

# OPTIMAL SIR ALGORITHM VS. FULLY ADAPTED AUXILIARY PARTICLE FILTER : A MATTER OF CONDITIONAL INDEPENDENCE

*François Desbouvries, Yohan Petetin and Emmanuel Monfrini*

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

{francois.desbouvries,yohan.petetin,emmanuel.monfrini}@it-sudparis.eu

## ABSTRACT

Particle filters (PF) and auxiliary particle filters (APF) are widely used sequential Monte Carlo (SMC) techniques. In this paper we comparatively analyse the Sampling Importance Resampling (SIR) PF with optimal conditional importance distribution (CID) and the fully adapted APF (FA-APF). Both algorithms share the same Sampling (S), Weighting (W) and Resampling (R) steps, and only differ in the order in which these steps are performed. The order of the operations is not insignificant : starting at time  $n - 1$  from a common set of particles, we show that one single updated particle at time  $n$  will marginally be sampled in both algorithms from the same probability density function (pdf), but as a whole the full set of particles will be conditionally independent if created by the FA-APF algorithm, and dependent if created by the SIR algorithm, which results in support degeneracy.

**Index Terms**— Sequential Monte Carlo, Particle Filtering, Auxiliary Particle Filtering, Conditional Independence.

## 1. INTRODUCTION

Let  $\{X_n\}_{n \in \mathbb{N}}$  and  $\{Y_n\}_{n \in \mathbb{N}}$  be respectively a hidden and an observed process. Let  $p(x_n | \mathbf{y}_{0:n})$  (or in short  $p_{n|n}$ ), say, denote the pdf (w.r.t. Lebesgue measure) of  $X_n$  given  $\mathbf{y}_{0:n} = \{y_i\}_{i=0}^n$ . We assume  $\{X_n, Y_n\}_{n \in \mathbb{N}}$  is a Hidden Markov Chain:  $p(\mathbf{x}_{0:n}, \mathbf{y}_{0:n}) = p(x_0) \prod_{i=1}^n p(x_i | x_{i-1}) \prod_{i=0}^n p(y_i | x_i)$ . Bayesian filtering consists in computing  $p_{n|n}$ , or at least some approximation of the measure  $\mu_{n|n}$  with density  $p_{n|n}$ . Recursive solutions are of particular interest, and indeed  $p_{n|n}$  can be computed from  $p_{n-1|n-1}$  by the well known recursion (here  $\mathcal{N}$  stands for numerator) :

$$p(x_n | \mathbf{y}_{0:n}) = \frac{p(y_n | x_n) \int p(x_n | x_{n-1}) p(x_{n-1} | \mathbf{y}_{0:n-1}) dx_{n-1}}{\int \mathcal{N} dx_n} \quad (1)$$

Many efforts have been devoted to the computation of eq. (1). If exact computing is difficult or impossible, one needs to resort to approximations. Among them PF and APF methods (see e.g. [1] [2]) are SMC methods which propagate a discrete approximation  $\hat{\mu}_{n|n}$  of  $\mu_{n|n}$ .

The aim of this paper is to comparatively analyse two closely related algorithms, the SIR algorithm with optimal CID [3] and the FA-APF [2]. We show that starting from a common set of points  $\mathbf{x}_{n-1}^{\{i\}} \stackrel{def}{=} \{x_{n-1}^i\}_{i=1}^N$  sampled from  $p_{n-1|n-1}$ , a new particle produced by either algorithm is marginally sampled from the same pdf, but the whole set of particles is (given  $\mathbf{x}_{n-1}^{\{i\}}$  and  $y_n$ ) independent if produced by FA-APF, while it is dependent if produced by SIR. This paper is organized as follows. The SIR and FA-APF algorithms are recalled in §2. Our analysis is performed in §3, and in §4 we compare both algorithms via simulations.

## 2. TWO CLOSELY RELATED SMC FILTERS

Let us first recall the well known SIR algorithm with optimal CID (see e.g. [3]). It consists of the succession of a sampling step  $S$ , a weighting step  $W$  and an optional<sup>1</sup> resampling step  $R$ . The algorithm (which will simply be denoted as SIR in the sequel) is as follows :

**SIR algorithm (with optimal CID and Resampling at each step).**

Let  $\hat{\mu}_{n-1|n-1} = \sum_{i=1}^N \frac{1}{N} \delta_{x_{n-1}^i}$  approximate  $\mu_{n-1|n-1}$ .

$S$ . For  $1 \leq i \leq N$ , sample  $\tilde{x}_n^i \sim p(\tilde{x}_n | x_{n-1}^i, y_n)$ ;

$W$ . For  $1 \leq i \leq N$ , compute  $w_n^i \propto p(y_n | x_{n-1}^i), \sum_{i=1}^N w_n^i = 1$ ;

$R$ . For  $1 \leq i \leq N$ , sample  $x_n^i \sim \sum_{i=1}^N w_n^i \delta_{\tilde{x}_n^i}(dx_n)$ .

Then  $\hat{\mu}_{n|n} = \sum_{i=1}^N \frac{1}{N} \delta_{x_n^i}$  approximates  $\mu_{n|n}$ .

Let us now briefly recall the principle of APF (see e.g. [2] [4], and also [5] for recent developments). If exact computing of (1) is not possible, a natural idea consists in replacing  $\mu_{n-1|n-1}$  by a discrete approximation  $\sum_{i=1}^N w_{n-1}^i \delta_{x_{n-1}^i}$ . Injecting into (1) we get a continuous approximation of  $p(x_n | \mathbf{y}_{0:n})$  :

$$\pi(x_n) = \sum_{i=1}^N \frac{w_{n-1}^i p(y_n | x_{n-1}^i)}{\sum_{i=1}^N w_{n-1}^i p(y_n | x_{n-1}^i)} p(x_n | x_{n-1}^i, y_n). \quad (2)$$

<sup>1</sup>For the purposes of this paper we will assume resampling at each step, but variants are available.

In the view of deriving an SMC algorithm one should get samples from (2) in order to further proceed at time  $n + 1$ . If direct sampling from  $\pi$  is difficult or impossible, one can approximate  $\pi$  by the mixture pdf

$$q(x_n) = \sum_{i=1}^N \frac{w_{n-1}^i \tau_{n-1}^i}{\sum_{i=1}^N w_{n-1}^i \tau_{n-1}^i} \tilde{q}(x_n | x_{n-1}^i), \quad (3)$$

sample from  $q$  (the so-called first-stage weights  $\tilde{w}_{n-1}^i \propto w_{n-1}^i \tau_{n-1}^i$  and pdf  $\tilde{q}$  are degrees of freedom used for designing this importance density), and use importance sampling (IS) in augmented dimension, which leads to the family of APF algorithms, see e.g. [2] for details.

Now, if direct sampling from  $\pi$  is indeed feasible then we are in the so-called *fully-adapted* (FA) case, in the terminology of [2]. Remember that to get a sample from a mixture  $\sum_{i=1}^N \alpha_i p_i(x)$  one can first draw an index  $j$  from  $\sum_{i=1}^N \alpha_i \delta_i$  ((W,R) steps) and then sample  $x$  from the  $p_j(x)$  which has just been selected (S step). Applying to (2) we get the

### FA-APF algorithm<sup>2</sup>

Let  $\hat{\mu}_{n-1|n-1} = \sum_{i=1}^N \frac{1}{N} \delta_{x_{n-1}^i}$  approximate  $\mu_{n-1|n-1}$ .

*W.* For  $1 \leq i \leq N$ , compute  $w_n^i \propto p(y_n | x_{n-1}^i), \sum_{i=1}^N w_n^i = 1$ ;

*R.* For  $1 \leq i \leq N$ , sample  $\tilde{x}_{n-1}^i \sim \sum_{i=1}^N w_n^i \delta_{x_{n-1}^i}(dx_{n-1})$ ;

*S.* For  $1 \leq i \leq N$ , sample  $x_n^i \sim p(x_n | \tilde{x}_{n-1}^i, y_n)$ .

Then  $\hat{\mu}_{n|n} = \sum_{i=1}^N \frac{1}{N} \delta_{x_n^i}$  approximates  $\mu_{n|n}$ .

As we see the FA-APF algorithm is a *reordering* of the SIR algorithm. In SIR the successive steps are  $S \rightarrow W \rightarrow R$  (or, equivalently,  $W \rightarrow S \rightarrow R$ : steps  $S$  and  $W$  commute since the incremental weight  $p(y_n | x_{n-1}^i)$  does not depend on the new particle  $\tilde{x}_n^i$ ), while the recursive loop of FA-APF is made of the successive steps  $W \rightarrow R \rightarrow S$ .

## 3. ANALYSING THE SIR AND FA-APF ALGORITHMS

In this section we analyse the SIR and FA-APF algorithms. Assume that at time  $n - 1$  we have  $N$  points  $\{x_{n-1}^i\}_{i=1}^N$  sampled from  $p_{n-1|n-1}$  (or at least from some approximation  $\hat{p}_{n-1|n-1}$  of  $p_{n-1|n-1}$ ). Starting from this common set of points, our aim is to compare the samples  $\mathbf{x}_n^{\{i\}}$  produced either by the SIR algorithm, or by the FA-APF one. To that aim we are going to compute, for both algorithms, the conditional pdf  $p(\mathbf{x}_n^{\{i\}} | \mathbf{x}_{n-1}^{\{i\}}, y_n)$ .

<sup>2</sup>In the view of §3 we set  $w_{n-1}^i = \frac{1}{N}$  for all  $i$ , i.e. we start from an unweighted pdf.

### 3.1. Reformulating the SIR and FA-APF algorithms

Let  $\alpha_n^i = \frac{p(y_n | x_{n-1}^i)}{\sum_{i=1}^N p(y_n | x_{n-1}^i)}$ ,  $p_i(x_n) = p(x_n | x_{n-1}^i, y_n)$  and  $\tilde{p}_i(x_n) = p(x_n | \tilde{x}_{n-1}^i, y_n)$ . We first rewrite the algorithms in terms of conditional pdf. Both algorithms compute  $\mathbf{x}_n^{\{i\}}$  from  $\mathbf{x}_{n-1}^{\{i\}}$  via an intermediate set of points  $\tilde{\mathbf{x}}_n^{\{i\}} \stackrel{\text{def}}{=} \{\tilde{x}_n^i\}_{i=1}^N$  (computed by the *S* step in SIR, and by the (W,R) steps in FA-APF), and so each algorithm can be written as the succession of two steps.

- Let us begin with SIR. In the (S) step, given  $\mathbf{x}_{n-1}^{\{i\}}$ , we independently draw  $N$  particles  $\tilde{\mathbf{x}}_n^{\{i\}}$ . We next weight them, and given  $\mathbf{x}_{n-1}^{\{i\}}$  and  $\tilde{\mathbf{x}}_n^{\{i\}}$ , we draw  $N$  independent samples  $\mathbf{x}_n^{\{i\}}$  from this reweighted distribution (R step). This algorithm can be rewritten as follows :

**SIR algorithm.** Let  $\mathbf{x}_{n-1}^{\{i\}} \stackrel{i.i.d.}{\sim} p_{n-1|n-1}$ . Then

$$\tilde{\mathbf{x}}_n^{\{i\}} \stackrel{S}{\sim} p(\tilde{\mathbf{x}}_n | \mathbf{x}_{n-1}^{\{i\}}, y_n) = \prod_{i=1}^N p_i(\tilde{x}_n^i); \quad (4)$$

$$\mathbf{x}_n^{\{i\}} \stackrel{(W,R)}{\sim} p(\mathbf{x}_n | \tilde{\mathbf{x}}_n^{\{i\}}, \mathbf{x}_{n-1}^{\{i\}}, y_n) = \prod_{i=1}^N \sum_{j=1}^N \alpha_n^j \delta_{\tilde{x}_n^j}(x_n^i). \quad (5)$$

- Now in FA-APF, given  $\mathbf{x}_{n-1}^{\{i\}}$  and  $y_n$ , we draw  $N$  independent points  $\mathbf{x}_n^{\{i\}}$  from  $\pi(x_n)$  defined in (2). The algorithm can be rewritten on the whole as follows :

**FA-APF algorithm.** Let  $\mathbf{x}_{n-1}^{\{i\}} \stackrel{i.i.d.}{\sim} p_{n-1|n-1}$ . Then

$$\tilde{\mathbf{x}}_{n-1}^{\{i\}} \stackrel{(W,R)}{\sim} p(\tilde{\mathbf{x}}_{n-1} | \mathbf{x}_{n-1}^{\{i\}}, y_n) = \prod_{i=1}^N \sum_{j=1}^N \alpha_n^j \delta_{x_{n-1}^j}(\tilde{x}_{n-1}^i) \quad (6)$$

$$\mathbf{x}_n^{\{i\}} \stackrel{S}{\sim} p(\mathbf{x}_n | \tilde{\mathbf{x}}_{n-1}^{\{i\}}, \mathbf{x}_{n-1}^{\{i\}}, y_n) = \prod_{i=1}^N \tilde{p}_i(x_n^i). \quad (7)$$

Note that in FA-APF we sample from the discrete pdf (6) and next from the continuous one (7), while this is the contrary in SIR ((4) is continuous and (5) discrete). As we will see, this will induce consequences on the degeneracy of the support of  $\{x_n^i\}_{i=1}^N$  produced by the SIR algorithm (see Remark 2 below).

### 3.2. Marginal conditional pdf

We are now ready to compute  $p(x_n | \mathbf{x}_{n-1}^{\{i\}}, y_n)$  for both algorithms :

**Proposition 1** *Let  $\{x_{n-1}^i\}_{i=1}^N \sim \hat{p}_{n-1|n-1}$ . Let  $x_{n,i}^{\text{SIR}}$  (resp.  $x_{n,i}^{\text{FA}}$ ) be a point computed by the SIR (resp. FA-APF) algorithm. Then both  $x_{n,i}^{\text{SIR}}$  and  $x_{n,i}^{\text{FA}}$  are sampled from  $p(x_n | \mathbf{x}_{n-1}^{\{i\}}, y_n) = \pi(x_n)$  defined in (2).*

**Proof 1** By construction each sample  $x_{n,i}^{\text{FA}}$  computed by FA-APF is drawn from  $\pi$ . Let us address SIR. For any Borel set  $A \in \mathcal{B}(\mathbb{R})$  let  $\mu(A) = \mathbb{P}(X_n^{\text{SIR}} \in A | \mathbf{x}_{n-1}^{\{i\}}, y_n)$ . Then

$$\begin{aligned} \mu(A) &= \int_{\mathbb{R}^N} \left[ \sum_{i=1}^N \alpha_n^i \delta_{\tilde{x}_n^i}(A) \right] \prod_{j=1}^N p_j(\tilde{x}_n^j) d\tilde{\mathbf{x}}_n^{\{j\}} \\ &= \sum_{i=1}^N \alpha_n^i \int_{\mathbb{R}^N} \delta_{\tilde{x}_n^i}(A) \prod_{j=1}^N p_j(\tilde{x}_n^j) d\tilde{\mathbf{x}}_n^{\{j\}} \\ &= \sum_{i=1}^N \alpha_n^i \int_{\mathbb{R}} \mathbb{1}_A(\tilde{x}_n^i) p_i(\tilde{x}_n^i) d\tilde{x}_n^i \\ &= \underbrace{\int_A \sum_{i=1}^N \alpha_n^i p_i(x_n) dx_n}_{\pi(x_n)}, \end{aligned}$$

so  $\mu$  has pdf  $\pi$  w.r.t. Lebesgue measure. ■

### 3.3. Joint conditional pdf

As we see from Proposition 1, given the old particles  $\mathbf{x}_{n-1}^{\{i\}}$  and the new observation  $y_n$ , each single point produced by both algorithms is marginally drawn from the same conditional pdf  $\pi$ . The difference appears when we consider the joint conditional pdf  $p(\mathbf{x}_n | \mathbf{x}_{n-1}^{\{i\}}, y_n)$ . We begin with the following result :

**Proposition 2** Let  $\{x_{n-1}^i\}_{i=1}^N \sim \hat{p}_{n-1|n-1}$ . Let  $X_n^1 = x_n^1$  and  $X_n^2$  be the first two points computed by the SIR algorithm. For any Borel set  $A \in \mathcal{B}(\mathbb{R})$  let  $\nu(A) = \mathbb{P}(X_n^2 \in A | \mathbf{x}_{n-1}^{\{i\}}, y_n, x_n^1)$ . Then  $\nu$  is the mixed (i.e., discrete and continuous) probability measure

$$\nu([x, x + dx]) = \alpha \delta_{x_n^1}([x, x + dx]) + (1 - \alpha) p(x) dx, \quad (8)$$

in which

$$\alpha = \mathbb{P}(X_n^2 = x_n^1 | \mathbf{x}_{n-1}^{\{i\}}, y_n, x_n^1) = \frac{\sum_{i=1}^N (\alpha_n^i)^2 p_i(x_n^1)}{\sum_{i=1}^N \alpha_n^i p_i(x_n^1)}, \quad (9)$$

$$p(x) = \sum_{i=1}^N \beta_n^i p_i(x), \quad (10)$$

$$\beta_n^i \propto \alpha_n^i \sum_{\substack{j=1 \\ j \neq i}}^N \alpha_n^j p_j(x_n^1), \quad \sum_{i=1}^N \beta_n^i = 1. \quad (11)$$

**Proof 2** Let  $B(x) = (-\infty, x)$ ,  $C(dx) = [x_n^1, x_n^1 + dx)$  and  $\mu_N(x, dx) = \mathbb{P}(X_n^2 \in B(x), X_n^1 \in C(dx) | \mathbf{x}_{n-1}^{\{i\}}, y_n)$ . As in Proposition 1 we have  $\mu_N(x, dx) =$

$$\int_{\mathbb{R}^N} \prod_{i=1}^N p_i(\tilde{x}_n^i) \left[ \sum_{k=1}^N \alpha_n^k \delta_{\tilde{x}_n^k}(B(x)) \right] \left[ \sum_{j=1}^N \alpha_n^j \delta_{\tilde{x}_n^j}(C(dx)) \right] d\tilde{\mathbf{x}}_n^{\{i\}} =$$

$$\begin{aligned} &\int_{\mathbb{R}^N} \prod_{i=1}^N p_i(\tilde{x}_n^i) \left( \sum_{p=1}^N (\alpha_n^p)^2 \delta_{\tilde{x}_n^p, \tilde{x}_n^p} [B(x) \times C(dx)] \right) d\tilde{\mathbf{x}}_n^{\{i\}} + \\ &\int_{\mathbb{R}^N} \prod_{q=1}^N p_q(\tilde{x}_n^q) \left( \sum_{k=1}^N \sum_{\substack{j=1 \\ j \neq k}}^N \alpha_n^k \alpha_n^j \delta_{\tilde{x}_n^k, \tilde{x}_n^j} [B(x) \times C(dx)] \right) d\tilde{\mathbf{x}}_n^{\{i\}} \\ &= \left[ \sum_{p=1}^N (\alpha_n^p)^2 \int_{D(x, dx)} p_p(\tilde{x}_n^p) d\tilde{x}_n^p \right] \mathbb{1}_{x > x_n^1} + \\ &\sum_{k=1}^N \sum_{\substack{j=1 \\ j \neq k}}^N \alpha_n^k \alpha_n^j \int_{B(x)} p_k(\tilde{x}_n^k) d\tilde{x}_n^k \int_{C(dx)} p_j(\tilde{x}_n^j) d\tilde{x}_n^j \end{aligned}$$

in which  $D(x, dx) = [x_n^1, \min(x_n^1 + dx, x))$ . A first-order Taylor series expansion of  $\mu_N(x, dx)$  gives  $\mu_N(x, dx) =$

$$\left[ \left( \sum_{p=1}^N (\alpha_n^p)^2 p_p(x_n^1) \right) \mathbb{1}_{x > x_n^1} + \sum_{k=1}^N \sum_{\substack{j=1 \\ j \neq k}}^N \alpha_n^k \alpha_n^j p_j(x_n^1) \int_{B(x)} p_k(\tilde{x}_n^k) d\tilde{x}_n^k \right] dx$$

+  $o(dx)$  with  $\frac{o(dx)}{dx} \rightarrow 0$  when  $dx \rightarrow 0$ . On the other hand, from Proposition 1 we have

$$\mu_D(dx) \stackrel{\text{def}}{=} \mathbb{P}(X_n^1 \in C(dx) | \mathbf{x}_{n-1}^{\{i\}}, y_n) = \pi(x_n^1) dx + o'(dx)$$

with  $\frac{o'(dx)}{dx} \rightarrow 0$  when  $dx \rightarrow 0$ . Finally  $\mathbb{P}(X_n^2 < x | \mathbf{x}_{n-1}^{\{i\}}, y_n, x_n^1) = \lim_{dx \rightarrow 0} \frac{\mu_N(x, dx)}{\mu_D(dx)} = \alpha \mathbb{1}_{x > x_n^1} + (1 - \alpha) \int_{B(x)} p(x) dx$  where  $\alpha$  and  $p$  are respectively given by (9) and (10). ■

**Remark 1 (conditional independence.)** Let us comment this result. The SIR algorithm draws particles  $\tilde{\mathbf{x}}_n^{\{i\}}$ , reweights them, and finally resamples, i.e. draws  $N$  samples  $\mathbf{x}_n^{\{i\}}$  which are independent given  $y_n$ ,  $\mathbf{x}_{n-1}^{\{i\}}$  and  $\tilde{\mathbf{x}}_n^{\{i\}}$  (see (5)). But these samples are no longer independent given  $y_n$  and  $\mathbf{x}_{n-1}^{\{i\}}$  only : since we resample from a discrete distribution  $\sum_{i=1}^N \alpha_n^i \delta_{\tilde{x}_n^i}$ , once the first particle  $x_n^1$  has been drawn, the second one  $x_n^2$  is either equal to  $x_n^1$  with probability  $\alpha$  (given by (9)), or can take any real value distributed according to the mixture pdf  $p(x)$  defined in (10).

**Proposition 3 (conditional independence.)** Let  $\{x_{n-1}^i\}_{i=1}^N \sim \hat{p}_{n-1|n-1}$ . Let  $\mathbf{x}_{n,\{i\}}^{\text{SIR}} = \{x_{n,i}^{\text{SIR}}\}_{i=1}^N$  (resp.  $\mathbf{x}_{n,\{i\}}^{\text{FA}} = \{x_{n,i}^{\text{FA}}\}_{i=1}^N$ ) be the  $N$  points computed by SIR (resp. FA-APF). Then given  $\mathbf{x}_{n-1}^{\{i\}}$  and  $y_n$ ,  $\mathbf{x}_{n,\{i\}}^{\text{FA}}$  are i.i.d. samples from  $\pi(x_n)$ , while  $\mathbf{x}_{n,\{i\}}^{\text{SIR}}$  are identically distributed from  $\pi(x_n)$  but are not independent.

**Proof 3** Given  $\mathbf{x}_{n-1}^{\{i\}}$  and  $y_n$ , the FA-APF algorithm draws independent samples by construction. That the  $N$  particles computed by the SIR algorithm are not independent is a consequence of proposition 2. ■

**Remark 2 (support degeneracy.)** As is well known, in a sequential IS algorithm weights degenerate with time, so a lot of effort is devoted to sampling particles which will be associated to vanishing weights. Resampling was introduced as a rescue to this problem, and indeed this technique eliminates particles associated with weights that are too low. But since in the SIR algorithm a point  $x_n^i$  can be sampled several times,  $\{x_n^i\}_{i=1}^N$  is a subset of  $\{\tilde{x}_n^i\}_{i=1}^N$ . So the SIR algorithm is not computationally efficient, because the particles  $\tilde{x}_n^i$  which finally will not be selected during the  $R$  step have indeed been drawn unnecessarily. More importantly, due to resampling the support of the distribution shrinks, which is another form of degeneracy. Quantitatively this support degeneracy can be measured by the (conditional) average size of the support after resampling  $E(N')$ , defined as the expectation of the number of points which have been (re-)selected at least once. Since the total number of outcomes of each particle after the  $R$  step follows a multinomial distribution with parameters  $\alpha_n^i$ , we have

$$E(N') = N - \sum_{i=1}^N (1 - \alpha_n^i)^N. \quad (12)$$

Now by contrast in FA-APF the support cannot degenerate. Of course, the  $R$  step still induces support shrinkage, so  $\{\tilde{x}_{n-1}^i\}_{i=1}^N$  is a subset of  $\{x_{n-1}^i\}_{i=1}^N$ ; but next each  $x_n^i$  is drawn from the continuous pdf  $\tilde{p}_i(x_n^i)$ , so particles  $x_n^i$  and  $x_n^j$  will be different even if  $\tilde{x}_{n-1}^i = \tilde{x}_{n-1}^j$ .

#### 4. SIMULATIONS

We consider the ARCH state space model :

$$x_{n+1} = \sqrt{\beta_0 + \beta_1 x_n^2} \times u_n \quad (13)$$

$$y_n = x_n + v_n \quad (14)$$

in which  $u_n \sim \mathcal{N}(0, 1)$ ,  $v_n \sim \mathcal{N}(0, R)$ , and  $x_0$ ,  $\{u_n\}$  and  $\{v_n\}$  are independent. For this model  $p(x_n|x_{n-1}, y_n) \sim \mathcal{N}(\frac{\beta_0 + \beta_1 x_{n-1}^2}{R + \beta_0 + \beta_1 x_{n-1}^2}, \frac{R(\beta_0 + \beta_1 x_{n-1}^2)}{R + \beta_0 + \beta_1 x_{n-1}^2})$  and  $p(y_n|x_{n-1}) \sim \mathcal{N}(0, R + \beta_0 + \beta_1 x_{n-1}^2)$ . We sample  $K$  trajectories, and define  $\mathcal{J}$  as the empirical standard deviation  $\mathcal{J} = \frac{1}{T} \sum_{n=1}^T [\frac{1}{K} \sum_{j=1}^K (\hat{x}_{n|n}^j - x_n^j)^2]^{\frac{1}{2}}$  ( $x_n^j$  and  $\hat{x}_{n|n}^j$  are respectively the true and estimated state at time  $n$  for trajectory  $j$ ). We set  $\beta_0 = 1$ ,  $\beta_1 = 0.1$ ,  $R = 3$ ,  $P = 400$  and  $T = 50$ . Fig. 1 displays  $\mathcal{J}$  against the number of particles  $N$ . FA-APF gives better results and the distance between the two methods decreases as  $N$  increases. Note that  $\mathcal{J}_{SIR}(400) = \mathcal{J}_{FA-APF}(200) \simeq 0.8970$ , so that similar performances are obtained by FA-APF as compared to SIR, but with half as many particles.

In the second experiment we set  $\beta_0 = 9$ ,  $\beta_1 = 5$  and  $R = 1$ . Fig. 2 displays the empirical MSE  $MSE(n) = \frac{1}{K} \sum_{j=1}^K (\hat{x}_{n|n}^j - x_n^j)^2$  as a function of time. As we see FA-APF outperforms SIR for almost all values of  $k$ . This is because we take few particles ( $N = 50$ ), and the support of

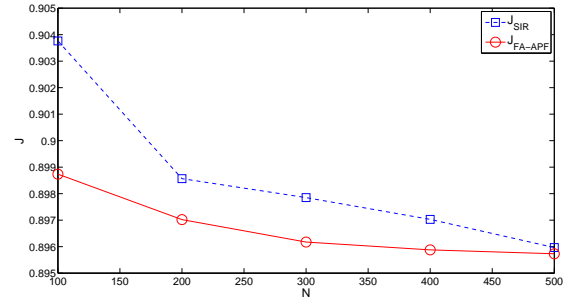


Fig. 1. Emp. Standard Deviation ( $\beta_0 = 1$ ,  $\beta_1 = 0.1$ ,  $R = 3$ )

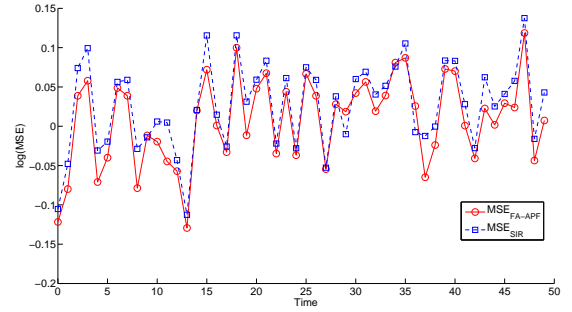


Fig. 2. Empirical MSE ( $\beta_0 = 9$ ,  $\beta_1 = 5$ ,  $R = 1$ )

particles for SIR is smaller than that of FA-APF (we checked that for SIR the empirical mean is located around 31, which is consistent with the theoretical value (12)).

#### 5. REFERENCES

- [1] A. Doucet, N. de Freitas, and N. Gordon, Eds., *Sequential Monte Carlo Methods in Practice*, Springer Verlag, New York, 2001.
- [2] M. K. Pitt and N. Shephard, “Filtering via simulation : Auxiliary particle filter,” *Journal of the American Statistical Association*, vol. 94, no. 446, pp. 550–99, june 1999.
- [3] 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.
- [4] P. Fearnhead, “Computational methods for complex stochastic systems: A review of some alternatives to MCMC,” *Statistics and Computing*, vol. 18, no. 2, pp. 151–71, 2008.
- [5] N. Whiteley and A. M. Johansen, “Recent developments in auxiliary particle filtering,” in *Inference and Learning in Dynamic Models*, Barber, Cemgil, and Chiappa, Eds. 2010, Cambridge University Press.