EXACT AND APPROXIMATE BAYESIAN SMOOTHING ALGORITHMS IN PARTIALLY OBSERVED MARKOV CHAINS

Boujemaa Ait-el-Fquih and François Desbouvries

GET / INT / Dépt. CITI and CNRS UMR 5157, 9 rue Charles Fourier, 91011 Evry, France

ABSTRACT

Let $\mathbf{x} = \{\mathbf{x}_n\}_{n\in\mathbb{N}}$ be a hidden process, $\mathbf{y} = \{\mathbf{y}_n\}_{n\in\mathbb{N}}$ an observed process and $\mathbf{r} = \{\mathbf{r}_n\}_{n\in\mathbb{N}}$ some auxiliary process. We assume that $\mathbf{t} = \{\mathbf{t}_n\}_{n\in\mathbb{N}}$ with $\mathbf{t}_n = (\mathbf{x}_n, \mathbf{r}_n, \mathbf{y}_{n-1})$ is a (Triplet) Markov Chain (TMC). TMC are more general than Hidden Markov Chains (HMC) and yet enable the development of efficient restoration and parameter estimation algorithms. This paper is devoted to Bayesian smoothing algorithms for TMC. We first propose twelve algorithms for general TMC. In the Gaussian case, they reduce to a set of algorithms which includes, among other solutions, extensions to TMC of classical Kalman-like smoothing algorithms such as the RTS algorithms, the Two-Filter algorithm or the Bryson and Frazier algorithm. We finally propose particle filtering (PF) approximations for the general case.

1. INTRODUCTION

HMC have been extensively studied for many years (see e.g. [1]). In an HMC $\mathbf{x}$ is first assumed to be a Markov chain (MC) (by the very meaning of the words "HMC"), and next the stochastic interactions of $\mathbf{x}$ and $\mathbf{y}$ are designed in such a way that $\mathbf{x}$ can be efficiently restored from $\mathbf{y}$. On the other hand, Pairwise [2] (PMC) and Triplet [3] MC (TMC) have been introduced recently. The TMC model describes the interactions between 3 processes : the hidden process $\mathbf{x}$, the observed process $\mathbf{y}$, and an auxiliary process $\mathbf{r} = \{\mathbf{r}_n\}_{n\in\mathbb{N}}$. The triplet $\mathbf{t} = (\mathbf{x}, \mathbf{r}, \mathbf{y})$ is called a TMC if $(\mathbf{x}, \mathbf{r}, \mathbf{y})$ is a (vector) MC. So a TMC can be viewed as a partially observed MC, in which one observes some component(s) $\mathbf{y}$ and one wants to restore (part of) the remaining ones $\mathbf{x}^* = (\mathbf{r}, \mathbf{x})$. The interest of TMC is twofold :

- As far as modeling is concerned, TMC are rather general models which include, in particular, some classical extensions of HMC [4] [5]. For instance, Hidden semi-Markov Chains are particular TMC with $\mathbf{x}$ and $\mathbf{r}$ discrete; Jump-Markov state space systems are particular TMC with $\mathbf{x}$ continuous and $\mathbf{r}$ discrete; state-space systems with colored process and/or measurement noise(s) are particular TMC with continuous $\mathbf{x}$ and $\mathbf{r}$.
- As far as restoration is concerned, in a TMC the variable $\mathbf{x}^*$ can be restored efficiently, and finally $\mathbf{x}$ is obtained by marginalization (such algorithms have been proposed in the discrete [3] or linear Gaussian [5] cases).

Let us now turn to the contribution of this paper. In §2 we first propose twelve smoothing algorithms for general continuous TMC. These algorithms are derived from Markovian properties of $\mathbf{t}$ only, considered as an MC both in the forward and backward directions. They can be classified into three classes : four forward filtering backward smoothing, four backward filtering forward smoothing, and four non recursive algorithms. We emphasize on the role played by four probability density functions (pdf) (α_n, β_n, γ_n and δ_n).

In §3 we address the particular case of Gaussian TMC. The general algorithms §2 reduce to a set of twelve specific algorithms (plus variations thereof) which include extensions to TMC of classical HMC smoothers, as well as original algorithms. Finally in §4 we propose PF solutions for the general (non linear and/or non Gaussian) case.

2. TMC BAYESIAN SMOOTHING ALGORITHMS

Let $\mathbf{x}_n \in \mathbb{R}^{n_\mathbf{x}}$ be the hidden process, $\mathbf{y}_n \in \mathbb{R}^{n_\mathbf{y}}$ the observation and $\mathbf{r}_n \in \mathbb{R}^{n_\mathbf{r}}$ the auxiliary process. Let $\mathbf{x}^*_n = (\mathbf{x}_n,\mathbf{r}_n)$ and $\mathbf{t}_n = (\mathbf{x}_n,\mathbf{r}_n,\mathbf{y}_{n-1})$. We assume that $\mathbf{t} = \{\mathbf{t}_n\}_{n\geq 0}$ (with $\mathbf{y}_{-1} = \mathbf{0}$) is an MC. Let $p(\mathbf{x}_{0:n})$ (resp. $p(\mathbf{x}^*_n|\mathbf{y}_{0:n})$), say, denote the pdf (w.r.t. Lebesgue measure) of $\mathbf{x}_{0:n}$ (resp. of $\mathbf{x}^*_n$ given $\mathbf{y}_{0:n}$); other pdfs of interest are defined similarly. The aim of this section is to propose general fixed-interval Bayesian smoothing algorithms for TMC, i.e. we want to compute the smoothing pdf $p(\mathbf{x}_n|\mathbf{y}_{0:N})$ for all n, $0 \leq n \leq N$. In the following we indeed focus on the computation of $p(\mathbf{x}^*_n|\mathbf{y}_{0:N})$; the pdf $p(\mathbf{x}_n|\mathbf{y}_{0:N})$ of interest is obtained by marginalization. The algorithms we propose can be classified into three families :

1. *Backward recursive algorithms.* These are two-pass algorithms, in which (i) $p(\mathbf{x}^*_n|\mathbf{y}_{0:N})$ is computed from $p(\mathbf{x}^*_{n+1}|\mathbf{y}_{0:N})$ via
$$p(\mathbf{x}^*_n|\mathbf{y}_{0:N}) = \int p(\mathbf{x}^*_{n+1}|\mathbf{y}_{0:N})p(\mathbf{x}^*_n|\mathbf{x}^*_{n+1}, \mathbf{y}_{0:N})d\mathbf{x}^*_{n+1} \quad (1)$$
(whence the term "backward recursive algorithm"); and (ii) $p(\mathbf{x}^*_n|\mathbf{x}^*_{n+1}, \mathbf{y}_{0:N})$ in (1) is computed in the forward direction;

2. *Forward recursive algorithms.* These are two-pass algorithms, in which (i) $p(\mathbf{x}_{n+1}^*|\mathbf{y}_{0:N})$ is computed from $p(\mathbf{x}_n^*|\mathbf{y}_{0:N})$ via

$$p(\mathbf{x}_{n+1}^*|\mathbf{y}_{0:N}) = \int p(\mathbf{x}_n^*|\mathbf{y}_{0:N})p(\mathbf{x}_{n+1}^*|\mathbf{x}_n^*, \mathbf{y}_{0:N})d\mathbf{x}_n^*, \quad (2)$$

and (ii) $p(\mathbf{x}_{n+1}^*|\mathbf{x}_n^*, \mathbf{y}_{0:N})$ in (2) is computed in the backward direction;

3. *Non-recursive algorithms.* $p(\mathbf{x}_n^*|\mathbf{y}_{0:N})$ is computed from two pdfs; one of them is computed recursively in the forward direction and the other recursively in the backward direction.

As we shall see, each one of the twelve algorithms (7)-(18) (which can all be derived from Bayes's rule, and from the fact that $\mathbf{t}$ is an MC both in the forward and in the backward directions) makes use of one (or two) out of the four pdfs $\alpha_n \stackrel{\text{def}}{=} p(\mathbf{x}_n^*|\mathbf{y}_{0:n-1})$, $\beta_n \stackrel{\text{def}}{=} p(\mathbf{y}_{n:N}|\mathbf{t}_n)$, $\gamma_n \stackrel{\text{def}}{=} p(\mathbf{x}_n^*|\mathbf{y}_{n-1:N})$ and $\delta_n \stackrel{\text{def}}{=} p(\mathbf{y}_{0:n-2}|\mathbf{t}_n)$. These pdfs, in turn, can be computed recursively (in the forward direction for α_n and δ_n, in the backward direction for β_n and γ_n); so for sake of clarity we first gather these recursions in §2.1.

2.1. Recursive algorithms for α_n, β_n, γ_n and δ_n

Let $\mathbf{t}$ be an MC. Then $\alpha_n = p(\mathbf{x}_n^*|\mathbf{y}_{0:n-1})$ and $\widetilde{\alpha}_n = p(\mathbf{x}_n^*|\mathbf{y}_{0:n})$ can be computed recursively (in the forward direction) as

$$\begin{cases} \widetilde{\alpha}_n = \frac{p(\mathbf{y}_n|\mathbf{t}_n)\alpha_n}{\int p(\mathbf{y}_n|\mathbf{t}_n)\alpha_n d\mathbf{x}_n^*} \\ \alpha_{n+1} = \int p(\mathbf{x}_{n+1}^*|\mathbf{t}_n, \mathbf{y}_n)\widetilde{\alpha}_n d\mathbf{x}_n^* \end{cases} ; \quad (3)$$

$\beta_n = p(\mathbf{y}_{n:N}|\mathbf{t}_n)$ and $\widetilde{\beta}_n = p(\mathbf{y}_{n+1:N}|\mathbf{t}_n, \mathbf{y}_n)$ can be computed recursively (in the backward direction) as

$$\begin{cases} \widetilde{\beta}_n = \int p(\mathbf{x}_{n+1}^*|\mathbf{t}_n, \mathbf{y}_n) \times \beta_{n+1}\, d\mathbf{x}_{n+1}^* \\ \beta_n = p(\mathbf{y}_n|\mathbf{t}_n)\, \widetilde{\beta}_n \end{cases} ; \quad (4)$$

$\gamma_n = p(\mathbf{x}_n^*|\mathbf{y}_{n-1:N})$ and $\widetilde{\gamma}_{n+1} = p(\mathbf{x}_{n+1}^*|\mathbf{y}_{n-1:N})$ can be computed recursively (in the backward direction) as

$$\begin{cases} \widetilde{\gamma}_{n+1} = \frac{p(\mathbf{y}_{n-1}|\mathbf{t}_{n+1})\gamma_{n+1}}{\int p(\mathbf{y}_{n-1}|\mathbf{t}_{n+1})\gamma_{n+1} d\mathbf{x}_{n+1}^*} \\ \gamma_n = \int p(\mathbf{x}_n^*|\mathbf{t}_{n+1}, \mathbf{y}_{n-1})\widetilde{\gamma}_{n+1} d\mathbf{x}_{n+1}^* \end{cases} ; \quad (5)$$

and $\delta_n = p(\mathbf{y}_{0:n-2}|\mathbf{t}_n)$ and $\widetilde{\delta}_{n+1} = p(\mathbf{y}_{0:n-2}|\mathbf{t}_{n+1}, \mathbf{y}_{n-1})$ can be computed recursively (in the forward direction) as

$$\begin{cases} \widetilde{\delta}_{n+1} = \int p(\mathbf{x}_n^*|\mathbf{t}_{n+1}, \mathbf{y}_{n-1}) \times \delta_n\, d\mathbf{x}_n^* \\ \delta_{n+1} = p(\mathbf{y}_{n-1}|\mathbf{t}_{n+1}) \times \widetilde{\delta}_{n+1} \end{cases} . \quad (6)$$

2.2. Backward recursive computation of the smoothing pdf

The aim of this section is to compute the backward conditional TMC pdf $p(\mathbf{x}_n^*|\mathbf{x}_{n+1}^*, \mathbf{y}_{0:N})$ in (1). Since $\mathbf{t}$ is an MC, $\mathbf{y}_{n+1:N}$ and $\mathbf{x}_n^*$ are independent conditionally on $(\mathbf{x}_{n+1}^*, \mathbf{y}_{0:n})$, so $p(\mathbf{x}_n^*|\mathbf{x}_{n+1}^*, \mathbf{y}_{0:N}) = p(\mathbf{x}_n^*|\mathbf{x}_{n+1}^*, \mathbf{y}_{0:n})$. As we now see, $p(\mathbf{x}_n^*|\mathbf{x}_{n+1}^*, \mathbf{y}_{0:n})$ can be computed from $\widetilde{\alpha}_n$, α_n or δ_n :

Proposition 1 *Let $\mathbf{t}$ be an MC. Then $\widetilde{\alpha}_n$ and α_{n+1} (resp. δ_n) can be computed in the forward direction by (3) (resp. (6)), and next $p(\mathbf{x}_n^*|\mathbf{x}_{n+1}^*, \mathbf{y}_{0:n})$ by*

$$p(\mathbf{x}_n^*|\mathbf{x}_{n+1}^*, \mathbf{y}_{0:n}) \propto p(\mathbf{x}_{n+1}^*|\mathbf{t}_n, \mathbf{y}_n)\widetilde{\alpha}_n, \quad (7)$$

$$\propto \begin{cases} p(\mathbf{x}_0^*, \mathbf{t}_1) & \text{if } n = 0 \\ p(\mathbf{x}_n^*|\mathbf{t}_{n+1}, \mathbf{y}_{n-1})\delta_n & \text{if } n \geq 1 \end{cases} \quad (8)$$

$$\propto \frac{p(\mathbf{x}_n^*|\mathbf{t}_{n+1}, \mathbf{y}_{n-1})\alpha_n}{p(\mathbf{x}_n^*|\mathbf{y}_{n-1})}, \quad (9)$$

$$\propto p(\mathbf{t}_{n+1}|\mathbf{t}_n)\delta_n p(\mathbf{x}_n^*|\mathbf{y}_{n-1}). \quad (10)$$

Finally $p(\mathbf{x}_n^*|\mathbf{y}_{0:N})$ can be computed by (1).

2.3. Forward recursive computation of the smoothing pdf

This section is parallel to §2.2. Our aim here is to compute the forward conditional TMC pdf $p(\mathbf{x}_{n+1}^*|\mathbf{x}_n^*, \mathbf{y}_{0:N})$ in (2). Since $\mathbf{t}$ is an MC, $\mathbf{y}_{0:n-2}$ and $\mathbf{x}_{n+1}^*$ are independent conditionally on $(\mathbf{x}_n^*, \mathbf{y}_{n-1:N})$, so $p(\mathbf{x}_{n+1}^*|\mathbf{x}_n^*, \mathbf{y}_{0:N}) = p(\mathbf{x}_{n+1}^*|\mathbf{x}_n^*, \mathbf{y}_{n-1:N})$. Now $p(\mathbf{x}_{n+1}^*|\mathbf{x}_n^*, \mathbf{y}_{n-1:N})$ can be computed from β_n, γ_n or $\widetilde{\gamma}_n$:

Proposition 2 *Let $\mathbf{t}$ be an MC. Then β_n (resp. γ_n and $\widetilde{\gamma}_{n+1}$) can be computed in the backward direction by (4) (resp. (5)), and next $p(\mathbf{x}_{n+1}^*|\mathbf{x}_n^*, \mathbf{y}_{n-1:N})$ by*

$$p(\mathbf{x}_{n+1}^*|\mathbf{x}_n^*, \mathbf{y}_{n-1:N}) \propto p(\mathbf{x}_{n+1}^*|\mathbf{t}_n, \mathbf{y}_n)\beta_{n+1}, \quad (11)$$

$$\propto p(\mathbf{x}_n^*|\mathbf{t}_{n+1}, \mathbf{y}_{n-1})\widetilde{\gamma}_{n+1}, \quad (12)$$

$$\propto \frac{p(\mathbf{x}_{n+1}^*|\mathbf{t}_n, \mathbf{y}_n)\gamma_{n+1}}{p(\mathbf{x}_{n+1}^*|\mathbf{y}_n)}, \quad (13)$$

$$\propto p(\mathbf{t}_n|\mathbf{t}_{n+1})\beta_{n+1}p(\mathbf{x}_{n+1}^*|\mathbf{y}_n). \quad (14)$$

Finally $p(\mathbf{x}_n^*|\mathbf{y}_{0:N})$ can be computed by (2).

2.4. Non recursive computation of the smoothing pdf

Let us now see that $p(\mathbf{x}_n^*|\mathbf{y}_{0:N})$ can be essentially computed as a (normalized) product of α_n (or δ_n) and β_n (or γ_n), which leads to four algorithms :

Proposition 3 *Let $\mathbf{t}$ be an MC. Then the smoothing pdf can be computed as*

$$p(\mathbf{x}_n^*|\mathbf{y}_{0:N}) \propto \alpha_n \times \beta_n, \quad (15)$$

$$\propto \gamma_n \times \delta_n, \quad (16)$$

$$\propto \frac{\alpha_n \times \gamma_n}{p(\mathbf{x}_n^*|\mathbf{y}_{n-1})}, \quad (17)$$

$$\propto \delta_n \times \beta_n \times p(\mathbf{x}_n^*|\mathbf{y}_{n-1}), \quad (18)$$

in which α_n (resp. δ_n) is computed in the forward direction by (3) (resp. (6)), and β_n (resp. γ_n) is computed in the backward direction by (4) (resp. (5)).

3. EXACT COMPUTATION : THE GAUSSIAN CASE

In the Gaussian case the equations in §2 can be computed explicitely, which yields twelve algorithms (plus variations thereof); when further particularized to HMC, some of these algorithms coincide with classical smoothing solutions already proposed in the literature. More precisely :

- Equation (3) reduces to an algorithm which extends to TMC the Kalman filter;
- equation (4) reduces to an algorithm which propagates $\arg\max_{\mathbf{x}_n^*} \beta_n$, and which generalizes to TMC the backward algorithm used in the two-filter smoother by Mayne [6] (see also [7, eqs. (10.4.14)-(10.4.15)]);
- equation (5) reduces to filter or information forms algorithms; in particular, the filter form algorithm has an HMC counterpart [7, §9.8];
- equation (6) reduces to an algorithm which propagates $\arg\max_{\mathbf{x}_n^*} \delta_n$. After some manipulations, one can show that it has a counterpart in the HMC case, introduced in the context of complementary models by Weinert (see [8, §3.2]);
- equations (1) and (7) reduce to an algorithm which extends to TMC the RTS algorithm [9];
- equations (1) and (8) reduce to an algorithm which extends to TMC an algorithm introduced by Weinert [8, p. 40];
- eqs. (2) and (11) reduce to an algorithm which extends to TMC an algorithm introduced in [10] [8, p. 35];
- equations (2) and (12) reduce to an algorithm which extends to TMC an algorithm partially found in [7, pp. 401, Exs. 10.12 & 10.14];
- equation (15) reduces to an algorithm which extends to TMC the two-filter algorithm [6] (see also [11]);
- equation (17) is the counterpart of an algorithm which extends to TMC the General two-filter algorithm [7, Thm. 10.4.1];
- finally, one can show after some computations that (18) reduces to an algorithm which extends to TMC an algorithm [8, §3.3] introduced in the context of complementary models.

4. APPROXIMATE COMPUTATION : PARTICLE SMOOTHING SOLUTIONS

In the general case, equations (3)-(18) are impossible (or difficult) to compute exactly, and PF (see e.g. [12] [13] [14]) is then one possible approximate solution. As we now see, the smoothing pdf can be approximated by the discrete pdf $p(\mathbf{x}_n^*|\mathbf{y}_{0:N}) \approx \sum_{i=1}^S w_n^{s,(i)}\ \delta(\mathbf{x}_n^* - \mathbf{x}_n^{*(i)})$. Let us give 3 examples. Section 4.1 deals with the approximation of the backward algorithm ((1) & (7)), §4.2 with the approximation of the forward algorithm ((2) & (13)), and §4.3 with an approximation of the non recursive algorithm (17).

4.1. Backward particle smoothing :

Let us first consider a PF approximation of (1) & (7). The associated PF is a two-pass algorithm. In the forward pass, a PF algorithm [15] propagates the filtering importance weights $w_n^{f,(i)}$ in the forward direction; next the smoothing weights $w_n^{s,(i)}$ are computed recursively in the backward direction.

Proposition 4 (Backward particle smoothing) *Let $p(\mathbf{x}_n^*|\mathbf{y}_{0:n}) \approx \sum_{i=1}^S w_n^{f,(i)} \delta(\mathbf{x}_n^* - \mathbf{x}_n^{*(i)})$ be computed recursively (until $n = N$) by [15]; Then $p(\mathbf{x}_n^*|\mathbf{y}_{0:N}) \approx \sum_{i=1}^S w_n^{s,(i)} \delta(\mathbf{x}_n^* - \mathbf{x}_n^{*(i)})$, in which $\{w_n^{s,(i)}\}_{i=1}^S$ can be computed recursively (from $n = N$ to $n = 0$) as*

1. *Initialization $n = N$: for $i = 1 : S$, $w_N^{s,(i)} = w_N^{f,(i)}$;*
2. *For all $n = N-1 : 0$ and $i = 1 : S$,*

$$w_n^{s,(i)} = \sum_{j=1}^S w_{n+1}^{s,(j)} \frac{w_n^{f,(i)} p(\mathbf{x}_{n+1}^{*(j)}|\mathbf{x}_n^{*(i)}, \mathbf{y}_n, \mathbf{y}_{n-1})}{\sum_{l=1}^S w_n^{f,(l)} p(\mathbf{x}_{n+1}^{*(j)}|\mathbf{x}_n^{*(l)}, \mathbf{y}_n, \mathbf{y}_{n-1})}. \tag{19}$$

Proof 1 *The forward PF algorithm propagates the approximation $p(\mathbf{x}_n^*|\mathbf{y}_{0:n}) \approx \sum_{i=1}^S w_n^{f,(i)} \delta(\mathbf{x}_n^* - \mathbf{x}_n^{*(i)})$. Then from (7), $p(\mathbf{x}_n^*|\mathbf{x}_{n+1}^*, \mathbf{y}_{0:n})$ is approximated by*

$$\sum_{i=1}^S \frac{w_n^{f,(i)} p(\mathbf{x}_{n+1}^*|\mathbf{x}_n^{*(i)}, \mathbf{y}_n, \mathbf{y}_{n-1}) \delta(\mathbf{x}_n^* - \mathbf{x}_n^{*(i)})}{\sum_{l=1}^S w_n^{f,(l)} p(\mathbf{x}_{n+1}^*|\mathbf{x}_n^{*(l)}, \mathbf{y}_n, \mathbf{y}_{n-1})}. \tag{20}$$

Let now $p(\mathbf{x}_{n+1}^*|\mathbf{y}_{0:N}) \approx \sum_{j=1}^S w_{n+1}^{s,(j)}\ \delta(\mathbf{x}_{n+1}^* - \mathbf{x}_{n+1}^{*(j)})$. From (20), (1) is approximated by $p(\mathbf{x}_n^*|\mathbf{y}_{0:N}) \approx \sum_{i=1}^S w_n^{s,(i)} \delta(\mathbf{x}_n^* - \mathbf{x}_n^{*(i)})$ in which $w_n^{s,(i)}$ is given by (19). ■

4.2. Forward particle smoothing :

Let us now approximate the algorithm ((2) & (13)). Similarly to §4.1, ((2) & (13)) reduce to a two-pass algorithm. In the backward pass, a backward PF algorithm (see §4.4) propagates an approximation of γ_{n+1}, and next the smoothing weights are computed in the forward direction :

Proposition 5 (Forward particle smoothing) *Let* $p(\mathbf{x}^*_{n+1}|\mathbf{y}_{n:N}) \approx \sum_{i=1}^{S} w_{n+1}^{b,(i)} \delta(\mathbf{x}^*_{n+1} - \mathbf{x}^{*(i)}_{n+1})$ *be computed recursively (until* $n = 0$*) by the algorithm of §4.4. Then* $p(\mathbf{x}^*_n|\mathbf{y}_{0:N}) \approx \sum_{i=1}^{S} w_n^{s,(i)} \delta(\mathbf{x}^*_n - \mathbf{x}^{*(i)}_n)$*, in which* $\{w_n^{s,(i)}\}_{i=1}^{S}$ *are computed recursively (from* $n = 0$ *to* $n = N$*) as*

1. *Initialization* $n = 0$ *: for* $i = 1$ *to* S, $w_0^{s,(i)} = w_0^{b,(i)}$.
2. *For all* $n = 1$ *to* N *and* $i = 1$ *to* S,

$$w_n^{s,(i)} = \sum_{j=1}^{S} w_{n-1}^{s,(j)} \frac{w_n^{b,(i)} \frac{p(\mathbf{x}_n^{*(i)}|\mathbf{x}_{n-1}^{*(j)}, \mathbf{y}_{n-1}, \mathbf{y}_{n-2})}{p(\mathbf{x}_n^{*(i)}|\mathbf{y}_{n-1})}}{\sum_{l=1}^{S} w_n^{b,(l)} \frac{p(\mathbf{x}_n^{*(l)}|\mathbf{x}_{n-1}^{*(j)}, \mathbf{y}_{n-1}, \mathbf{y}_{n-2})}{p(\mathbf{x}_n^{*(l)}|\mathbf{y}_{n-1})}}. \tag{21}$$

4.3. Non recursive particle smoothing :

Let us finally propose a PF approximation of (17). The smoothing weights $w_n^{s,(i)}$ are computed from the forward ones $w_n^{f,(i)}$ (computed by a forward PF algorithm [15]) and the backward ones $w_n^{b,(i)}$ (computed by the algorithm of §4.4) :

Proposition 6 (Non recursive particle smoothing) *At time* n*, the smoothing weights* $\{w_n^{s,(i)}\}_{i=1}^{S}$ *can be computed as*

$$w_n^{s,(i)} = \frac{w_n^{b,(i)} \sum_{j=1}^{S} w_{n-1}^{f,(j)} \frac{p(\mathbf{x}_n^{*(i)}|\mathbf{x}_{n-1}^{*(j)}, \mathbf{y}_{n-1}, \mathbf{y}_{n-2})}{p(\mathbf{x}_n^{*(i)}|\mathbf{y}_{n-1})}}{\sum_{l=1}^{S} w_n^{b,(l)} \sum_{j=1}^{S} w_{n-1}^{f,(j)} \frac{p(\mathbf{x}_n^{*(l)}|\mathbf{x}_{n-1}^{*(j)}, \mathbf{y}_{n-1}, \mathbf{y}_{n-2})}{p(\mathbf{x}_n^{*(l)}|\mathbf{y}_{n-1})}}, \tag{22}$$

in which $w_n^{f,(i)}$ *are computed in the forward direction [15], and* $w_n^{b,(i)}$ *in the backward direction (see Proposition 7).*

4.4. Backward TMC PF

The algorithms of §4.2 and 4.3 make use of a backward PF algorithm (for computing an approximation of $\gamma_n = p(\mathbf{x}^*_n|\mathbf{y}_{n-1:N})$) which we now derive by following [15]. Let $\mathbf{t}_n^{(i)} = [(\mathbf{x}_n^{*(i)})^T, \mathbf{y}_{n-1}^T]^T$ and let $p(\mathbf{x}^*_{n+1:N+1}|\mathbf{y}_{n:N}) \approx \sum_{i=1}^{S} w_{n+1}^{b,(i)} \delta(\mathbf{x}^*_{n+1:N+1} - \mathbf{x}^{*(i)}_{n+1:N+1})$, in which $\widetilde{w}_{n+1}^{b,(i)} = \frac{p(\mathbf{t}^{(i)}_{n+1:N+1})}{q(\mathbf{x}^{*(i)}_{n+1:N+1}|\mathbf{y}_{n:N})}$, $w_{n+1}^{b,(i)} = \frac{\widetilde{w}_{n+1}^{b,(i)}}{\sum_{j=1}^{S} \widetilde{w}_{n+1}^{b,(j)}}$, and $\{\mathbf{x}^{*(i)}_{n+1:N+1}\}_{i=1}^{S}$ are drawn from some importance function $q(\mathbf{x}^*_{n+1:N+1}|\mathbf{y}_{n:N})$. Let us assume that q factorizes as $q(\mathbf{x}^*_{n:N+1}|\mathbf{y}_{n-1:N}) = q(\mathbf{x}^*_n|\mathbf{x}^*_{n+1:N+1}, \mathbf{y}_{n-1:N}) \times q(\mathbf{x}^*_{n+1:N+1}|\mathbf{y}_{n:N})$. Then the importance weights $w_n^{b,(i)}$ and the particles $\mathbf{x}_n^{*(i)}$ can be propagated as follows :

Proposition 7 (Backward TMC PF) *For all* $n = N$ *to* 0 *and* $i = 1$ *to* S,

- *sample* $\mathbf{x}_n^{*(i)} \sim q(\mathbf{x}^*_n|\mathbf{x}^{*(i)}_{n+1:N+1}, \mathbf{y}_{n-1:N})$;
- *compute* $\widetilde{w}_n^{b,(i)} \frac{p(\mathbf{t}_n^{(i)}|\mathbf{t}_{n+1}^{(i)})}{q(\mathbf{x}_n^{*(i)}|\mathbf{x}^{*(i)}_{n+1:N+1}, \mathbf{y}_{n-1:N})} \times \widetilde{w}_{n+1}^{b,(i)}$ *and* $w_n^{b,(i)} = \frac{\widetilde{w}_n^{b,(i)}}{\sum_{i=1}^{S} \widetilde{w}_n^{b,(i)}}$.

5. REFERENCES

[1] Y. Ephraim and N. Merhav, "Hidden Markov processes," *IEEE Tr. Info. Th.*, vol. 48, no. 6, pp. 1518–69.

[2] W. Pieczynski, "Pairwise Markov chains," *IEEE Tr. PAMI*, vol. 25, no. 5, pp. 634–39, May 2003.

[3] W. Pieczynski, C. Hulard, and T. Veit, "Triplet Markov chains in hidden signal restoration," in *SPIE Int. Symp. Remote Sensing*, Crete, Grece, Sept. 22-27 2002.

[4] W. Pieczynski and F. Desbouvries, "On Triplet Markov chains," in *Proc. Int. Symp. on Applied Stochastic Models and Data Analysis*, Brest, France, May 17-20 2005.

[5] B. Ait-El-Fquih and F. Desbouvries, "Kalman filtering in triplet Markov chains," *IEEE Transactions on Signal Processing*, vol. 54, no. 8, pp. 2957–63, August 2006.

[6] D. Q. Mayne, "A solution to the smoothing problem for linear dynamical systems," *Automatica*, vol. 4, pp. 73–92, 1966.

[7] T. Kailath, A. H. Sayed, and B. Hassibi, *Linear estimation*, Prentice Hall, Upper Saddle River, NJ, 2000.

[8] H. L. Weinert, *Fixed interval smoothing for state-space models*, Kluwer Academic Publishers, Boston, 2001.

[9] H. E. Rauch, F. Tung, and C. T. Striebel, "Maximum likelihood estimates of linear dynamic systems," *AIAA J.*, vol. 3, no. 8, pp. 1445–1450, August 1965.

[10] U. Desai, H. Weinert, and G. Yusypchuk, "Discrete-time complementary models and smoothing algorithms : The correlated noise case," *IEEE Transactions on Automatic Control*, vol. 28, no. 4, pp. 536–539, April 1983.

[11] D. C. Fraser and J. E. Potter, "The optimum linear smoother as a combination of two optimum linear filters," *IEEE Tr. Aut. Control*, pp. 387–90, Aug. 1969.

[12] M. Sanjeev Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear / non-Gaussian Bayesian tracking," *IEEE Tr. Signal Processing*, vol. 50, no. 2, pp. 174–188, February 2002.

[13] A. Doucet, N. de Freitas, and N. Gordon, Eds., *Sequential Monte Carlo Methods in Practice*, Springer Verlag, New York, 2001.

[14] M. Klaas, M. Briers, N. de Freitas, A. Doucet, S. Maskell, and D. Lang, "Fast particle smoothing : If I had a million particles," in *Proceedings of the ICML*, Pittsburgh, PA, 2006.

[15] F. Desbouvries and W. Pieczynski, "Particle filtering in pairwise and triplet Markov chains," in *Proc. IEEE - EURASIP Workshop on Nonlinear Signal and Image Processing*, Grado-Gorizia, Italy, June 8-11 2003.