Direct vs. indirect sequential Monte-Carlo filters

Yohan Petetin and François Desbouvries

Telecom SudParis/CITI Department and CNRS UMR 5157,
9 rue Charles Fourier, 91011 Evry, France
yohan.petetin@telecom-sudparis.eu, francois.desbouvries@it-sudparis.eu

Abstract. We address the recursive computation of the a posteriori filtering pdf $p_{n|n}$ in a Hidden Markov Chain (HMC). Classically $p_{n|n}$ is computed via the recursion $p_{n-1|n-1} \rightarrow p_{n|n-1} \rightarrow p_{n|n}$. In this paper we explore direct, prediction-based (P-based) and smoothing-based (S-based) alternative loops for propagating $p_{n|n}$. We next address sequential Monte Carlo (SMC) implementations of these filtering paths, and compare our algorithms via simulations.
Keywords: Sequential Monte Carlo, Particle filtering, Sampling Importance Resampling.

1 Introduction

Let $(\mathbf{x}, \mathbf{y})$ be an HMC : $p(\mathbf{x}_{0:n}, \mathbf{y}_{0:n}) = p(\mathbf{x}_0) \prod_{i=1}^{n} p(\mathbf{x}_i|\mathbf{x}_{i-1}) \prod_{i=0}^{n} p(\mathbf{y}_i|\mathbf{x}_i)$. Let $p_{n|m}$ be a shorthand notation for $p(\mathbf{x}_n|\mathbf{y}_{0:m})$. Bayesian filtering consists in computing $p_{n|n}$, or at least some approximation of the measure $p(d\mathbf{x}_n|\mathbf{y}_{0:n})$ with pdf $p(\mathbf{x}_n|\mathbf{y}_{0:n})$. $p_{n|n}$ can be computed from $p_{n-1|n-1}$ by using the path $p_{n-1|n-1} \stackrel{P}{\rightarrow} p_{n|n-1} \stackrel{U}{\rightarrow} p_{n|n}$, in which we first predict state $\mathbf{x}_n$, based on the same measurements (whence superscript P), and then update the measurements set $\{\mathbf{y}_k\}_{k=0}^{n-1}$ with the new data $\mathbf{y}_n$ (whence superscript U). This path is described by the well known equation (here $\mathcal{N}$ stands for numerator) :

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

However, computing (1) is often impossible in practice, so many approximate techniques have been developed. Among them, particle filters (PF) [5] [1] are SMC methods which propagate a discrete approximation of $p(d\mathbf{x}_n|\mathbf{y}_{0:n})$.

In this paper we do not try to further improve the PF algorithms based on (1); we rather focus on (1) itself, or indeed explore alternate paths for computing $p_{n|n}$ recursively, even if $p_{n|n}$ is obtained as a byproduct.

Let us consider only those paths in which one time index is incremented at a time. The first alternative is $p_{n-1|n-1} \rightarrow p_{n-1|n} \rightarrow p_{n|n}$. Both paths compute $p_{n|n}$ recursively and differ only by the intermediate step which is

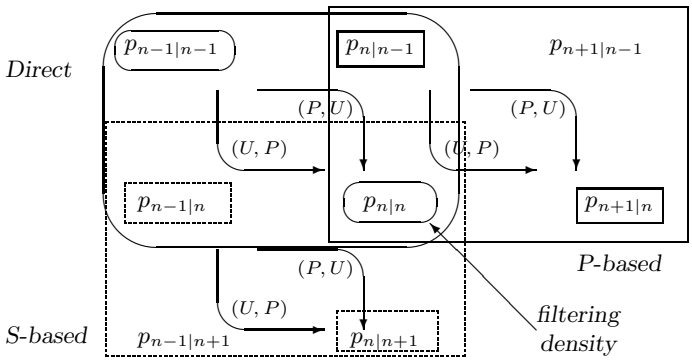

Fig. 1. The direct, P-based and S-based filtering paths.

either the one step predictive pdf $p_{n|n-1}$, or the one step smoothing pdf $p_{n-1|n}$. Now, in turn $p_{n|n-1}$ and $p_{n-1|n}$ can be propagated via the two paths obtained by moving one index and next the other. This observation yields six paths for computing $p_{n|n}$ recursively; the two paths already mentioned are "direct", i.e. $p_{n|n}$ is computed as the output of a loop with input $p_{n-1|n-1}$; two other paths are P-based, i.e. $p_{n|n}$ is computed indirectly from $p_{n|n-1}$, but the recursion itself now acts on $p_{n|n-1}$; and two paths are S-based, see Fig. 1. Out of these 6 paths only 4 are distinct, because the two paths at the boundary direct/P-based and direct/S-based coincide (for instance, the direct path $p_{n-1|n-1} \to p_{n|n-1} \to p_{n|n}$ coincides, up to a shift in time, with the P-based path $p_{n|n-1} \to p_{n|n} \to p_{n+1|n}$). The paper is organized as follows. We recall the four direct and indirect filtering paths in §2 and consider their SMC implementations in §3 (see [3] for details). §4 is devoted to simulations.

2 Direct, P-based and S-based paths

- The direct path $p_{n-1|n-1} \to p_{n|n-1} \to p_{n|n}$ is described by (1). Since it involves the one-step ahead prediction pdf $p_{n|n-1}$ we will call it 1-P;
- The alternate direct path $p_{n-1|n-1} \to p_{n-1|n} \to p_{n|n}$ involves the one step backward smoothing pdf $p_{n-1|n}$ and will thus be denoted as 1-S :

$$p(\mathbf{x}_n|\mathbf{y}_{0:n}) = \int p(\mathbf{x}_n|\mathbf{x}_{n-1},\mathbf{y}_n)\, \underbrace{[\frac{p(\mathbf{y}_n|\mathbf{x}_{n-1})p(\mathbf{x}_{n-1}|\mathbf{y}_{0:n-1})}{p(\mathbf{y}_n|\mathbf{y}_{0:n-1}) = \int \mathcal{N} d\mathbf{x}_{n-1}}]}_{p(\mathbf{x}_{n-1}|\mathbf{y}_{0:n})} d\mathbf{x}_{n-1}; \quad (2)$$

- P-based paths compute $p_{n|n}$, but via a recursive loop involving $p_{n|n-1}$. Path $p_{n|n-1} \to p_{n|n} \to p_{n+1|n}$ coincides with 1-P (up to a shift in time). The other P-based path $p_{n|n-1} \to p_{n+1|n-1} \to p_{n+1|n}$ involves the two-

step ahead prediction pdf $p_{n+1|n-1}$ and will thus be denoted by 2-P :

$$p(\mathbf{x}_{n+1}|\mathbf{y}_{0:n}) = \frac{p(\mathbf{y}_n|\mathbf{x}_{n+1},\mathbf{y}_{0:n-1}) \overbrace{\int p(\mathbf{x}_{n+1}|\mathbf{x}_n)p(\mathbf{x}_n|\mathbf{y}_{0:n-1})d\mathbf{x}_n}^{p(\mathbf{x}_{n+1}|\mathbf{y}_{0:n-1})}}{p(\mathbf{y}_n|\mathbf{y}_{0:n-1}) = \int \mathcal{N} d\mathbf{x}_{n+1}}; \quad (3)$$

- S-based paths compute $p_{n|n}$, but via a recursive loop involving $p_{n-1|n}$. Path $p_{n-1|n} \to p_{n|n} \to p_{n|n+1}$ coincides with (2) (up to a shift in time). The other S-based path $p_{n-1|n} \to p_{n-1|n+1} \to p_{n|n+1}$ involves the two-step backward smoothing pdf $p_{n-1|n+1}$ and will be denoted by 2-S :

$$p(\mathbf{x}_n|\mathbf{y}_{0:n+1}) = \int p(\mathbf{x}_n|\mathbf{x}_{n-1},\mathbf{y}_n,\mathbf{y}_{n+1}) \underbrace{\frac{p(\mathbf{y}_{n+1}|\mathbf{x}_{n-1},\mathbf{y}_n)p(\mathbf{x}_{n-1}|\mathbf{y}_{0:n})}{p(\mathbf{y}_{n+1}|\mathbf{y}_{0:n}) = \int \mathcal{N} d\mathbf{x}_{n-1}}}_{p(\mathbf{x}_{n-1}|\mathbf{y}_{0:n+1})} d\mathbf{x}_{n-1}. \quad (4)$$

3 SMC implementations

3.1 A practical toolbox

Each path (1) to (4) is made of the succession of a propagation step P (which transforms some pdf $p(\mathbf{x}_1)$ into $p(\mathbf{x}_1) \overset{P}{\mapsto} p(\mathbf{x}_2) = \int p(\mathbf{x}_2|\mathbf{x}_1)p(\mathbf{x}_1)d\mathbf{x}_1$), and of a Bayesian or updating step U (which transforms some density $p(\mathbf{x})$ into $p(\mathbf{x}) \overset{U}{\mapsto} p(\mathbf{x}|\mathbf{y}) \propto p(\mathbf{y}|\mathbf{x})p(\mathbf{x})$), or vice versa. Let us recall how we can propagate a set of points sampled from $p(\mathbf{x}_1)$ (resp. from $p(\mathbf{x})$) into a set of points sampled (at least approximatively) from $p(\mathbf{x}_2)$ (resp. from $p(\mathbf{x}|\mathbf{y})$).

1. *Propagating*. Starting from N i.i.d. samples $\{\mathbf{x}_1^i\}_{i=1}^N \sim p(\mathbf{x}_1)$, we get N i.i.d. samples $\{\mathbf{x}_2^i\}_{i=1}^N \sim p(\mathbf{x}_2)$ by sampling, for each i, $\mathbf{x}_2^i$ from $p(\mathbf{x}_2|\mathbf{x}_1^i)$. This is nothing but the Sampling step S of PF algorithms;
2. *Updating* [8]. Starting from $\{\mathbf{x}^i\}_{i=1}^N \sim p(\mathbf{x})$, we get N points $\{\tilde{\mathbf{x}}^i\}_{i=1}^N$ (approximately) independently sampled from $p(\mathbf{x}|\mathbf{y})$ by associating to each sample $\mathbf{x}^i$ a weight proportional to $p(\mathbf{y}|\mathbf{x}^i)$, and then sampling $\{\tilde{\mathbf{x}}^i\}_{i=1}^N \sim \sum_{i=1}^N \frac{p(\mathbf{y}|\mathbf{x}^i)}{\sum_{i=1}^N p(\mathbf{y}|\mathbf{x}^i)} \delta_{\mathbf{x}^i}(d\mathbf{x})$. We just described nothing but the Weighting step W, followed by the Resampling step R of PF algorithms.

3.2 SMC algorithms

We now routinely derive generic SMC implementations of (1) to (4).

1-P. (1) gives the Bootstrap [6] : Let $p(d\mathbf{x}_{n-1}|\mathbf{y}_{0:n-1}) \approx \sum_{i=1}^N \frac{1}{N} \delta_{\mathbf{x}_{n-1}^i}(d\mathbf{x}_{n-1})$.
S. For $1 \leq i \leq N$, sample $\tilde{\mathbf{x}}_n^i$ from $p(\mathbf{x}_n|\mathbf{x}_{n-1}^i)$;

W. For $1 \leq i \leq N$, compute $w_n^i \propto p(\mathbf{y}_n|\tilde{\mathbf{x}}_n^i)$, $\sum_i w_n^i = 1$;
R. For $1 \leq i \leq N$, sample $\mathbf{x}_n^i$ from $\sum_{i=1}^N w_n^i \delta_{\tilde{\mathbf{x}}_n^i}(d\mathbf{x}_n)$.

1-*S*. (2) gives [2, Algorithm 8.1.1. p. 253], which is a *reordering* of the SIR algorithm with optimal importance distribution $p(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{y}_n)$ (and systematic resampling) (see [4], where the successive steps are $S \to W \to R$):
Let $p(d\mathbf{x}_{n-1}|\mathbf{y}_{0:n-1}) \approx \sum_{i=1}^N \frac{1}{N}\delta_{\mathbf{x}_{n-1}^i}(d\mathbf{x}_{n-1})$;
W. For $1 \leq i \leq N$, compute $w_n^i \propto p(\mathbf{y}_n|\mathbf{x}_{n-1}^i)$, $\sum_{i=1}^N w_n^i = 1$;
R. For $1 \leq i \leq N$, sample $\tilde{\mathbf{x}}_{n-1}^i \sim \sum_{i=1}^N w_n^i \delta_{\mathbf{x}_{n-1}^i}(d\mathbf{x}_{n-1})$;
S. For $1 \leq i \leq N$, sample $\mathbf{x}_n^i$ from $p(\mathbf{x}_n|\tilde{\mathbf{x}}_{n-1}^i, \mathbf{y}_n)$.

2-*P*. Implementing (3) would require the knowledge of $p(\mathbf{y}_n|\mathbf{x}_{n+1}, \mathbf{y}_{0:n-1})$, but this pdf is not directly available. We thus consider the alternative path $p_{n|n-1} \to p_{n,n+1|n-1} \to p_{n,n+1|n} \to p_{n+1|n}$, given by

$$p(\mathbf{x}_{n+1}|\mathbf{y}_{0:n}) = \int \frac{p(\mathbf{y}_n|\mathbf{x}_n)\overbrace{[p(\mathbf{x}_{n+1}|\mathbf{x}_n)p(\mathbf{x}_n|\mathbf{y}_{0:n-1})]}^{p(\mathbf{x}_n,\mathbf{x}_{n+1}|\mathbf{y}_{0:n-1})}}{p(\mathbf{y}_n|\mathbf{y}_{0:n-1}) = \int \mathcal{N} d\mathbf{x}_n d\mathbf{x}_{n+1}} \, d\mathbf{x}_n. \quad (5)$$

Let us implement (5). Let $p(d\mathbf{x}_n|\mathbf{y}_{0:n-1}) \approx \sum_{i=1}^N \frac{1}{N}\delta_{\mathbf{x}_n^i}(d\mathbf{x}_n)$.
S. For $1 \leq i \leq N$, sample $\tilde{\mathbf{x}}_{n+1}^i \sim p(\mathbf{x}_{n+1}|\mathbf{x}_n^i)$;
W. For $1 \leq i \leq N$, compute $w_n^i \propto p(\mathbf{y}_n|\mathbf{x}_n^i)$, $\sum_{i=1}^N w_n^i = 1$;
R. For $1 \leq i \leq N$, sample $\mathbf{x}_{n+1}^i$ from $\sum_{i=1}^N w_n^i \delta_{\tilde{\mathbf{x}}_{n+1}^i}(d\mathbf{x}_{n+1})$.

Filtering. $p(d\mathbf{x}_n|\mathbf{y}_{0:n}) \approx \sum_{i=1}^N w_n^i \delta_{\mathbf{x}_n^i}(d\mathbf{x}_n)$.

2-*S*. We now implement (4). Let $p(d\mathbf{x}_{n-1}|\mathbf{y}_{0:n}) \approx \sum_{i=1}^N \frac{1}{N} \delta_{\mathbf{x}_{n-1}^i}(d\mathbf{x}_{n-1})$.
W. For $1 \leq i \leq N$, compute $w_{n+1}^i \propto p(\mathbf{y}_{n+1}|\mathbf{x}_{n-1}^i, \mathbf{y}_n)$, $\sum_{i=1}^N w_{n+1}^i = 1$;
R. For $1 \leq i \leq N$, sample from $\sum_{i=1}^N w_{n+1}^i \delta_{\mathbf{x}_{n-1}^i}(d\mathbf{x}_{n-1})$. We get N points $\tilde{\mathbf{x}}_{n-1}^i$, (approximately) distributed $\sim p(d\mathbf{x}_{n-1}|\mathbf{y}_{0:n+1})$;
S. For $1 \leq i \leq N$, sample $\mathbf{x}_n^i \sim p(\mathbf{x}_n|\tilde{\mathbf{x}}_{n-1}^i, \mathbf{y}_n, \mathbf{y}_{n+1})$; then $p(d\mathbf{x}_n|\mathbf{y}_{0:n+1}) \approx \sum_{i=1}^N \frac{1}{N}\delta_{\mathbf{x}_n^i}(d\mathbf{x}_n)$.
Filtering. For $1 \leq i \leq N$, sample $\overline{\mathbf{x}}_{n+1}^i \sim p(\mathbf{x}_{n+1}|\mathbf{x}_n^i, \mathbf{y}_{n+1})$; then $p(d\mathbf{x}_{n+1}|\mathbf{y}_{0:n+1}) \approx \sum_{i=1}^N \frac{1}{N}\delta_{\overline{\mathbf{x}}_{n+1}^i}(d\mathbf{x}_{n+1})$.

4 Simulations

4.1 Simulations, linear model

We first consider the state-space model : $x_{n+1} = 0.2x_n + u_n$, $y_n = 5x_n + v_n$, in which $u_n \sim \mathcal{N}(0, Q)$ and $v_n \sim \mathcal{N}(0, R)$ are i.i.d., mutually independent and independent of $x_0 \sim \mathcal{N}(0.5, 0.5)$. Even though exact Kalman filtering (KF) is available, we compare to that benchmark solution the four SMC algorithms

2-P, 1-P, 1-S, 2-S, and the SIR algorithm with optimal importance function $p(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{y}_n)$ (simply denoted SIR). Let $\mathcal{J} = \frac{1}{50}\sum_{n=1}^{50}(\frac{1}{200}\sum_{j=1}^{200}(\widehat{x}_{n|n}^j - x_n^j)^2)^{\frac{1}{2}}$ (200 is the number of realizations). Let $N = 50$ and $R = 2$. As we see from Table 1 S-based algorithms outperform P-based ones, and for a class of algorithms (P- or S-based) better results occur when updating precedes propagation. For $Q = 0.1$ all algorithms are similar, but the P-based ones degrade as Q increases, for strong variations of x_n are better tracked when y_n (for SIR or 1-S) or y_n and y_{n+1} (for 2-S) are taken into account. Next in Fig. 2 $Q = 0.1$ and $\mathcal{J}$ evolves with N. The ordering of the algorithms is maintained, but when N increases SIR, 1-S, 2-S and KF become very close.

	2-P	1-P	SIR	1-S	2-S	KF
$Q = 0.1$	0.221996933	0.217401933	0.215515333	0.2136542	0.213415733	0.2116127
$Q = 1$	0.4304519	0.2955335	0.2745811	0.2724687	0.2723657	0.2696133
$Q = 3$	0.8114436	0.3258072	0.2827361	0.2801141	0.2773301	0.2766857
$Q = 5$	1.0114103	0.3932067	0.2856878	0.2840456	0.2834111	0.2810438
$Q = 10$	1.5200521	0.4607805	0.2870748	0.2853349	0.2848457	0.2822257

Table 1. Empirical standard deviation $\mathcal{J}$, linear model.

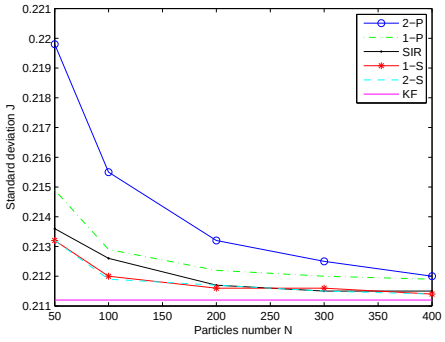

Fig. 2. Empirical standard deviation $\mathcal{J}$, linear model.

4.2 Simulations, Kitagawa model

Let us now consider the model

$$\begin{cases} x_{n+1} = f_n(x_n) + u_n \\ y_n \quad\quad = x_n^2/20 + v_n \end{cases}, \tag{6}$$

with $f_n(x_n) = 0.5x_n + 25\frac{x_n}{1+x_n^2} + 8\cos(1.2(n+1))$, and $u_n \sim \mathcal{N}(0, Q)$ and $v_n \sim \mathcal{N}(0, R)$ are i.i.d., mutually independent and independent of $x_0 \sim \mathcal{N}(0.5, 0.5)$. In (6) $p(x_n|x_{n-1}, y_n)$ and $p(y_n|x_{n-1})$ cannot be computed exactly. So we use some approximation (linearization [4], [7], EMM [7] or UPF [9]).

	2-P	1-P	SIR(EMM)	SIR(UPF)	1-S(EMM)	1-S(UPF)
$R = 0.5$	5.0124197	3.7637961	3.4181757	3.5046666	2.7191797	2.734824
$R = 1$	4.422289	3.6087193	3.447493	3.6076351	2.8830467	2.9981435
$R = 5$	4.1079475	3.6386135	3.705746	3.6928957	3.3874559	3.3369996
$R = 10$	4.4682053	4.0435623	4.020258	4.0743315	3.7412504	3.8619037
$R = 20$	5.0894144	5.0229244	4.8433825	4.8105053	4.7580243	4.8140369

Table 2. Empirical standard deviation $\mathcal{J}$, Kitagawa model.

	2-P	1-P	SIR(EMM)	SIR(UPF)	1-S(EMM)	1-S(UPF)
$N = 50$	7.1328	5.5397	5.2445	5.1169	4.7151	4.8542
$N = 100$	5.9246	4.9741	4.8057	4.9631	4.6840	4.8111
$N = 150$	5.4544	4.7726	4.8290	4.7241	4.5828	4.5029
$N = 200$	5.1171	4.6771	4.5407	4.7033	4.6551	4.5327
$N = 300$	4.9559	4.5545	4.4494	4.4312	4.4449	4.4393

Table 3. Empirical standard deviation $\mathcal{J}$, Kitagawa model.

Let $Q = 1$, $N = 50$, and in UPF $\alpha = 1$ and $\beta = 0$. Table 2 displays $\mathcal{J}$ for different values of R. The ordering of the algorithms is maintained; the difference between SIR and 1-S becomes significant; and for 1-S and SIR EMM provides better results than UPF. Note that in (6) f_n has strong variations, so the influence of the new data y_n is essential. This explains the difference between P-based algorithms and the SIR and 1-S ones, at least when R is small. On the other hand if the observations become very noisy ($R = 10$) the performance of 2-P is unaltered, while SIR and 1-S degrade.

Next in Table 3 we set $Q = 10$ and $R = 1$ and we see how $\mathcal{J}$ evolves with N. Let now $\alpha = 0.94$ and $\beta = 0$. As above, 1-S outperforms the P-based and the SIR algorithms, and we observe e.g. that 1-S(EMM) with $N = 50$ particles gives about the same result as 1-P with $N = 200$ particles.

4.3 Simulations, semi-linear models

In §4.2 we compared the P-based, SIR and 1-S algorithms, but not the 2-S one, because in (6) $p(x_n|x_{n-1}, y_n, y_{n+1})$ and $p(y_{n+1}|x_{n-1}, y_n)$ are difficult to

implement. Yet 2-S can be used in some situations. Let us first consider the non linear model with linear measurements equation

$$\begin{cases} x_{n+1} = f_n(x_n) + u_n \\ y_n \quad\;\; = 0.5x_n + v_n \end{cases}, \tag{7}$$

in which $u_n \sim \mathcal{N}(0, Q)$ and $v_n \sim \mathcal{N}(0, R)$ are i.i.d., mutually independent and independent of $x_0 \sim \mathcal{N}(0, 1)$ (the first equation of (6) and (7) coincide).

In (7) $p(x_n|x_{n-1}, y_n)$ and $p(y_n|x_{n-1})$ can be computed easily. $p(x_n|x_{n-1},$ $y_n,$ $y_{n+1})$ cannot, but the problem of computing $p(x_n|x_{n-1},$ $y_n,$ $y_{n+1})$ from $(p(x_n|x_{n-1}, y_n),$ $p(y_{n+1}|x_n))$ is the same as that of computing $p(x_n|x_{n-1}, y_n)$ from $(p(x_n|x_{n-1}), p(y_n|x_n))$ and so the approximation techniques recalled in §4.2 can be adapted to (7) (a difference however is that the exact moments of $p(x_n|x_{n-1}, y_n, y_{n+1})$ cannot be computed in (7)).

Let $R = 2$ and $N = 50$. For 2-S we use either a second-order Taylor series expansion, or UPF with $\alpha = 0.73$ and $\beta = \alpha^2 - 1$. Table 4 displays $\mathcal{J}$ as a function of Q. As we can see, the ordering 2-P < 1-P < SIR < 1-S is maintained. 2-S outperforms 1-S if Q is small, but 1-S performs better if Q increases. The reason why is that in (7) $p(x_n|x_{n-1}, y_n)$ (used in 1-S) can be computed exactly but $p(x_n|x_{n-1}, y_n, y_{n+1})$ (used in 2-S) cannot. Since all techniques indeed approximate f_n (up to the first orders), the results strongly depend on this function. In (6) f_n has strong variations; all orders matter, so all approximations of f_n at some point becomes very poor outside of a small neighbourghood of that point, and such situations do happen if Q gets large.

	2-P	1-P	SIR	1-S	2-S(Taylor)	2-S(UPF)
$Q = 0.1$	1.649036	1.5608385	1.5513972	1.1820396	1.1017785	1.0886346
$Q = 1$	2.1070664	1.8257239	1.6739058	1.5607033	2.4151141	1.6287723
$Q = 10$	2.8134363	2.4198548	2.3303342	2.2850875	2.8866604	2.4475675
$Q = 50$	3.8612771	2.9077013	2.7306043	2.70971	2.7548658	2.7188303

Table 4. Empirical standard deviation $\mathcal{J}$, semi-linear model.

Finally let us consider the semi-linear model (7), but in which the evolution equation is replaced by $x_{n+1} = \arctan x_n + u_n$. Let $\alpha = 0.88$, $\beta = \alpha^2 - 1$, $N = 50$ and $R = 2$. Table 6 displays $\mathcal{J}$ in terms of Q. By contrast with (7), function $f_n = \arctan$ is now very smooth. As a result all algorithms give satisfactory results, especially if Q is low. Also observe that the ordering of the algorithms is maintained, and in particular that 2-S outperforms 1-S, even when Q is large. The reason why is that for the arctan function limited order approximations are valid in a large domain, so the necessity of approximating $p(x_n|x_{n-1}, y_n, y_{n+1})$ is no longer a handicap of 2-S w.r.t. 1-S.

	2-P	1-P	SIR	1-S	2-S(UPF)
$Q = 1$	1.1240331	1.1202899	1.1118075	1.1041993	1.1033117
$Q = 5$	1.861067	1.8353582	1.8334354	1.8134666	1.8107098
$Q = 20$	2.611825	2.4849116	2.439016	2.4268744	2.4144096
$Q = 50$	3.1980866	2.761685	2.6592343	2.6345335	2.6329076

Table 5. Empirical standard deviation $\mathcal{J}$, alternate semi-linear model.

5 Conclusion

We explored direct and indirect paths (and their generic SMC implementations) for computing $p_{n|n}$ recursively. These algorithms remain PF, in the sense that their aim is to compute $p_{n|n}$, but possibly via a predictive or smoothing distribution. Our algorithms were validated by simulations. S-based algorithms outperform P-based ones, and in each class of algorithms better results are obtained (under fair conditions, i.e. when the necessary approximations are valid) when updating precedes propagation.

References

1. Arulampalam, M. S., Maskell, S., Gordon, N., and Clapp, T., "A Tutorial on Particle Filters for Online Nonlinear / Non-Gaussian Bayesian Tracking". *IEEE Transactions on Signal Processing* 50-2:174-188, February (2002).
2. Cappé, 0., Moulines E. and Rydén T., *Inference in Hidden Markov Models.* Springer-Verlag (2005).
3. Desbouvries, F. and Ait-El-Fquih, B., "Direct, prediction-based and smoothing-based particle filter algorithms". *Proc. 4th world conference of the Int. Ass. for Statistical Computing (IASC 2008)* Yokohama, Japan, Dec. 5-8 (2008).
4. Doucet, A., Godsill, S. J. and Andrieu, C., "On sequential Monte Carlo sampling methods for Bayesian filtering". *Statistics and Computing*, 10:197-208 (2000).
5. Doucet A., de Freitas N. and Gordon N. (eds.), *Sequential Monte Carlo methods in practice.* Springer Verlag (2001).
6. Gordon, N.J., Salmond, D.J. and Smith, A.F.M., "Novel approach to non linear/non-Gaussian Bayesian state estimation". *IEE Pr.-F.* 140:107-113 (1993).
7. Saha, S., Manda, P.K., Boers, Y., Driessen, H. and Bagchi, A., "Gaussian Proposal density using moment matching in SMC methods" *Statistics and Computing.* 19-2:203-208, June (2009).
8. Smith, A.F.M. and Gelfand, A.E., "Bayesian statistics without tears: a sampling-resampling perspective". *The American Statistician.* 46-2:84-87 (1992).
9. van der Merwe, R., Doucet, A., de Freitas, N. and Wan, E., "The unscented Particle Filter". *Advances in Neural Information Processing Systems.* (2001).