In Search for Improved Auxiliary Particle Filters

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

[⋆] IMT Lille Douai & CRIStAL, Villeneuve d'Ascq (France).
[†] University Carlos III of Madrid, Leganés (Spain).
[‡] Stony Brook University, Stony Brook (USA).

***Abstract*—In designing a particle filter, the most important task is choosing the importance function that can generate good particles. If the importance function, also called proposal, does a satisfactory job, the particles of the filter are placed in parts where the explored state space has high probability mass. Further, the weights of these particles are not too disparate in values. An important class of particle filtering that uses a clever approach to create good importance functions is known as auxiliary particle filtering. In this paper, we first analyze the approximations used for computing the particle weights of the standard auxiliary particle filter. We show that these approximations can be detrimental to the performance of the auxiliary particle filter. Further, we propose a more comprehensive evaluation of the weights, which leads to a much enhanced performance of the auxiliary particle filter. We also demonstrate the improvements with computer simulations.**

I. Introduction

Particle filters are a well-known Monte Carlo methodology for sequential estimation of hidden processes (states) based on state-space models. Particle filters provide approximations of filtering and predictive probability density functions (pdfs) recursively [1], [2], [3]. The approximations are based on samples (particles) of the states and weights associated to them. Particle filters are based on a clever combination of the sequential importance sampling (SIS) technique and resampling [4], [5], [6], [7].

Since the publication of [8], the number of works on particle filtering has been continuously increasing and the number of areas where they have been applied has not stopped growing. The reason behind this popularity is their ability to work in highly nonlinear and non-Gaussian settings and their relative simplicity of implementation. By now, however, it has also been established that they suffer from two notorious problems: sample degeneracy and impoverishment [6], [7], [9].

Sample degeneracy is an inherent drawback of the SIS process. Indeed, after a few recursions of the sequential procedure, only a very small number of particles have non-negligible weights, which means that they are the only particles that approximate the desired pdf. This often results in a rather poor approximation of all the ensuing estimates. A standard recipe for preventing sample degeneracy is implementing a step known as resampling. With resampling, copies of particles with large weights are kept and those with low weights are discarded [10], [11].

The use of resampling, however, introduces another problem, often referred to as sample impoverishment, path degeneracy, or loss of diversity in the population of particles. When the degeneracy problem is more serious, then the impoverishment after resampling will be more serious as well. The loss of diversity is due to the distribution of the resampled particles, which can differ substantially from the target [12], [13, Appendix 3]. The loss of diversity, if significant, may lead to loss of the tracked state.

The problem of degeneracy is often overcome by applying the resampling only intermittently. In other words, one implements resampling when some conditions are met. One such condition is based on the concept of effective sample size (ESS), which is a measure of the "quality" of the samples and obtained from their weights. If the ESS is smaller than a pre-specified threshold, resampling is applied [14], [15]. Another simple strategy consists in applying a regularization in the resampling procedure, i.e., in applying an additional perturbation to the resampled particles (it is based on the kernel density estimation idea) [4], [7], [16]. Another approach is based on the use of a nonlinear transformation of the weights in order to force flatness in the discrete probability mass defined by the weights, before applying the resampling step [2], [17]. In adaptive importance sampling (AIS) schemes [18], also involving resampling steps, the same effect can be obtained by a more sophisticated definition of the importance weights [19], [20].

One can easily argue that the best way to mitigate sample impoverishment is by generating good particles. The optimal theoretical solution consists in using the optimal proposal density, which provides the smallest variance of the importance weights [4], [6], [7]. Unfortunately, the use of the optimal proposal density is impossible in most practical cases. Thus, one resorts to approximations of the optimal proposal density [6], [7], [16]. Another possibility is to apply Markov Chain Monte Carlo (MCMC) sampling within the particle filtering (e.g, as in the resample-move algorithm [21]). Other techniques suggest the application of optimization procedures for modifying the positions of the particles (see for instance [22]). In the same fashion, there are methods that push the particles to be in high-likelihood regions by using bridging densities and tempering strategies, in essence, introducing intermediate distributions between the prior and the likelihood

V.E. acknowledges support from the *Agence Nationale de la Recherche* of France under PISCES project (ANR-17-CE40-0031-01), the Fulbright program, and the Marie Curie Fellowship (FP7/2007-2013) under REA grant agreement n. PCOFUND-GA-2013-609102, through the PRESTIGE program.

ISBN 978-90-827970-1-5 © EURASIP 2018

[9], [23].

A method that has the same aim and that is of interest in this paper is the auxiliary particle filtering method [24]. The auxiliary particle filter (APF) tries to mimic the particle propagation obtained by drawing from the optimal proposal pdf, at the expense of an additional computational cost. Namely, it places first new particles deterministically at locations with high probabilities, and computes their weights by using the new observations. These weights are then exploited for deciding which particles from the previous time step to propagate. The APF has also attracted high interest, which is attested by a continuous stream of papers where the main idea behind them is exploited in creative ways. Some very recent contributions include [25], [26], [27].

In this paper, we return to the classical APF and investigate the approximations used to compute the weights of its intermediate samples and particles. We show that when these approximations are too harsh, the filter cannot perform well. If by contrast, these weights are computed more accurately, with the APF we can achieve very much improved performance over the standard particle filter.

The paper is organized as follows. In the next section, we review the standard auxiliary particle filter. In the following section, we present the derivation of an improved auxiliary particle filter. In Section IV, we compare the performance of the proposed filter with that of the standard APF and the bootstrap particle filter (BPF). In Section V, we make some final comments.

II. Background

A. Bayesian filtering in dynamical models

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

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

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

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

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

The stochastic filtering problem consists of the probabilistic estimation of the hidden state $\mathbf{x}_t$ conditioned on all the observations available up to time t denoted by $\boldsymbol{y}_{1:t}$, i.e., the computation of the filtering pdf $p(\mathbf{x}_t|\boldsymbol{y}_{1:t})$. At each time t, the filtering task requires two steps. First, the propagation step computes the predictive pdf of the state, $p(\mathbf{x}_t|\boldsymbol{y}_{1:t-1})$,

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

Algorithm 1: Auxiliary Particle Filter

1) Initialization. At time $t = 0$, draw M i.i.d. samples, $\mathbf{x}_0^{(m)}$, from the distribution $p(\mathbf{x}_0)$, and set $w_0^{(m)} = \frac{1}{M}$, $m = 1, ..., M$.
2) Recursive step. Let $\{\mathbf{x}_{t-1}^{(m)}\}_{m=1}^M$ be the particles (samples) generated at time $t-1$. At time t, proceed with the steps below.
   a) Compute the mean of the pdf $p(\mathbf{x}_t|\mathbf{x}_{t-1}^{(m)})$ as
   $$\bar{\mathbf{x}}_t^{(m)} = \mathbb{E}_{p(\mathbf{x}_t|\mathbf{x}_{t-1}^{(m)})}[\mathbf{x}_t], \qquad m = 1, ..., M. \tag{6}$$
   b) Compute the normalized weights of each kernel in the mixture as
   $$\lambda_t^{(m)} \propto p(\boldsymbol{y}_t|\bar{\mathbf{x}}_t^{(m)})w_{t-1}^{(m)}, \qquad m = 1, ..., M, \tag{7}$$
   and build the importance sampling (IS) proposal as
   $$q^{\text{APF}}(\mathbf{x}_t) = \sum_{m=1}^{M} \lambda_t^{(m)} p(\mathbf{x}_t|\mathbf{x}_{t-1}^{(m)}). \tag{8}$$
   c) Draw M i.i.d. samples from $q^{\text{APF}}(\mathbf{x}_t)$ in two steps:
      i) select the indexes $i^{(m)}$, $m = 1, ...M$, with pmf given by $\mathbb{P}(i^{(m)} = j) = \lambda_t^{(j)}$, $j \in \{1, ..., M\}$.
      ii) simulate $\mathbf{x}_t^{(m)} \sim p(\mathbf{x}_t|\mathbf{x}_{t-1}^{(i^m)})$, $\qquad m = 1, ...M$.
   d) Compute the weights as
   $$w_t^{(m)} = \frac{p(\boldsymbol{y}_t|\mathbf{x}_t^{(m)})}{p(\boldsymbol{y}_t|\bar{\mathbf{x}}_t^{(i^m)})}, \qquad m = 1, ..., M. \tag{9}$$
   e) Normalize the weights
   $$\bar{w}_t^{(m)} = \frac{w_t^{(m)}}{\sum_{k=1}^M w_t^{(k)}}, \qquad m = 1, ..., M. \tag{10}$$

Second, using Bayes' theorem together with Eq. (4), the filtering distribution is obtained at the update state as

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

Unfortunately the two previous steps require intractable integrals for most models. Hence, the filtering distribution cannot be exactly obtained, and Monte Carlo approximations are required.

B. The auxiliary particle filter

Particle filters are Monte Carlo algorithms for approximating the sequence of filtering distributions $p(\mathbf{x}_t|\boldsymbol{y}_{1:t})$ in a sequential manner. At each time step, the distribution is approximated with a random set of weighted particles. The BPF is arguably the simplest version of the algorithm [28]. The APF aims at improving the way that the samples are simulated at each time step. To that end, unlike the BPF, the APF uses the new observation $\boldsymbol{y}_t$ in the prediction step. The outline of the APF is described in Algorithm 1. Eq. (7) computes what in the literature is usually called pre-weights. Here, we prefer to call them mixture weights, since they are useful to define the importance function (or proposal) $q^{\text{APF}}(\mathbf{x}_t)$ in Eq. (8). Note that as a result, the APF approximates the filtered distribution $p(\mathbf{x}_t|\boldsymbol{y}_{1:t})$ at time t with the random measure $p^M(\mathbf{x}_t) := \sum_{m=1}^M \bar{w}_t^{(m)} \delta(\mathbf{x}_t - \bar{\mathbf{x}}_t^{(m)})$.

ISBN 978-90-827970-1-5

III. Improved Auxiliary Particle Filter

In this section, we propose a novel *improved* APF (IAPF) based on a different importance function. We justify the choice of this function theoretically and derive the APF as a particular case of the IAPF, stating the implicit approximations in this derivation. Moreover, we discuss the mixture importance function used in the BPS, APS, and IAPF. The IAPF is displayed in Algorithm 2. In step 2(a), the means of the M kernels $p(\mathbf{x}_t|\mathbf{x}_{t-1}^{(m)})$ are computed. This set of kernels builds the mixture proposal

$$q^{\text{IAPF}}(\mathbf{x}_t) = \sum_{m=1}^{M} \lambda_t'^{(m)} p(\mathbf{x}_t|\mathbf{x}_{t-1}^{(m)}). \tag{11}$$

The set of weights in the mixture, $\{\lambda_t'^{(m)}\}_{m=1}^M$, in Eq. (16) is justified as follows. The targeted distribution at time t is

$$\begin{aligned} p(\mathbf{x}_t|\boldsymbol{y}_{1:t}) &\propto p(\boldsymbol{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\boldsymbol{y}_{1:t-1}) &(12)\\ &\approx p(\boldsymbol{y}_t|\mathbf{x}_t)\sum_{m=1}^{M} \bar{w}_{t-1}^{(m)} p(\mathbf{x}_t|\mathbf{x}_{t-1}^{(m)}), &(13)\end{aligned}$$

where the state predictive pdf is approximated from the previous filtering distribution $p^M(\mathbf{x}_t|\boldsymbol{y}_{1:t-1}) = \sum_{m=1}^M \bar{w}_{t-1}^{(m)} p(\mathbf{x}_t|\mathbf{x}_{t-1}^{(m)})$. The one-step optimal importance sampling (IS) proposal is proportional to the targeted distribution, i.e., $q^{\text{opt}}(\mathbf{x}_t) \propto p(\boldsymbol{y}_t|\mathbf{x}_t)\sum_{m=1}^M \bar{w}_{t-1}^{(m)} p(\mathbf{x}_t|\mathbf{x}_{t-1}^{(m)})$. Since the APF proposal of Eq. (11) is a mixture of M kernels, $p(\mathbf{x}_t|\mathbf{x}_{t-1}^{(m)})$, of identical bandwidths, using kernel density estimation (KDE) arguments, the weight $\lambda'^{(m)}$ of each kernel is computed as the ratio between the filtering distribution and the equal-weighted mixture of kernels as

$$\lambda_t'^{(m)} \propto \frac{p(\bar{\mathbf{x}}_t^{(m)}|\boldsymbol{y}_{1:t})}{\frac{1}{M}\sum_{j=1}^M p(\bar{\mathbf{x}}_t^{(m)}|\mathbf{x}_{t-1}^{(j)})}, \tag{14}$$

where substituting Eq. (13) in the numerator and evaluating at $\bar{\mathbf{x}}_t^{(m)}$ yields Eq. (16). In other words, the mixture weights $\{\lambda_t'^{(m)}\}_{m=1}^M$ are chosen so $q^{\text{IAPF}}(\mathbf{x}_t)$ reconstructs the target with the set of kernels. In step 2(c), the new particles are independently drawn from the proposal $q^{\text{IAPF}}(\mathbf{x}_t)$. The IS weight of Eq. (18) is obtained by simple IS arguments, i.e., with the targeted distribution in the numerator given by Eq. (13) and the proposal in the denominator given by Eq. (11), both evaluated at each particle $\mathbf{x}_t^{(m)}$.

A. Novel derivation of the APF from the IAPF

There are two differences between the APF and the novel IAPF. First, note that both proposals $q^{\text{APF}}(\mathbf{x}_t)$ and $q^{\text{IAPF}}(\mathbf{x}_t)$ are mixtures of the same kernels, but with different weights of the mixands (see Eqs. (7) and (16)). We now explore the connection between the mixture weights of both algorithms. Let us suppose that the kernels $\{p(\mathbf{x}_t|\mathbf{x}_{t-1}^{(m)})\}_{m=1}^M$ do not have

Algorithm 2: Improved Auxiliary Particle Filter

1) Initialization. At time $t = 0$, draw M i.i.d. samples, $\mathbf{x}_0^{(m)}$, from the distribution $p(\mathbf{x}_0)$, and set $w_0^{(m)} = \frac{1}{M}$, $m = 1, ..., M$.
2) Recursive step. Let $\{\mathbf{x}_{t-1}^{(m)}\}_{m=1}^M$ be the particles (samples) generated at time $t-1$. At time t, proceed with the steps below.
   a) Compute the mean of the pdf $p(\mathbf{x}_t|\mathbf{x}_{t-1}^{(m)})$ as
   $$\bar{\mathbf{x}}_t^{(m)} = \mathbb{E}_{p(\mathbf{x}_t|\mathbf{x}_{t-1}^{(m)})}[\mathbf{x}_t], \qquad m = 1, ..., M. \tag{15}$$
   b) Compute the normalized weights of each kernel in the mixture as
   $$\lambda_t'^{(m)} \propto \frac{p(\boldsymbol{y}_t|\bar{\mathbf{x}}_t^{(m)})\sum_{j=1}^M w_{t-1}'^{(j)} p(\bar{\mathbf{x}}_t^{(m)}|\mathbf{x}_{t-1}^{(j)})}{\sum_{j=1}^M p(\bar{\mathbf{x}}_t^{(m)}|\mathbf{x}_{t-1}^{(j)})} \tag{16}$$
   and build the IS proposal as
   $$q^{\text{IAPF}}(\mathbf{x}_t) = \sum_{m=1}^{M} \lambda_t'^{(m)} p(\mathbf{x}_t|\mathbf{x}_{t-1}^{(m)}). \tag{17}$$
   c) Draw M i.i.d. samples from $q^{\text{IAPF}}(\mathbf{x}_t)$ in two steps:
      i) select the indexes $i^{(m)}$, $m = 1, ...M$, with pmf given by $\mathbb{P}(i^{(m)} = j) = \lambda_t'^{(j)}$, $j \in \{1, ..., M\}$.
      ii) simulate $\mathbf{x}_t^{(m)} \sim p(\mathbf{x}_t|\mathbf{x}_{t-1}^{(i^m)})$, $\qquad m = 1, ...M$.
   d) Compute the weights as
   $$w_t'^{(m)} = \frac{p(\boldsymbol{y}_t|\mathbf{x}_t^{(m)})\sum_{j=1}^M w_{t-1}'^{(j)} p(\mathbf{x}_t^{(m)}|\mathbf{x}_{t-1}^{(j)})}{\sum_{j=1}^M \lambda_t'^{(j)} p(\mathbf{x}_t^{(m)}|\mathbf{x}_{t-1}^{(j)})}. \tag{18}$$
   e) Normalize the weights
   $$\bar{w}_t'^{(m)} = \frac{w_t'^{(m)}}{\sum_{k=1}^M w_t'^{(k)}}, \qquad m = 1, ..., M. \tag{19}$$

significant overlap among them. Then, the mixture weights of the IAPF can be approximated as

$$\begin{aligned} \lambda_t'^{(m)} &\propto \frac{p(\boldsymbol{y}_t|\bar{\mathbf{x}}_t^{(m)})\sum_{j=1}^M w_{t-1}^{(j)} p(\bar{\mathbf{x}}_t^{(m)}|\mathbf{x}_{t-1}^{(j)})}{\sum_{j=1}^M p(\bar{\mathbf{x}}_t^{(m)}|\mathbf{x}_{t-1}^{(j)})} &(20)\\ &\approx \frac{p(\boldsymbol{y}_t|\bar{\mathbf{x}}_t^{(m)}) w_{t-1}'^{(m)} p(\bar{\mathbf{x}}_t^{(m)}|\mathbf{x}_{t-1}^{(m)})}{p(\bar{\mathbf{x}}_t^{(m)}|\mathbf{x}_{t-1}^{(m)})} &(21)\\ &= p(\boldsymbol{y}_t|\bar{\mathbf{x}}_t^{(m)}) w_{t-1}'^{(m)} = \lambda_t^{(m)}, &(22)\end{aligned}$$

where the approximation comes from the assumption that $p(\bar{\mathbf{x}}_t^{(m)}|\mathbf{x}_{t-1}^{(j)}) \approx 0$, for $j \neq m$. In other words, each value $\bar{\mathbf{x}}_t^{(m)}$ is evaluated in all M kernels, but all the evaluations are considered as zero except the evaluation in its own kernel. The second difference is in the IS weights. Due to similar arguments, and still assuming no significant overlap between kernels, the IS weights in Eq. (18) are approximated as

$$\begin{aligned} w_t'^{(m)} &= \frac{p(\boldsymbol{y}_t|\mathbf{x}_t^{(m)})\sum_{j=1}^M w_{t-1}'^{(j)} p(\mathbf{x}_t^{(m)}|\mathbf{x}_{t-1}^{(j)})}{\sum_{j=1}^M \lambda_t'^{(j)} p(\mathbf{x}_t^{(m)}|\mathbf{x}_{t-1}^{(j)})} &(23)\\ &\approx \frac{p(\boldsymbol{y}_t|\mathbf{x}_t^{(m)}) w_{t-1}'^{(i^m)} p(\mathbf{x}_t^{(m)}|\mathbf{x}_{t-1}^{(i^m)})}{p(\boldsymbol{y}_t|\bar{\mathbf{x}}_t^{(i^m)}) w_{t-1}'^{(i^m)} p(\mathbf{x}_t^{(m)}|\mathbf{x}_{t-1}^{(i^m)})} &(24)\\ &= \frac{p(\boldsymbol{y}_t|\mathbf{x}_t^{(m)})}{p(\boldsymbol{y}_t|\bar{\mathbf{x}}_t^{(i^m)})} = w_t^{(m)}, &(25)\end{aligned}$$

ISBN 978-90-827970-1-5 © EURASIP 2018

where the approximation of Eq. (24) assumes that $p(x_t^{(m)}|\mathbf{x}_{t-1}^{(j)}) \approx 0$, for $j \neq i^{(m)}$, i.e., each particle $x_t^{(m)}$ only takes relevant value when it is evaluated at the kernel where it was drawn from.

B. Discussion

The APF can be seen as a particular case of the IAPF when the kernels $\{p(\mathbf{x}_t|\mathbf{x}_{t-1}^{(m)})\}_{m=1}^{M}$ do not have significant overlap among them. Note that it requires less computations than the IAPF due to the adopted assumption. However, when this assumption is violated, i.e., if the overlap of the kernels is significant, the APF implicitly uses a proposal distribution potentially very different from the target. This can happen when the number of particles is large and when the transition kernel is broad, i.e., when there is a moderate to big uncertainty in the transition model. Due to IS arguments, this mismatch between the target and the proposal can dramatically increase the errors of the filter.

IV. Numerical results

A. Example 1: Channel estimation in a communication system

We consider a wireless communication system in which the unknown dynamic channel has a memory of a length $d_x - 1$. The goal consists in tracking the channel vector $\mathbf{x}_t \in \mathbb{R}^{d_x}$, where d_x is the number of dynamic coefficients to be estimated. The channel is estimated by sequentially transmitting symbols (pilots) that are also known to the receiver. We suppose a linear-Gaussian system described by

$$\mathbf{x}_t = \mathbf{F}\mathbf{x}_{t-1} + \mathbf{v}_t, \tag{26}$$

$$y_t = \mathbf{g}_t^\top \mathbf{x}_t + \mathbf{n}_t, \tag{27}$$

where $\mathbf{g}_t = [d_t, d_{t-1}, ..., d_{t-d_x+1}]^\top$, is the column vector that contains the last d_x transmitted pilots. The symbols are independent and $d_t \in \{-1, +1\}$ with equal probability. We simulate the system with $\mathbf{F} = 0.7\mathbf{I}$, $\mathbf{\Sigma}_v = 5\mathbf{I}$, $\mathbf{\Sigma}_n = \sigma_n^2 = 0.5$, and $p(\mathbf{x}_0) = \mathbf{I}$. We set $T = 200$ time steps. The choice of this model allows us to compute the ground truth with the Kalman filter. Table I shows the mean square error (MSE) in the approximation of the mean of the filtering distribution w.r.t. the mean provided by the Kalman filter. All filters are implemented with $M = 100$ particles. In Table II, the filters are run with $M = 1000$ particles for different state values of state dimension $d_x \in \{1, 2, 3, 5, 10\}$. The results are averaged over 50 runs. Note that the novel IAPF outperforms the other filters in all scenarios except in $d_x = 10$ with $M = 1000$. In all cases, the performance of the IAPF is significantly better than the APF, which is consistent with the theoretical justification in previous section.

B. Example 2: Stochastic growth model

Let us consider the non-linear growth model (see for instance [29], [30]) described by

$$x_t = \frac{x_{t-1}}{2} + \frac{25x_{t-1}}{1 + x_{t-1}^2} + 8\cos(\phi t) + u_t, \tag{28}$$

$$y_t = \frac{x_t^2}{20} + v_t, \tag{29}$$

d_x (dimension)	1	2	3	5	10
MSE (BPF)	0.0272	0.3762	0.9657	1.4705	2.9592
MSE (APF)	0.0709	0.8041	1.6041	2.2132	3.7187
MSE (IAPF)	**0.0062**	**0.1764**	**0.5176**	**0.8041**	**2.6931**

TABLE I
MSE IN THE APPROXIMATION OF THE POSTERIOR MEAN. $M = 100$ PARTICLES WERE USED IN ALL FILTERS.

d_x (dimension)	1	2	3	5	10
MSE (BPF)	0.0012	0.0318	0.0996	0.2159	**0.3280**
MSE (APF)	0.0189	0.3893	0.8854	1.3307	1.8961
MSE (IAPF)	**0.0006**	**0.0150**	**0.0389**	**0.1204**	0.4742

TABLE II
MSE IN THE APPROXIMATION OF THE POSTERIOR MEAN. $M = 1000$ PARTICLES WERE USED IN ALL FILTERS.

where $\phi = 0.4$ is a frequency parameter (in rad/s), and u_t and v_t denote independent zero-mean univariate Gaussian r.v.'s with variance $\sigma_u^2 = 1$ and $\sigma_v^2 = 0.1$, respectively. The model was run from $t = 1, 2, ..., T$, with $T = 100$. We run all filters with $M = 100$ particles. Figure 1 shows the evolution of the true underlying state $\mathbf{x}_t$, and the approximations of the mean posterior given by BPF, APF, and IAPF. In this specific run, both BPF and IAPF are able to track the state, while APF provides significant errors.

V. Conclusions

The APF is an important type of filter in the class of particle filters. In theory, it should perform better than the BPF because it uses a better importance function than the BPF. A standard implementation of the APF involves a computation of the particle weights that is based on a series of approximations. These approximations can result in a much deteriorated performance of the APF. In this paper, we presented an alternative and more accurate way of (a) building the importance function and (b) computing the particle weights, which in turn, leads to improved performance of the filter. The simulation results demonstrated that the proposed filter outperforms both the BPF and the standard APF.

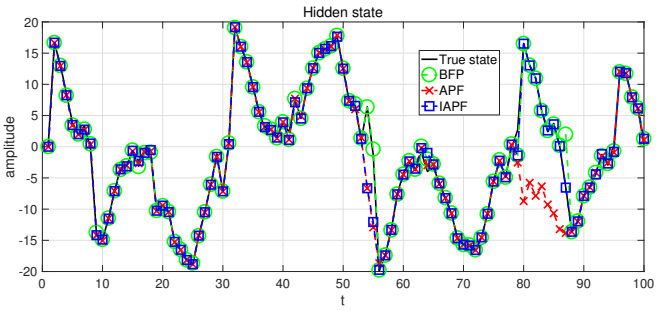

Fig. 1. Evolution of the hidden state in the stochastic growth model, and the mean approximations given by BPF, APF, and IAPF.

ISBN 978-90-827970-1-5 © EURASIP 2018

REFERENCES

[1] J. Candy, *Bayesian signal processing: classical, modern and particle filtering methods*. England: John Wiley & Sons, 2009.

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

[3] C. P. Robert and G. Casella, *Monte Carlo Statistical Methods*. Springer, 2004.

[4] 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.

[5] T. Bertozzi, D. L. Ruyet, G. Rigal, and H. Vu-Thien, "On particle filtering for digital communications," in *Proceedings of the IEEE Workshop on SPAWC*, June 2003.

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

[7] 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.

[8] 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, no. 2. IET, 1993, pp. 107–113.

[9] V. Elvira, L. Martino, D. Luengo, and M. F. Bugallo, "Population Monte Carlo schemes with reduced path degeneracy," *IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, 2017.

[10] M. Bolić, P. M. Djurić, and S. Hong, "New resampling algorithms for particle filters," in *Proceedings of the IEEE ICASSP*, April 2003.

[11] T. Li, M. Bolić, and P. M. Djurić, "Resampling methods for particle filtering," *IEEE Signal Processing Magazine*, vol. 32, no. 3, pp. 70–86, May 2015.

[12] L. Martino, V. Elvira, and F. Louzada, "Weighting a resampled particle in Sequential Monte Carlo," *IEEE Statistical Signal Processing Workshop, (SSP)*, vol. 122, pp. 1–5, 2016.

[13] L. Martino, V. Elvira, D. Luengo, and J. Corander, "Layered adaptive importance sampling," *Statistics and Computing*, vol. 27, no. 3, pp. 599–623, 2017.

[14] A. Kong, "A note on importance sampling using standardized weights," *Technical Report 348, Department of Statistics, University of Chicago*, 1992.

[15] L. Martino, V. Elvira, and M. F. Louzada, "Effective Sample Size for importance sampling based on the discrepancy measures," *Signal Processing*, vol. 131, pp. 386–401, 2017.

[16] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P.-J. Nordlund, "Particle filters for positioning, navigation and tracking," *IEEE Transactions Signal Processing*, vol. 50, no. 2, pp. 425–437, February 2002.

[17] E. Koblents and J. Míguez, "A population Monte Carlo scheme with transformed weights and its application to stochastic kinetic models," *Statistics and Computing*, vol. 25, no. 2, pp. 407–425, 2015.

[18] M. F. Bugallo, V. Elvira, L. Martino, D. Luengo, J. Míguez, 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.

[19] V. Elvira, L. Martino, D. Luengo, and M. F. Bugallo", "Generalized multiple importance sampling," *arXiv:1511.03095*, 2015.

[20] V. Elvira, L. Martino, D. Luengo, and M. F. Bugallo, "Improving Population Monte Carlo: Alternative weighting and resampling schemes," *Signal Processing*, vol. 131, no. 12, pp. 77–91, 2017.

[21] C. Berzuini and W. Gilks, "Resample-move filtering with cross-model jumps," in *Sequential Monte Carlo Methods in Practice*, A. Doucet, N. de Freitas, and N. Gordon, Eds. Springer, 2001, ch. 6.

[22] A. O. D and J. Míguez, "Nudging the Particle Filter," *IEEE Transactions Signal Processing*, arXiv:1708.07801 2017.

[23] A. Gelman and X. Meng, "Simulating normalizing constants: From importance sampling to bridge sampling to path sampling," *Statistical Science*, vol. 13, no. 2, pp. 163–185, 1998.

[24] M. K. Pitt and N. Shephard, "Auxiliary variable based particle filters," in *Sequential Monte Carlo Methods in Practice*, A. Doucet, N. de Freitas, and N. Gordon, Eds. Springer, 2001, ch. 13, pp. 273–293.

[25] P. Guarniero, A. M. Johansen, and A. Lee, "The iterated auxiliary particle filter," *Journal of the American Statistical Association*, pp. 1–12, 2017.

[26] B. Li, C. Liu, and W.-H. Chen, "An auxiliary particle filtering algorithm with inequality constraints," *IEEE Transactions on Automatic Control*, vol. 62, no. 9, pp. 4639–4646, 2017.

[27] L. Úbeda-Medina, Á. F. García-Fernández, and J. Grajal, "Adaptive auxiliary particle filter for track-before-detect with multiple targets," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 53, no. 5, pp. 2317–2330, 2017.

[28] 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.

[29] 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.

[30] P. M. Djurić and J. Míguez, "Assessment of nonlinear dynamic models by Kolmogorov–Smirnov statistics," *IEEE Transactions on Signal Processing*, vol. 58, no. 10, pp. 5069–5079, 2010.

ISBN 978-90-827970-1-5 © EURASIP 2018