Elucidating the auxiliary particle filter via multiple importance sampling

Víctor Elvira[⋆], Luca Martino[†], Mónica F. Bugallo[‡], Petar M. Djurić[‡]

[⋆] University of Edinburgh (UK).

[†] Universidad Rey Juan Carlos, Madrid (Spain)

[‡] Stony Brook University, Stony Brook (USA)

Sequential Monte Carlo methods, also known as particle filtering, have seen an explosion of development both in theory and applications. The publication of [1], sparked huge interest in the area of sequential signal processing, and in particular in sequential filtering. Ever since, the number of publications where particle filtering plays a prominent role has continued to grow. An early reference of development is [2] and later tutorials include [3], [4], [5], [6], [7], [8], [9]. With particle filtering, we estimate probability density functions (pdfs) of interest by probability mass functions (pmfs) whose masses are placed at randomly chosen locations (particles) and weights assigned to the particles. The particle filter (PF) proposed in [1] is often called the bootstrap particle filter (BPF), and although it is not optimal, it is the most often used filter by practitioners. A filter that became also popular is known as the auxiliary particle filter (APF) and was proposed in [10]. With the APF, the objective is to generate better particles at each time step than with the BPF and thereby improve the accuracy of the filtering. In these notes, we derive the APF from a new perspective, one based on interpreting the APF from the multiple importance sampling (MIS) paradigm. The derivation also shows its relationship with the BPF.

Relevance

State-space models are ubiquitously used in many fields of science and engineering for modeling complex dynamical systems. Once the models are formulated, the usual goal is to estimate the state of the model, which evolves over time, from observations that are sequentially acquired and that are functions of the state. With the Bayesian approach, the estimation is carried out through the posterior distribution of the state, which can be obtained analytically only for linear and Gaussian models (where

the Kalman filter is applied [11]). By contrast PFs allow for the approximation of the posterior for virtually any state-space model, even models used for most complex systems. The BPF is a simple implementation of the PF but advanced filters such as the APF are necessary to boost the performance in complex and higher dimensional scenarios. While APF often obtains better results than the BPF, in some settings the performance is similar or even worse. The novel derivation presented in this paper will help the reader to (1) get some new insights about the APF, and (2) better understand when it is advised to use the APF. Note that the APF has been used in many real-world problems, such as mathematical finance [12], tracking applications [13], sensor networks [14] or electronics [15], among many others.

PREREQUISITES

This article is designed for researchers and students with a basic knowledge in particle filtering and importance sampling (IS). It aims at providing a new interpretation of the popular APF and a better understanding of this specific type of PF. The reader is expected to be familiar with basic notions of PFs [4], including the basic derivations of the BPF [16], [17], and some familiarity with the APF is preferable [10]. Finally, the novel perspective presented in this paper will be better understood with some basic knowledge of the importance sampling (IS) methodology [18].

PROBLEM STATEMENT

Bayesian filtering in dynamical models

We consider the following Markovian state-space model in discrete time ($t \in \mathbb{N}^+$):

$$\mathbf{x}_0 \quad \sim \quad p(\mathbf{x}_0), \tag{1}$$

$$\mathbf{x}_t \quad \sim \quad p(\mathbf{x}_t|\mathbf{x}_{t-1}), \tag{2}$$

$$\mathbf{y}_t \quad \sim \quad p(\mathbf{y}_t|\mathbf{x}_t), \tag{3}$$

where $\mathbf{x}_t \in \mathbb{R}^{d_x}$ represents the hidden (and random) system state at time instant t; $p(\mathbf{x}_0)$ is the a priori pdf of the state at t_0; $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ denotes the transition kernel (i.e., the conditional density of the state $\mathbf{x}_t$ given $\mathbf{x}_{t-1}$); $\mathbf{y}_t \in \mathbb{R}^{d_y}$ is the observation vector at time instant t, and is assumed to be conditionally independent of all other observations given the state $\mathbf{x}_t$; and $p(\mathbf{y}_t|\mathbf{x}_t)$ is the conditional pdf of $\mathbf{y}_t$ given $\mathbf{x}_t$, (i.e., the observation kernel) and is often referred as the *likelihood* of $\mathbf{x}_t$. The model described by Eqs. (1)–(3) includes a broad class of systems, both linear and nonlinear, with Gaussian or non-Gaussian perturbations.

The filtering problem consists of the probabilistic estimation of the hidden state $\mathbf{x}_t$ conditioned on all the observations available up to time instant t. For simplicity in the notation, we denote this set of observations $\mathbf{y}_{1:t} = \{\mathbf{y}_\tau\}_{\tau=1}^t$. Hence, the

goal is the computation of the filtering pdf $p(\mathbf{x}_t|\mathbf{y}_{1:t})$. When the observations are received sequentially, the additional goal is to process the observations recursively. To this end, the filtering task requires two steps at each time instant t:

1) **Propagation step.** The predictive pdf of the state, $p(\mathbf{x}_t|\mathbf{y}_{1:t-1})$, is computed as

$$p(\mathbf{x}_t|\mathbf{y}_{1:t-1}) = \int p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})d\mathbf{x}_{t-1}. \tag{4}$$

2) **Update step.** According to Bayes' theorem and from Eqs. (3) and (4), the filtering distribution is obtained by

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) \propto p(\mathbf{y}_t|\mathbf{x}_t) \int p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})d\mathbf{x}_{t-1}. \tag{5}$$

Unfortunately, these two steps require solving intractable integrals for most models of interest, and as a result, the filtering distribution cannot be exactly obtained. Thus, one has to search for approximate solutions, and to that end, Monte Carlo-based methods offer a rich trove of possibilities.

Particle filtering

Particle filtering is a Monte Carlo technique where the distributions are approximated by sets of weighted random samples in a sequential manner. For simplicity, here we assume that all algorithms always have M particles at each time instant t, although strategies for adapting the number of particles with time have been lately proposed [19].

Bootstrap particle filter: The BPF is arguably the most well known PF algorithm. It is often called sequential importance resampling filter, or in brief SIR filter, since it implements a sequential importance sampling step followed by a resampling step. The outline of the BPF is described in Algorithm 1. Note that the weights of Eq. (7) are normalized in the sense that $\sum_{m=1}^{M} w_t^{(m)} = 1$.

The auxiliary particle filter: The APF aims at improving the "quality" of the samples that are generated at each time instant. Here "quality" means how representative the drawn samples are of the target pdf. In principle, they are more representative if they come from parts of $\mathbb{R}^{d_x}$ that contain higher probability masses as measured by the target pdf. To get this improvement, unlike the BPF, the APF uses the new observation $\mathbf{y}_t$ in the prediction step, which is implemented by way of a delayed resampling (avoided at the end of the previous time instant).

The outline of the APF is described in Algorithm 2. First, Eq. (10) is used to generate pre-particles (particle projections) based on the expected value given the particles from the previous step. Note that the values $\bar{\mathbf{x}}_t^{(m)}$, $m = 1, 2, \cdots, M$, represent the centers of the kernels that compose the usual PF approximation of the predictive distribution of x_t,

$$p(\mathbf{x}_t|\mathbf{y}_{1:t-1}) \approx p^M(\mathbf{x}_t|\mathbf{y}_{1:t-1}) = \sum_{m=1}^{M} w_{t-1}^{(m)}p(\mathbf{x}_t|\mathbf{x}_{t-1}^{(m)}). \tag{9}$$

Algorithm 1: Bootstrap Particle Filter

1) Initialization. At time instant $t = 0$, draw M i.i.d. samples, $\mathbf{x}_0^{(m)}$, $m = 1, \ldots, M$, from the distribution $p(\mathbf{x}_0)$.

2) Recursive step. Let $\{\tilde{\mathbf{x}}_{t-1}^{(m)}\}_{m=1}^M$ be the particles (samples) resampled at time instant $t - 1$. At time instant t, proceed with the steps below.

    a) **Propagation step.** Propagate the particles as

$$\mathbf{x}_t^{(m)} \sim p(\mathbf{x}_t | \tilde{\mathbf{x}}_{t-1}^{(m)}), \qquad m = 1, ..., M. \tag{6}$$

    b) **Update step.** Compute the normalized weights as

$$w_t^{(m)} \propto p(\mathbf{y}_t | \mathbf{x}_t^{(m)}), \qquad m = 1, ..., M. \tag{7}$$

    c) **Resampling step.** Resample M times from the approximation of the filtering distribution as

$$\tilde{\mathbf{x}}_t^{(m)} \sim \sum_{j=1}^M w_t^{(j)} \delta(\mathbf{x}_t - \mathbf{x}_t^{(j)}). \tag{8}$$

Then, Eq. (11) is applied to compute the weights of $\bar{\mathbf{x}}_t^{(m)}$ using the current observation $\mathbf{y}_t$. In the literature, these weights are called pre-weights. The pre-weights are employed to perform a delayed resampling in Step 2(b). Hence, the particles are replicated taking into account not only the previous observation (like in BPF) but also using the information of the current observation. The final set of particles at time t, $\mathbf{x}_t^{(m)}$, comes from the propagation described by Eq. (12). Finally, the particles receive a weight computed according to Eq. (13). Note that this weight is proportional to the likelihood at time instant t, but inversely proportional to $p(\mathbf{y}_t | \bar{\mathbf{x}}_t^{(i^{(m)})})$. One can interpret that the mth particle is replicated proportionally to $w_{t-1}^{(m)} p(\mathbf{y}_t | \bar{\mathbf{x}}_t^{(m)})$ instead of $w_{t-1}^{(m)}$ (as in the BPF), and that the final IS weight "discounts" this augmentation of the pre-weight by dividing the likelihood of the particle by the factor $p(\mathbf{y}_t | \bar{\mathbf{x}}_t^{(i^{(m)})})$. Note that as a result, the APF approximates the filtering distribution at time instant t, $p(\mathbf{x}_t | \mathbf{y}_{1:t})$, with the random measure $p^M(\mathbf{x}_t | \mathbf{y}_{1:t}) = \sum_{m=1}^M w_t^{(m)} \delta(\mathbf{x}_t - \mathbf{x}_t^{(m)})$.

SOLUTION

The usual description of the APF presented in Algorithm 1 does not allow for a clear comparison with the BPF, and in particular, neither the full justification of the derivation of the pre-weights of Eq. (11), nor the demonstration of the validity of the weights of Eq. (13). In the following, we develop a generic framework based on MIS that eases the derivation of several existing PF algorithms. The MIS perspective also makes easier the understanding of some of the challenges of the APF and the proposing of novel algorithms.

> **Algorithm 2: Auxiliary Particle Filter**
>
> 1) Initialization. At time instant $t = 0$, draw M i.i.d. samples, $\mathbf{x}_0^{(m)}$, $m = 1, \ldots, M$, from the distribution $p(\mathbf{x}_0)$.
>
> 2) Recursive step. Let $\{\tilde{\mathbf{x}}_{t-1}^{(m)}, \bar{w}_{t-1}\}_{m=1}^{M}$ be the set of weighted particles (samples) generated at time instant $t - 1$. At time instant t, proceed with the steps below.
>
>   a) **Pre-weights computation step.**
>
>   i) Compute the mean of the pdf $p(\mathbf{x}_t | \mathbf{x}_{t-1}^{(m)})$ as
>
> $$\bar{\mathbf{x}}_t^{(m)} = \mathbb{E}_{p(\mathbf{x}_t | \mathbf{x}_{t-1}^{(m)})}[\mathbf{x}_t], \qquad m = 1, ..., M. \tag{10}$$
>
>   ii) Compute the normalized pre-weights of each kernel in the mixture as
>
> $$\lambda_t^{(m)} \propto p(\mathbf{y}_t | \bar{\mathbf{x}}_t^{(m)}) \bar{w}_{t-1}^{(m)}, \qquad m = 1, ..., M. \tag{11}$$
>
>   b) **Delayed resampling step.** Sample the indexes $i^{(m)}$, $m = 1, ... M$, with probability mass function (pmf) given by $\mathbb{P}(i^{(m)} = j) = \lambda_t^{(j)}$, $j \in \{1, ..., M\}$.
>
>   c) **Propagation step.** Simulate
>
> $$\mathbf{x}_t^{(m)} \sim p(\mathbf{x}_t | \mathbf{x}_{t-1}^{(i^{(m)})}), \qquad m = 1, ... M. \tag{12}$$
>
>   d) **Update step.** Compute the normalized weights as
>
> $$w_t^{(m)} \propto \frac{p(\mathbf{y}_t | \mathbf{x}_t^{(m)})}{p(\mathbf{y}_t | \bar{\mathbf{x}}_t^{(i^{(m)})})}, \qquad m = 1, ..., M. \tag{13}$$

Common Framework

The novel generic framework is presented in Algorithm 3. Interestingly, it avoids the explicit use of the resampling step which usually hinders the connection with IS.[1] Therefore, the generic algorithm is presented in a simpler adaptation-sampling-weighting manner. The initialization of the algorithm is performed as usual, and the recursive step also uses as basis the set of available weighted particles $\{\mathbf{x}_{t-1}^{(m)}, w_{t-1}\}_{m=1}^{M}$ from time $t - 1$. In Eq. (14), a proposal $\psi(\mathbf{x}_t)$ is selected/adapted. Note that this proposal is composed of the transition kernels $\{p(\mathbf{x}_t | \mathbf{x}_{t-1}^{(m)})\}_{m=1}^{M}$, whose positions depend on each of the M particles at $t - 1$. The different algorithms differ in the selection of the associated coefficients $\{\lambda_t^{(m)}\}_{m=1}^{M}$ (see next section). In Eq. (18), the M new particles are i.i.d. simulated from ψ. Finally, the normalized IS weights are proportionally computed as in Eq. (22). Note that this weight is supported by simple IS arguments. The numerator is proportional to the filtering distribution $p(\mathbf{x}_t | \mathbf{y}_{1:t-1}) \propto p(\mathbf{y}_t | \mathbf{x}_t) p(\mathbf{x}_t | \mathbf{y}_{1:t-1})$, where $p(\mathbf{x}_t | \mathbf{y}_{1:t-1})$ is substituted by the standard particle approximation of the predictive distribution given by Eq. (9). The denominator is the proposal $\psi(\mathbf{x}_t)$. Both numerator and denominator are evaluated at each particle (i.e., the usual targeted-divided-by-proposal IS weight).

---

[1] However, note that sampling from a proposal mixture is equivalent to a resampling step followed by a propagation step.

Algorithm 3: MIS interpretation of particle filtering

1) **Initialization.** At time $t = 0$, draw M i.i.d. samples, $\mathbf{x}_0^{(m)}$, $m = 1, \ldots, M$, from the distribution $p(\mathbf{x}_0)$, and set $\lambda_1^{(m)} = 1/M$.

2) **Recursive step.** Let $\{\mathbf{x}_{t-1}^{(m)}, w_{t-1}\}_{m=1}^M$ be the set of weighted particles (samples) generated at time $t - 1$. At time t, proceed with the steps below.

    a) **Proposal adaptation/selection.** Select the MIS proposal of the form

$$\psi_t(\mathbf{x}_t) = \sum_{m=1}^M \lambda_t^{(m)} p(\mathbf{x}_t | \mathbf{x}_{t-1}^{(m)}), \tag{14}$$

    where $\{p(\mathbf{x}_t | \mathbf{x}_{t-1}^{(m)})\}_{m=1}^M$ are the transition kernels centered at each of the M particles of $t-1$, and $\{\lambda_t^{(m)}\}_{m=1}^M$ are the associated coefficients computed by

$$\lambda_t^{(m)} = w_{t-1}^{(m)}, \qquad m = 1, ..., M, \quad \text{if the applied filter is the BPF,} \tag{15}$$

$$\lambda_t^{(m)} \propto p(\mathbf{y}_t | \bar{\mathbf{x}}_t^{(m)}) w_{t-1}^{(m)}, \qquad m = 1, ..., M, \quad \text{if the applied filter is the APF,} \tag{16}$$

    where

$$\bar{\mathbf{x}}_t^{(m)} = \mathbb{E}_{p(\mathbf{x}_t | \mathbf{x}_{t-1}^{(m)})} [\mathbf{x}_t], \qquad m = 1, ..., M. \tag{17}$$

    are the means of the kernels.

    b) **Sampling.** Draw samples according to

$$\mathbf{x}_t^{(m)} \sim \psi_t(\mathbf{x}_t), \qquad m = 1, ..., M. \tag{18}$$

    c) **Weighting.** Compute the normalized IS weights by

$$w_t^{(m)} \quad \propto \quad \frac{p(\mathbf{x}_t^{(m)} | \mathbf{y}_{1:t})}{\psi_t(\mathbf{x}_t^{(m)})} \tag{19}$$

$$= \quad \frac{p(\mathbf{y}_t | \mathbf{x}_t^{(m)}) p(\mathbf{x}_t^{(m)} | \mathbf{y}_{1:t-1})}{\psi_t(\mathbf{x}_t^{(m)})} \tag{20}$$

$$\approx \quad \frac{p(\mathbf{y}_t | \mathbf{x}_t^{(m)}) \sum_{j=1}^M w_{t-1}^{(j)} p(\mathbf{x}_t^{(m)} | \mathbf{x}_{t-1}^{(j)})}{\psi_t(\mathbf{x}_t^{(m)})} \tag{21}$$

$$= \quad \frac{p(\mathbf{y}_t | \mathbf{x}_t^{(m)}) \sum_{j=1}^M w_{t-1}^{(j)} p(\mathbf{x}_t^{(m)} | \mathbf{x}_{t-1}^{(j)})}{\sum_{j=1}^M \lambda_t^{(j)} p(\mathbf{x}_t^{(m)} | \mathbf{x}_{t-1}^{(j)})} \tag{22}$$

$$\approx \quad \frac{p(\mathbf{y}_t | \mathbf{x}_t^{(m)}) w_{t-1}^{(m)}}{\lambda_t^{(m)}}, \qquad m = 1, ..., M. \tag{23}$$

Multiple importance sampling perspective

In Alg. 3, we have described a generic PF from the MIS perspective where the traditional propagation-update-resampling steps are now replaced by adaptation-sampling-weighting steps. The sampling in the generic PF framework is performed by Eq. (18), where M samples are simulated from the mixture proposal.

Let us now focus on the adaptation/selection of the proposal, and its impact on the weighting step of Eq. (22). It is well

	BPF	APF	IAPF
$\lambda_t^{(m)}$	$w_{t-1}^{(m)}$	$\propto p(\mathbf{y}_t|\bar{\mathbf{x}}_t^{(m)})w_{t-1}^{(m)}$	$\propto \dfrac{p(\mathbf{y}_t|\bar{\mathbf{x}}_t^{(m)})\sum_{j=1}^{M}w_{t-1}^{(j)}p(\bar{\mathbf{x}}_t^{(m)}|\mathbf{x}_{t-1}^{(j)})}{\sum_{j=1}^{M}p(\bar{\mathbf{x}}_t^{(m)}|\mathbf{x}_{t-1}^{(j)})}$
$w_t^{(m)}$	$\propto p(\mathbf{y}_t|\mathbf{x}_t^{(m)})$	$\propto \dfrac{p(\mathbf{y}_t|\mathbf{x}_t^{(m)})}{p(\mathbf{y}_t|\bar{\mathbf{x}}_t^{(im)})}$	$\propto \dfrac{p(\mathbf{y}_t|\mathbf{x}_t^{(m)})\sum_{j=1}^{M}w_{t-1}^{(j)}p(\mathbf{x}_t^{(m)}|\mathbf{x}_{t-1}^{(j)})}{\sum_{j=1}^{M}\lambda_t^{(j)}p(\mathbf{x}_t^{(m)}|\mathbf{x}_{t-1}^{(j)})}$

TABLE I

THE COEFFICIENTS $\lambda_t^{(m)}$ OF THE PROPOSAL MIXTURE, AND THE IS WEIGHTS $\bar{w}_t^{(m)}$ FOR THE BPF, APF, AND IAPF. NOTE THAT IN ALL CASES, THE WEIGHTS MUST BE NORMALIZED SUCH $\sum_{m=1}^{M}\lambda_t^{(m)}=1$ AND $\sum_{m=1}^{M}\bar{w}_t^{(m)}=1$.

known that in MIS, when we draw i.i.d. samples from a mixture, the performance improves when the discrepancy between the proposal and the target density decreases [18]. In other words, it is important that the numerator of Eq. (21), $p(\mathbf{x}_t|\mathbf{y}_{1:t})$ and the denominator, $\psi(\mathbf{x}_t)$, are as similar as possible. Note that, in the IS weight, the numerator in (22) is the product of the likelihood and a mixture of weighted kernels, while the denominator is another mixture with the same kernels. The fact that both mixtures have the same kernels with (potentially) different coefficients is key in the derivation of the different algorithms, and in the understanding of their performance under different models.

Table I summarizes the choice of pre-weights and weights of the BPF, APF, and the improved APF (IAPF) [20], a recent algorithm that also fits in the MIS framework presented above. In the following, we provide details of the new interpretations of each algorithm.

BPF from the MIS perspective

Let us interpret the BPF of Algorithm 1 from the generic Algorithm 3. First, set $\lambda_t^{(j)}=w_{t-1}^{(j)}$ in Eq. (14). Sampling M times from this mixture $\psi_t(\mathbf{x}_t)=\sum_{m=1}^{M}w_{t-1}^{(m)}p(\mathbf{x}_t|\mathbf{x}_{t-1}^{(m)})$ is equivalent to the resampling step at time instant $t-1$ from Eq. (7) followed by the propagation of particles in Eq. (6). If we plug $\psi_t(\mathbf{x}_t)$ in the denominator of Eq. (22), since $\psi(\mathbf{x}_t)=p(\mathbf{x}_t^{(m)}|\mathbf{y}_{1:t-1})$, it cancels out the right-hand-side of the numerator, and only the likelihood in the numerator remains. This allows us to recover the traditional BPF weight $w_t^{(m)}\propto p(\mathbf{y}_t|\mathbf{x}_t^{(m)})$ of Eq. (7), which is simply the likelihood evaluated at the associated particle (cf. (23)). We point out that for the BPF, the approximation in (23) becomes an equality because the summations in the numerator and denominator of (22) are identical, and before we implement the approximation in (23), they cancel each other.

In BPF, the strategy in the choice of the coefficients $\lambda_t^{(m)}=w_{t-1}^{(m)}$ in the proposal mixture is matching the mixture of the numerator of Eq. (22), i.e., the likelihood in the numerator can be seen as the mismatch factor between the numerator and denominator. This explains the challenges of BPF when there are very informative observations: the likelihood can amplify

severely some kernels w.r.t. others, increasing the mismatch between the target and the proposal distributions.

Figure 1 shows the traditional and the MIS perspective of the BFP. On the left side, we display the traditional propagation, weighting, and resampling steps, which are repeated at each time step. On the right side, we display the MIS interpretation, with the proposal selection, the sampling from a mixture, and the weighting. In the MIS perspective, the sampling step from the mixture of proposals (ψ_t in Eq. (14)) is equivalent to the resampling followed by the propagation of the kernels in the traditional perspective. In the BPF, there is no proposal/selection adaptation, since the mixture proposal $\psi_t(\mathbf{x}_t)$ depends exclusively on the kernels and the IS weights of the previous time step.
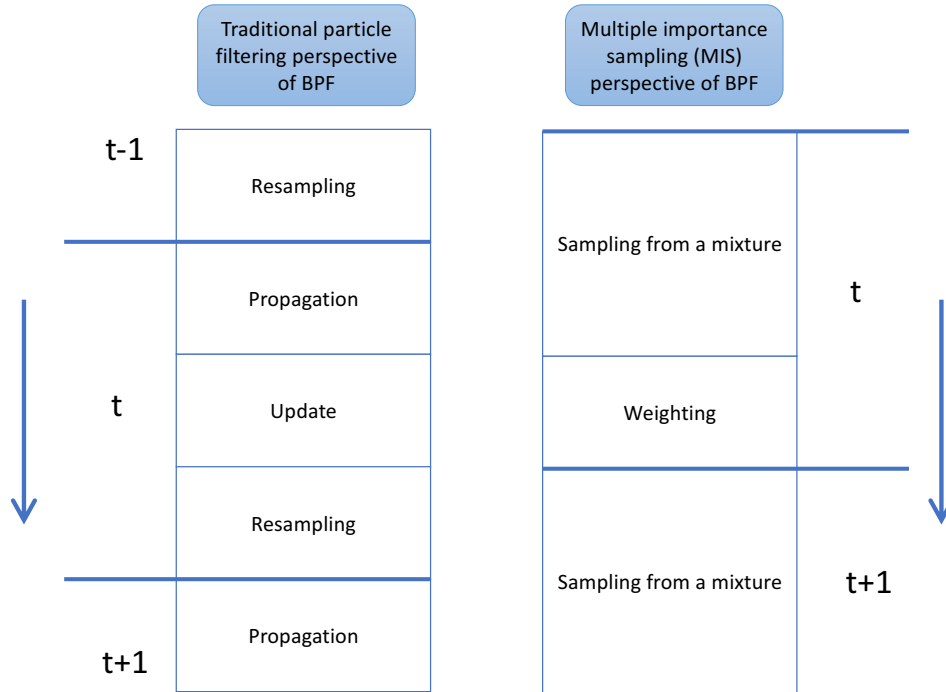
Fig. 1. Two different perspectives for explaining the BPF. On the left side, the traditional perspective with the propagation, update, and resampling steps. On the right side, the MIS perspective with the sampling from a mixture and the weighting steps.

APF from the MIS perspective

From the MIS perspective, the choice of APF seems more adequate than that of the BPF. Even though the APF still contains the same kernels in the proposal mixture, its coefficients $\lambda_t^{(m)} \propto p(\mathbf{y}_t|\bar{\mathbf{x}}_t^{(m)})w_{t-1}^{(m)}$ are those of the BPF but amplified by the factor $p(\mathbf{y}_t|\bar{\mathbf{x}}_t^{(m)})$. This factor is simply the likelihood evaluated at the mean of the mth kernel. For instance, for additive Gaussian perturbations in the state model, the mixture proposal of APF is the same as the mixture proposal of the BPF where each kernel has been amplified by the value of the function $p(\mathbf{y}_t|\mathbf{x}_t)$ at its center. Interestingly, this approach now tries to minimize the mismatch of $\psi(\mathbf{x}_t)$ with the whole numerator in Eq. (22) and not only with a part of it, that is, with the factor

that corresponds to the predictive pdf $p(x_t|y_{1:t-1})$. In many state-space models of interest, the APF has improved performance due to this better matching mixture proposal. Note that this approximation of the target is especially good when the kernels are "far apart" (more precisely, when their distance is high w.r.t. their width).

Within the MIS perspective, we also derive the weights of APF from Eq. (22) under the assumption of well separated kernels. Let us consider that the mth particle $x_t^{(m)}$ has been simulated from the kernel $i^{(m)}$. If the other kernels $p(\mathbf{x}_t|\mathbf{x}_{t-1}^{(j)})$, with $j = 1, ..., M$ and $j \neq i^{(m)}$, take small values when evaluated at $\bar{\mathbf{x}}_t^{(i^{(m)})}$ (which is equivalent to our previous assumption of the kernels being separated enough), one can approximate the weight as

$$w_t^{(m)} \propto \frac{p(\mathbf{x}_t^{(m)}|\mathbf{y}_{1:t})}{\psi_t(\mathbf{x}_t^{(m)})} \tag{24}$$

$$\approx \frac{p(\mathbf{y}_t|\mathbf{x}_t^{(m)}) \sum_{j=1}^{M} w_{t-1}^{(j)} p(\mathbf{x}_t^{(m)}|\mathbf{x}_{t-1}^{(j)})}{\sum_{j=1}^{M} \lambda_t^{(j)} p(\mathbf{x}_t^{(m)}|\mathbf{x}_{t-1}^{(j)})} \tag{25}$$

$$\approx \frac{p(\mathbf{y}_t|\mathbf{x}_t^{(m)}) w_{t-1}^{(i^{(m)})} p(\mathbf{x}_t^{(m)}|\mathbf{x}_{t-1}^{(i^{(m)})})}{\lambda_t^{(i^{(m)})} p(\mathbf{x}_t^{(m)}|\mathbf{x}_{t-1}^{(i^{(m)})})} \tag{26}$$

$$\propto \frac{p(\mathbf{y}_t|\mathbf{x}_t^{(m)}) w_{t-1}^{(i^{(m)})} p(\mathbf{x}_t^{(m)}|\mathbf{x}_{t-1}^{(i^{(m)})})}{p(\mathbf{y}_t|\bar{\mathbf{x}}_t^{(i^m)}) w_{t-1}^{(i^{(m)})} p(\mathbf{x}_t^{(m)}|\mathbf{x}_{t-1}^{(i^{(m)})})} \tag{27}$$

$$= \frac{p(\mathbf{y}_t|\mathbf{x}_t^{(m)})}{p(\mathbf{y}_t|\bar{\mathbf{x}}_t^{(i^m)})}, \tag{28}$$

recovering the traditional APF of Eq. (13) (or Eq. (23)). The first approximation in (25) comes from approximating the predictive distribution $p(\mathbf{x}_t|\mathbf{y}_{1:t})$. The second approximation in (27) assumes that the evaluation of a sample in the whole mixture is equivalent to the evaluation in the kernel where the sample was simulated from. Hence, the latter approximation assumes that the kernels have negligible overlaps, which can be a too strict assumption and would explain why for some models, the APF can perform worse than the BPF.

Figure 2 displays the two interpretations of the APF described above. On the left side, we display the traditional pre-weights computation, delayed resampling, propagation, and weight computation, which are repeated at each time step. On the right side, we see the MIS approach, with the proposal selection, the (mixture) sampling, and the weighting steps. Again, the update step in the former perspective is equivalent to the weighting step in the latter perspective. In the MIS perspective, the sampling from the mixture of proposals is again equivalent to the resampling followed by the propagation of the kernels in the traditional perspective. The difference now (w.r.t. the BPF), is that in the traditional perspective there is a pre-weighting step for modifying the weights before the resampling step. This pre-weighting step is clearly equivalent to selecting the coefficients $\{\lambda_t^{(j)}\}_{j=1}^{M}$ of the kernels in the mixture proposal $\psi_t(\mathbf{x}_t)$ in the MIS perspective.
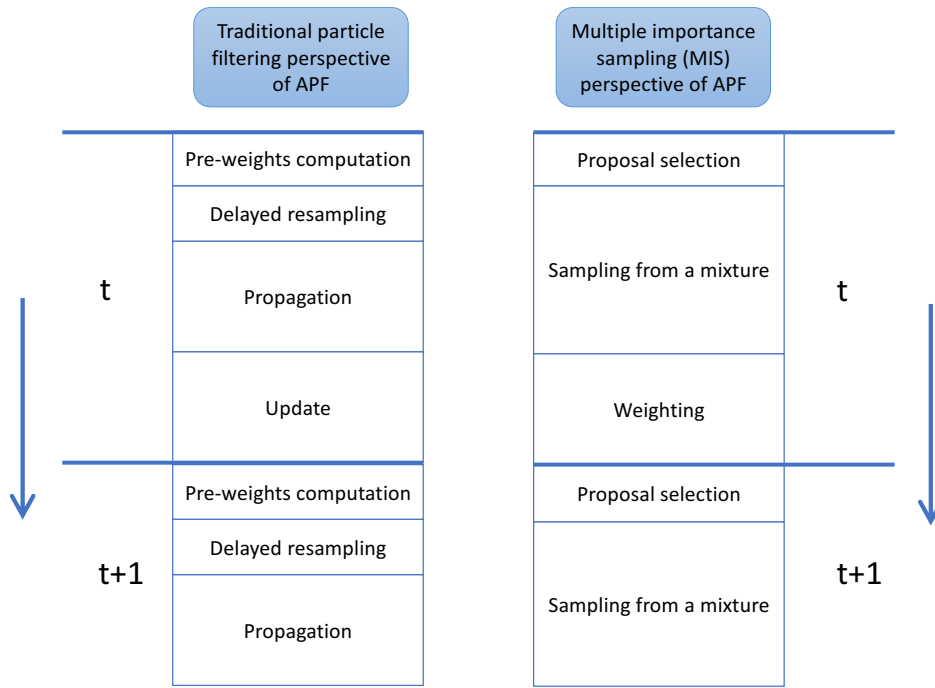
Fig. 2. Two different perspectives for explaining the APF. On the left side, the traditional interpretation with the pre-weighting, delayed resampling, propagation, and weighting steps. On the right side, the MIS perspective with the proposal selection, the (mixture) sampling, and the weighting steps.

IAPF from the MIS perspective

The improved APF (IAPF) has been recently proposed in order to reduce the flaws of the APF that we have listed in the previous section [20]. It takes into account the overlap of the kernels in order to both select the coefficients of the mixture proposal, and approximate the IS weights. Algorithm 4 describes the IAPF within the novel MIS framework. First, the coefficient of the mth kernel of the mixture in Eq. (30) is computed by evaluating the ratio between the target distribution and the equally-weighted mixture of proposals at the center of the kernel. This allows for taking into account the position of the rest of the kernels when computing the coefficient of a specific kernel (unlike in APF). Second, the weight in (32) is computed as it is traditionally done in IS, i.e., it is simply the ratio between the target and the whole proposal mixture evaluated at the sample.

Note that the IAPF becomes the APF when the overlap between the kernels decreases. In [20], BPF, APF, and IAPF, are tested on a linear-Gaussian model, where the transition kernels are particularly wide, and the observations are very informative (low observation variance). This scenario is in principle particularly adverse for the APF, since the assumptions on the selection of the proposal do not hold and the approximation of the IS weights in Eq. (27) is poor. This explains why the IAPF can obtain much smaller MSE than the APF. Note however that the computation of the weights of IAPF in Eq. (32) requires $2M^2$

extra kernel evaluations unlike the IS weights of APF in Eq. (28). Similarly, the computation of the coefficients $\lambda_t^{(m)}$ is also more costly in IAPF. The IAPF can also be explained from the two perspectives (traditional and MIS) described in Fig. 2 for the APF.

A NUMERICAL EXAMPLE

We now demonstrate the differences in the functions used for generation of new particles among the BPF, APF and IAPF. We use a toy example, where the model is linear and Gaussian, and where we focus on the transition of a generic PF, with $M = 4$ particles, from time instant $t-1$ to t. The transition kernels are Gaussian, $p(x_t|x_{t-1}^{(m)}) = \mathcal{N}(x_t|x_{t-1}^{(m)}, \sigma_x^2)$, the likelihood function $p(y_t|x_t) = \mathcal{N}(x_t; y_t, \sigma_y^2)$ is also Gaussian, and the weights at $t-1$ are $w_{t-1} = [0.03, 0.16, 0.16, 0.65]$.

Figure 3(a) displays the weighted kernels $\{w_{t-1}^{(m)} p(x_t|x_{t-1}^{(m)})\}_{m=1}^M$, and the likelihood function. The green circles represent the value of the likelihood at the center of each kernel, i.e., $\{p(y_t|\bar{x}_t^{(m)})\}_{m=1}^4$, where $\bar{x}_t^{(m)}$ is defined in Eq. (17). Figure 3(b) shows the target distribution $p(x_t|y_{1:t}) = p(y_t|x_t)p(x_t|y_{1:t-1}) \approx p(y_t|x_t) \sum_{m=1}^4 w_t^{(m)} p(x_t|x_{t-1}^{(m)})$, and the resulting BPF proposal, whereas Fig. 3(c) displays the same for the APF, and Fig. 3(d) depicts the IAPF proposal. We recall that in all cases,

the proposal $\psi_t(\mathbf{x}_t)$ has the form of Eq. (14), with $\lambda_t^{(m)}$ defined in Table I for each algorithm. In this example, the proposal of the BPF presents a large mismatch with the target distribution, while the IAPF mimics it very accurately. The APF achieves an intermediate performance in terms of proposal adequacy. The MIS perspective helps to understand why a filter that fails at replicating well the target with the proposal $\psi_t(\mathbf{x}_t)$ will have a poor performance. See for instance the numerical example in [20, Section IV-A], where there is a sharp likelihood (because of the small noise variance), and a high width of the kernels (due to large transition noise). This setup is particularly adverse for APF according to the MIS derivation in the previous section, since the kernels will have a big overlap, and hence the assumption for the approximation in Eq. (27) does not hold. Therefore, the proposal selection $\psi_t(\mathbf{x}_t)$ will not be close to the target, which explains the poor performance of APF in this example.

Conclusions

In these notes, we derived the auxiliary particle filter (APF) from the perspective of multiple importance sampling. With the new interpretation, we provided insights about the assumptions and approximations that are required to derive the APF. The insights are of great use for understanding when the APF does not have a satisfactory performance. The derivation also shows the relationship between the APF and the BPF.

Acknowledgments

References

[1] N. Gordon, D. Salmond, and A. F. M. Smith, "Novel approach to nonlinear and non-Gaussian Bayesian state estimation," *IEE Proceedings-F Radar and Signal Processing*, vol. 140, pp. 107–113, 1993.

[2] A. Doucet, N. de Freitas, and N. Gordon, Eds., *Sequential Monte Carlo Methods in Practice*. New York (USA): Springer, 2001.

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

[4] P. M. Djurić, J. H. Kotecha, J. Zhang, Y. Huang, T. Ghirmai, M. F. Bugallo, and J. Míguez, "Particle filtering," *IEEE Signal Processing Magazine*, vol. 20, no. 5, pp. 19–38, September 2003.

[5] O. Cappé, S. J. Godsill, and E. Moulines, "An overview of existing methods and recent advances in sequential Monte Carlo," *Proceedings of the IEEE*, vol. 95, no. 5, pp. 899–924, 2007.

[6] A. Doucet and A. M. Johansen, "A tutorial on particle filtering and smoothing: fifteen years later," *technical report*, 2008.

[7] O. Hlinka, F. Hlawatsch, and P. M. Djuric, "Distributed particle filtering in agent networks: A survey, classification, and comparison," *IEEE Signal Processing Magazine*, vol. 30, no. 1, pp. 61–81, 2013.
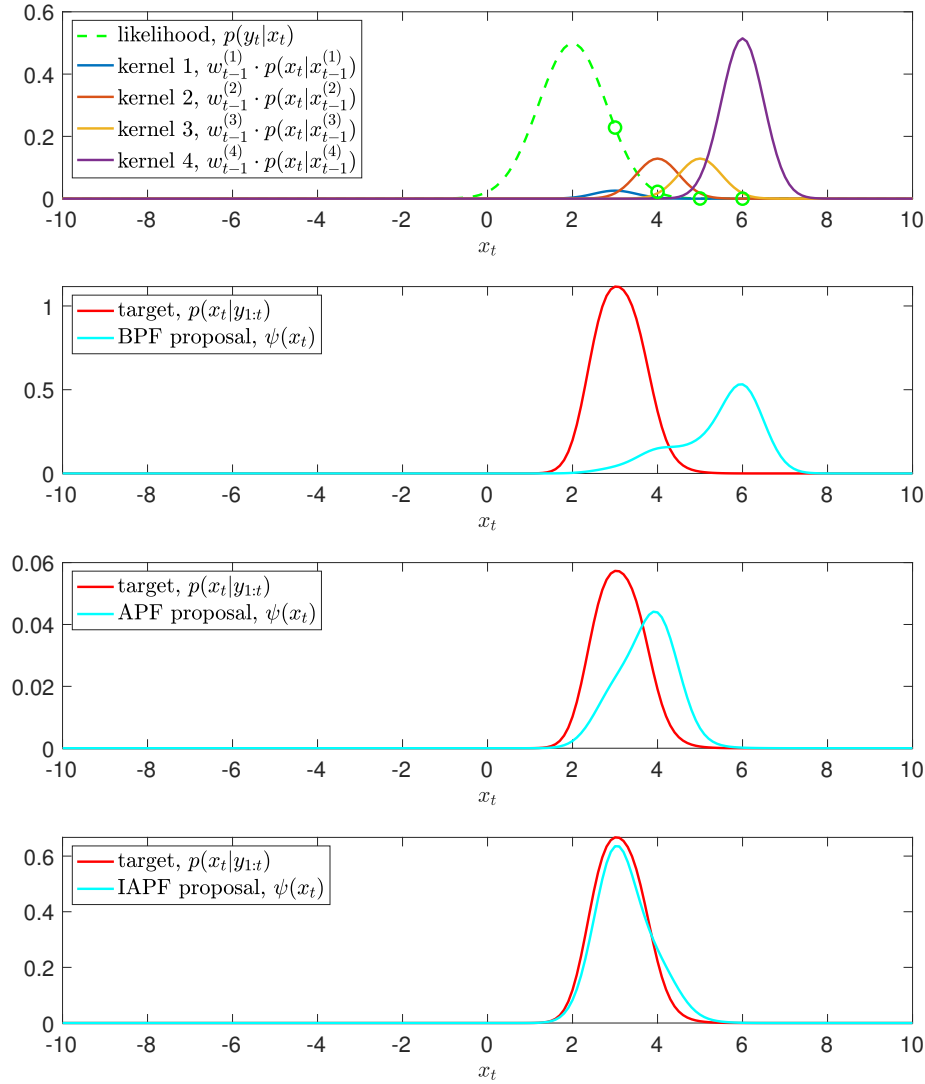
Fig. 3. Example with $M = 4$ particles, and Gaussian kernels $p(x_t|x_{t-1}^{(m)}) = \mathcal{N}(x_t|x_{t-1}^{(m)}, \sigma_x^2)$. (a) Weighted kernels $\{\bar{w}_{t-1}^{(m)} p(x_t|x_{t-1}^{(m)})\}_{m=1}^M$, and likelihood $p(y_t|x_t) = \mathcal{N}(x_t; y_t, \sigma_y^2)$. In green circles, the value of the likelihood at the center of each kernel, i.e., $\{p(y_t|\bar{x}_t^{(m)})\}_{m=1}^4$. (b) Target distribution $p(x_t|y_{1:t}) = p(y_t|x_t)p(x_t|y_{1:t-1}) \approx p(y_t|x_t) \sum_{m=1}^4 \bar{w}_t^{(m)} p(x_t|x_{t-1}^{(m)})$, and BPF proposal ($\lambda_t^{\text{BPF}} = [0.03, \ 0.16, \ 0.16, \ 0.65]$). (c) Target distribution and APF proposal ($\lambda_t^{\text{APF}} = [0.6713, \ 0.3221, \ 0.0065, \ 0.0001]$). (d) Target distribution and and IAPF proposal ($\lambda_t^{\text{IAPF}} = [0.7657, 0.2276, 0.0066, 0.0001]$).

[8] P. Del Moral and A. Doucet, "Particle methods: An introduction with applications," in *ESAIM: Proceedings*, vol. 44. EDP Sciences, 2014, pp. 1–46.

[9] X. Wang, T. Li, S. Sun, and J. Corchado, "A survey of recent advances in particle filters and remaining challenges for multitarget tracking," *Sensors*, vol. 17, no. 12, p. 2707, 2017.

[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] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, vol. 82, pp. 35–45, 1960.

[12] N. Whiteley, A. M. Johansen, and S. Godsill, "Monte carlo filtering of piecewise deterministic processes," *Journal of Computational and Graphical Statistics*, vol. 20, no. 1, pp. 119–139, 2011.

[13] R. Karlsson and N. Bergman, "Auxiliary particle filters for tracking a maneuvering target," in *Proceedings of the 39th IEEE Conference on Decision and Control (Cat. No. 00CH37187)*, vol. 4.   IEEE, 2000, pp. 3891–3895.

[14] P. M. Djuric, M. Vemula, and M. F. Bugallo, "Target tracking by particle filtering in binary sensor networks," *IEEE Transactions on signal processing*, vol. 56, no. 6, pp. 2229–2238, 2008.

[15] M. S. Haque, S. Choi, and J. Baek, "Auxiliary particle filtering-based estimation of remaining useful life of igbt," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 3, pp. 2693–2703, 2018.

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

[17] L. Martino, J. Read, V. Elvira, and F. Louzada", "Cooperative parallel particle filters for on-line model selection and applications to urban mobility," *Digital Signal Processing*, vol. 60, pp. 172–185, 2017.

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

[19] V. Elvira, J. Míguez, and P. 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.

[20] V. Elvira, L. Martino, M. F. Bugallo, and P. Djurić, "In search for improved auxiliary particle filters," in *Signal Processing Conference (EUSIPCO), 2018 Proceedings of the 26th European*.   IEEE, 2018, pp. 1–5.