,i}) \cdot r \,\mathrm{d}r$ we get

$$\frac{\partial D_W}{\partial \underline{x}_i} = \frac{2}{VN} \sum_{v=1}^{V} \underline{p}_v \left(\underline{p}_v^\top \underline{x}_i - r_{v,i}^* \right) \quad . \tag{18}$$

The values $r_{v,i}^*$ can be seen as the optimal solutions for the corresponding one-dimensional problem and are calculated as the mean of the interval that was assigned to this sample.

It can be observed, that (18) is equivalent to the partial derivatives of the least-squares problem

$$\underline{x}_i^* = \arg\min_{\underline{x}_i} \frac{1}{VN} \sum_{v=1}^{V} \| \underline{p}_v^\top \underline{x}_i - r_{v,i}^* \|_2^2 \quad . \tag{19}$$

Instead of using a general iterative solver, this least-squares problem can be solved with a specialized linear solver like conjugate gradient, improving performance. These small least-squares problems for each of the samples positions can also be combined into one large problem, that solves for all sample positions at once, as has been done in [3]. When using the EM-style alternating algorithm, there is one major difference between using this least-squares method and Newton's method. The least-squares method will always calculate the optimal solution for a fixed transport plan, after which the transport plan is updated with the new sample positions. Iterative solvers on the other hand usually only take a small step toward the optimum per iteration. The transport plan can then be recalculated after each small change, which might lead to a different solution than least-squares.

V. Progressive Filtering

If the prior distribution and likelihood function have very little overlap it may happen that resampling the posterior after the filter step, does not give satisfactory results. In this case too much information about the true underlying posterior density was already lost during weighting of the samples and a lot of samples have a very small weight.

Progressive filtering tries to solve this problem by gradually applying the likelihood to the samples, taking care to not lose too much information in each progression step [7], [8]. This is done by introducing a progression parameter γ and splitting the likelihood into a product of wider functions. With $\gamma_r > 0$ and $\Gamma = \sum_{r=1}^{R} \gamma_r = 1$ it leads to the Bayesian filter step

$$f^e(\underline{x}) = \eta \cdot \mathcal{L}_{\hat{\underline{z}}}(\underline{x}) \cdot f^p(\underline{x}) \tag{20}$$

$$= \eta \cdot \prod_{r=1}^{R} \left[\mathcal{L}_{\hat{\underline{z}}}(\underline{x}) \right]^{\gamma_r} \cdot f^p(\underline{x}) \quad . \tag{21}$$

In the first progression step, the prior samples are weighted only with a part of the likelihood $\left[\mathcal{L}_{\hat{\underline{z}}}(\underline{x}; \hat{\underline{y}}) \right]^{\gamma_1}$. The resulting density can then be resampled and the next progression step is applied with the partial likelihood $\prod_{r=1}^{2} \left[\mathcal{L}_{\hat{\underline{z}}}(\underline{x}; \hat{\underline{y}}) \right]^{\gamma_r}$. This is repeated until the complete likelihood is processed, see algorithm 1.

By choosing appropriate progression step γ_r, the amount of information loss due to the weighting can be reduced and the amount of "dead" particle can be minimized. One strategy to select the size of the progression step is to look at the quotient of the samples with the smallest and largest likelihood [22]

$$q = \left[\mathcal{L}_{\hat{\underline{z}}}(\underline{x}_{\min}) / \mathcal{L}_{\hat{\underline{z}}}(\underline{x}_{\max}) \right]^{\gamma_r} \quad . \tag{22}$$

These samples $\underline{x}_{\min}$ and $\underline{x}_{max}$ can either be found by evaluating the likelihood or in the case of a Gaussian likelihood using the Mahalanobis distance. In this case, they will be the samples with the largest and smallest Mahalanobis distance to the mean of the likelihood. The optimal γ_r can now be found by fixing the quotient q and solving (22) for γ_r

$$\gamma_r = \log(q) / \log(\mathcal{L}_{\hat{\underline{z}}}(\underline{x}_{\min}) / \mathcal{L}_{\hat{\underline{z}}}(\underline{x}_{\max})) \quad . \tag{23}$$

Algorithm 1 Proposed progressive filter step.

```
function ProgressiveFilterStep($\underline{\mathcal{X}}$, $\underline{\hat{z}}$, $q$)
    $\Gamma \leftarrow 0.0$
    while $\Gamma < 1.0$ do
        $\gamma_r \leftarrow \log(q)/\log(\mathcal{L}_{\underline{\hat{z}}}(\underline{x}_{\min})/\mathcal{L}_{\underline{\hat{z}}}(\underline{x}_{\max}))$
        if $\Gamma + \gamma_r > 1.0$ then
            $\gamma_r \leftarrow 1.0 - \Gamma$
        end if
        $\underline{w} \leftarrow \left[\mathcal{L}_{\underline{\hat{z}}}(\underline{x})\right]^{\gamma_r}$
        $\underline{\mathcal{X}} \leftarrow$ resamplePCD($\underline{w}$, $\underline{\mathcal{X}}$)
        $\Gamma \leftarrow \Gamma + \gamma_r$
    end while
return $p$
end function
```

Choosing the quotient q is a trade-off between information loss and number of progression steps. If it is set too small, too much information is and therefore the advantage of the progressive filter is lost. If it is set too big, only small changes in the density are allowed in each progression step, resulting in an increased number of steps. This can be somewhat remedied by setting a minimum value for γ_r at the cost of losing more information. In our practical experiment, q was set to 0.5 and a minimum value of $\gamma_r = 0.001$ was chosen.

VI. Evaluation

The progressive filter scheme was implemented with the proposed PCD-based resampling with interpolation between progression steps. The filter performance when using the two selected distances, 2-Wasserstein distance and Cramér-von-Mises distance for resampling as well as the effectiveness of the progression steps are evaluated.

A. One-dimensional Example

We first compare the results of a single filter step of a one-dimensional problem with a cubic measurement equation $h(x) = x^3$. The initial state is set to a Gaussian distribution with mean 0.0 and variance 1.0. From this distribution, 40 samples are randomly drawn as initial particles. Then a filter step with measurement $y = 1.5$ and measurement variance 2.0 is applied. The results of this experiment are depicted in Fig. 3. The posterior distribution without progressive filtering was resampled once after the application of the likelihood with the Cramér-von-Mises distance as optimality measure and gives a relatively good approximation of the true density. Still some clumps and gaps in the samples are present, because of the random nature of the original samples. Both of the progressively filtered posteriors are almost completely without clumps, especially the one that was resampled with the Cramér-von-Mises distance. They smoothly approximate the underlying ground truth density.

B. Two-dimensional Examples

In fig. 1 the progressive filter was applied to a cubic filtering problem in two dimensions with good results. In a second experiment, the performance of the proposed filter was evaluated on the two-dimensional point tracking problem seen in Fig. 5. The ground truth movement starts at the origin $[0.0, 0.0]^\top$ and then continues in a straight line towards $[2.25, 4.5]^\top$. A new measurement of the points position is received and processed at each location of a red diamond. Each measurement consists of the Euclidean distance to three landmarks (black triangles) with additive Gaussian noise with variance 0.01. Between each filter step a prediction step is performed, adding Gaussian noise with variance 0.1 to the particles. For all experiments 20 initial particles are drawn from the initial state distribution using the LCD-procedure implemented in [23]. The problem was run with four different filter setups.

We compared a SIR particle filter (SIR-PF) against a particle filter with PCD-based resampling, but without progression steps (PCD-PF), and against two of the proposed progressive particle filters. One of the progressive filters used the Cramér-von-Mises distance (CVM-PPF) for resampling, while the other one used the 2-Wasserstein distance (WS-PPF). All filters were run 100 times with different random measurements and the average MSE to the ground truth position was recorded, see Fig. 4.

Because of the relatively low number of particles used and the narrow likelihood of the problem, the SIR particle filter exhibits severe sample degeneracy after only two filter steps. This also reflects in the average MSE, which increases with each filter step performed. Exchanging the importance resampling with our PCD-based resampling method yields a slightly more spread out posterior than the SIR-PF, see Fig. 5b. After the likelihood application still only very few samples have a nonzero weight, causing a high tracking deviation from the ground truth.

Both of the progressive filters on the other hand, do not exhibit sample degeneracy and are able to follow the track more accurately. Both filters hover around an average MSE of slightly above 0.1. Cramér-von-Mises and 2-Wasserstein distance seem both to be well suited to be used in the proposed resampling scheme and filter. There is no significant difference in the MSE of the CMV-PPF and WS-PPF.

VII. Conclusion

We introduced a new kind of progressive particle filter making use of PCDs for resampling in-between progression steps. An interpolation strategy was proposed to enable PCDs-based resampling for the weighted Dirac mixtures occurring in the filter step. The Cramér-von-Mises and 2-Wasserstein distance were investigated as one-dimensional metrics to use with PCDs.

The proposed filter was compared to a standard SIR particle filter in a simulated tracking problem. It showed a higher accuracy, resilience toward sample degeneracy, and a visually better approximation of the posterior density than the SIR particle filter. One drawback of the proposed filter is its computational cost, which is significantly higher than that of the SIR particle filter. While importance resampling is linear in the number of samples N, PCD-based resampling is additionally dependent on the number of projection directions V and the distance used in the optimization. The filter step then adds additional cost with each progression step that is applied.

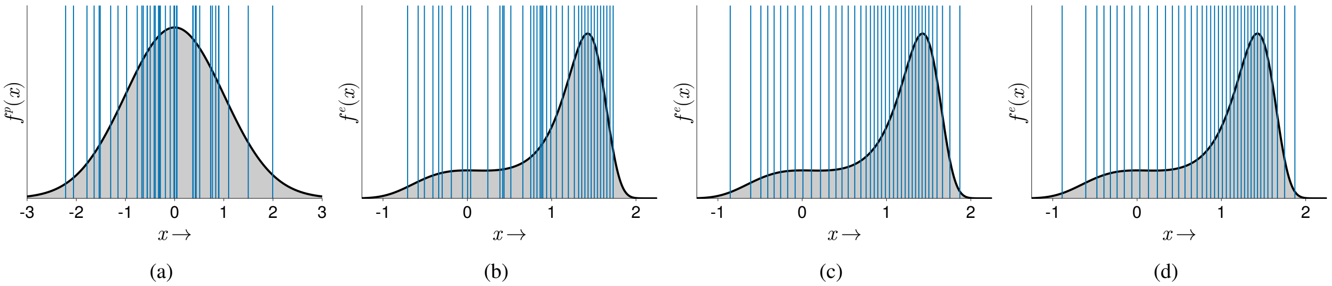


Fig. 3: (a) Prior distribution and estimated posterior distributions of a one-dimensional cubic filtering problem. The filters were run (b) without progressive filtering using the Cramér-von-Mises distance for resampling and with progressive filtering using (c) Cramér-von-Mises distance and (d) 2-Wasserstein distance for resampling. For the progressive filter a maximum ratio of 0.5 between the largest and smallest weight was chosen, which resulted in 12 progression steps. The weighted posterior was interpolated using the midpoints between samples.

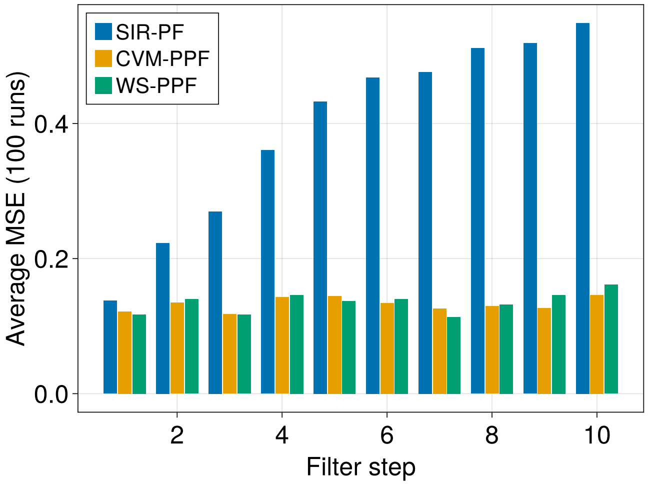


Fig. 4: Average MSE of a SIR particle filter and the proposed progressive particle filters over 100 runs with random measurements on the problem in Fig. 5.

By using a parallelized implementation at least the dependency on the number of directions can be reduced. For example, many intermediate results for each projection direction can be calculated in parallel without any dependencies between them, making some workloads embarrassingly parallel.

Some further improvements to the filter could be made by introducing a deterministic prediction step. This would make the whole filtering process deterministic. Some different solution strategies for the nested minimization problem, when optimizing the Wasserstein distance could lead to faster solutions. Finally, trying some more sophisticated interpolation methods for the one-dimensional subspaces as well as the high-dimensional space could increase the accuracy of the resampling result.

VIII. Acknowledgement

This work has been supported by the project "Jung bleiben mit Robotern" (JuBot) funded by the Carl-Zeiss-Foundation.

References

[1] S. Julier and J. Uhlmann, "Unscented Filtering and Nonlinear Estimation," *Proceedings of the IEEE*, vol. 92, no. 3, pp. 401–422, 2004.

[2] 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), July 2013.

[3] D. Prossel and U. D. Hanebeck, "Dirac Mixture Reduction Using Wasserstein Distances on Projected Cumulative Distributions," in *Proceedings of the 25th International Conference on Information Fusion (Fusion 2022)*, (Linköping, Sweden), July 2022.

[4] T. Li, M. Bolic, and P. M. Djuric, "Resampling Methods for Particle Filtering: Classification, Implementation, and Strategies," *IEEE Signal processing magazine*, vol. 32, no. 3, pp. 70–86, 2015.

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

[6] U. D. Hanebeck, "Deterministic Sampling of Multivariate Densities Based on Projected Cumulative Distributions," in *2020 54th Annual Conference on Information Sciences and Systems (CISS)*, (Princeton, NJ, USA), pp. 1–6, IEEE, Mar. 2020.

[7] U. D. Hanebeck, "PGF 42: Progressive Gaussian Filtering with a Twist," in *Proceedings of the 16th International Conference on Information Fusion*, pp. 1103–1110, 2013.

[8] D. Frisch and U. D. Hanebeck, "Progressive Bayesian Filtering with Coupled Gaussian and Dirac Mixtures," in *IEEE 23rd International Conference on Information Fusion, FUSION 2020, Rustenburg, South Africa, July 6-9, 2020*, pp. 1–8, IEEE, 2020.

[9] N. Oudjane and C. Musso, "Progressive Correction for Regularized Particle Filters," in *Proceedings of the Third International Conference on Information Fusion*, vol. 2, pp. THB2–10, IEEE, 2000.

[10] J. Gall, J. Potthoff, C. Schnörr, B. Rosenhahn, and H.-P. Seidel, "Interacting and Annealing Particle Filters: Mathematics and a Recipe for Applications," *Journal of Mathematical Imaging and Vision*, vol. 28, pp. 1–18, 2007.

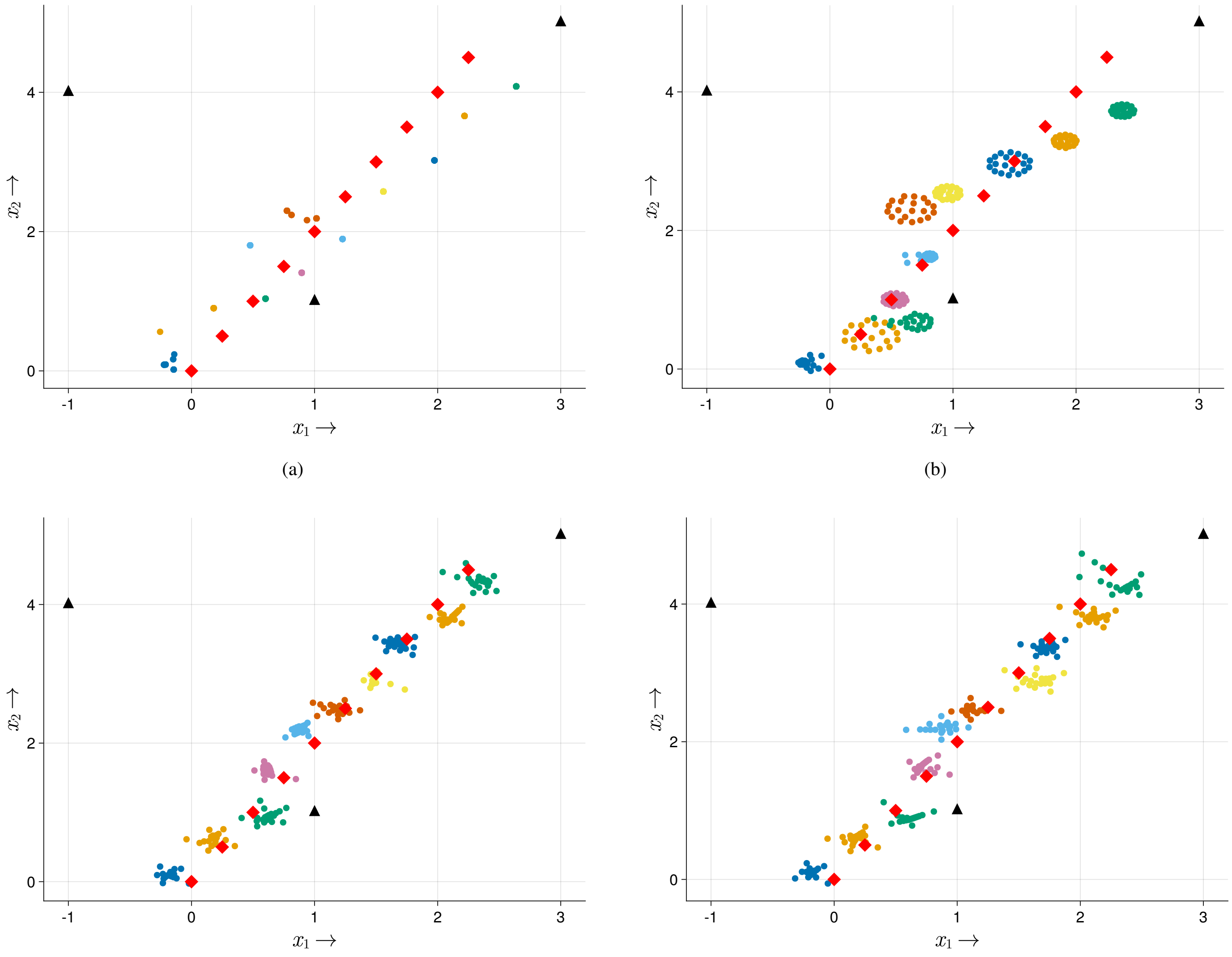
[11] U. D. Hanebeck, "Progressive Bayesian Particle Flows Based on Optimal Transport Map Sequences," in *Proceedings of the 26th International Conference on Information Fusion (Fusion 2023)*, (Charleston, USA), June 2023.

[12] F. Daum and J. Huang, "Particle Flow for Nonlinear Filters with Log-homotopy," in *Signal and Data Processing of Small Targets 2008*, vol. 6969, pp. 414–425, SPIE, 2008.

[13] J. Heng, A. Doucet, and Y. Pokern, "Gibbs Flow for Approximate Transport with Applications to Bayesian Computation," *Journal of the Royal Statistical Society Series B: Statistical Methodology*, vol. 83, no. 1, pp. 156–187, 2021.

[14] U. D. Hanebeck, "FLUX: Progressive State Estimation Based on Zakai-type Distributed Ordinary Differential Equations," *arXiv preprint arXiv:1808.02825*, 2018.

[15] A. Petersen and H.-G. Müller, "Wasserstein Covariance for Multiple Random Densities," *Biometrika*, vol. 106, pp. 339–351, 04 2019.



(a)

(b)

(c)

(d)

Fig. 5: Result of tracking a point following the red diamonds with distance measurements to landmarks (triangles). (a) A standard SIR particle filter suffers from particle degeneracy on this problem. (b) No progression steps are performed and the filter is not able to track the point due to the small overlap between prior distribution and likelihood. (c) Progression steps and resampling with Cramér-von-Mises distance is performed. (d) Progression steps and resampling with 2-Wasserstein distance is performed. In all experiments 20 particles were used.

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

[17] D. Frisch and U. D. Hanebeck, "Deterministic Sampling on the Circle Using Projected Cumulative Distributions," *arXiv:2102.04528 [cs, eess]*, Feb. 2021.

[18] J. Rabin, G. Peyré, J. Delon, and M. Bernot, "Wasserstein Barycenter and Its Application to Texture Mixing," in *Scale Space and Variational Methods in Computer Vision* (A. M. Bruckstein, B. M. ter Haar Romeny, A. M. Bronstein, and M. M. Bronstein, eds.), vol. 6667, pp. 435–446, Berlin, Heidelberg: Springer Berlin Heidelberg, 2012.

[19] S. Kolouri, K. Nadjahi, S. Shahrampour, and U. Şimşekli, "Generalized Sliced Probability Metrics," in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4513–4517, May 2022.

[20] E. Schubert, J. Sander, M. Ester, H. P. Kriegel, and X. Xu, "DBSCAN Revisited, Revisited: Why and How You Should (Still) Use DBSCAN," *ACM Trans. Database Syst.*, vol. 42, jul 2017.

[21] H. D. Sherali and F. L. Nordai, "NP-Hard, Capacitated, Balanced p-Median Problems on a Chain Graph with a Continuum of Link Demands," *Mathematics of Operations Research*, Feb. 1988.

[22] J. Steinbring and U. D. Hanebeck, "Progressive Gaussian Filtering Using Explicit Likelihoods," in *17th International Conference on Information Fusion (FUSION)*, pp. 1–8, 2014.

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