

A PARTICLE SMOOTHING IMPLEMENTATION OF THE FULLY-ADAPTED AUXILIARY PARTICLE FILTER : AN ALTERNATIVE TO AUXILIARY PARTICLE FILTERS

Yohan Petetin, François Desbouvries

Telecom Institute / Telecom SudParis / CITI Department and CNRS UMR 5157,
9 rue Charles Fourier, 91011 Evry, France

ABSTRACT

The Fully Adapted Auxiliary Particle Filter (FA-APF) is a well known Sequential Monte Carlo (SMC) algorithm for computing recursively the filtering pdf in a Hidden Markov Chain (HMC) model. However, in most of cases, the FA-APF cannot be used directly because the required functions are unavailable. To cope with this issue, the Auxiliary Particle Filter (APF) uses Importance Sampling (IS) with two degrees of freedom. APF techniques need an importance distribution and also a reliable approximation of the predictive likelihood. In this paper, we propose a class of SMC algorithms which also try to mimic the FA-APF but which has the advantage not to require any approximation of the predictive likelihood. The performances of our solution as compared to the APF algorithm is provided by simulations.

Index Terms— Particle Filtering, Sequential Monte Carlo, Auxiliary Particle Filter, Importance Sampling

1. INTRODUCTION

Let $\mathbf{x}_n \in \mathbb{R}^m$ and $\mathbf{y}_n \in \mathbb{R}^p$ be respectively a hidden and observed process. Let $p(\mathbf{x}_n|\mathbf{y}_{0:n})$, say, denote the pdf (w.r.t. Lebesgue measure) of \mathbf{x}_n given $\mathbf{y}_{0:n} = \{\mathbf{y}_i\}_{i=0}^n$, and $p(d\mathbf{x}) = p(\mathbf{x})d\mathbf{x}$ the continuous measure with density $p(\mathbf{x})$. We assume that $\{\mathbf{x}_n, \mathbf{y}_n\}$ is an HMC :

$$p(\mathbf{x}_{0:n}, \mathbf{y}_{0:n}) = p(\mathbf{x}_0) \prod_{i=1}^n p(\mathbf{x}_i|\mathbf{x}_{i-1}) \prod_{i=0}^n p(\mathbf{y}_i|\mathbf{x}_i). \quad (1)$$

Bayesian Filtering consists in computing recursively $p(\mathbf{x}_n|\mathbf{y}_{0:n})$ (we note $p_{n|j} = p(\mathbf{x}_n|\mathbf{y}_{0:j})$). If $p_{n|n}$ cannot be computed exactly, one can resort to Monte Carlo based solutions such as particles filters (PF) (see e.g [1] or [2]) or APF [3] which propagate a set of N samples and their associated weights $\{\mathbf{x}_n^i, w_n^i\}_{i=1}^N$.

Let us briefly recall the principle of APF. We start for the

classical recursion (here \mathcal{N} stands for numerator) :

$$p(\mathbf{x}_n|\mathbf{y}_{0:n}) = \frac{\overbrace{p(\mathbf{x}_n|\mathbf{y}_{0:n-1})}^{p(\mathbf{x}_n|x_n)} \int p(\mathbf{x}_n|\mathbf{x}_{n-1})p(\mathbf{x}_{n-1}|\mathbf{y}_{0:n-1})d\mathbf{x}_{n-1}}{p(\mathbf{y}_n|\mathbf{y}_{0:n-1}) = \int \mathcal{N}d\mathbf{x}_n} \quad (2)$$

If we plug a discrete approximation $\sum_{i=1}^N w_{n-1}^i \delta_{\mathbf{x}_{n-1}^i}(d\mathbf{x}_{n-1})$ of $p(\mathbf{x}_{n-1}|\mathbf{y}_{0:n-1})$ in (2), it can be rewritten as

$$\pi(\mathbf{x}_n|\mathbf{y}_{0:n}) = \sum_{i=1}^N \frac{w_{n-1}^i p(\mathbf{y}_n|\mathbf{x}_{n-1}^i)}{\sum_{i=1}^N w_{n-1}^i p(\mathbf{y}_n|\mathbf{x}_{n-1}^i)} p(\mathbf{x}_n|\mathbf{x}_{n-1}^i, \mathbf{y}_n). \quad (3)$$

Pdf $\pi(\mathbf{x}_n|\mathbf{y}_{0:n})$ is a finite mixture continuous density approximating $p(\mathbf{x}_n|\mathbf{y}_{0:n})$. In the view of deriving an SMC filter one should get samples from (3) in order to further proceed at time $n+1$. We obtain the so-called FA-APF in the terminology of Pitt and Shephard [3]. However, when the predictive likelihood $p(\mathbf{y}_n|\mathbf{x}_{n-1}^i)$ is not available and/or it is not possible to get samples from $p(\mathbf{x}_n|\mathbf{x}_{n-1}^i, \mathbf{y}_n)$, one solution consists in using importance sampling (IS) in augmented dimension with the following importance distribution :

$$q(\mathbf{x}_n) = \sum_{i=1}^N \frac{w_{n-1}^i \tau(\mathbf{x}_{n-1}^i)}{\sum_{i=1}^N w_{n-1}^i \tau(\mathbf{x}_{n-1}^i)} \tilde{q}(\mathbf{x}_n|\mathbf{x}_{n-1}^i) \quad (4)$$

In summary, assuming that at $n-1$ $\{\mathbf{x}_{n-1}^i, w_{n-1}^i\}_{i=1}^N$ approximates $p(d\mathbf{x}_{n-1}|\mathbf{y}_{0:n-1})$, then an iteration of the APF algorithm is the succession of the three following steps :

APF Algorithm

1. $\tilde{\mathbf{x}}_{n-1}^i \sim \sum_{i=1}^N \frac{w_{n-1}^i \tau(\mathbf{x}_{n-1}^i)}{\sum_{i=1}^N w_{n-1}^i \tau(\mathbf{x}_{n-1}^i)} \delta_{\mathbf{x}_{n-1}^i}(d\mathbf{x}_{n-1})$ (Re-sampling Step)
2. $\mathbf{x}_n \sim \tilde{q}(\mathbf{x}_n|\tilde{\mathbf{x}}_{n-1}^i)$
3. $w_n^i \propto \frac{p(\mathbf{x}_n^i|\tilde{\mathbf{x}}_{n-1}^i)p(\mathbf{y}_n|\mathbf{x}_n^i)}{\tau(\tilde{\mathbf{x}}_{n-1}^i)\tilde{q}(\mathbf{x}_n^i|\tilde{\mathbf{x}}_{n-1}^i)}$, $\sum_{i=1}^N w_n^i = 1$.

The performance of the APF depends on the choice of the first stage weights $\tau(\mathbf{x}_{n-1}^i)$ and of the importance distribution $\tilde{q}(\mathbf{x}_n|\mathbf{x}_{n-1})$. A recent contribution [4] suggests to ap-

We would like to thank the French MOD DGA/MRIS for financial support of the Ph.D. of Y.Petetin.

proach respectively $p(\mathbf{y}_n|\mathbf{x}_{n-1})$ and $p(\mathbf{x}_n|\mathbf{x}_{n-1}^i, \mathbf{y}_n)$ however in practice it can be very difficult to derive a good approximation of $p(\mathbf{y}_n|\mathbf{x}_{n-1})$. When the objective is to evaluate directly a function $\Phi = \int f(\mathbf{x}_n)p(\mathbf{x}_n|\mathbf{y}_{0:n})d\mathbf{x}_n$, an optimal choice of the first stage weights, based on an asymptotical variance analysis, has also been proposed in [5].

In this paper, we show that the APF is not the only workable approximation of the FA-APF algorithm and more precisely we focus on the approximate computation of the measure $p(d\mathbf{x}_n|\mathbf{y}_{0:n})$ from $p(d\mathbf{x}_{n-1}|\mathbf{y}_{0:n-1})$. We derive a Particle Smoothing APF (PS-APF) which imitates the FA-APF in a sense which will be specified later. Our algorithm has the advantage to circumvent the critical approximation of the predictive likelihood $p(\mathbf{y}_n|\mathbf{x}_{n-1})$. This is of practical interest when it is difficult to derive an approximation of this predictive likelihood or when such a procedure is computationally intensive. Moreover, we show that our approach outperforms the APF when the required parameters are not correctly approximated.

2. A PARTICLE SMOOTHING IMPLEMENTATION OF THE FA-APF

Before introducing our method, let us first revisit the FA-APF as an SMC algorithm propagating samples from $p_{n-1|n-1}$ into samples from $p_{n-1|n}$ and then $p_{n|n}$. More precisely equation (2) can be rewritten as

$$p(\mathbf{x}_n|\mathbf{y}_{0:n}) = \int p(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{y}_n) \times \underbrace{\left[\frac{p(\mathbf{y}_n|\mathbf{x}_{n-1})p(\mathbf{x}_{n-1}|\mathbf{y}_{0:n-1})}{p(\mathbf{y}_n|\mathbf{y}_{0:n-1})} \right]}_{p(\mathbf{x}_{n-1}|\mathbf{y}_{0:n})} d\mathbf{x}_{n-1}. \quad (5)$$

In the view of deriving an SMC filter from (5) we can first sample,

$$\tilde{\mathbf{x}}_{n-1}^i \sim \sum_{i=1}^N \frac{p(\mathbf{y}_n|\mathbf{x}_{n-1}^i)}{\sum_{i=1}^N p(\mathbf{y}_n|\mathbf{x}_{n-1}^i)} \delta_{\mathbf{x}_{n-1}^i}(d\mathbf{x}_{n-1})$$

in order to get samples approximately drawn from $p_{n-1|n}$, and next sample \mathbf{x}_n^i from $p(\mathbf{x}_n|\tilde{\mathbf{x}}_{n-1}^i, \mathbf{y}_n)$ in order to get samples from $p_{n|n}$. Of course, this procedure coincides with the FA-APF described previously.

Now, let us assume that $p(\mathbf{y}_n|\mathbf{x}_{n-1})$ is not computable and/or it is not possible to get samples from $p(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{y}_n)$. As we now show, it remains possible to get approximately samples $\{\tilde{\mathbf{x}}_{n-1}^i\}$ drawn from $p_{n-1|n}$ and next (possibly weighted) samples $\{\mathbf{x}_n^i\}$ from $p_{n|n}$. Let $\bar{q}(\mathbf{x}_n|\mathbf{x}_{n-1})$ be some importance distribution. Firstly, we use a one step backward marginal smoother of a SIR algorithm to obtain samples from $p_{n-1|n}$:

Particle Smoothing Step

1. $\bar{\mathbf{x}}_n^i \sim \bar{q}(\mathbf{x}_n|\mathbf{x}_{n-1}^i)$
2. $\bar{w}_n^i \propto w_{n-1}^i \frac{p(\bar{\mathbf{x}}_n^i|\mathbf{x}_{n-1}^i)p(\mathbf{y}_n|\bar{\mathbf{x}}_n^i)}{\bar{q}(\bar{\mathbf{x}}_n^i|\mathbf{x}_{n-1}^i)}$, $\sum_{i=1}^N \bar{w}_n^i = 1$.
3. $\tilde{\mathbf{x}}_{n-1}^i \sim \sum_{i=1}^N \bar{w}_n^i \delta_{\bar{\mathbf{x}}_{n-1}^i}(d\mathbf{x}_{n-1})$.

By construction samples $\{\tilde{\mathbf{x}}_{n-1}^i\}_{i=1}^N$ are drawn approximately from $p(d\mathbf{x}_{n-1}|\mathbf{y}_{0:n})$ and we look for obtaining weighted samples $(\tilde{\mathbf{x}}_{n-1}^i, \mathbf{x}_n^i)$ drawn from $p(\mathbf{x}_{n-1}, \mathbf{x}_n|\mathbf{y}_{0:n}) \propto p(\mathbf{x}_{n-1}|\mathbf{y}_{0:n})p(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{y}_n)$. Of course, this second step is straightforward if one can sample from $p(\mathbf{x}_n|\tilde{\mathbf{x}}_{n-1}, \mathbf{y}_n)$. If however it is not possible to sample from this distribution, we can call for IS again with a importance distribution $q(\mathbf{x}_n|\mathbf{x}_{n-1}) = \hat{p}(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{y}_n)$ close enough to $p(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{y}_n)$. The associated weights are :

$$\begin{aligned} w_n^i &\propto \frac{p(\tilde{\mathbf{x}}_{n-1}^i, \mathbf{x}_n^i|\mathbf{y}_{0:n})}{p(\tilde{\mathbf{x}}_{n-1}^i|\mathbf{y}_{0:n})q(\mathbf{x}_n^i|\tilde{\mathbf{x}}_{n-1}^i)} \\ &\propto \frac{p(\mathbf{x}_n^i|\tilde{\mathbf{x}}_{n-1}^i)p(\mathbf{y}_n|\mathbf{x}_n^i)}{p(\mathbf{y}_n|\tilde{\mathbf{x}}_{n-1}^i)q(\mathbf{x}_n^i|\tilde{\mathbf{x}}_{n-1}^i)} \\ &\propto \frac{p(\mathbf{x}_n^i|\tilde{\mathbf{x}}_{n-1}^i, \mathbf{y}_n)}{q(\mathbf{x}_n^i|\tilde{\mathbf{x}}_{n-1}^i)}, \quad \sum_{i=1}^N w_n^i = 1 \end{aligned} \quad (6)$$

At this point, several cases can occur :

1. Assume first that $p(\mathbf{x}_n^i|\tilde{\mathbf{x}}_{n-1}^i, \mathbf{y}_n)$ is computable at any point $(\mathbf{x}_{n-1}^i, \mathbf{x}_n^i)$. Then weights w_n^i in (6) are computable. In this case there is a connection between our algorithm and the APF, see Remarks 2 and 3 below.
2. Assume now that $p(\mathbf{x}_n^i|\tilde{\mathbf{x}}_{n-1}^i, \mathbf{y}_n)$ is not computable but we have a good approximation $\hat{p}(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{y}_n)$ of this pdf. Then we can reasonably assume that weights are equal to $\frac{1}{N}$.
3. Otherwise, we can introduce an MCMC method to adjust the samples \mathbf{x}_n^i . We then set $w_n^i = \frac{1}{N}$ for all i . The idea to use an MCMC method in PF is not new (see e.g [1, chap. 6]) and is generally applied directly after a SIR algorithm. Here, by contrast, we first draw a full set of samples \mathbf{x}_n^i before moving particles with an MCMC method in order to have a set of different particles \mathbf{x}_n^i like in the FA-APF (see also Remark 1). The MCMC algorithm is used here to circumvent the impossibility to obtain directly weighted samples \mathbf{x}_n^i drawn from $p_{n|n}$ using IS.

Let us sum up the methodology of the PS-APF filter :

PS-APF algorithm

- Obtain samples $\tilde{\mathbf{x}}_{n-1}^i \sim p(\mathbf{x}_{n-1}|\mathbf{y}_{0:n})$ using the one step backward marginal smoother of a SIR algorithm with importance distribution $\bar{q}(\mathbf{x}_n|\mathbf{x}_{n-1})$.

- Draw $\mathbf{x}_n^i \sim \hat{p}(\mathbf{x}_n|\tilde{\mathbf{x}}_{n-1}^i, \mathbf{y}_n)$, then if the weights of equation (6) are not available, use (if needed) a few MCMC steps to adjust \mathbf{x}_n^i and set $w_n^i = 1/N$.

3. DISCUSSION AND CONNECTION TO THE APF

Both algorithms are composed of a first Weighting Resampling step (which actually is the Particle Smoothing step for PS-APF) then a sampling-weighting step (with optional MCMC method). This construction leads to similarities between our algorithm and the (FA)-APF, illustrated by following remarks:

Remark 1 *The FA-APF is a particular case of our class of algorithms. Indeed, let us choose $\bar{q}(\mathbf{x}_n|\mathbf{x}_{n-1}) = q(\mathbf{x}_n|\mathbf{x}_{n-1}) = p(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{y}_n)$ in the PS-step and in the sampling step of PS-APF. Then the first stage weights \bar{w}_n^i of the PS step, are proportional to $p(\mathbf{y}_n|\mathbf{x}_{n-1}^i)$ and the second stage weights w_n^i are equal to $\frac{1}{N}$. Since the first stage weights of the PS step do not depend on $\bar{\mathbf{x}}_n^i$, the first sampling step can be avoided and the computational efforts are reduced.*

Remark 2 *The APF is also a particular solution of PS-APF algorithms. If we approximate the product $p(\mathbf{x}_n|\mathbf{x}_{n-1}) p(\mathbf{y}_n|\mathbf{x}_n) = p(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{y}_n) p(\mathbf{y}_n|\mathbf{x}_{n-1})$ by the approximated product $p(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{y}_n) \hat{p}(\mathbf{y}_n|\mathbf{x}_{n-1})$ in our algorithm, and we select $\bar{q}(\mathbf{x}_n|\mathbf{x}_{n-1}) = p(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{y}_n)$ in the PS step, then the PS-APF corresponds to the APF with $\tau(\mathbf{x}_{n-1}^i) = \hat{p}(\mathbf{y}_n|\mathbf{x}_{n-1}^i)$ and $\tilde{q}(\mathbf{x}_n|\mathbf{x}_{n-1}^i) = q(\mathbf{x}_n|\mathbf{x}_{n-1}^i)$. Indeed, under this approximation, it is easy to check that the first and second stage weights of the APF and the PS-APF are equal and that samples $\{\mathbf{x}_n^i\}$ are drawn from the same distribution.*

Remark 3 *If $p(\mathbf{x}_n^i|\mathbf{x}_{n-1}^i, \mathbf{y}_n)$ is computable then the predictive likelihood $p(\mathbf{y}_n|\mathbf{x}_{n-1})$ is computable. So in this case, case 1 of the discussion about weights w_n^i coincides with the particular APF in which $\tau(\mathbf{x}_{n-1}^i) = p(\mathbf{y}_n|\mathbf{x}_{n-1}^i)$ and $\tilde{q}(\mathbf{x}_n|\mathbf{x}_{n-1}^i) = q(\mathbf{x}_n|\mathbf{x}_{n-1}^i)$.*

Let us now focus on the difference between the PS-APF and the classical APF. From a practical point of view, we never use any approximation of $p(\mathbf{y}_n|\mathbf{x}_{n-1})$. Furthermore, a major difference between the two methodologies comes from the intermediate set of particles $\{\tilde{\mathbf{x}}_{n-1}^i\}_{i=1}^N$ which is approximately sampled from a pdf proportional to $\hat{p}(\mathbf{y}_n|\mathbf{x}_{n-1}) p(\mathbf{x}_{n-1}|\mathbf{y}_{0:n-1})$ for the APF and, by contrast, approximately sampled from $p(\mathbf{x}_{n-1}|\mathbf{y}_{0:n})$ for PS-APF. This difference is critical for the performances of the two algorithms, since the interpretation of the FA-APF in section 2 shows that the aim of this first step is to guide particles into promising regions. The APF tries to approach directly the likelihood $p(\mathbf{y}_n|\mathbf{x}_{n-1})$ then selecting new particles $\tilde{\mathbf{x}}_{n-1}^i$ whereas in our algorithm, a set of particles $\{\tilde{\mathbf{x}}_n^i\}_{i=1}^N$ is used to explore the new state space at time n and select particles \mathbf{x}_{n-1}^i corresponding to a promising trajectory. Of course,

$p(\mathbf{x}_{n-1}|\mathbf{y}_{0:n}) \propto p(\mathbf{y}_n|\mathbf{x}_{n-1})p(\mathbf{x}_{n-1}|\mathbf{y}_{0:n})$ but if we have at our disposal a poor approximation $\hat{p}(\mathbf{y}_n|\mathbf{x}_{n-1})$ then the intermediate samples produced by the APF will be far from being sampled from $p(\mathbf{x}_{n-1}|\mathbf{y}_{0:n})$, which means in other words that $\tilde{\mathbf{x}}_{n-1}^i$ are not in promising regions or are few diversified : this will produce few efficient samples \mathbf{x}_n^i . For example, taking $\tau_n(\mathbf{x}_{n-1}^i) = p(\mathbf{y}_n|\mu_n^i)$ [3] where μ_n^i is the mean or the mode of $p(\mathbf{x}_n|\mathbf{x}_{n-1}^i)$ in some model can lead to poor performances.

4. PRACTICAL CONSIDERATIONS

Let us specify how to choose the importance distributions $\bar{q}(\mathbf{x}_n|\mathbf{x}_{n-1})$ and $q(\mathbf{x}_n|\mathbf{x}_{n-1})$ in the PS-APF. It is well known that we should take $\bar{q}(\mathbf{x}_n|\mathbf{x}_{n-1}) = \hat{p}(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{y}_n)$ if we want to try to minimize the variance of \bar{w}_n^i conditionally on $\mathbf{x}_{0:n-1}^i$ and \mathbf{y}_n [6]. Heuristically, it is important to take into \mathbf{y}_n account to guide particles $\bar{\mathbf{x}}_{n-1}^i$. For the second distribution $q(\mathbf{x}_n|\mathbf{x}_{n-1})$ we should ideally sample from $p(\mathbf{x}_n|\tilde{\mathbf{x}}_{n-1}^i, \mathbf{y}_n)$ and as this choice is often impossible we draw samples from $\hat{p}(\mathbf{x}_n|\tilde{\mathbf{x}}_{n-1}^i, \mathbf{y}_n)$. Therefore, the same distribution could be used for the two steps of our algorithm. Let us thus briefly remind some approximation techniques of $\hat{p}(\mathbf{y}_n|\mathbf{x}_{n-1})$ for the APF and $\hat{p}(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{y}_n)$ for both algorithms. Roughly speaking, most of techniques consist in first approximating locally $p(\mathbf{x}_n, \mathbf{y}_n|\mathbf{x}_{n-1}^i)$ by a Gaussian pdf, the moments of which are approximated by using a n -th degree Taylor polynomial [6] [7] or the Unscented Transformation for particle filters [8] applied to the state and/or observation equation(s), then in deducing a local approximation of $p(\mathbf{y}_n|\mathbf{x}_{n-1}^i)$ and $p(\mathbf{x}_n|\mathbf{x}_{n-1}^i, \mathbf{y}_n)$. The approximation of $p(\mathbf{y}_n|\mathbf{x}_{n-1}^i)$ by $\hat{p}(\mathbf{y}_n|\mathbf{x}_{n-1}^i)$ can be used as a choice of the first stage weights τ_n^i in the APF. Note that the knowledge of the first and second moment order of a pdf can lead to a poor approximation of this pdf but enables to getting samples approximately drawn from it. It why the proposed method is less restrictive.

5. SIMULATIONS

Let us consider the Kitagawa model :

$$\begin{cases} \mathbf{x}_{n+1} &= 0.5\mathbf{x}_n + 25\frac{\mathbf{x}_n}{1+\mathbf{x}_n^2} + 8 \cos(1.2(n+1)) + \mathbf{u}_n \\ \mathbf{y}_n &= \frac{\mathbf{x}_n^2}{20} + \mathbf{v}_n \end{cases}, \quad (7)$$

in which \mathbf{u}_n and \mathbf{v}_n are i.i.d., mutually independent and independent of \mathbf{x}_0 , with $\mathbf{x}_0 \sim \mathcal{N}(0, 1)$. Let also $\mathbf{u}_n \sim \mathcal{N}(0, Q)$ and $\mathbf{v}_n \sim \mathcal{N}(0, R)$, $R = 1$. Let \mathcal{J} be defined as $\mathcal{J} = \frac{1}{T} \sum_{n=1}^T \left[\frac{1}{P} \sum_{j=1}^P (\hat{x}_{n|n}^j - x_n^j)^2 \right]^{\frac{1}{2}}$ where $\hat{x}_{n|n}^j$ and x_n^j are respectively the estimated state and the true state at time n according the first n observations, for the j -th realization. \mathcal{J} is averaged over $P = 1000$ realizations, and $T = 50$ time indices. Exact first and second moments of distributions concerned are calculable [7], so when it is necessary, same ap-

proximations are used for the PS-APF and the APF and we note for simplicity $\hat{p}_{EMM}(\cdot) = \hat{p}(\cdot)$. We compare two variants of the APF and three variants of the PS-APF. Parameters of each algorithm are described in Table 1 and we indicate for the PS-APF the number of MCMC (here we use an independent Metropolis Hasting algorithm) steps used (0 or 1).

| | $\tau(\mathbf{x}_{n-1}^i)$ | $\tilde{q}(\mathbf{x}_n \mathbf{x}_{n-1}^i)$ |
|--------------------------|--|--|
| APF _{class} | $p(\mathbf{y}_n \mu_n^i)$ | $p(\mathbf{x}_n \mathbf{x}_{n-1}^i)$ |
| APF _{EMM} | $\hat{p}(\mathbf{y}_n \mathbf{x}_{n-1}^i)$ | $\hat{p}(\mathbf{x}_n \mathbf{x}_{n-1}^i, \mathbf{y}_n)$ |
| | $\tilde{q}(\mathbf{x}_n \mathbf{x}_{n-1}^i)$ | $q(\mathbf{x}_n \mathbf{x}_{n-1}^i)$ |
| PS-APF _{boot-1} | $p(\mathbf{x}_n \mathbf{x}_{n-1}^i)$ | $\hat{p}(\mathbf{x}_n \mathbf{x}_{n-1}^i, \mathbf{y}_n)$ |
| PS-APF _{EMM-0} | $\hat{p}(\mathbf{x}_n \mathbf{x}_{n-1}^i, \mathbf{y}_n)$ | $\hat{p}(\mathbf{x}_n \mathbf{x}_{n-1}^i, \mathbf{y}_n)$ |
| PS-APF _{EMM-1} | $\hat{p}(\mathbf{x}_n \mathbf{x}_{n-1}^i, \mathbf{y}_n)$ | $\hat{p}(\mathbf{x}_n \mathbf{x}_{n-1}^i, \mathbf{y}_n)$ |

Table 1. Parameters of algorithms used for simulations

We first fix the number of particles $N = 200$ then Q varies. Results are presented in Table 2. When Q is small, all algorithms present equivalent results since approximated pdfs are close to the true pdfs. Indeed, it can be shown that EMM implies an approximation on $p(\mathbf{y}_n|\mathbf{x}_n)$ which is valid when Q is small. Consequently, $\bar{w}_n^i \approx \hat{p}(\mathbf{y}_n|\mathbf{x}_{n-1}^i)$ (by construction, $\hat{p}(\mathbf{x}_n|\mathbf{x}_{n-1}^i) = p(\mathbf{x}_n|\mathbf{x}_{n-1}^i)$) and second stage weights of APF are close to $\frac{1}{N}$. The MCMC step does not bring any improvement, for the same reasons. When Q increases, these approximations become rough and the PS-APF method gives better results, even if we do not use an MCMC step. Of course, MCMC step improves performances since this time it is less reasonable to assume that weights w_n^i (see equation (6)) are equal to $\frac{1}{N}$ before adjusting samples $\{\mathbf{x}_n^i\}$. For PS-APF_{boot-1}, performances are damaged because particles \mathbf{x}_{n-1}^i are not as well guided in promising regions than in the other algorithms since particles $\bar{\mathbf{x}}_n^i$ are sampled according the transition density.

Finally we take $Q = 10$ and the number of particles N varies (Figure 1). For clarity, we do not present results of the algorithm APF_{class}, but it leads to poor performances when Q is high (see results of Table 2). Figure 1 shows that PS-APF methods outperform the APF when the number of particles is weak, except for the PS-APF_{boot-1}. However, this phenomenon is alleviated when $N \rightarrow \infty$ and for example, when $N = 4000$, PS-APF_{boot-1} gives same results than PS-APF_{EMM-1} (it does not appear on this figure).

6. REFERENCES

[1] A. Doucet, N. de Freitas, and N. Gordon, Eds., *Sequential Monte Carlo Methods in Practice*, Statistics for Engineering and Information Science. Springer Verlag, New York, 2001.

| | $Q = 0.1$ | $Q = 1$ | $Q = 5$ | $Q = 10$ |
|--------------------------|-----------|---------|---------|----------|
| APF _{class} | 1.3061 | 2.5800 | 4.1330 | 5.1853 |
| APF _{EMM} | 1.3297 | 2.4875 | 3.6734 | 4.6746 |
| PS-APF _{boot-1} | 1.3264 | 2.5329 | 3.7347 | 4.6614 |
| PS-APF _{EMM-0} | 1.2776 | 2.4923 | 3.6671 | 4.4872 |
| PS-APF _{EMM-1} | 1.3075 | 2.4639 | 3.6114 | 4.4461 |

Table 2. \mathcal{J} - Non linear Model - $N = 200$ - $R = 1$

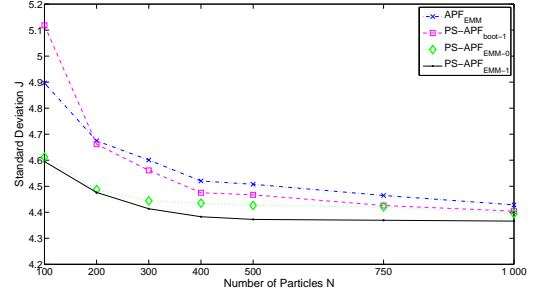


Fig. 1. Empirical standard deviation, non linear model.

- [2] M. Sanjeev 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, February 2002.
- [3] M. K. Pitt and N. Shephard, “Filtering via simulation : Auxiliary particle filter,” *Journal of the American Statistical Association*, vol. 94, no. 446, pp. 590–99, June 1999.
- [4] A. M. Johansen and A. Doucet, “A note on the auxiliary particle filter,” *Statistics and Probability Letters*, vol. 78, no. 12, pp. 1498–1504, September 2008.
- [5] R. Douc, É. Moulines, and J. Olsson, “Optimality of the auxiliary particle filter,” *Probability and Mathematical Statistics*, vol. 29, no. 1, pp. 1–28, 2009.
- [6] A. Doucet, S. J. Godsill, and C. Andrieu, “On sequential Monte Carlo sampling methods for Bayesian filtering,” *Statistics and Computing*, vol. 10, pp. 197–208, 2000.
- [7] S. Saha, P. K. Manda, Y. Boers, H. Driessen, and A. Bagchi, “Gaussian Proposal density using moment matching in smc methods,” *Statistics and Computing*, vol. 19-2, pp. 203–208, 2009.
- [8] R. van der Merwe, A. Doucet, N. de Freitas, and E. Wan, “The Unscented Particle Filter,” *Advances in Neural Information Processing Systems*, 2000.