Optimized Auxiliary Particle Filters

Anonymous Author
Anonymous Institution

Abstract

Auxiliary particle filters (APFs) are a class of sequential Monte Carlo (SMC) methods for Bayesian inference in state-space models. In their original derivation, APFs operate in an extended space using an auxiliary variable to improve the inference. Later works have re-interpreted APFs from a multiple importance sampling perspective. In this perspective, the proposal is a mixture composed of kernels and weights that are selected by taking into account the latest observation.
In this work, we further exploit this perspective by proposing an online, flexible framework for APFs that adapts the mixture proposal by convex optimization and allows for a controllable computational complexity. We minimize the discrepancy between the proposal and the filtering distribution at a set of relevant points, which are chosen by leveraging the structure of SMC. We compare our method to state-of-the-art particle filters, showing better performance in challenging and widely used dynamical models.

1 INTRODUCTION

State-space models (SSMs) allow a mathematical description of complex dynamical systems which are very relevant in computational statistics, machine learning and signal processing, among many other fields [1, 2]. Particle filters (PF) or sequential Monte Carlo methods (SMC) are the *de facto* family of algorithms to perform learning tasks in virtually any SSM, e.g., filtering, prediction, or parameter estimation [3, 4, 5, 6]. PFs have been used extensively for solving complex real-world problems in robotics [7], object tracking [8, 9, 10], image processing [11, 12], financial econometrics [13, 14], and even modelling of spread of infectious diseases [15, 16, 17]. PFs are also used for problems beyond the classical SSM setting. For instance, they have been recently applied in reinforcement learning [18, 19, 20], generative modelling [21, 22, 23], and more generally for approximate Bayesian inference in large probabilistic models [24, 25, 26, 27].

PFs are Monte Carlo methods [28] that approximate probability density functions (pdfs) of interest with M particles. The *bootstrap PF* (BPF) [29] is the most popular algorithm, because of its simplicity and reasonable performance in several settings. However, alternatives are needed for challenging applications that require models with complex posterior distributions. Most notably, the *auxiliary PF* (APF) [30] was designed to deal with very informative observations. Since it was proposed, the APF has been re-interpreted under different perspectives [31, 32]. However, the APF still presents an insufficient behavior in many applications of interest, and the development of better PFs is crucial to address new challenges in sequential learning.

In this paper, we develop a flexible framework named *optimized* APF (OAPF) for accurate inference in SSMs. The OAPF framework is motivated by recent advances in multiple importance sampling (MIS) [33, 34, 35], implementing within PF a mixture proposal sampling and a weighting scheme that allows for a variance reduction in the importance weights, the key aim in SMC methods [36].

The structure of the paper is as follows. In Section 2, we review SSMs and give a brief overview on PFs. In Section 3, we derive our OAPF framework, discussing the design choices and providing a theoretical analysis of its estimators. In Section 5, we show improved results against common particle filters and the recent improved APF [37] in challenging and widely used nonlinear state-space models such as a stochastic Lorenz 63 model and a multivariate stochastic volatility model. We conclude the paper in Section 6 with some final remarks.

Preliminary work. Under review by AISTATS 2021. Do not distribute.

Contributions. (1) We develop the optimized auxiliary particle filter (OAPF) framework, which encompasses other particle filters as special cases and allows the development of new algorithms with improved estimators. Our framework has a flexible mixture proposal distribution which appears in the importance weights, provably reducing their variance.

(2) We prove that the resulting marginal likelihood estimators are unbiased and consistent, generalizing the APF estimator in [38]. The marginal likelihood is a key quantity in PF, directly related to the variance of the importance weights, and needed for model selection and parameter estimation.

(3) We propose strategies to select kernels and mixture weights in the proposal. The mixture weights are optimized by matching proposal and posterior at a set of relevant points. Crucially, this allows us to find mixture weights as a solution to a *convex* optimization problem. Therefore, our strategy allows for optimizing the proposal in very generic models (transition and observation pdfs), while avoiding black-box non-convex optimization methods that are common in for instance in variational inference [39, 40, 41]. Further, we allow for a flexible choice of the number of kernels, detaching this choice from the number of particles unlike previous works (see for instance [42]).

(4) We propose specific implementations of our framework and show their effectiveness with widely used state-space models. We compare to BPF, APF and to the improved APF (IAPF) [37], a recent algorithm which provides the state-of-the-art in terms of importance weight variance. We show evidence for better estimates in OAPF with similar computational complexity.

2 BACKGROUND

2.1 State-Space Models and Particle Filtering

State-space models (SSM) describe the temporal evolution of a system in a probabilistic manner. They are composed of a stochastic discrete-time Markovian process of a (potentially multivariate) *hidden state* $\{\mathbf{x}_t\}_{t\geq 1}$, which can only be observed via corresponding noisy measurements $\{\mathbf{y}_t\}_{t\geq 1}$. SSMs are fully specified by a prior probability density function (pdf), $p(\mathbf{x}_0)$, and by the *transition* and *observation* kernels, $f(\mathbf{x}_t|\mathbf{x}_{t-1})$ and $g(\mathbf{y}_t|\mathbf{x}_t)$, respectively, defined for $t \geq 1$. In these models, the *filtering* task consists in the sequential estimation of the filtering density $p(\mathbf{x}_t|\mathbf{y}_{1:t})$, as well as expectations of the form

$$I(h_t) \triangleq \mathbb{E}_{p(\mathbf{x}_t|\mathbf{y}_{1:t})}[h_t(\mathbf{x}_t)] = \int h_t(\mathbf{x}_t)\, p(\mathbf{x}_t|\mathbf{y}_{1:t})\, d\mathbf{x}_t, \tag{1}$$

for (integrable) functions of interest. For most SSMs of interests, the filtering pdf is intractable and one needs to resort to approximate inference. In this context, particle filters (PFs) are the most popular inferential methods, approximating the filtering pdf with a set of random particles (Monte Carlo samples). PFs are a sequential implementation of importance sampling (IS), generating at each time step M particles $\{\mathbf{x}_t^{(m)}\}_{m=1}^M$ from a proposal pdf $q(\mathbf{x}_t)$ and assigning them normalized importance weights $w_t^{(m)}$. The unnormalized importance weights can be computed by updating the previous weights as

$$\widetilde{w}_t^{(m)} = w_{t-1}^{(m)} \frac{g(\mathbf{y}_t|\mathbf{x}_t^{(m)}) f(\mathbf{x}_t^{(m)}|\mathbf{x}_{t-1}^{(m)})}{q(\mathbf{x}_t^{(m)}|\mathbf{y}_t, \mathbf{x}_{t-1}^{(m)})}, \tag{2}$$

which can be derived by factorizing the joint proposal in a recursive way $q(\mathbf{x}_{1:t-1}|\mathbf{y}_{1:t})q(\mathbf{x}_t|\mathbf{y}_t, \mathbf{x}_{t-1})$ and targeting $p(\mathbf{x}_{1:t}|\mathbf{y}_{1:t})$ [1]. Therefore, a particle filter maintains a set of normalized weights and particles $\{w_t^{(m)}, \mathbf{x}_t^{(m)}\}_{m=1}^M$ as a representation of the filtering pdf, updating weights at each time step with as in Eq. (2). The most popular choice for $q(\mathbf{x}_t|\mathbf{y}_t, \mathbf{x}_{t-1})$ is $f(\mathbf{x}_t|\mathbf{x}_{t-1})$ and leads to the concrete bootstrap particle filter (BPF) [29]. The advantage of this choice is that the weights in (2) simply become $w_{t-1}^{(m)} g(\mathbf{y}_t|\mathbf{x}_t^{(m)})$. In practice, particle filters suffer from the *weight degeneracy* effect [1], consisting on few normalized weights taking all probability mass (i.e., the posterior is approximated with very few samples). In the BPF, a resampling step is introduced to mitigate this effect (see more details in [43]). In some implementations, the resampling step is performed only when the effective sample size $\text{ESS} = \frac{1}{\sum_{m=1}^M \left(w_t^{(m)}\right)^2}$ is below some threshold [3, 36, 1].

2.2 Auxiliary Particle Filters

Auxiliary PFs (APFs) were introduced to alleviate some of the limitations of existing PF methods [30]. For instance, it is well known that informative likelihoods often impact negatively the ability of the standard BPF to reconstruct the filtering pdf [36, 44, 45].[1] Intuitively, the reason is that the resampling step at the end of the recursion at time $t-1$ does not take into account the new observation $\mathbf{y}_t$. In the standard APF, the resampling step at $t-1$ is delayed until the

[1] Informally, an informative likelihood refers to a peaky likelihood that heavily influences the shape of the posterior.

new observation $\mathbf{y}_t$ is available. Then the resampling is performed with modified unnormalized weights

$$\widetilde{\lambda}_t^{(m)} = w_{t-1}^{(m)} g(\mathbf{y}_t|\boldsymbol{\mu}_t^{(m)}), \qquad \lambda_t^{(m)} = \frac{\widetilde{\lambda}_t^{(m)}}{\sum_{m=1}^{M} \widetilde{\lambda}_t^{(m)}}, \tag{3}$$

where $\boldsymbol{\mu}_t^{(m)} = \mathbb{E}_{f(\mathbf{x}_t|\mathbf{x}_{t-1}^{(m)})}[\mathbf{x}_t]$. Then the particles are propagated using the transition kernel $f(\mathbf{x}_t|\mathbf{x}_{t-1})$ as in BPF. Finally, the importance weights are chosen as

$$\widetilde{w}_{t-1}^{(m)} = \frac{g(\mathbf{y}_t|\mathbf{x}_t^{(m)})}{g(\mathbf{y}_t|\boldsymbol{\mu}_t^{(i^{(m)})})}, \tag{4}$$

where $i^{(m)}$ denotes the index of the ancestor that generates the m-th resampled particle. Intuitively, this can be seen as scaling down the BPF weights, taking into account that particles have been already resampled in large number in regions of high likelihood [42]. A different interpretation of APFs is possible from the multiple importance sampling (MIS) perspective [42]. Note that MIS refers to the different sampling and weighting schemes that are possible in the presence of multiple proposals in IS [46, 35]. In this perspective, a resampling step followed by a propagation step is considered to be simply a single sampling step from a mixture pdf. Moreover, it shows that the mixture weights and importance weights can be derived by using approximations that in some models are too loose, which explains the poor behavior of APF in many settings. The improved APF (IAPF) overcomes these approximations at the expense of increasing the computational complexity of calculating $\lambda_t^{(m)}$ [37]. The MIS perspective is also related to auxiliary marginal particle filters (AMPF) [32], where a similar importance weight is derived. The AMPF interpret that the inference is performed in the marginal space of $\mathbf{x}_t$ (marginalizing the auxiliary variable), which guarantees to reduce the variance of the importance weights (it is a Rao-Blackwellization that can be proved by the variance decomposition lemma [32]).

3 OPTIMIZED AUXILIARY PARTICLE FILTERS

3.1 The OAPF Framework

In this Section, we present our new framework for *optimized auxiliary particle filters* (OAPFs). The OAPF framework builds upon the MIS perspective, considering a generic mixture as proposal where all samples are (independently) simulated from. More precisely, we consider the generic mixture proposal at each time step t as

$$\psi_t(\mathbf{x}_t) = \sum_{k=1}^{K} \lambda_t^{(k)} q_t^{(k)}(\mathbf{x}_t), \tag{5}$$

with associated mixture weights $\lambda_t^{(k)}$. To the best of our knowledge, the OAPF is the first method to detach the choice of K from the number of samples M (i.e., $K \neq M$ in the general case).

Algorithm 1: Optimized Auxiliary Particle Filter

Input: prior, transition, and observation pdfs $p(\mathbf{x}_0), f(\mathbf{x}_t|\mathbf{x}_{t-1}), g(\mathbf{y}_t|\mathbf{x}_t)$, and sequence of observations $\mathbf{y}_{1:T}$

Output: set of weighted samples for each time step $\{\mathbf{x}_t^{(m)}, \widetilde{w}_t^{(m)}\}_{m=1,t=1}^{M,T}$

Draw M samples from prior: $\mathbf{x}_0^{(m)} \sim p(\mathbf{x}_0)$ and set $w_0^{(m)} = 1/M$;

foreach $t = 1, \ldots, T$ **do**

(a) optimization step: optimize the mixture proposal by selecting K kernels $q_t^{(k)}$ and choosing their associated mixture weight $\lambda_t^{(k)}$ that compose the mixture proposal ψ_t (see Section 3.2)

(b) sampling step: simulate M particles $\mathbf{x}_t^{(m)}$ from the proposal as

$$\mathbf{x}_t^{(m)} \sim \psi_t(\mathbf{x}_t) \tag{6}$$

(c) weighting step: calculate new importance weights as:

$$\widetilde{w}_t^{(m)} = \frac{g(\mathbf{y}_t|\mathbf{x}_t^{(m)}) \sum_{i=1}^{M} w_{t-1}^{(i)} f(\mathbf{x}_t^{(m)}|\mathbf{x}_{t-1}^{(i)})}{\sum_{k=1}^{K} \lambda_t^{(k)} q_t^{(k)}(\mathbf{x}_t^{(m)})} \tag{7}$$

end

The OAPF framework is described in Algorithm 1. The method starts by simulating M samples from the prior pdf, and then at each time t, it consists of the three following stages: (a) optimization, (b) sampling, and (c) weighting steps. Note that this structure keeps also some ties with adaptive IS (AIS) algorithms [47]. First, the optimization step adapts the mixture proposal of Eq. (5). This procedure is discussed in detail in the next Section. Second, the new M particles are simulated from the mixture proposal. Third, the importance weights are calculated as in Eq. (7). Note that, although unusual in PF, they follow a standard structure in MIS, with the whole mixture in the denominator. It is worth remarking that the numerator does not evaluate the true filtering pdf but only an (unnormalized) approximation. However, the importance weights are still *proper*([48]), as we show below.

Note that the importance weights play a crucial role both in the estimators of generic moments of the approximate distributions and also in the behavior of the PF for the next time step. Hence, reducing the vari-

ance of the importance weights is the ultimate goal in PF. Since this variance depends on the discrepancy between the proposal and target pdfs [49, 50] the benefit of considering a mixture proposal in Eq. 5 and for the importance weights in Eq. (7) is twofold. First, mixtures are a flexible way to approximate a large collection of pdfs (see for instance [51, 52, 53]). Second, while PFs work implicitly with mixture proposal, only few of them use them in the denominator of the importance weights [32, 37, 42]. Placing the whole mixture in the denominator, as we do in OAPF, is known to reduce variance in MIS [35], even yielding to zero-variance importance weights in the case of perfect matching between proposal and target pdfs.

In OAPF, the standard IS estimators can be built. More precisely, moments of the filtering pdf as in Eq. (1) can be approximated by the self-normalized IS (SNIS) estimator as

$$\widehat{I}\left(h_t\right) = \sum_{m=1}^{M} w_t^{(m)} h_t(\mathbf{x}_t^{(m)}), \tag{8}$$

where $w_t^{(m)} = \frac{\widetilde{w}_t^{(m)}}{\sum_{j=1}^{M} \widetilde{w}_t^{(j)}}$ are the normalized weights. Finally, the weights of OAPF can be used to build an unbiased estimator of $p(\mathbf{y}_{1:t})$, which is crucial for many statistical tasks such as model selection [54, 55, 56]. The OAPF estimator of the partial normalizing constant at time t, $p(\mathbf{y}_t|\mathbf{y}_{1:t-1})$, is given by

$$\widehat{p}(\mathbf{y}_t|\mathbf{y}_{1:t-1}) \triangleq \frac{1}{M} \sum_{m=1}^{M} \widetilde{w}_t^{(m)}. \tag{9}$$

Since the marginal likelihood admits a decomposition as $p(\mathbf{y}_{1:T}) = p(\mathbf{y}_1) \prod_{t=1}^{T} p(\mathbf{y}_t|\mathbf{y}_{1:t-1})$, then we can build the OAPF estimator as

$$\widehat{p}(\mathbf{y}_{1:T}) \triangleq \widehat{p}(\mathbf{y}_1) \prod_{t=2}^{T} \widehat{p}(\mathbf{y}_t|\mathbf{y}_{1:t-1}). \tag{10}$$

The functional form of the OAPF estimator is similar to other PFs and can be justified by standard IS arguments, but the computation of the importance weights $\widetilde{w}_t^{(m)}$ differs from other methods as discussed above. In the following, we prove that the estimator $\widehat{p}(\mathbf{y}_{1:T})$ is unbiased and consistent, which turns the SNIS estimator of Eq. (8) consistent.

Theorem 1 *For any set of mixture proposals* $\{\psi(\mathbf{x}_t)\}_{t=1}^{T}$ *fulfilling standard regularity conditions in IS, the normalizing constant estimator in Eq.* (10) *is unbiased and consistent, i.e.,* $\mathbb{E}[\widehat{p}(\mathbf{y}_{1:T})] = p(\mathbf{y}_{1:T})$ *and* $\lim_{M\to\infty} \widehat{p}(\mathbf{y}_{1:T}) = p(\mathbf{y}_{1:T})$ *a.s. for any* $T \in \mathbb{R}^+$.

Proof: The proof is presented in the supplementary material as well as a description of the regularity conditions. □

Note that the consistency of the SNIS estimator in Eq. (8) is also guaranteed by standard IS arguments (we complete this discussion in the supplement).

Finally, note that the minimization of the variance of the normalizing constant is equivalent to minimizing the variance of the importance weights $\widetilde{w}_t$, [57, 36]. The OAPF explicitly aims at reducing this variance by minimizing the mismatch between the target pdf and the mixture proposal. In the supplement, we also present a proof showing that the variance the OAPF weights in Eq. (7) is always less than those of APF in Eq. (4), when $K = M$ and the mixture weights are the same.

3.2 Selection of the Mixture Weights

In this Section we discuss an approach to select the weights of the mixture proposal ψ_t. The ultimate goal is to select them so that the proposal is as close as possible to the approximate filtering posterior. To achieve this, we posit that these two distributions should be equal at a set of E *evaluation points* $\{\mathbf{z}_t^{(e)}\}_{e=1}^{E}$. We will show that this approach is flexible and brings several advantages. For simplicity, consider these points to be the centers of the K kernels $q^{(k)}$ in the proposal (5), i.e. $\{\mathbf{z}_t^{(e)}\}_{e=1}^{E} = \{\boldsymbol{\mu}_t^{(k)}\}_{k=1}$ and so $K = E$. To make this more concrete, the K kernels could be chosen to be, for instance, some subset of size K of the M transition kernels available from the previous particles. Note that APF and improved APF [42] also recommend the use of transition kernel centers. Alternatively, we could also use "optimal" SMC kernels [36] which are defined via the intractable function $p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{y}_t)$. Our framework allows for generic choices so these restrictions are not necessary, and we continue this Section in a generic setting. We expand this discussion in Section 3.3.

Now that we have chosen the K mixture kernels and the E evaluation points, we can satisfy the condition previously mentioned and build a *linear* system of E equations as:

$$\sum_{k=1}^{K} \lambda_t^{(k)} q_t^{(k)}(\mathbf{z}_t^{(e)}) = g(\mathbf{y}_t|\mathbf{z}_t^{(e)}) \sum_{m=1}^{M} w_{t-1}^{(m)} f(\mathbf{z}_t^{(e)}|\mathbf{x}_{t-1}^{(m)}), \quad e = 1, \ldots, E, \tag{11}$$

where the K mixture weights $\lambda_t^{(k)}$ are unknown. For a unique solution to exist is necessary that $K = E$, but in general we do not need to restrict to this case. Below, we show how to turn this problem into a (constrained) *convex* optimization problem. Let us first

define the following vectors:

$$\begin{aligned}
\boldsymbol{\lambda} &\triangleq \left(\lambda_t^{(1)}, \lambda_t^{(2)}, \ldots, \lambda_t^{(K)}\right)^\top, \\
\mathbf{w} &\triangleq \left(w_{t-1}^{(1)}, w_{t-1}^{(2)}, \ldots w_{t-1}^{(M)}\right)^\top, \\
\mathbf{f}^{(e)} &\triangleq \left(f(\mathbf{z}_t^{(e)}|\mathbf{x}_{t-1}^{(1)}), \ldots, f(\mathbf{z}_t^{(e)}|\mathbf{x}_{t-1}^{(M)})\right)^\top, \\
\mathbf{q}^{(e)} &\triangleq \left(q^{(1)}(\mathbf{z}_t^{(e)}), \ldots, q^{(K)}(\mathbf{z}_t^{(e)})\right)^\top.
\end{aligned}$$

Then, we can re-write Eq. (11) as

$$\mathbf{q}^{(e)^\top}\boldsymbol{\lambda} = \underbrace{g(\mathbf{y}_t|\mathbf{z}_t^{(e)}) \odot \mathbf{w}^\top \mathbf{f}^{(e)}}_{\widetilde{\boldsymbol{\pi}}^{(e)}}, \tag{12}$$

for $e = 1, \ldots, E$, where $\odot$ is elementwise multiplication and defining additionally the right-hand side to be $\widetilde{\boldsymbol{\pi}}^{(e)}$. More compactly, the above can be re-expressed in matrix form as:

$$\overbrace{\begin{bmatrix} — & \mathbf{q}^{(1)^\top} & — \\ \vdots & \vdots & \vdots \\ — & \mathbf{q}^{(E)^\top} & — \end{bmatrix}}^{E\times K} \overbrace{\begin{bmatrix} | \\ \boldsymbol{\lambda} \\ | \end{bmatrix}}^{K\times 1} = \overbrace{\begin{bmatrix} — & g(\mathbf{y}_t|\mathbf{z}_t^{(1)}) \odot \mathbf{f}^{(1)^\top} & — \\ \vdots & \vdots & \vdots \\ — & g(\mathbf{y}_t|\mathbf{z}_t^{(M)}) \odot \mathbf{f}^{(E)^\top} & — \end{bmatrix}}^{E\times M} \overbrace{\begin{bmatrix} | \\ \mathbf{w} \\ | \end{bmatrix}}^{M\times 1} \tag{13}$$

defining $\mathbf{Q}$ as the $E \times K$ matrix on the left-hand side of (13) and $\widetilde{\boldsymbol{\pi}}$ as the resulting $E \times 1$ vector on the right-hand side. We now define a generic constrained optimization problem as for the mixture weights as:

$$\boldsymbol{\lambda}^* = \arg\min_{\boldsymbol{\lambda}} \mathcal{L}\left(\mathbf{Q}\boldsymbol{\lambda}, \widetilde{\boldsymbol{\pi}}\right), \tag{14}$$

where $\mathcal{L}(\cdot)$ a generic loss function. The optimization will be constrained since $\boldsymbol{\lambda}$ will be used for resampling, and therefore needs to have non-negative elements.[2] In in the next Section, we present a possible strategy to implement $\mathcal{L}(\cdot)$ and solve the optimization problem.

Optimization via Non-Negative Least Squares (NNLS) The previous problem can be encoded as a non-negative least squares problem by taking the squared distance of the pdfs at the E evaluation points $\{\mathbf{z}_t^{(e)}\}_{e=1}^E$. Taking squared differences between left-hand side and right-hand side of (13) leads to:

$$\begin{aligned}
\boldsymbol{\lambda}^* = &\arg\min_{\boldsymbol{\lambda}} \|\mathbf{Q}\boldsymbol{\lambda} - \widetilde{\boldsymbol{\pi}}\|_2^2 \\
&\text{subject to} : \boldsymbol{\lambda} \in \mathbb{R}_{\geq 0}^K.
\end{aligned}$$

[2]The resulting values can be normalized afterwards so they parametrize the mixture proposal in Eq. (5).

This problem is a (constrained) quadratic program, therefore it is convex and the non-negativity constraints form a convex feasible set. When $\mathbf{Q}$ has full column rank, then there is a unique solution. Theoretical results on NNLS have shown that, especially when the dimension of $\mathbf{Q}$ is large (large K and E in our case), the solutions tend to be very sparse [58, 59, 60]. The optimization problem can be solved by the widely used algorithm in [61] (see also a faster version in [62]), as well as concurrent work on exact sparse NNLS [63] and strong GPU accelerations [64, 65]

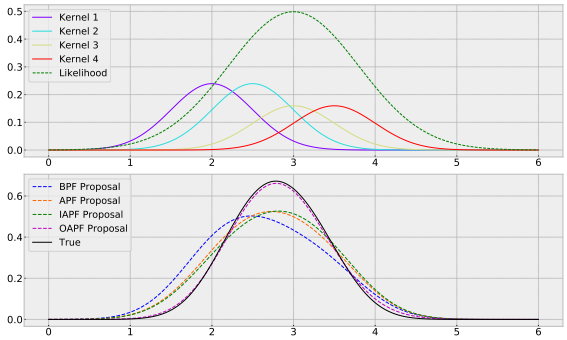
3.3 Selection of Kernels, Evaluation Points and Computational Complexity

The computational complexity of the strategy of the NNLS algorithm is worst-case $\mathcal{O}(KE^2 + MK)$ (calculation of mixture weights and weighting step) runtime, and $\mathcal{O}(KE)$ space to store the matrix $\mathbf{Q}$, and thus depends on the choices of K and E. The selection of the number of kernels K depends on the shape of the target pdf but also in the positions of the mixture kernels. In our experiments, we will take a simple approach by inheriting the standard transition kernels used for instance in BPF or APF. This allows to reduce K and E w.r.t. M in some cases, as we show in the numerical experiments. As a general guideline, a good starting point is setting $K = E$ and choosing the evaluation points as the center of the kernels, retaining those kernels associated with the E greatest values of the approximate filtering pdf. More strategies could be explored in further work.

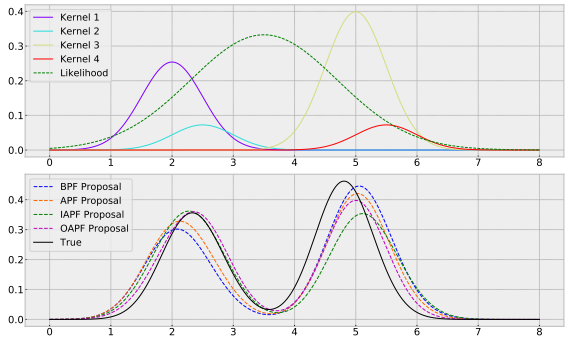
Finally, we recall that in NNLS, the solutions are in general sparse as explained in 3.2, so the *effective* number of kernels (i.e., those with $\lambda_t^{(k)} > 0$) is often much less than K [60]. This greatly simplifies the mixture sampling stage and also reduces the runtime of the weighting step in Eq. (7). Note that the weighting step could be reduced from $\mathcal{O}(MK)$ to $\mathcal{O}(M \log K)$ with dual-tree methods [32].

4 RELATED WORK

The OAPF follows a different approach w.r.t. most papers in the literature by interpreting the M samples to be simulated from a mixture proposal, and hence explicitly including that proposal in the denominator. This perspective is connected to the auxiliary marginal PF (AMPF) [32] and improved APF (IAPF) [37] algorithms, and is supported by recent advances in MIS [35] (see also the discussion of the variance reduction in [32]), and it also links with the re-interpretation of BPF and APF [42]. The selection of the mixture weights is also linked with other works. For instance, a flexible framework named *twisted* APFs is developed



(a) In this first example we choose a unimodal posterior. OAPF substantially outperforms the other algorithms.



(b) OAPF is the only algorithm who can match well *both* modes simultaneously with this multimodal posterior.

Figure 1: **Experiment 1 (Toy Example.)** In this experiment we show that OAPF proposals are closer to true posteriors compared to its competitors. We calculated χ^2-divergence for these examples in Table 1. Note that here OAPF uses transition kernels for the proposal and their centers as evaluation points. We provide all parameters for reproducibility in the supplement.

in [66], where APFs are interpreted as a special case of changing the distribution targeted in IS (this interpretation appeared first in [44, 36]. In this method, the computation is done in an offline fashion (see an extension of this line in [67]). In [68], they develop a PF framework with a different approach, preemptively moving a subset of particles to a region of high likelihood with gradient methods. In [69], the method adapts a mixture of kernels in a more generic setup (sequential Monte Carlo samplers), focussing in the choice of kernels.

5 EXPERIMENTS

We compare OAPF with BPF, APF as well as the recent improved APF (IAPF) [37], which also uses a mixture in the denominator of the importance weights and can be seen as a special case in our framework. The IAPF strictly improves over APF and BPF in most settings [37]. Note that the simple BPF can sometimes perform unexpectedly well, as it is well known in the PF community. We evaluate our framework in 4 sets of experiments (see below).

Our aim is to show the benefits of OAPF in terms of variance of importance weights, which is crucial in particle filters: the weights are used not only for approximating integrals of interest but also for building better particle approximations in the next time steps. Therefore, we choose metrics that are directly connected to the variance of the importance weights: χ^2-divergence between mixture proposal and filtering pdfs, error in the estimation of the marginal likelihood, and effective sample size (ESS) [1]. The setup of the experiments is as follows:

- **Experiment 1: Toy example**. We show visually that the mixture proposal in OAPF reconstructs the posterior better than its competitors, both with unimodal and multimodal posteriors. Numerically we show an improved χ^2-divergence between proposal and filtering pdfs, which directly translates into lower variance of importance weights.

- **Experiment 2: Linear dynamical model.** We exploit the closed-form solution of the linear dynamical model, perhaps the most known SSM and widely used for instance in object tracking [1]. This allows an evaluation of the estimators of the marginal likelihood, $p(\mathbf{y}_{1:t})$. We show that OAPF reaches better solutions with, while highly reducing the runtime.

- **Experiment 3: Stochastic Lorenz 63 model.** Transitioning to more challenging non-linear non-Gaussian models, we show an improved performance on discretized version of this popular chaotic dynamical system [70], which is used for instance in atmospheric models for weather forecasting [71, 72]. We compare the PFs in terms of the ESS, which is widely used as a proxy for the weight variance when the true normalizing constant is not available (as it is the case here).

- **Experiment 4: Stochastic volatility model.** Finally, we perform inference for a multivariate stochastic volatility model used in related work on APFs [73]. Here, as in Experiment 4, we look at ESS and show improved performance against all other algorithms.

We consider time-series with $T = 100$ time steps, except otherwise stated. We let $d_{\mathbf{x}_t}$ be the dimension

of the hidden state, i.e., $\mathbf{x}_t \in \mathbb{R}^{d_{\mathbf{x}_t}}$. In general we use $M = 100$ particles, except for $d_{\mathbf{x}_t} = 10$ where $M = 1000$. Due to the curse of dimensionality [74], a general reduction in performance for all methods is expected as $d_{\mathbf{x}_t}$ grows. For linear dynamical models, we show improved estimates with significantly reduced runtime than all other algorithms, including IAPF. For the more challenging models, we achieve better ESS than the competitors. Note that in those experiments, we set $K = E = M$. However, due to the high sparsity of solutions in OAPF, our effective K is much lower. For $M = 1000$ particles we report an average of 88% sparsity, while for $M = 100$ an average of 65%. All averages, standard errors and boxplots are obtained with 100 independent Monte Carlo runs.[3]

Table 1: **Experiment 1 (Toy example).** χ^2-div. between filtering and mixture proposal pdfs in Figure 1.

Method	χ^2-div. (Fig. 1a)	χ^2-div. (Fig. 1b)
BPF	0.1662	0.2245
APF	0.0916	0.1633
IAPF	0.0870	0.2402
OAPF	**0.0069**	**0.0819**

1. Toy Example. The main goal of this toy example is to illustrate that the OAPF mixture proposal better reconstructs the filtering pdf. We also measure the χ^2-divergence between both pdfs. We consider a single iteration of each PF algorithm and build artificial proposals by multiplying a mixture of 4 Gaussians with a Gaussian likelihood. Results from the two experiments with the above setting are shown in Figure 1. We select the means of the transition kernel as evaluation points, and set $K = E = M = 4$. In Figure 1a, we show the results for a unimodal posterior. This setting is advantageous for the IAPF, as transition kernels significantly overlap (see [42] for more details). The likelihood is sufficiently informative, which explains why APF outperforms BPF. Figure 1b shows a more complex multimodal posterior with a more diffused likelihood. Interestingly, we find that IAPF can perform even worse than APF, while our OAPF does not suffer from this issue. Table 1 quantifies (for both settings) the mismatch between mixture proposal and filtering pdfs in terms of χ^2-divergence, confirming the visual analysis of Fig. 1a.

2. Linear Dynamical Model. The linear dynamical model is arguably the most popular SSM, routinely being the first choice to assess PFs [2, 75, 76]. It has been applied in a wide range of applications (e.g., robotics [1]). This model is particularly useful for validating PFs since it is one of the few models admitting closed-form solutions of the filtering distribution and the normalizing constant (via the celebrated *Kalman filter* [77]). The defining transition and observation equations are standard (see e.g., [1]) and require mainly the selection of observation and transition covariances (we provide more details in the supplement). We used observation covariance is $5\mathbf{I}$ and transition covariance $2.5\mathbf{I}$. This is again a favourable setting for the competitor IAPF because of the wide transition kernel w.r.t. observation kernel. We calculate the normalized mean-squared error (NMSE) between our unbiased estimates of the marginal likelihood $p(\mathbf{y}_{1:t})$ and the true value for $d_{\mathbf{x}_t} \in \{2, 5, 10\}$. We define NSME as mean-squared error divided by true value of $p(\mathbf{y}_{1:t})$. In Figure 2 we show boxplots of NSME for all methods with $d_{\mathbf{x}_t} = 10$. We set $M = 100$ and $M = 1000$. In both cases, the proposed OAPF outperforms all the competitors, while setting a low $K = E = 5$, which is translated into large computational savings. Similar conclusions can be extracted for other choices of the model parameters. In the supplement, we include complementary results for this model with $d_{\mathbf{x}_t} \in \{2, 5\}$.

3. Stochastic Lorenz 63 Model. The Lorenz 63 [70] is a *chaotic* system, since slightly different initial conditions generate extremely different trajectories. Due to this difficulty, this model is often used to evaluate PFs [78, 68]. We consider a discretized version of the state dynamics using an Euler-Maryuyama scheme and observations with additive noise. The hidden state is three dimensional $\mathbf{x}_t = [x^{[1]}, x^{[2]}, x^{[3]}]$ and the transition dynamics are defined by the differential equations:

$$\mathrm{d}x^{[1]} = \sigma(x^{[2]} - x^{[1]})\mathrm{d}\tau + \mathrm{d}w_{x^{[1]}} \tag{15}$$

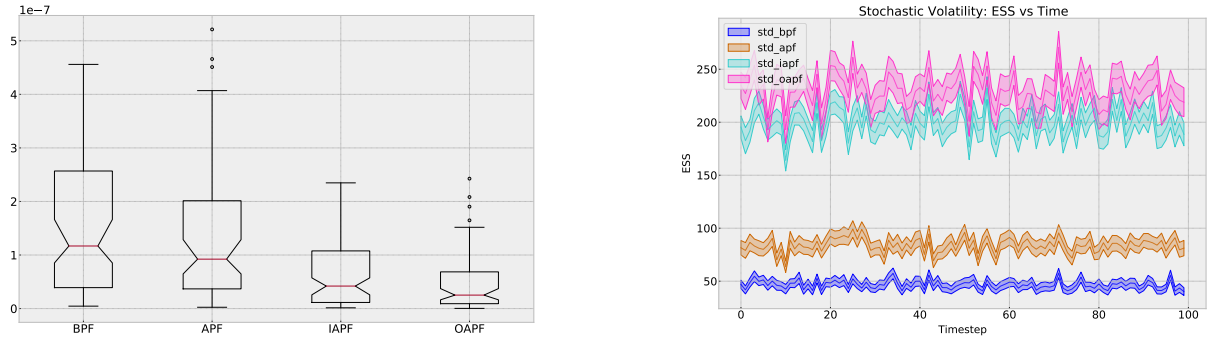
$$\mathrm{d}x^{[2])} = (\rho x^{[1]} - x^{[3]}x^{[1]} - x^{[2]})\mathrm{d}\tau + \mathrm{d}w_{x^{[2]}} \tag{16}$$

$$\mathrm{d}x^{[3]} = (x^{[1]}x^{[2]} - \beta x^{[3]})\mathrm{d}\tau + \mathrm{d}w_{x^{[3]}} \tag{17}$$

where τ denotes continuous time, $w_{x^{[1]}}, w_{x^{[2]}}, w_{x^{[3]}}$ are independent one-dimensional standard Wiener processes and (σ, ρ, β) are parameters of the model. We use the an increment Δt in the discretization, and partially observe the hidden state (only the first dimension) with scalar $y_t \sim \mathcal{N}_{y_t}(x^{(1)}, \sigma^2_{y_t} = 1)$, using the standard values for (σ, ρ, β) (see supplement). The results of averaged ESS with two different values of $\Delta t \in \{0.01, 0.008\}$ are shown in Table 2. Note that even these small changes in Δt cause very different trajectories, as we also show in the supplement.

4. Stochastic Volatility Model. We perform inference in a multivariate stochastic volatility model

[3]The code of all experiments will be made available upon publication.



(a) (b)

Figure 2: **Experiment 2 (Linear dynamical model)** 2a: boxplot of normalized MSE for (log) marginal likelihood estimates with $d_{\mathbf{x}_t} = 10$. Note that in this experiment, $K = E = 5$ so OAPF runs with greatly reduced runtime. **Experiment 4 (Stochastic volatility)** 2b: ESS over timesteps with $d_{\mathbf{x}_t} = 10$.

Table 2: **Experiment 3 (Lorenz)**. $T = 1000$ timesteps, $M = 100$ particles. Averaged ESS and standard errors.

Method	$\Delta t = 0.01$	$\Delta t = 0.008$
BPF	57.7 ± 0.2	58.1 ± 0.2
APF	55.1 ± 0.2	55.2 ± 0.2
IAPF	70.1 ± 0.1	71.0 ± 0.1
OAPF	$\mathbf{76.7 \pm 0.1}$	$\mathbf{76.4 \pm 0.1}$

Table 3: **Experiment 4 (Stochastic Volatility)**. Note that when $d_{\mathbf{x}_t} = 10$ then $M = 1000$. Averaged ESS and standard errors.

.

Method	$d_{\mathbf{x}_t} = 2$	$d_{\mathbf{x}_t} = 5$	$d_{\mathbf{x}_t} = 10$
BPF	50.8 ± 0.2	21.2 ± 0.4	46.6 ± 0.5
APF	59.7 ± 0.2	31.9 ± 0.4	83.9 ± 0.6
IAPF	80.5 ± 0.1	49.4 ± 0.5	199.9 ± 1.7
OAPF	$\mathbf{92.6 \pm 0.1}$	$\mathbf{59.5 \pm 0.7}$	$\mathbf{239.5 \pm 2.4}$

(SVM), a type of stochastic process where the variance is a latent variable that follows itself a stochastic process. These are extremely useful models to apply for many tasks in econometrics, e.g., for predicting the volatility of a heteroskedastic sequence such as returns on equity indices or currency exchanges [79]. SVMs are often used to evaluate particle filters [30, 80, 81, 32, 73]. We employ the version in [82], which is also used in related work on APFs [66]. It is defined by the following prior, transition and observation pdfs:

$$p(\mathbf{x}_0) = \mathcal{N}_{\mathbf{x}_0}(\mathbf{m}, \mathbf{U}_0), \tag{18}$$

$$f(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}_{\mathbf{x}_t}(\mathbf{m} + \text{diag}(\phi)(\mathbf{x}_{t-1} - \mathbf{m}), \mathbf{U}), \tag{19}$$

$$g(\mathbf{y}_t|\mathbf{x}_t) = \mathcal{N}_{\mathbf{y}_t}(\mathbf{0}, \exp(\text{diag}(\mathbf{x}_t))). \tag{20}$$

Table 3 shows averaged ESS for $d_{\mathbf{x}_t} = (2, 5, 10)$ while Figure 2b shows the averaged ESS over time for $d_{\mathbf{x}_t} = 10$. The parameters for this experiment are set to $\mathbf{m} = \mathbf{0}, \mathbf{U}_0 = \mathbf{I}, \mathbf{U} = \mathbf{I}, \phi = \mathbf{1}$. For results with additional parameters, see supplementary.

6 CONCLUSIONS

In this paper we have proposed OAPF, a flexible framework for particle filtering that uses a generic mixture distribution as a proposal and includes previous methods as special cases. The framework allows for the development of particle filters with improved performance, and we provide a explicit implementation. We have proved the unbiasedness of the OAPF marginal likelihood estimator for any mixture proposal that fulfills standard IS requirements. We also show the effectiveness of OAPF in reducing the varince of the IS estimators. In OAPF, we directly optimize the mixture proposal to the posterior in an online fashion, rather than making specific analytic choices of mixture weights like in AMPF or IAPF. Conversely to most other methods that optimize a proposal (e.g., variational inference), our optimization strategy is convex, directly addresses the ultimate goal of minimizing the variance of the importance weights, and can deal with any likelihood and transition models all, at the same time, without resorting to black-box, non-convex methods (see for instance [39, 41]). We have shown an improved performance of the proposed implementation of the OAPF across a series of challenging state-space models and metrics, comparing with BPF, APF and the competitive IAPF. Finally, the flexibility and the strong theoretical guarantees of OAPF pave the way for new methodological advances within this framework.

References

[1] S. Särkkä, *Bayesian filtering and smoothing*, vol. 3. Cambridge University Press, 2013.

[2] M. Hirt and P. Dellaportas, "Scalable bayesian learning for state space models using variational inference with smc samplers," vol. 89 of *Proceedings of Machine Learning Research*, pp. 76–86, PMLR, 16–18 Apr 2019.

[3] A. Doucet, N. De Freitas, and N. Gordon, "An introduction to sequential monte carlo methods," in *Sequential Monte Carlo methods in practice*, pp. 3–14, Springer, 2001.

[4] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking," *IEEE Transactions on signal processing*, vol. 50, no. 2, pp. 174–188, 2002.

[5] P. M. Djuric, J. H. Kotecha, J. Zhang, Y. Huang, T. Ghirmai, M. F. Bugallo, and J. Miguez, "Particle filtering," *IEEE signal processing magazine*, vol. 20, no. 5, pp. 19–38, 2003.

[6] P. Del Moral, "Feynman-kac formulae," in *Feynman-Kac Formulae*, pp. 47–93, Springer, 2004.

[7] S. Thrun, "Particle filters in robotics," in *Proceedings of the Eighteenth conference on Uncertainty in artificial intelligence*, pp. 511–518, Morgan Kaufmann Publishers Inc., 2002.

[8] N. Vlassis, B. Terwijn, and B. Krose, "Auxiliary particle filter robot localization from high-dimensional sensor observations," in *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)*, vol. 1, pp. 7–12, IEEE, 2002.

[9] C. Kwok, D. Fox, and M. Meila, "Real-time particle filters," in *Advances in neural information processing systems*, pp. 1081–1088, 2003.

[10] A. A. Wardhana, E. Clearesta, A. Widyotriatmo, *et al.*, "Mobile robot localization using modified particle filter," in *2013 3rd International Conference on Instrumentation Control and Automation (ICA)*, pp. 161–164, IEEE, 2013.

[11] K. Nummiaro, E. Koller-Meier, and L. Van Gool, "An adaptive color-based particle filter," *Image and vision computing*, vol. 21, no. 1, pp. 99–110, 2003.

[12] G. Zhu, D. Liang, Y. Liu, Q. Huang, and W. Gao, "Improving particle filter with support vector regression for efficient visual tracking," in *IEEE International Conference on Image Processing 2005*, vol. 2, pp. II–422, IEEE, 2005.

[13] H. F. Lopes and R. S. Tsay, "Particle filters and bayesian inference in financial econometrics," *Journal of Forecasting*, vol. 30, no. 1, pp. 168–209, 2011.

[14] M. P. Rios and H. F. Lopes, "The extended liu and west filter: Parameter learning in markov switching stochastic volatility models," in *State-Space Models*, pp. 23–61, Springer, 2013.

[15] M. Keeling and P. Rohani, "Stochastic dynamics," *Modeling Infectious Diseases in Humans and Animals*, pp. 190–230, 2007.

[16] D. M. Sheinson, J. Niemi, and W. Meiring, "Comparison of the performance of particle filter algorithms applied to tracking of a disease epidemic," *Mathematical biosciences*, vol. 255, pp. 21–32, 2014.

[17] P. Dawson, R. Gailis, and A. Meehan, "Detecting disease outbreaks using a combined bayesian network and particle filter approach," *Journal of theoretical biology*, vol. 370, pp. 171–183, 2015.

[18] A. Piché, V. Thomas, C. Ibrahim, Y. Bengio, and C. Pal, "Probabilistic planning with sequential monte carlo methods," in *International Conference on Learning Representations*, 2019.

[19] C. J. Maddison, D. Lawson, G. Tucker, N. Heess, A. Doucet, A. Mnih, and Y. W. Teh, "Particle value functions," in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*, OpenReview.net, 2017.

[20] Y. Wang, B. Liu, J. Wu, Y. Zhu, S. S. Du, L. Fei-Fei, and J. B. Tenenbaum, "Dualsmc: Tunneling differentiable filtering and planning under continuous pomdps." Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20, 2020.

[21] D. Lawson, G. Tucker, C. A. Naesseth, C. J. Maddison, R. P. Adams, and Y. W. Teh, "Twisted variational sequential monte carlo," in *Third workshop on Bayesian Deep Learning (NeurIPS)*, 2018.

[22] T. A. Le, M. Igl, T. Rainforth, T. Jin, and F. Wood, "Auto-encoding sequential monte

carlo," in *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, OpenReview.net, 2018.

[23] A. K. Moretti, Z. Wang, L. Wu, and I. Pe'er, "Smoothing nonlinear variational objectives with sequential monte carlo," in *Deep Generative Models for Highly Structured Data, ICLR 2019 Workshop, New Orleans, Louisiana, United States, May 6, 2019*, OpenReview.net, 2019.

[24] S. S. Gu, Z. Ghahramani, and R. E. Turner, "Neural adaptive sequential monte carlo," in *Advances in neural information processing systems*, pp. 2629–2637, 2015.

[25] C. Naesseth, S. Linderman, R. Ranganath, and D. Blei, "Variational sequential monte carlo," vol. 84 of *Proceedings of Machine Learning Research*, (Playa Blanca, Lanzarote, Canary Islands), pp. 968–977, PMLR, 09–11 Apr 2018.

[26] X. Ma, P. Karkus, D. Hsu, and W. S. Lee, "Particle filter recurrent neural networks," in *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020*, pp. 5101–5108, AAAI Press, 2020.

[27] Y. Li and M. Coates, "Particle filtering with invertible particle flow," *IEEE Transactions on Signal Processing*, vol. 65, no. 15, pp. 4102–4116, 2017.

[28] N. Metropolis and S. Ulam, "The monte carlo method," *Journal of the American statistical association*, vol. 44, no. 247, pp. 335–341, 1949.

[29] N. J. Gordon, D. J. Salmond, and A. F. Smith, "Novel approach to nonlinear/non-gaussian bayesian state estimation," in *IEE proceedings F (radar and signal processing)*, vol. 140, pp. 107–113, IET, 1993.

[30] 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.

[31] S. Godsill and T. Clapp, "Improvement strategies for monte carlo particle filters," in *Sequential Monte Carlo methods in practice*, pp. 139–158, Springer, 2001.

[32] M. Klaas, N. de Freitas, and A. Doucet, "Toward practical n2 monte carlo: the marginal particle filter," in *Proceedings of the Twenty-First Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-05)*, (Arlington, Virginia), pp. 308–315, AUAI Press, 2005.

[33] A. B. Owen, "Monte carlo theory, methods and examples," *Monte Carlo Theory, Methods and Examples. Art Owen*, 2013.

[34] M. Sbert, V. Havran, and L. Szirmay-Kalos, "Multiple importance sampling revisited: breaking the bounds," *EURASIP Journal on Advances in Signal Processing*, vol. 2018, no. 1, p. 15, 2018.

[35] V. Elvira, L. Martino, D. Luengo, M. F. Bugallo, *et al.*, "Generalized multiple importance sampling," *Statistical Science*, vol. 34, no. 1, pp. 129–155, 2019.

[36] A. Doucet and A. M. Johansen, "A tutorial on particle filtering and smoothing: Fifteen years later," *Handbook of nonlinear filtering*, vol. 12, no. 656-704, p. 3, 2009.

[37] V. Elvira, L. Martino, M. F. Bugallo, and P. M. Djurić, "In search for improved auxiliary particle filters," in *2018 26th European Signal Processing Conference (EUSIPCO)*, pp. 1637–1641, IEEE, 2018.

[38] M. K. Pitt, R. dos Santos Silva, P. Giordani, and R. Kohn, "On some properties of markov chain monte carlo simulation methods based on the particle filter," *Journal of Econometrics*, vol. 171, no. 2, pp. 134–151, 2012.

[39] E. Archer, I. M. Park, L. Buesing, J. Cunningham, and L. Paninski, "Black box variational inference for state space models," *arXiv preprint arXiv:1511.07367*, 2015.

[40] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, "Variational inference: A review for statisticians," *Journal of the American statistical Association*, vol. 112, no. 518, pp. 859–877, 2017.

[41] A. B. Dieng, D. Tran, R. Ranganath, J. Paisley, and D. Blei, "Variational inference via χ upper bound minimization," in *Advances in Neural Information Processing Systems*, pp. 2732–2741, 2017.

[42] V. Elvira, L. Martino, M. F. Bugallo, and P. M. Djuric, "Elucidating the auxiliary particle filter via multiple importance sampling [lecture notes]," *IEEE Signal Processing Magazine*, vol. 36, no. 6, pp. 145–152, 2019.

[43] 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.

[44] A. M. Johansen and A. Doucet, "A note on auxiliary particle filters," *Statistics & Probability Letters*, vol. 78, no. 12, pp. 1498–1504, 2008.

[45] N. Whiteley and A. M. Johansen, "Auxiliary particle filtering: recent developments," *Bayesian time series models. Cambridge University Press, Cambridge*, 2011.

[46] E. Veach and L. J. Guibas, "Optimally combining sampling techniques for monte carlo rendering," in *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pp. 419–428, 1995.

[47] M. F. Bugallo, V. Elvira, L. Martino, D. Luengo, J. Miguez, and P. M. Djuric, "Adaptive importance sampling: The past, the present, and the future," *IEEE Signal Processing Magazine*, vol. 34, no. 4, pp. 60–79, 2017.

[48] J. S. Liu, *Monte Carlo Strategies in Scientific Computing*. Springer, 2004.

[49] E. K. Ryu and S. P. Boyd, "Adaptive importance sampling via stochastic convex programming," *arXiv preprint arXiv:1412.4845*, 2014.

[50] J. Míguez, "On the performance of nonlinear importance samplers and population Monte Carlo schemes," in *2017 22nd International Conference on Digital Signal Processing (DSP)*, pp. 1–5, IEEE, 2017.

[51] W. Feller, *An Introduction to Probability and Its Applications*, vol. II of *Wiley Publication in Mathematical Statistics*. Wiley India Pvt. Limited, 1966.

[52] H. W. Sorenson and D. L. Alspach, "Recursive Bayesian estimation using Gaussian sums," *Automatica*, pp. 465–479, 1971.

[53] J. H. Kotecha and P. M. Djuric, "Gaussian sum particle filtering," *IEEE Transactions on signal processing*, vol. 51, no. 10, pp. 2602–2612, 2003.

[54] C. Andrieu, A. Doucet, and R. Holenstein, "Particle markov chain monte carlo methods," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 72, no. 3, pp. 269–342, 2010.

[55] N. Kantas, A. Doucet, S. S. Singh, J. Maciejowski, N. Chopin, *et al.*, "On particle methods for parameter estimation in state-space models," *Statistical science*, vol. 30, no. 3, pp. 328–351, 2015.

[56] D. Luengo, L. Martino, M. Bugallo, V. Elvira, and S. Särkkä, "A survey of monte carlo methods for parameter estimation," *EURASIP Journal on Advances in Signal Processing*, vol. 2020, no. 1, pp. 1–62, 2020.

[57] A. Doucet, S. Godsill, and C. Andrieu, "On sequential monte carlo sampling methods for bayesian filtering," *Statistics and computing*, vol. 10, no. 3, pp. 197–208, 2000.

[58] M. Slawski and M. Hein, "Sparse recovery by thresholded non-negative least squares," in *Advances in Neural Information Processing Systems*, pp. 1926–1934, 2011.

[59] M. Slawski, M. Hein, *et al.*, "Non-negative least squares for high-dimensional linear models: Consistency and sparse recovery without regularization," *Electronic Journal of Statistics*, vol. 7, pp. 3004–3056, 2013.

[60] N. Meinshausen *et al.*, "Sign-constrained least squares estimation for high-dimensional regression," *Electronic Journal of Statistics*, vol. 7, pp. 1607–1631, 2013.

[61] C. L. Lawson and R. J. Hanson, *Solving least squares problems*. SIAM, 1995.

[62] R. Bro and S. De Jong, "A fast non-negativity-constrained least squares algorithm," *Journal of Chemometrics: A Journal of the Chemometrics Society*, vol. 11, no. 5, pp. 393–401, 1997.

[63] N. Nadisic, A. Vandaele, N. Gillis, and J. E. Cohen, "Exact sparse nonnegative least squares," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5395–5399, IEEE, 2020.

[64] Y. Luo and R. Duraiswami, "Efficient parallel nonnegative least squares on multicore architectures," *SIAM Journal on Scientific Computing*, vol. 33, no. 5, pp. 2848–2863, 2011.

[65] V. Kysenko, K. Rupp, O. Marchenko, S. Selberherr, and A. Anisimov, "Gpu-accelerated nonnegative matrix factorization for text mining," in *International Conference on Application of Natural Language to Information Systems*, pp. 158–163, Springer, 2012.

[66] P. Guarniero, *The Iterated Auxiliary Particle Filter and Applications to State Space Models and Diffusion Processes*. PhD thesis, University of Warwick, 2017.

[67] J. Heng, A. N. Bishop, G. Deligiannidis, A. Doucet, *et al.*, "Controlled sequential monte carlo," *Annals of Statistics*, vol. 48, no. 5, pp. 2904–2929, 2020.

[68] Ö. D. Akyildiz and J. Míguez, "Nudging the particle filter," *Statistics and Computing*, vol. 30, no. 2, pp. 305–330, 2020.

[69] J. Cornebise, E. Moulines, and J. Olsson, "Adaptive sequential monte carlo by means of mixture of experts," *Statistics and Computing*, vol. 24, no. 3, pp. 317–337, 2014.

[70] E. N. Lorenz, "Deterministic nonperiodic flow," *Journal of Atmospheric Sciences*, vol. 20, no. 2, pp. 130–141, 1963.

[71] E. Ott, B. R. Hunt, I. Szunyogh, A. V. Zimin, E. J. Kostelich, M. Corazza, E. Kalnay, D. Patil, and J. A. Yorke, "A local ensemble kalman filter for atmospheric data assimilation," *Tellus A: Dynamic Meteorology and Oceanography*, vol. 56, no. 5, pp. 415–428, 2004.

[72] H. C. Yeong, R. T. Beeson, N. S. Namachchivaya, and N. Perkowski, "Particle filters with nudging in multiscale chaotic systems: With application to the lorenz'96 atmospheric model," *Journal of Nonlinear Science*, pp. 1–34, 2020.

[73] P. Guarniero, A. M. Johansen, and A. Lee, "The iterated auxiliary particle filter," *Journal of the American Statistical Association*, vol. 112, no. 520, pp. 1636–1647, 2017.

[74] D. J. MacKay and D. J. Mac Kay, *Information theory, inference and learning algorithms*. Cambridge university press, 2003.

[75] S. M. Schmon, A. Doucet, and G. Deligiannidis, "Bernoulli race particle filters," vol. 89 of *Proceedings of Machine Learning Research*, pp. 2350–2358, PMLR, 16–18 Apr 2019.

[76] A. Boustati, Ö. D. Akyildiz, T. Damoulas, and A. Johansen, "Generalized bayesian filtering via sequential monte carlo," in *Advances in neural information processing systems*, 2020.

[77] R. E. Kalman *et al.*, "A new approach to linear filtering and prediction problems," *Journal of basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[78] V. Elvira, J. Míguez, and P. M. Djurić, "Adapting the number of particles in sequential monte carlo methods through an online scheme for convergence assessment," *IEEE Transactions on Signal Processing*, vol. 65, no. 7, pp. 1781–1794, 2017.

[79] A. G. Wilson and Z. Ghahramani, "Copula processes," in *Advances in Neural Information Processing Systems*, pp. 2460–2468, 2010.

[80] J. R. Stroud, N. G. Polson, and P. Müller, "Practical filtering for stochastic volatility models," *State space and unobserved components models*, pp. 236–247, 2004.

[81] 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.

[82] S. Chib, Y. Omori, and M. Asai, "Multivariate stochastic volatility," in *Handbook of Financial Time Series*, pp. 365–400, Springer, 2009.