Deterministic Proposal Sampling 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@kit.edu

***Abstract*—Particle filters are an important class of algorithms for Bayesian estimation. One of their drawbacks is the so-called particle degeneration where only very few particles with a meaningful weight remain after the filter step. This effect is typically remedied by regularly resampling the particles, yielding a set of equally weighted particles. This paper investigates an approach to deterministically sample particles from the proposal distribution in such a way to automatically have equally weighted particles at the end of the filter step. The proposed method is first motivated and presented for the one-dimensional case. Using the Radon transform and projected cumulative distributions, the one-dimensional algorithm is extended to multivariate problems. Some examples of the usefulness of the proposed algorithm are also shown.**

***Index Terms*—Particle Filter, upsampling, deterministic sampling, Radon transform.**

I. Introduction

Bayesian state estimation combines uncertain prior information about a system with noisy measurements to produce an improved estimate of the state of the system. Closed-form solutions to this problem are available only for a limited number of systems. For example, with Gaussian prior distributions and linear measurement equations, the Kalman filter emerges as the optimal solution. When dealing with general non-linear systems and arbitrary priors, different approximations can be made to arrive at closed-form approximate solutions. Two popular Kalman filter variants, the extended Kalman filter and the unscented Kalman filter, for example, only take the first two moments into account for the estimation and linearize the measurement function.

The particle filter goes a different route from these by approximating the posterior distribution with weighted particles, which are essentially samples of the underlying continuous posterior distribution. With an increasing number of particles N, it can be shown to converge asymptotically to the exact solution. Common particle filters that use random samples have a convergence rate for the error in $\mathcal{O}(N^{-1/2})$ [1]. This may be improved up to $\mathcal{O}(N^{-1})$ by using quasi-random sequences [2][3] or other kinds of deterministic samples [4][5].

A common challenge for all particle filters is the so-called sample degeneration, where the number of particles contributing to the solution decreases over time. This effect is typically remedied by regularly resampling the particles to remove low-weight particles while keeping the overall number of particles the same.

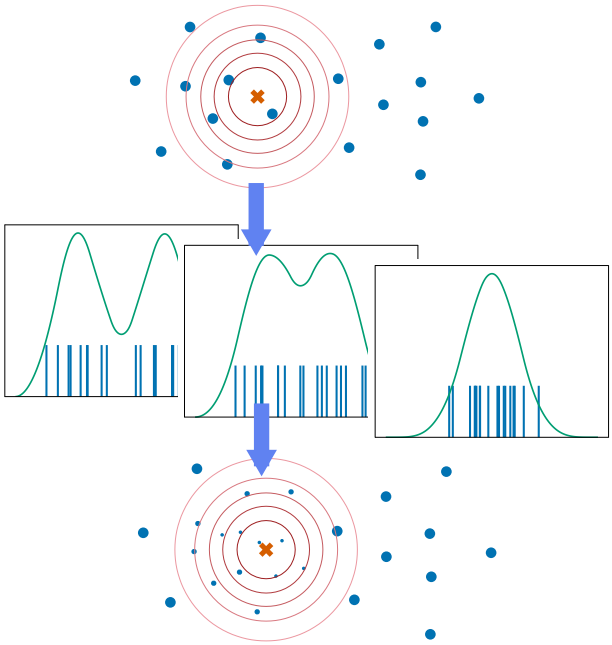


Fig. 1: Core idea of the proposed algorithm. Equally weighted particles of a proposal distribution (blue) are transformed with the Radon transform. The proposal density is estimated in each of the resulting one-dimensional sub-spaces. The number of particles is increased in areas of high likelihood, in this case around the measurement located at the red cross.

Progressive particle filters and particle flow filters are based on the idea of continuously morphing a prior distribution of particles into the posterior distribution. Particle flow filters formulate this as differential equations that have to be solved to find the posterior [6]. Progressive particle filters instead divide this transformation into a sequence of smaller transformations that are applied iteratively, often with some kind of resampling step in between [7][8]. In [9] a progressive particle filter with a deterministic resampling scheme based on Projected Cumulative Distributions (PCDs) was proposed.

Another method of combating sample degeneration is to consider the future likelihood update already during sampling

from the prior distribution. Auxiliary particle filters [10] fall into this category. They are based on the idea that no prior samples are needed in low-likelihood regions as they would not be resampled after the update anyway. Therefore, an auxiliary distribution is constructed and sampled to prefer samples from high-likelihood regions.

This paper proposes a novel deterministic sampling scheme to sample a proposal distribution for use in a particle filter. Unlike random sampling, the weights assigned to the samples can be selected freely when using deterministic sampling. This additional degree of freedom allows to increase the number of particles in areas where the posterior particles are expected to have a large weight. It is proposed to choose the weights of the proposal particles inversely proportional to the weighting factor used in the filter step. In case of the bootstrap particle filter, this factor coincides with the measurement likelihood. This weighting scheme leads to posterior particles that are closer to equally weighted than when using random samples and increases the number of effective particles in the filter. As the proposal distribution is often only available in the form of samples, a novel upsampling method is developed to adjust the number of particles and their weights. It builds on the idea presented in [9] where multivariate distributions are upsampled using estimated PCDs, which were first presented in [5]. Instead of piecewise constant functions as in [9], piecewise polynomials similar to [11] are used for density estimation. The paper first gives a brief introduction to particle filtering and the reasoning behind choosing the particle weights. The proposed sampling scheme is then derived for one-dimensional problems, before it is extended to multivariate densities using PCDs. An application of the resulting algorithm to a Gaussian mixture prior is visualized in Fig. 1. Finally, the improvement in the number of effective posterior particles, when using these pre-weighted particles is shown on some simulated examples.

II. Problem Formulation

Bayesian estimation is concerned with calculating the posterior probability density function (pdf) of the state of a system, given a prior pdf, a system model, a measurement model, and some measurements. For the purpose of this paper, the system model is the identity without any corruption by noise and only one timestep is considered. In other words, the focus lies on a single filter step given the prior pdf $f(\underline{x})$ of the state $\underline{x}$, a measurement $\underline{y}$, and a likelihood function $f(\underline{y}|\underline{x})$. The underbar notation $\underline{x}$ denotes the variable as a vector. The posterior pdf $f(\underline{x}|\underline{y})$ can then be calculated by applying Bayes rule and normalizing the result

$$f(\underline{x}|\underline{y}) = \eta \cdot f(\underline{x}) \cdot f(\underline{y}|\underline{x}) \ . \tag{1}$$

The normalization factor η ensures that $f(\underline{x}|\underline{y})$ integrates to one and is therefore a valid pdf.

In the bootstrap particle filter, the prior and posterior densities are represented by weighted samples of the respective distributions called particles. The application of Bayes rule in this case is straightforward by taking the prior distribution

$$f(\underline{x}) = \sum_{i=1}^{N} w_i^p \, \delta(\underline{x} - \underline{x}_i) \ , \tag{2}$$

expressed as a Dirac mixture density, and multiplying the prior particle weights w_i^p with the likelihood function

$$\begin{aligned} f(\underline{x}|\underline{y}) &= \eta \sum_{i=1}^{N} w_i^p \, f(\underline{y}|\underline{x}) \cdot \delta(\underline{x} - \underline{x}_i) \\ &= \eta \sum_{i=1}^{N} w_i^e \, \delta(\underline{x} - \underline{x}_i) \ . \end{aligned} \tag{3}$$

Note that only the particle weights change from prior to posterior weights w_i^e, while the particle positions stay the same and all new information from the measurement is encoded in the weights.

This filtering scheme is known as the bootstrap particle filter. Its simplicity makes it very easy to use but leads to considerable particle degeneration when the prior and posterior distributions are far apart. When placed in a general particle filter framework [12], it uses the standard proposal or transition density as the so-called proposal distribution. It is identical to the prior pdf $f(\underline{x})$ here, as the considered system does not evolve. By choosing a well-suited proposal distribution for a given problem, the performance of a particle filter can be significantly improved. It should typically be as close as possible to the posterior distribution while still being fast to sample.

Using a general proposal density $\pi(\underline{x}|\underline{y})$, the weight update becomes

$$w_i^e = w_i^p \, \frac{f(\underline{y}|\underline{x}_i) \cdot f(\underline{x}_i)}{\pi(\underline{x}_i|\underline{y})} \ . \tag{4}$$

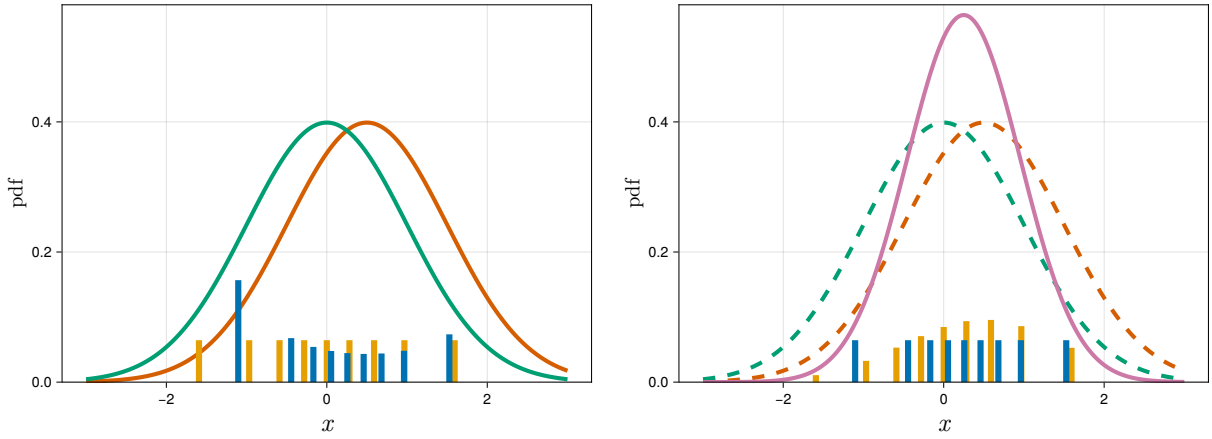
By identifying the proposal distribution $\pi(x|y)$ as a prior distribution $\tilde{f}(x)$ and the weight update factor in (4) as a likelihood function

$$\tilde{f}(\underline{y}|\underline{x}_i) = \frac{f(\underline{y}|\underline{x}_i) \cdot f(\underline{x}_i)}{\pi(\underline{x}_i|\underline{y})} \tag{5}$$

this general update can also be interpreted as the multiplication of a prior distribution with a likelihood as in (3). This paper uses the bootstrap particle filter as an example to derive the proposed methods, but, through the identification above, these can be easily extended to different proposal distributions.

Sampling from the prior distribution is usually done with random sampling methods like Markov-Chain-Monte-Carlo. This results in particles, where each particle has the same weight. When considering deterministic sampling from a distribution through optimization, as in [13] or [5], an additional degree of freedom is available because the weight of each particle can be chosen individually. The optimal deterministic particles drawn from a prior distribution have weights according to

$$w_i^p = \frac{1}{f(\underline{y}|\underline{x}_i)} \ , \tag{6}$$



(a) before filter step

(b) after filter step

Fig. 2: Comparison of equally weighted particles (yellow) and particles weighted according to the inverse likelihood function (blue) before and after the filter step in a bootstrap particle filter. The particles were drawn deterministically from the proposal distribution (green) and the blue samples were weighted with the inverse of the likelihood function (red). The ground truth posterior distribution is shown in pink.

the inverse of the likelihood function. This assumes that the likelihood is larger than zero at all particle locations, requiring a reasonable overlap between the prior distribution and the likelihood function. After multiplication of the weights in (6) with the likelihood function and subsequent normalization, the posterior particles are equally weighted. This makes resampling unnecessary and eliminates sample degeneration.

One way to draw such samples from a one-dimensional distribution that is available in closed form is to minimize the Cramér-von-Mises distance

$$D(f,g) = \int_{-\infty}^{\infty} \big(F(x) - G(x)\big)^2 \,\mathrm{d}x \tag{7}$$

between the underlying continuous distribution and a Dirac mixture density representing the samples with cumulative distribution function (cdf) $F(x)$ and $G(x)$. Specifically, the sample positions are obtained by solving the optimization problem

$$\underset{x_i}{\arg\min} \int_{-\infty}^{\infty} \left(F(x) - \sum_{i=1}^{N} \frac{\big(f(y|x_i)\big)^{-1}}{\sum_{j=1}^{N} \big(f(y|x_j)\big)^{-1}} H(x - x_i) \right)^2 \mathrm{d}x \tag{8}$$

with the Heaviside step function

$$H(x) = \begin{cases} 0 & x < 0 \\ \frac{1}{2} & x = 0 \\ 1 & x > 0 \end{cases} . \tag{9}$$

Given that the integral exists and converges (see [14] for more details) and given a suitable starting solution that avoids very small and very large likelihood values, this problem can be solved with standard unconstrained optimization algorithms like L-BFGS.

An example of particles in a bootstrap particle filter sampled and weighted in this way is shown in blue in Fig. 2. Compared to the equally weighted particles in yellow, the blue particles are denser, where the posterior is larger, and are equally weighted after the filter step. Unfortunately, this approach breaks down relatively fast in practical applications. As the likelihood goes to zero in low-likelihood regions, its inverse grows to infinity. This in turn leads to inverses close to zero in high-likelihood regions when normalizing back to a sum of one. The solution to (8) would not be useful for filtering in this case.

III. Weight Selection Scheme

As discussed above, exact weighting of the particles with the inverse likelihood (6) is not really practical. However, it is still advantageous to reduce the variance of the posterior weights as much as possible and get them close to being equally weighted [10]. This means having densely sampled particles with small weights in areas of high likelihood and avoiding the problems caused by low likelihood regions discussed above.

In the following, an algorithm is introduced that selects particle weights roughly proportional to the inverse likelihood. The starting point of the proposed algorithm is a number M of equally weighted samples of the prior distribution. The likelihood function is evaluated at each sample position. Based on these provisional weights, each particle is replaced with a number of particles according to Algorithm 1.

This algorithm iteratively adds a new particle at the location of the particle with the current largest weight. The current weights are calculated by dividing the original weight of each particle by the number of particles at that location. New particles are added, up to a maximum number of particles N. Overall this increases the number of particles from M to N by splitting particles with high weights and leaving particles with small weights alone. After this procedure, the new particles

Algorithm 1 Particle replacement algorithm to increase the number of particles from M to N. The $\arg\max$ function returns the index of the largest entry in the vector and $\oslash$ denotes element-wise division.

Input vector of M particle weights $\underline{w}$
Output vector of number of replacement particles $\underline{n}$

$\underline{n} \leftarrow \underline{1} \in \mathbb{R}^M$
while $\mathrm{sum}(\underline{n}) < N$ **do**
  $i \leftarrow \arg\max(\underline{w} \oslash \underline{n})$
  $\underline{n}[i] \leftarrow \underline{n}[i] + 1$
end while
return $\underline{n}$

are spread by adding a small perturbation and used as initial solution to the following optimization problem

$$\arg\min_{x_i} \int_{-\infty}^{\infty} \left(F(x) - \sum_{i=1}^{N} w_i \, H(x - x_i) \right)^2 \mathrm{d}x \ . \tag{10}$$

This is similar to (8), but with the weights w_i determined by Algorithm 1. The solution can then be obtained through inverse transform sampling of $F(x)$ [11].

IV. One-dimensional Density Estimation

In real-world applications, the prior distribution is often not available as a closed-form function, but can only be sampled or is already available in sampled form. It has the form of a Dirac mixture distribution as in (2). The cdf of this distribution is a sum of step functions. Substituting such a function for $F(x)$ in (10) and solving the optimization problem often leads to undesired results, for example clumped up particles. The reason for this is that the particles are drawn from the discrete Dirac mixture density, when they should actually be drawn from the unknown underlying continuous distribution.

This problem was also discussed in [9] and solved there by interpolating the Dirac mixture density with a piecewise constant function before sampling. In [11], a different density estimation algorithm was proposed based on polynomial spline segments instead of a constant function. While this seems to give relatively accurate estimation results, it only works in one dimension and introduces some hard constraints based on the assumption that the samples were drawn by minimizing the Cramér-von-Mises distance.

Nevertheless, this method, in a slightly modified form, is used here to estimate the continuous pdf of the underlying prior distribution based on the initial particles. A very important adaptation that is done is to soften the mentioned hard constraints.

The estimated density $\hat{f}(x)$ is parametrized as a sum of S squared piecewise polynomials $r_s(x)$

$$\hat{f}(x) = \sum_{s=1}^{S} \hat{f}_s(x) = \sum_{s=1}^{S} r_s^2(x) \tag{11}$$

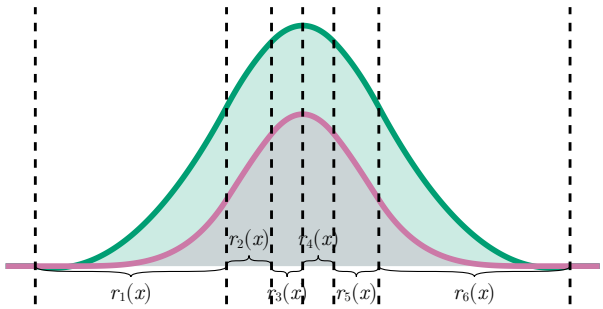


Fig. 3: Density parametrization based on piecewise squared polynomial splines. Estimated density $\hat{f}(x)$ in pink and root representations $r_s(x)$ in green.

where

$$r_s(x) = \begin{cases} \sum_{i=1}^{D} c_{s,i} x^{D-1} & \text{if } x > \xi_s \wedge x \leq \xi_{s+1} \\ 0 & \text{else} \end{cases} \ . \tag{12}$$

with the degree of the polynomial D and the nodes $\xi_1, \dots \xi_{S+1}$. These nodes are sorted in ascending order and chosen such, that all samples are covered. Continuity constraints on the function value and first derivative are added at each node to ensure a continuous estimate. The representation is visualized in Fig. 3.

This parameterization inherently ensures non-negativity of the resulting density, instead of needing to add explicit constraints on the polynomial coefficients. It also leads to easily computable cdfs due to the ease of integrating polynomials. The density estimation itself is formulated as an optimization problem to find the optimal coefficients of the polynomials $r_s(x)$. In [11] the Fisher information number

$$I\Big(\hat{f}\big(r_s(x)\big)\Big) = \sum_{s=1}^{S} \int_{-\infty}^{\infty} \big(r_s'(x)\big)^2 \, \mathrm{d}x \tag{13}$$

is used as the objective function and additional constraints are added to force the estimated density to intersect the empirical cdf exactly at each sample location. The Fisher information number can be interpreted as a measure of the roughness of a function and is used as a regularizer to find a unique solution [15]. The constraints were chosen to motivate density estimation as the inverse operation to deterministic sampling. They force the estimated cdf to follow the empirical cdf very strictly. This is often too restrictive if the samples do not follow the assumptions exactly. Therefore, they are replaced by the objective to minimize the squared distance between the empirical and estimated cdf at each sample location

$$\hat{D}(\hat{f}, X) = \sum_{i=1}^{M} \big(\hat{F}(x_i) - H(x - x_i)\big)^2 \tag{14}$$

for a set of samples $X = \{x_1, \dots, x_M\}$. The new optimization objective is the weighted sum of this distance $\hat{D}(\hat{f}, X)$ and the

Fisher information number $I(\hat{f})$. The complete minimization problem for density estimation is

$$\underset{\hat{f}(r_s(x;\underline{c}_s))}{\arg\min}\ \hat{D}(\hat{f}, X) + \lambda I(\hat{f}) \tag{15}$$

$$\begin{aligned} \text{s.t.}\quad & \hat{f}_j(\xi_j) = \hat{f}_{j+1}(\xi_j) \quad \forall j = 2, \dots, S \\ & \hat{f}'_j(\xi_j) = \hat{f}'_{j+1}(\xi_j) \quad \forall j = 2, \dots, S \\ & \int_{-\infty}^{\infty} \hat{f}(x)\,\mathrm{d}x = 1 \end{aligned} \tag{16}$$

with a regularization factor λ, continuity constraints at the node locations ξ_j, and the integration constraint to get a valid pdf. The solution to this problem is the distribution with the smoothest cdf that the given samples could reasonably be drawn from according to $\hat{D}$. Increased smoothness can be traded for a larger distance between samples and continuous estimate through the hyperparameter λ. In practice, this problem can be solved, for example, with an interior point algorithm that is capable of handling nonlinear objectives and equality constraints.

V. Multivariate extension with Projected Cumulative Distributions

The estimation algorithm described above as well as deterministic sampling using cdfs does unfortunately not generalize to high-dimensional problems in a straightforward way. One reason for this is that cdfs are not uniquely defined in more than one dimension, necessitating alternative approaches [16]. More importantly, it is also difficult to define multivariate and smooth piecewise polynomial splines analogous to (11). The Radon transform [17] is a tool to break down high-dimensional problems into (infinitely) many one-dimensional problems. The so-called sliced Wasserstein distance [18] is a popular variant of this method that has gained traction as a loss function in approaches based on artificial neural networks [19][20]. The Radon transform has also been used for deterministic sampling [5], [21]. In [22], the derivation of generalized sliced probability metrics from one-dimensional metrics is discussed.

Starting with a d-dimensional probability distribution with pdf $f(\underline{x})$, its projection onto the line given by the unit vector $\underline{u} \in \mathbb{S}^{d-1}$ has the pdf

$$f(r|\underline{u}) = \int_{\mathbb{R}^d} f(\underline{t}) \cdot \delta(r - \underline{u}^\top \underline{t})\,\mathrm{d}\underline{t}\ . \tag{17}$$

The Radon transform is obtained when considering the projections onto all directions $\underline{u} \in \mathbb{S}^{d-1}$.

A distance measure between multivariate distributions with pdfs $f(\underline{x})$ and $g(\underline{x})$ can then be derived by averaging the distances of all one-dimensional projections

$$D_d^\infty(f, g) = \frac{1}{A_d} \int_{\mathbb{S}^{d-1}} D_1\big(f(r|\underline{u}), g(r|\underline{u})\big)\,\mathrm{d}\underline{u}\ . \tag{18}$$

A_d is the surface of the hypersphere $\mathbb{S}^{d-1}$. In practical implementations, the number of projections has to be limited to a finite value, replacing the integral in (18) with a sum

$$D_d^L(f, g) = \frac{1}{L} \sum_{l=1}^{L} D_1\big(f(r|\underline{u}_l), g(r|\underline{u}_l)\big) \tag{19}$$

Algorithm 2 Proposed algorithm to sample N particles weighted approximately inversely to the likelihood function from a prior distribution initially represented by M particles.

Input M particles of prior distribution
Output N new particles

1: Evaluate likelihood for each particle yielding $\underline{w}$
2: Determine new weights using Algorithm 1 and $\underline{w}$
3: Project particle positions onto each direction $\underline{u}_l$
4: Fit polynomial splines $\hat{f}(r|\underline{u}_l)$ to projected particles
5: Draw N new particles by solving (20)

and using unit vectors $\underline{u}_l$ uniformly sampled from $\mathbb{S}^{d-1}$. Choosing the Cramér-von-Mises distance (7) for D_1 gives the modified Cramér-von-Mises distance applicable to multivariate distributions suggested in [5].

This distance measure enables deterministic sampling of distributions where only the one-dimensional cdfs of the projected distributions are needed. These can be estimated by projecting the samples and applying the algorithm for one-dimensional density estimation described in Section IV.

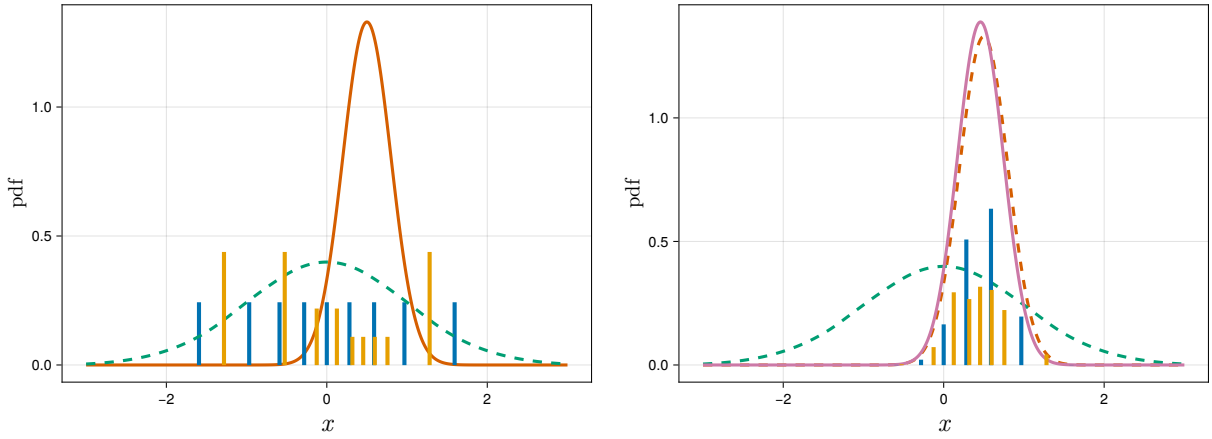
All that remains is to put the pieces together into the final algorithm for multivariate distributions Algorithm 2. Starting with M samples of an otherwise unknown prior distribution, the quantity of target samples N, and a way to evaluate the likelihood function, the number of replacements for each particle and their weights w_i^p are determined according to Algorithm 1. The initial M samples are projected onto a number L of unit vectors sampled from $\mathbb{S}^{d-1}$ where d is the dimension of the state space. For each of the projections, the according density $\hat{f}(r|\underline{u}_l)$ is estimated by solving the minimization problem (16). The corresponding estimated cdfs $\hat{F}(r|\underline{u}_l)$ and the newly determined particle weights are then used to set up the following optimization problem to find the positions $\underline{x}_i$ of N new particles

$$\underset{\underline{x}_i}{\arg\min} \frac{1}{L} \int_{-\infty}^{\infty} \left(\hat{F}(r|\underline{u}_l) - \sum_{i=1}^{N} w_i H(r - \underline{u}_l^\top \underline{x}_i) \right)^2 \mathrm{d}x\ . \tag{20}$$

The solution to this problem is obtained using Newton's method, as suggested in [5].

The complete algorithm shown in Algorithm 2 takes M particles propagated through the prediction step of a filter and a measurement as input and returns equally weighted posterior particles. It can be used as a drop-in replacement for the particle filter step by setting $N = M$, otherwise an additional reduction step is necessary to reduce the number of particles back to N after the filter step. In contrast to conventional particle filter steps that resample the particles after weighting them with the likelihood, a higher resolution of particles is achieved in areas of high likelihood. This results in a generally more accurate representation of the posterior distribution.

The cost of a better accuracy is an increase in computational complexity, especially compared to simple resampling schemes such as importance resampling. The three last lines of Algorithm 2 depend on the number of projection directions



(a) before weighting

(b) after weighting

Fig. 4: Comparison between equally weighted prior particles in blue and particles drawn with the proposed method in yellow. Both sample sets were drawn from the prior distribution (dashed green) and weighted with the inverse of the likelihood (red) to get the posterior distribution. The analytic posterior is drawn in pink.

L. This number can be seen as a way to trade accuracy and speed. Systems with a higher number of state dimensions and more intricate state distributions also require more projections to represent these. In line 4, an optimization problem is solved for each of the projections, where the objective function scales linearly with the number of initial particles, giving a runtime complexity of $O(LM)$. To solve the optimization in line 5 of Algorithm 2, a Newton step is calculated in each projection to iteratively update the position of the particles. This gives a complexity of $O(LN)$ depending on the number of projection directions and the number of new particles. Something that was omitted in these considerations is the runtime of the optimization algorithms used, which comes in addition to the above.

VI. Experiments

A. One-dimensional Sanity Check

To show the principal workings of the proposed algorithm, a bootstrap particle filter with one-dimensional states and measurements is considered. The standard Gaussian $\mathcal{N}(0, 1)$ is used as prior distribution and the likelihood function $p(y|x)$ is also Gaussian with mean 0.5 and standard deviation 0.3. Fig. 4a shows the situation before the multiplication of the particle weights with the likelihood. The blue samples are equally weighted and were deterministically drawn from the prior density (dashed green) using (10). The yellow samples were drawn according to the proposed algorithm starting with five deterministically drawn samples from the prior density. The algorithm has no knowledge of the actual underlying density, and only the samples and the likelihood function (red) were given as input. The resulting particles are approximately weighted with the inverse of the inverse likelihood. When comparing the according posteriors in Fig. 4b it can be seen that there are more yellow samples with significant weights than blue samples. Additionally, the yellow samples are closer to equally weighted than the blue ones. This suggests that the proposed algorithm works as expected.

To quantify the effect of the improved particle sampling scheme, the number of effective particles

$$N_{\text{eff}} = \sum_{i=1}^{N} \frac{1}{w_i^2} \tag{21}$$

is calculated and compared to equally weighted particles. This value is N when all particles have the same weight and goes to one, as more particles have weights close to zero. In the example, we have $N_{\text{eff}} \approx 3.2$ for the equally weighted particles, which increases to $N_{\text{eff}} \approx 5.7$ when using the improved particles. This further underpins the usefulness of the presented procedure.

B. Two-dimensional Distance Measurement

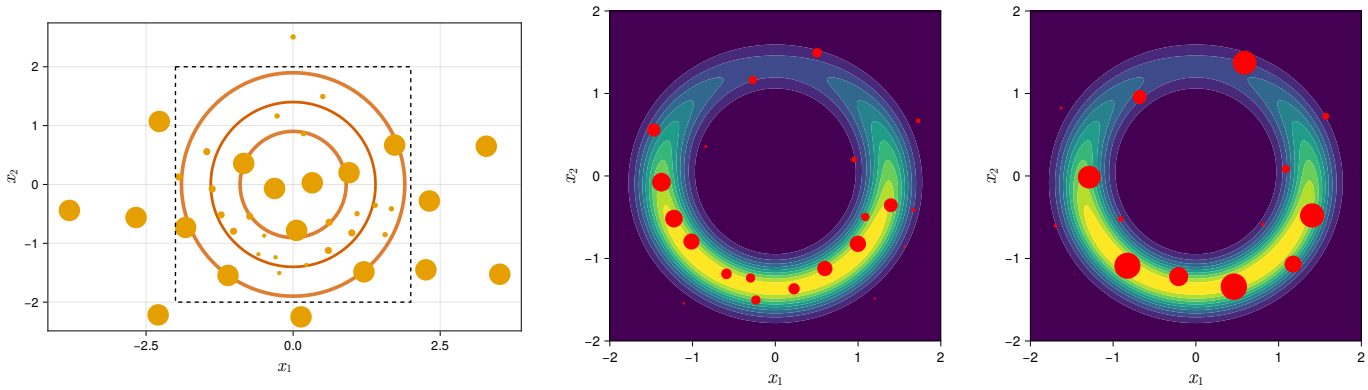
Next, a two-dimensional bootstrap particle filter with distance measurements is examined. The prior distribution is the two-dimensional Gaussian distribution

$$f(\underline{x}) = \mathcal{N}\left(\begin{bmatrix} 0 \\ -1 \end{bmatrix}, \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}\right) \ . \tag{22}$$

The likelihood function is based on a Euclidean distance measurement and given as

$$f(y|\underline{x}) = \mathcal{N}(y; \|\underline{x}\|_2, 0.2) \ . \tag{23}$$

Fig. 5a shows the prior pre-weighted particles sampled with the proposed method and the mean and 3σ-interval of the likelihood function. The size of the dots denotes the particle weight. The starting point were 24 equally weighted prior particles (not shown) which were increased to 40 weighted particles. The posterior particles are overlaid onto the ground truth posterior distribution in Fig. 5b. This result is compared to a posterior that was calculated using equally weighted prior samples Fig. 5c. The proposed method leads to more particles



(a) Prior Samples and likelihood

(b) Posterior distribution (pre-weighted)

(c) Posterior distribution (equally weighted)

Fig. 5: Example application of the proposed method in the context of a bootstrap particle filter with a distance measurement. The prior samples (a) are pre-weighted according to the inverse likelihood. The mean and rough spread of the likelihood function is drawn in red. There are more posterior particles with a significant weight in (c), where the samples in (a) were used, as in (b) where equally weighted prior samples were used. The dashed square in (a) marks the the part of the plot shown in (b) and (c).

with a significant weight remaining in the posterior distribution. This is also evident in the number of effective particles, which goes from 10.5 to 17 when using the improved sampling.

C. Scaling to Higher Dimensions

To investigate the scaling of the algorithm to higher dimensions in a bootstrap particle filter. A d-dimensional Gaussian mixture distribution with two components is used as prior distribution. Both components have the identity matrix as covariance matrix. The first component is centered in the origin, while the mean of the second component has its first coordinate offset by 2. The likelihood function is a d-dimensional Gaussian with the same mean as the second component of the prior distribution and a diagonal covariance matrix with variance 0.5 in every dimension.

The proposed particle sampling method is compared to equally weighted particles sampled with the method in [5]. For both methods, 1000 random unit vectors were used as projection directions. The number of effective particles in the posterior distribution was calculated Fig. 6. As expected, N_{eff} decreases with an increasing number of dimensions, as on average fewer particles lie in the relevant regions. The curves for both methods are more or less parallel, while the proposed method has more effective particles in all dimensions. This suggests that it should provide more accurate results than equally weighted samples, but cannot overcome the curse of dimensionality.

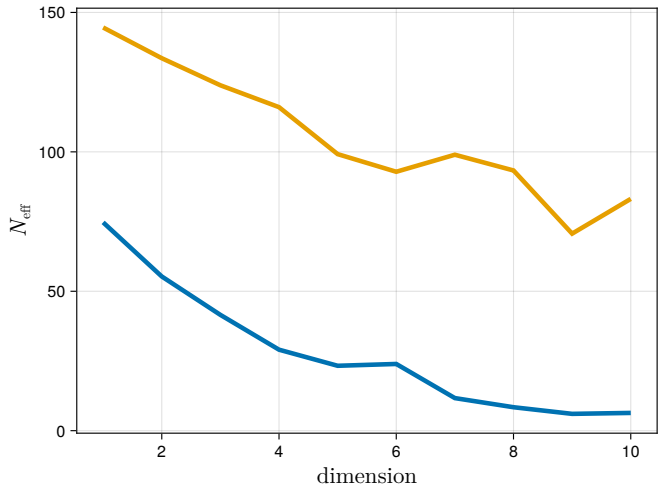


Fig. 6: Number of effective particles (N_{eff}) over the dimensions for the setup described in Section VI-C. Value for equally weighted particles in blue and for particles drawn with the proposed method in yellow.

VII. Conclusion

In this paper, a novel algorithm was introduced to enhance given samples of a prior distribution by considering particle weights. The particles are deterministically sampled by solving an optimization problem. This makes it possible to select the particle weights inversely proportional to the likelihood function that is applied in the filter step. This method of pre-weighting the particles is shown to increase the number of effective particles in the posterior distribution, when compared to equally weighted particles.

The proposed algorithm starts with some number of equally weighted particles and iteratively splits them in high-weight regions. Future research may look at methods to additionally merge particles in low-weight regions. It is in general desirable to get closer to the optimal pre-weighted particles, that lead to equally weighted posterior particles.

VIII. Acknowledgment

This work has been supported by the Carl Zeiss Foundation under the JuBot project.

References

[1] D. Crisan and A. Doucet, "A survey of convergence results on particle filtering methods for practitioners," *IEEE Transactions on Signal Processing*, vol. 50, no. 3, pp. 736–746, 2002.

[2] D. Guo and X. Wang, "Quasi-Monte Carlo filtering in nonlinear dynamic systems," *IEEE Transactions on Signal Processing*, vol. 54, no. 6, pp. 2087–2098, 2006.

[3] P. Fearnhead, "Using random quasi-Monte-Carlo within particle filters, with application to financial time series," *Journal of Computational and Graphical Statistics*, vol. 14, no. 4, pp. 751–769, 2005.

[4] D. Frisch and U. D. Hanebeck, "The generalized Fibonacci grid as low-discrepancy point set for optimal deterministic Gaussian sampling," *Journal of Advances in Information Fusion*, vol. 18, no. 1, pp. 16–34, Jun. 2023.

[5] U. D. Hanebeck, "Deterministic sampling of multivariate densities based on Projected Cumulative Distributions," in *Proceedings of the 54th Annual Conference on Information Sciences and Systems (CISS 2020)*, Princeton, New Jersey, USA, Mar. 2020.

[6] 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., International Society for Optics and Photonics, vol. 7697, SPIE, 2010, p. 769 704.

[7] J. Steinbring and U. D. Hanebeck, "Progressive Gaussian filtering using explicit likelihoods," in *17th International Conference on Information Fusion (FUSION)*, IEEE, 2014, pp. 1–8.

[8] N. Oudjane and C. Musso, "Progressive correction for regularized particle filters," in *Proceedings of the Third International Conference on Information Fusion*, IEEE, vol. 2, 2000, THB2–10.

[9] D. Prossel and U. D. Hanebeck, "Progressive particle filtering using Projected Cumulative Distributions," in *Proceedings of the combined IEEE 2023 Symposium Sensor Data Fusion and International Conference on Multisensor Fusion and Integration (SDF-MFI 2023)*, Bonn, Germany, Nov. 2023.

[10] M. K. Pitt and N. Shephard, "Filtering via simulation: Auxiliary particle filters," *Journal of the American statistical association*, vol. 94, no. 446, pp. 590–599, 1999.

[11] D. Prossel and U. D. Hanebeck, "Spline-based density estimation minimizing Fisher information," in *Proceedings of the 27th International Conference on Information Fusion (FUSION 2024)*, Venice, Italy, Jul. 2024.

[12] C. Snyder, "Particle filters, the "optimal" proposal and high-dimensional systems," in *Proceedings of the ECMWF Seminar on Data Assimilation for atmosphere and ocean*, Citeseer, 2011, pp. 1–10.

[13] 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.

[14] A. Barbiero and A. Hitaj, "Discrete approximations of continuous probability distributions obtained by minimizing Cramér-von Mises-type distances," *Statistical Papers*, vol. 64, no. 5, pp. 1669–1697, 2023.

[15] I. J. Good and R. A. Gaskins, "Nonparametric roughness penalties for probability densities," *Biometrika*, vol. 58, no. 2, pp. 255–277, 1971.

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

[17] J. Radon, "Über die Bestimmung von Funktionen durch ihre Integralwerte längs gewisser Mannigfaltigkeiten," *Berichte über die Verhandlungen der Königlich-Sächsischen Gesellschaft der Wissenschaften zu Leipzig. Mathematisch-Physische Klasse*, vol. 69, pp. 262–277, 1917.

[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: Third International Conference, SSVM 2011, Ein-Gedi, Israel, May 29–June 2, 2011, Revised Selected Papers 3*, Springer, 2012, pp. 435–446.

[19] I. Deshpande, Z. Zhang, and A. G. Schwing, "Generative modeling using the sliced Wasserstein distance," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 3483–3491.

[20] E. Heitz, K. Vanhoey, T. Chambon, and L. Belcour, "A sliced Wasserstein loss for neural texture synthesis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 9412–9420.

[21] D. Prossel and U. D. Hanebeck, "Dirac mixture reduction using Wasserstein distances on Projected Cumulative Distributions," en, in *2022 25th International Conference on Information Fusion (FUSION)*, Linköping, Sweden: IEEE, Jul. 2022, pp. 1–8.

[22] 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)*, IEEE, 2022, pp. 4513–4517.