BAYESIAN SMOOTHING ALGORITHMS IN PAIRWISE AND TRIPLET MARKOV CHAINS

B. Ait-El-Fquih & F. Desbouvries

GET/INT/CITI and CNRS UMR 5157
9 rue Charles Fourier, 91011 Evry, France
Boujemaa.Ait_Elfquih@int-evry.fr, Francois.Desbouvries@int-evry.fr

ABSTRACT

An important problem in signal processing consists in estimating an unobservable process $\mathbf{x} = \{\mathbf{x}_n\}_{n \in \mathbb{N}}$ from an observed process $\mathbf{y} = \{\mathbf{y}_n\}_{n \in \mathbb{N}}$. In Linear Gaussian Hidden Markov Chains (LGHMC), recursive solutions are given by Kalman-like Bayesian restoration algorithms. In this paper, we consider the more general framework of Linear Gaussian Triplet Markov Chains (LGTMC), i.e. of models in which the triplet $(\mathbf{x}, \mathbf{r}, \mathbf{y})$ (where $\mathbf{r} = \{\mathbf{r}_n\}_{n \in \mathbb{N}}$ is some additional process) is Markovian and Gaussian. We address fixed-interval smoothing algorithms, and we extend to LGTMC the RTS algorithm by Rauch, Tung and Striebel, as well as the Two-Filter algorithm by Mayne and Fraser and Potter.

1. INTRODUCTION

An important problem in signal processing consists in recursively estimating an unobservable process $\mathbf{x} = \{\mathbf{x}_n\}_{n \in \mathbb{N}}$ from an observed process $\mathbf{y} = \{\mathbf{y}_n\}_{n \in \mathbb{N}}$. This is done classically in the framework of Hidden Markov Chains (HMC), which have been extensively studied for many years (see e.g. the recent tutorial [1]).

In this paper we deal with the recently introduced Pairwise [2] (PMC) and Triplet [3] Markov Chains (TMC). In the PMC model we assume that the pair $(\mathbf{x}, \mathbf{y})$ is a Markov Chain (MC), and in the TMC model that the triplet $(\mathbf{x}, \mathbf{r}, \mathbf{y})$ (in which $\mathbf{r} = \{\mathbf{r}_n\}_{n \in \mathbb{N}}$ is some additional process) is an MC. These models are more general than the HMC model and yet enable the development of efficient restoration algorithms of the hidden process $\mathbf{x}$. In particular, a Kalman-like filtering algorithm for PMC (resp. for TMC) has been proposed in [4] (resp. in [5]). The aim of this paper is to propose Bayesian smoothing algorithms for PMC and TMC.

The paper is organized as follows. In section 2 we briefly recall the Kalman filter as well as some of its many extensions. We next recall the three embedded HMC, PMC and TMC models, and for illustrative purposes we show that two classical extensions of the standard state-space system used in Kalman filtering (namely the jump-Markov processes, and the state-space systems with colored process and measurement noises) are some particular TMC. In section 3 we propose smoothing algorithms for general PMC and TMC. Finally in section 4 we consider the particular case of linear and Gaussian TMC; the algorithms of section 3 then reduce to an extension of the RTS algorithm [6] by Rauch, Tung and Striebel, on the one hand, and of the Two-Filter algorithm by Mayne [7] and Fraser and Potter [8], on the other hand.

2. PARTIALLY OBSERVED MARKOV CHAINS

2.1. The Kalman filter and its extensions

Let us consider the classical state-space system :

$$\left\{ \begin{array}{lcl} \mathbf{x}_{n+1} & = & \mathbf{F}_n \mathbf{x}_n + \mathbf{G}_n \mathbf{u}_n \\ \mathbf{y}_n & = & \mathbf{H}_n \mathbf{x}_n + \mathbf{J}_n \mathbf{v}_n \end{array} \right. , \quad (1)$$

in which $\mathbf{x}_n \in \mathbb{R}^{n_\mathbf{x}}$ is the state, $\mathbf{y}_n \in \mathbb{R}^{n_\mathbf{y}}$ is the observation, $\mathbf{u}_n \in \mathbb{R}^{n_\mathbf{u}}$ is the process noise and $\mathbf{v}_n \in \mathbb{R}^{n_\mathbf{v}}$ is the measurement noise. The processes $\mathbf{u} = \{\mathbf{u_n}\}_{n \in \mathbb{N}}$ and $\mathbf{v} = \{\mathbf{v_n}\}_{n \in \mathbb{N}}$ are assumed to be independent, jointly independent and independent of $\mathbf{x}_0$.

Let $\mathbf{x}_{0:n} = \{\mathbf{x}_i\}_{i=0}^n$ and $\mathbf{y}_{0:n} = \{\mathbf{y}_i\}_{i=0}^n$. Let also $p(\mathbf{x}_n)$, $p(\mathbf{x}_{0:n})$ and $p(\mathbf{x}_n|\mathbf{y}_{0:n})$, say, denote the probability density function (pdf) (w.r.t. Lebesgue measure) of $\mathbf{x}_n$, the pdf of $\mathbf{x}_{0:n}$, and the pdf of $\mathbf{x}_n$, conditional on $\mathbf{y}_{0:n}$, respectively; the other pdf are defined similarly. The filtering problem consists in computing the posterior pdf $p(\mathbf{x}_n|\mathbf{y}_{0:n})$. If furthermore $\mathbf{x}_0$ and $\mathbf{w}_n = (\mathbf{u_n}, \mathbf{v_n})$ are Gaussian, then $p(\mathbf{x}_n|\mathbf{y}_{0:n})$ is also Gaussian and is thus described by its mean and covariance matrix. Propagating $p(\mathbf{x}_n|\mathbf{y}_{0:n})$ over time amounts to propagating these parameters, and the filtering algorithm reduces to the celebrated Kalman filter [9], see also [10].

Since this pionnering work, the Kalman filter has been generalized in many directions. To name just a few examples, robust (i.e., square-root type) or fast (i.e., Chandrasekhar type) algorithms have been proposed; restoration algorithms for smoothing or prediction problems have been developed; the independence assumptions on $\mathbf{u}$ and $\mathbf{v}$ have been dropped, leading to linear models with colored process and/or measurement noises; and the extension of (1) to non-linear and/or non-Gaussian systems has been addressed, leading to approximate solutions such as the extended Kalman filter and more recently particle filters. The literature on all these extensions is vast (see e.g. [11] [12] [13] as well as the references therein).

2.2. Embedded Markovian models : HMC $\subset$ PMC $\subset$ TMC

On the other hand, yet another direction in which it is possible to extend the Kalman filter consists in releasing some conditional independence assumptions among $\mathbf{x}$ and $\mathbf{y}$. Let us first come back to model (1). From (1), we get

$$p(\mathbf{x}_{n+1}|\mathbf{x}_{0:n}) = p(\mathbf{x}_{n+1}|\mathbf{x}_n) \; ; \quad (2)$$

$$p(\mathbf{y}_{0:n}|\mathbf{x}_{0:n}) = \prod_{i=0}^{n} p(\mathbf{y}_i|\mathbf{x}_{0:n}) \; ; \quad (3)$$

$$p(\mathbf{y}_i|\mathbf{x}_{0:n}) = p(\mathbf{y}_i|\mathbf{x}_i) \text{ for all } i, \; 0 \le i \le n \; . \quad (4)$$

In other words, $\mathbf{x}$ is an MC, and since $\mathbf{x}$ is known only through the observed process $\mathbf{y}$, (1) is an HMC.

In an HMC $\mathbf{x}$ is first assumed to be an 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, a PMC is a model in which the pair $(\mathbf{x}, \mathbf{y})$ is assumed to be an MC. So in a PMC $\mathbf{x}$ and $\mathbf{y}$ are modeled altogether (and in a symmetric way), and a PMC can indeed be seen as a partially observed vector MC, in which we observe one component $\mathbf{y}$ and we want to restore the other one $\mathbf{x}$.

Now if (2) to (4) hold, then $(\mathbf{x}, \mathbf{y})$ is an MC, so any HMC is also a PMC. The converse is not true, because if $(\mathbf{x}, \mathbf{y})$ is a (vector) MC then the marginal process $\mathbf{x}$ is not necessarily an MC; moreover, conditionnally on $\mathbf{x}_{0:n}$, the variables $\{\mathbf{y}_i\}_{i=0}^{n}$ form an MC and thus are not necessarily independent [4]. On the other hand, due to the symmetry of the PMC model, the conditional law of $\mathbf{x}_{0:n}$ given $\mathbf{y}_{0:n}$ is also Markovian. This key computational property (which in the context of HMC is well known, see e.g. [1, eq. (5.21) p. 1539]), in turn, enables the derivation of efficient HMC-like restoration algorithms. In particular, in the linear Gaussian case, the extension to PMC of the Kalman filter has been considered in [14, Corollary 1 p. 72] and [4][1].

The PMC model can be further generalized to the TMC model [3] which we now recall. A TMC is a stochastic dynamical model which describes the interactions between 3 processes : the hidden process $\mathbf{x}$, the observed process $\mathbf{y}$, and a third process $\mathbf{r}$ which, depending on the application, can have different physical meanings (see e.g. the examples given below). By definition, the triplet $\mathbf{t} = (\mathbf{x}, \mathbf{r}, \mathbf{y})$ is a TMC if $(\mathbf{x}, \mathbf{r}, \mathbf{y})$ is a (vector) Markov chain. The interest of TMC is twofold :

- As far as modeling is concerned, if $(\mathbf{r}, (\mathbf{x}, \mathbf{y}))$ is an MC then the marginal process $(\mathbf{x}, \mathbf{y})$ is not necessarily an MC, so TMC are not necessarily PMC;
- As far as restoration is concerned, the TMC $(\mathbf{r}, \mathbf{x}, \mathbf{y})$ can be viewed as the PMC $((\mathbf{r}, \mathbf{x}), \mathbf{y})$; so $\mathbf{x}^* = (\mathbf{r}, \mathbf{x})$ can be restored from $\mathbf{y}$ by a PMC algorithm, and finally $\mathbf{x}$ is obtained by marginalization (such algorithms have been proposed in the discrete [3] or linear Gaussian [5] cases).

Finally, let us notice that in practice computer experiments have demonstrated the superiority of PMC [15] (resp. TMC [16]) over HMC in the context of image segmentation.

2.3. Some classical TMC

It happens that some classical generalizations of the state-space system (1) are particular TMC. For an illustrative purpose, let us leave in this section the general discussion and consider the following two examples (the first one with a discrete latent process $\mathbf{r}$, the second one with a continuous latent process $\mathbf{r}$). Other examples of particular TMC models can be found in [5] [17].

2.3.1. Jump-Markov model

Let us consider the jump-Markov model (see e.g. [18] [19]):

$$\begin{cases} \mathbf{x}_{n+1} = \mathbf{F}(r_{n+1})\mathbf{x}_n + \mathbf{G}(r_{n+1})\mathbf{u}_n \\ \mathbf{y}_n = \mathbf{H}(r_n)\mathbf{x}_n + \mathbf{J}(r_n)\mathbf{v}_n \end{cases} , \quad (5)$$

in which $\mathbf{u} = \{\mathbf{u}_n\}_{n\in\mathbb{N}}$ and $\mathbf{v} = \{\mathbf{v}_n\}_{n\in\mathbb{N}}$ are zero-mean, Gaussian, independent, jointly independent and independent of $\mathbf{x}_0$, and $r = \{r_n\}_{n\in\mathbb{N}}$ is a scalar discrete MC. Conditionnally on r, (5) is a classical linear state-space system (1), and as such an HMC. Now if we consider the whole model $\mathbf{x}$, $\mathbf{y}$ and r, then the process $\mathbf{t} = \{\mathbf{t}_n\}$ with $\mathbf{t}_n = (\mathbf{x}_n, r_n, \mathbf{y}_n)$ is a particular TMC.

2.3.2. State-space model with colored noises

Let us once again consider model (1), but in which we now assume that

$$\underbrace{\begin{bmatrix} \mathbf{u}_{n+1} \\ \mathbf{v}_{n+1} \end{bmatrix}}_{\mathbf{r}_{n+1}} = \underbrace{\begin{bmatrix} \mathbf{A}_n^{\mathbf{u},\mathbf{u}} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_n^{\mathbf{v},\mathbf{v}} \end{bmatrix}}_{\mathbf{A}_n} \begin{bmatrix} \mathbf{u}_n \\ \mathbf{v}_n \end{bmatrix} + \underbrace{\begin{bmatrix} \xi_n^{\mathbf{u}} \\ \xi_n^{\mathbf{v}} \end{bmatrix}}_{\xi_n},$$

where $\xi^{\mathbf{u}} = \{\xi_n^{\mathbf{u}}\}_{n\in\mathbb{N}}$ (resp. $\xi^{\mathbf{v}} = \{\xi_n^{\mathbf{v}}\}_{n\in\mathbb{N}}$) is zero-mean, independent and independent of $\mathbf{u}_0$ (resp. of $\mathbf{v}_0$), and $\xi^{\mathbf{u}}$

[1] more precisely, if (2) to (4) hold then both $\{(\mathbf{x}_n, \mathbf{y}_n)\}_{n\in\mathbb{N}}$ and $\{(\mathbf{x}_{n+1}, \mathbf{y}_n)\}_{n\in\mathbb{N}}$ are MC; the difference between [14] and [4] stems from these two candidate definitions of a PMC (see also the remark at the end of [4, section 3]).

and $\xi^{\mathbf{v}}$ are independent. Each one of the two processes $\mathbf{u} = \{\mathbf{u}_n\}_{n\in\mathbb{N}}$ and $\mathbf{v} = \{\mathbf{v}_n\}_{n\in\mathbb{N}}$ is thus an MC, and $\mathbf{u}$ is independent of $\mathbf{v}$. Such a model has been introduced by Sorenson [20] (see also [21, ch. 5]). It is no longer an HMC ($\mathbf{x}$ is not an MC), but the whole model $\mathbf{t}_n = (\mathbf{x}_n, \mathbf{r}_n, \mathbf{y}_{n-1})$ can be rewritten as

$$\begin{bmatrix} \mathbf{x}_{n+1} \\ \mathbf{r}_{n+1} \\ \mathbf{y}_n \end{bmatrix} = \begin{bmatrix} \mathbf{F}_n & \overline{\mathbf{G}}_n & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_n & \mathbf{0} \\ \mathbf{H}_n & \overline{\mathbf{J}}_n & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x}_n \\ \mathbf{r}_n \\ \mathbf{y}_{n-1} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \xi_n \\ \mathbf{0} \end{bmatrix}$$

(with $\overline{\mathbf{G}}_n = [\mathbf{G}_n, \mathbf{0}]$ and $\overline{\mathbf{J}}_n = [\mathbf{0}, \mathbf{J}_n]$), and so $\mathbf{t} = \{\mathbf{t}_n\}$ is a TMC.

3. SMOOTHING ALGORITHMS FOR PMC AND TMC

Let again $\mathbf{x} = \{\mathbf{x}_n\}_{n\in\mathbb{N}}$ be the hidden state process, $\mathbf{y} = \{\mathbf{y}_n\}_{n\in\mathbb{N}}$ the observed process and $\mathbf{r} = \{\mathbf{r}_n\}_{n\in\mathbb{N}}$ the additional process. From now on we say that the process $\mathbf{t} = \{\mathbf{t}_n\}_{n\in\mathbb{N}}$, with $\mathbf{t}_n = (\mathbf{x}_n, \mathbf{r}_n, \mathbf{y}_{n-1}) = (\mathbf{x}_n^*, \mathbf{y}_{n-1})$ is a TMC if $\mathbf{t}$ is an MC. The aim of this section is to derive Bayesian fixed-interval smoothing algorithms for TMC.

Let us first briefly recall the existing fast Bayesian smoothing algorithms in the HMC framework :

- In the case of a discrete hidden process $\mathbf{x}$, the forward-backward (Baum-Welch) algorithm [22] [23] enables to compute $p(\mathbf{x}_n|\mathbf{y}_{0:N})$ (for an arbitrary n, $0 \le n \le N$), while the Viterbi algorithm [24] [25] computes $\arg\max_{\mathbf{x}_{0:N}}(p(\mathbf{x}_{0:N}|\mathbf{y}_{0:N}))$;
- In the case of a continuous hidden process $\mathbf{x}$, a number of fixed-interval smoothing algorithms have been proposed (see e.g. [12, ch. 10] [26] for recent surveys), among which the RTS algorithm [6] by Rauch, Tung and Striebel, and the Two-Filter smoother by Mayne [7] and Fraser and Potter [8];
- Though originally derived independently, these discrete and continuous range approaches have been reconciled recently; in particular, the forward-backward algorithm is implemented by the Two-Filter smoother, and the Viterbi recursions are implemented by the RTS smoother [27] [28].

From now on we consider smoothing algorithms for PMC or TMC. Generalizations of the forward-backward algorithm for PMC [2] and TMC [3] with discrete hidden process have already been developped. On the other hand, Kalman-like filtering algorithms have been proposed for PMC [4] and TMC [5] with continuous hidden process, but no smoothing algorithm has been derived yet. From now on we shall thus propose the TMC RTS smoother (see §3.1 and §4.1) and the TMC Two-Filter smoother (see §3.2 and §4.2). In the following, we focus on the computation of $p(\mathbf{x}_n^*|\mathbf{y}_{0:N})$ for some n, $0 \le n \le N$; finally the pdf $p(\mathbf{x}_n|\mathbf{y}_{0:N})$ of interest is obtained by marginalization.

3.1. The Triplet RTS algorithm

Proposition 1 *Let $\mathbf{t}$ be a TMC. Then $p(\mathbf{x}_n^*|\mathbf{y}_{0:N})$ can be computed recursively as*

$$p(\mathbf{x}_n^*|\mathbf{y}_{0:N}) = \int p(\mathbf{x}_{n+1}^*|\mathbf{y}_{0:N}) p(\mathbf{x}_n^*|\mathbf{x}_{n+1}^*, \mathbf{y}_{0:n}) d\mathbf{x}_{n+1}^*, \tag{6}$$

in which $p(\mathbf{x}_n^*|\mathbf{x}_{n+1}^*, \mathbf{y}_{0:n})$ can be computed as

$$p(\mathbf{x}_n^*|\mathbf{x}_{n+1}^*, \mathbf{y}_{0:n}) = \frac{p(\mathbf{x}_n^*|\mathbf{y}_{0:n}) p(\mathbf{x}_{n+1}^*|\mathbf{t}_n, \mathbf{y}_n)}{\int p(\mathbf{x}_n^*|\mathbf{y}_{0:n}) p(\mathbf{x}_{n+1}^*|\mathbf{t}_n, \mathbf{y}_n) d\mathbf{x}_n^*}. \tag{7}$$

Proof 1 *We start from*

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

Since $\mathbf{t}$ is an MC, $p(\mathbf{x}_n^*|\mathbf{x}_{n+1}^*, \mathbf{y}_{0:N}) = \frac{p(\mathbf{x}_n^*, \mathbf{x}_{n+1}^*, \mathbf{y}_{0:N})}{\int p(\mathbf{x}_n^*, \mathbf{x}_{n+1}^*, \mathbf{y}_{0:N}) d\mathbf{x}_n^*}$

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

$= p(\mathbf{x}_n^*|\mathbf{x}_{n+1}^*, \mathbf{y}_{0:n})$, *whence (6). On the other hand,*

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

and

$$p(\mathbf{x}_{n+1}^*|\mathbf{x}_n^*, \mathbf{y}_{0:n}) = \frac{p(\mathbf{x}_{n+1}^*, \mathbf{y}_n|\mathbf{x}_n^*, \mathbf{y}_{0:n-1})}{p(\mathbf{y}_n|\mathbf{x}_n^*, \mathbf{y}_{0:n-1})} = \frac{p(\mathbf{t}_{n+1}|\mathbf{t}_n)}{p(\mathbf{y}_n|\mathbf{t}_n)},$$

whence (7). ■

3.2. The Triplet Forward-Backward algorithm

Proposition 2 *Let $\mathbf{t}$ be a TMC. Let*

$$\alpha(\mathbf{x}_n^*) = p(\mathbf{x}_n^*, \mathbf{y}_{0:n-1}), \tag{8}$$

$$\beta(\mathbf{x}_n^*) = p(\mathbf{y}_{n:N}|\mathbf{x}_n^*, \mathbf{y}_{n-1}). \tag{9}$$

Then $p(\mathbf{x}_n^*|\mathbf{y}_{0:N})$ can be computed as

$$p(\mathbf{x}_n^*|\mathbf{y}_{0:N}) = \frac{\alpha(\mathbf{x}_n^*)\beta(\mathbf{x}_n^*)}{\int \alpha(\mathbf{x}_n^*)\beta(\mathbf{x}_n^*) d\mathbf{x}_n^*}, \tag{10}$$

in which the triplet forward (resp. backward) pdf $\alpha(\mathbf{x}_n^*)$ (resp. $\beta(\mathbf{x}_n^*)$) can be computed recursively via

$$\alpha(\mathbf{x}_{n+1}^*) = \int p(\mathbf{t}_{n+1}|\mathbf{t}_n) \alpha(\mathbf{x}_n^*) d\mathbf{x}_n^*, \tag{11}$$

$$\beta(\mathbf{x}_n^*) = \int p(\mathbf{t}_{n+1}|\mathbf{t}_n) \beta(\mathbf{x}_{n+1}^*) d\mathbf{x}_{n+1}^*. \tag{12}$$

Proof 2 *Equations (8)- (12) hold because of Bayes's rule and because* $\mathbf{t}$ *is an MC. Firstly,*

$$\begin{aligned} p(\mathbf{x}_n^*|\mathbf{y}_{0:N}) &= \frac{p(\mathbf{x}_n^*, \mathbf{y}_{0:n-1}, \mathbf{y}_{n:N})}{p(\mathbf{y}_{0:N})} \\ &\propto \underbrace{p(\mathbf{x}_n^*, \mathbf{y}_{0:n-1})}_{\alpha(\mathbf{x}_n^*)} \underbrace{p(\mathbf{y}_{n:N}|\mathbf{x}_n^*, \mathbf{y}_{n-1})}_{\beta(\mathbf{x}_n^*)}. \end{aligned}$$

Next (11) and (12) are obtained from

$$\alpha(\mathbf{x}_{n+1}^*) = \int \underbrace{p(\mathbf{x}_{n+1}^*, \mathbf{y}_n|\mathbf{x}_n^*, \mathbf{y}_{0:n-1})}_{p(\mathbf{t}_{n+1}|\mathbf{t}_n)} p(\mathbf{x}_n^*, \mathbf{y}_{0:n-1}) d\mathbf{x}_n^*,$$

$$\beta(\mathbf{x}_n^*) = \int p(\mathbf{t}_{n+1}|\mathbf{t}_n) \underbrace{p(\mathbf{y}_{n+1:N}|\mathbf{t}_{n+1}, \mathbf{t}_n)}_{\beta(\mathbf{x}_{n+1}^*)} d\mathbf{x}_{n+1}^*.$$

■

4. THE LINEAR GAUSSIAN CASE

From now on we further assume that the TMC process $\mathbf{t}$ is also Gaussian. More precisely, we assume that

$$\underbrace{\begin{bmatrix} \mathbf{x}_{n+1}^* \\ \mathbf{y}_n \end{bmatrix}}_{\mathbf{t}_{n+1}} = \underbrace{\begin{bmatrix} \mathcal{F}_n^{\mathbf{x}^*,\mathbf{x}^*} & \mathcal{F}_n^{\mathbf{x}^*,\mathbf{y}} \\ \mathcal{F}_n^{\mathbf{y},\mathbf{x}^*} & \mathcal{F}_n^{\mathbf{y},\mathbf{y}} \end{bmatrix}}_{\mathcal{F}_n} \begin{bmatrix} \mathbf{x}_n^* \\ \mathbf{y}_{n-1} \end{bmatrix} + \underbrace{\begin{bmatrix} \mathbf{w}_n^{\mathbf{x}^*} \\ \mathbf{w}_n^{\mathbf{y}} \end{bmatrix}}_{\mathbf{w}_n}, \quad (13)$$

in which $\mathbf{w} = \{\mathbf{w}_n\}_{n\in\mathbb{N}}$ is independent and independent of $\mathbf{t}_0$, $\mathbf{y}_{-1} = \mathbf{0}$, and

$$\mathbf{w}_n \sim \mathcal{N}(\mathbf{0}, \underbrace{\begin{bmatrix} \mathcal{Q}_n^{\mathbf{x}^*,\mathbf{x}^*} & \mathcal{Q}_n^{\mathbf{x}^*,\mathbf{y}} \\ \mathcal{Q}_n^{\mathbf{y},\mathbf{x}^*} & \mathcal{Q}_n^{\mathbf{y},\mathbf{y}} \end{bmatrix}}_{\mathcal{Q}_n}). \quad (14)$$

Then $p(\mathbf{x}_n^*|\mathbf{y}_{0:N})$ is also Gaussian, and computing this pdf amounts to computing its parameters. Let us thus introduce the following notation. For $0 \leq i \leq j \leq N$, let $\mathbf{y}_{i:j} = \{\mathbf{y}_m\}_{i\leq m\leq j}$, and let

$$p(\mathbf{x}_n^*|\mathbf{y}_{i:j}) \sim \mathcal{N}(\widehat{\mathbf{x}}_{n|i:j}^*, \mathbf{P}_{n|i:j}^{\mathbf{x}^*,\mathbf{x}^*}). \quad (15)$$

4.1. The Triplet RTS algorithm

Proposition 3 (The Triplet RTS algorithm) *Let (13) hold. Let*

$$\mathbf{K}_{n|0:N}^* = \mathbf{P}_{n|0:n}^{\mathbf{x}^*,\mathbf{x}^*} [\mathcal{F}_n^{\mathbf{x}^*,\mathbf{x}^*} - \mathcal{Q}_n^{\mathbf{x}^*,\mathbf{y}} (\mathcal{Q}_n^{\mathbf{y},\mathbf{y}})^{-1} \mathcal{F}_n^{\mathbf{y},\mathbf{x}^*}]^T \mathbf{P}_{n+1|0:n}^{\mathbf{x}^*,\mathbf{x}^*-1}. \quad (16)$$

Then (6) reduces to

$$\begin{aligned} \widehat{\mathbf{x}}_{n|0:N}^* &= \widehat{\mathbf{x}}_{n|0:n}^* + \mathbf{K}_{n|0:N}^* [\widehat{\mathbf{x}}_{n+1|0:N}^* - \widehat{\mathbf{x}}_{n+1|0:n}^*], \quad (17) \\ \mathbf{P}_{n|0:N}^{\mathbf{x}^*,\mathbf{x}^*} &= \mathbf{P}_{n|0:n}^{\mathbf{x}^*,\mathbf{x}^*} - \mathbf{K}_{n|0:N}^* \mathbf{P}_{n+1|0:n}^{\mathbf{x}^*,\mathbf{x}^*} \mathbf{K}_{n|0:N}^{*T} \\ &+ \mathbf{K}_{n|0:N}^* \mathbf{P}_{n+1|0:N}^{\mathbf{x}^*,\mathbf{x}^*} \mathbf{K}_{n|0:N}^{*T}. \quad (18) \end{aligned}$$

So $(\widehat{\mathbf{x}}_{n|0:N}^*, \mathbf{P}_{n|0:N}^{\mathbf{x}^*,\mathbf{x}^*})$ *can be computed recursively (in the backward direction) provided* $(\widehat{\mathbf{x}}_{n|0:n}^*, \mathbf{P}_{n|0:n}^{\mathbf{x}^*,\mathbf{x}^*})$ *and* $(\widehat{\mathbf{x}}_{n+1|0:n}^*, \mathbf{P}_{n+1|0:n}^{\mathbf{x}^*,\mathbf{x}^*})$ *are known; these, in turn, can be computed recursively (in the forward direction) by the TMC Kalman filter algorithm, see [4] [5].*

Proof 3 *We first need to compute (7). From (13), we get*

$$p(\mathbf{t}_{n+1}|\mathbf{t}_n) \sim \mathcal{N}(\mathcal{F}_n\mathbf{t}_n, \mathcal{Q}_n). \quad (19)$$

Using Proposition 6 (see section 5 below), we get

$$p(\mathbf{x}_{n+1}^*|\mathbf{x}_n^*, \mathbf{y}_{0:n}) \sim \mathcal{N}(\mathbf{A}_n\mathbf{x}_n^* + \mathbf{b}_n \ , \ \overline{\mathcal{Q}}_n^{\mathbf{x}^*,\mathbf{x}^*}), \quad (20)$$

with

$$\begin{aligned} \mathbf{A}_n &= \mathcal{F}_n^{\mathbf{x}^*,\mathbf{x}^*} - \mathcal{Q}_n^{\mathbf{x}^*,\mathbf{y}} (\mathcal{Q}_n^{\mathbf{y},\mathbf{y}})^{-1} \mathcal{F}_n^{\mathbf{y},\mathbf{x}^*}, \\ \mathbf{b}_n &= \mathcal{Q}_n^{\mathbf{x}^*,\mathbf{y}} (\mathcal{Q}_n^{\mathbf{y},\mathbf{y}})^{-1} \mathbf{y}_n \\ &+ [\mathcal{F}_n^{\mathbf{x}^*,\mathbf{y}} - \mathcal{Q}_n^{\mathbf{x}^*,\mathbf{y}} (\mathcal{Q}_n^{\mathbf{y},\mathbf{y}})^{-1} \mathcal{F}_n^{\mathbf{y},\mathbf{y}}] \ \mathbf{y}_{n-1}, \\ \overline{\mathcal{Q}}_n^{\mathbf{x}^*,\mathbf{x}^*} &= \mathcal{Q}_n^{\mathbf{x}^*,\mathbf{x}^*} - \mathcal{Q}_n^{\mathbf{x}^*,\mathbf{y}} (\mathcal{Q}_n^{\mathbf{y},\mathbf{y}})^{-1} \mathcal{Q}_n^{\mathbf{y},\mathbf{x}^*}. \end{aligned}$$

On the other hand, from (15)

$$p(\mathbf{x}_n^*|\mathbf{y}_{0:n}) \sim \mathcal{N}(\widehat{\mathbf{x}}_{n|0:n}^*, \mathbf{P}_{n|0:n}^{\mathbf{x}^*,\mathbf{x}^*}).$$

Using (20) and Proposition 7, we get

$$p(\mathbf{x}_n^*, \mathbf{x}_{n+1}^*|\mathbf{y}_{0:n}) \sim \mathcal{N}(\begin{bmatrix} \widehat{\mathbf{x}}_{n|0:n}^* \\ \widehat{\mathbf{x}}_{n+1|0:n}^* \end{bmatrix}, \begin{bmatrix} \mathbf{P}_{n|0:n}^{\mathbf{x}^*,\mathbf{x}^*} & \mathbf{P}_{n|0:n}^{\mathbf{x}^*,\mathbf{x}^*}\mathbf{A}_n^T \\ \mathbf{A}_n\mathbf{P}_{n|0:n}^{\mathbf{x}^*,\mathbf{x}^*} & \mathbf{P}_{n+1|0:n}^{\mathbf{x}^*,\mathbf{x}^*} \end{bmatrix})$$

with $\widehat{\mathbf{x}}_{n+1|0:n}^* = \mathbf{A}_n\widehat{\mathbf{x}}_{n|0:n}^* + \mathbf{b}_n$, $\mathbf{P}_{n+1|0:n}^{\mathbf{x}^*,\mathbf{x}^*} = \overline{\mathcal{Q}}_n^{\mathbf{x}^*,\mathbf{x}^*} + \mathbf{A}_n\mathbf{P}_{n|0:n}^{\mathbf{x}^*,\mathbf{x}^*}\mathbf{A}_n^T$. *Using Proposition 6, we get*

$$\begin{aligned} p(\mathbf{x}_n^*|\mathbf{x}_{n+1}^*, \mathbf{y}_{0:n}) \sim \mathcal{N}(&\widehat{\mathbf{x}}_{n|0:n}^* + \mathbf{K}_{n|0:N}^*(\mathbf{x}_{n+1}^* - \widehat{\mathbf{x}}_{n+1|0:n}^*), \\ &\mathbf{P}_{n|0:n}^{\mathbf{x}^*,\mathbf{x}^*} - \mathbf{K}_{n|0:N}^*\mathbf{P}_{n+1|0:n}^{\mathbf{x}^*,\mathbf{x}^*}\mathbf{K}_{n|0:N}^{*T}) \quad (21) \end{aligned}$$

in which $\mathbf{K}_{n|0:N}^*$ *is given by (16).*

It remains to compute equation (6). Using (15), (21) and Proposition 7, we finally get (17) and (18). ■

4.2. The Triplet Two-Filter algorithm

Proposition 4 (The Triplet Two-Filter algorithm) *Let (13) hold. Then (8) - (10) reduce to*

$$\begin{aligned} \mathbf{P}_{n|0:N}^{\mathbf{x}^*,\mathbf{x}^*-1}\widehat{\mathbf{x}}_{n|0:N}^* &= \mathbf{P}_{n|0:n-1}^{\mathbf{x}^*,\mathbf{x}^*-1}\widehat{\mathbf{x}}_{n|0:n-1}^* + \mathbf{P}_{n|n-1:N}^{\mathbf{x}^*,\mathbf{x}^*-1}\widehat{\mathbf{x}}_{n|n-1:N}^* \\ &- \mathbf{P}_{n|n-1}^{\mathbf{x}^*,\mathbf{x}^*-1}\widehat{\mathbf{x}}_{n|n-1}^*, \quad (22) \\ \mathbf{P}_{n|0:N}^{\mathbf{x}^*,\mathbf{x}^*-1} &= \mathbf{P}_{n|0:n-1}^{\mathbf{x}^*,\mathbf{x}^*-1} + \mathbf{P}_{n|n-1:N}^{\mathbf{x}^*,\mathbf{x}^*-1} - \mathbf{P}_{n|n-1}^{\mathbf{x}^*,\mathbf{x}^*-1}. \quad (23) \end{aligned}$$

Proof 4 *The proof is omitted for want of space.* ■

As we can see, the TMC Two-Filter algorithm can be used for computing $p(\mathbf{x}^*_n|\mathbf{y}_{0:N})$ once we know how to compute $(\widehat{\mathbf{x}}^*_{n|0:n-1}, \mathbf{P}^{\mathbf{x}^*,\mathbf{x}^*}_{n|0:n-1})$, $(\widehat{\mathbf{x}}^*_{n|n-1:N}, \mathbf{P}^{\mathbf{x}^*,\mathbf{x}^*}_{n|n-1:N})$ and $(\widehat{\mathbf{x}}^*_{n|n-1}, \mathbf{P}^{\mathbf{x}^*,\mathbf{x}^*}_{n|n-1})$. Let us now consider these points.

Firstly, as in the TMC RTS algorithm, $(\widehat{\mathbf{x}}^*_{n|0:n-1}, \mathbf{P}^{\mathbf{x}^*,\mathbf{x}^*}_{n|0:n-1})$ can be computed recursively by the TMC Kalman filter [4] [5].

Let us now address the computation of $(\widehat{\mathbf{x}}^*_{n|n-1}, \mathbf{P}^{\mathbf{x}^*,\mathbf{x}^*}_{n|n-1})$. Let

$$p(\mathbf{t}_n) \sim \mathcal{N}(\underbrace{\begin{bmatrix} \widehat{\mathbf{x}}^*_n \\ \widehat{\mathbf{y}}_{n-1} \end{bmatrix}}_{\widehat{\mathbf{t}}_n}, \underbrace{\begin{bmatrix} \mathbf{P}^{\mathbf{x}^*,\mathbf{x}^*}_n & \mathbf{P}^{\mathbf{x}^*,\mathbf{y}}_n \\ \mathbf{P}^{\mathbf{y},\mathbf{x}^*}_n & \mathbf{P}^{\mathbf{y},\mathbf{y}}_{n-1} \end{bmatrix}}_{\mathbf{P}^{\mathbf{t}}_n}). \quad (24)$$

From (13), we can compute $\widehat{\mathbf{t}}_n$ and $\mathbf{P}^{\mathbf{t}}_n$ recursively by

$$\widehat{\mathbf{t}}_{n+1} = \mathcal{F}_n \widehat{\mathbf{t}}_n, \quad (25)$$
$$\mathbf{P}^{\mathbf{t}}_{n+1} = \mathcal{F}_n \mathbf{P}^{\mathbf{t}}_n \mathcal{F}_n + \mathcal{Q}_n. \quad (26)$$

Using Proposition 6, we get

$$\widehat{\mathbf{x}}^*_{n|n-1} = \widehat{\mathbf{x}}^*_n + \mathbf{P}^{\mathbf{x}^*,\mathbf{y}}_n (\mathbf{P}^{\mathbf{y},\mathbf{y}}_{n-1})^{-1} (\mathbf{y}_{n-1} - \widehat{\mathbf{y}}_{n-1}), \quad (27)$$
$$\mathbf{P}^{\mathbf{x}^*,\mathbf{x}^*}_{n|n-1} = \mathbf{P}^{\mathbf{x}^*,\mathbf{x}^*}_n - \mathbf{P}^{\mathbf{x}^*,\mathbf{y}}_n (\mathbf{P}^{\mathbf{y},\mathbf{y}}_{n-1})^{-1} \mathbf{P}^{\mathbf{y},\mathbf{x}^*}_n. \quad (28)$$

Let us finally address the computation of $(\widehat{\mathbf{x}}^*_{n|n-1:N}, \mathbf{P}^{\mathbf{x}^*,\mathbf{x}^*}_{n|n-1:N})$. These parameters can be computed recursively by the TMC backward Kalman filter :

Proposition 5 (The TMC Backward Kalman filter) *Let*

$$\widetilde{\mathcal{F}}_{n+1} = \mathbf{P}^{\mathbf{t}}_n \mathcal{F}^T_n (\mathbf{P}^{\mathbf{t}}_{n+1})^{-1} = \begin{bmatrix} \widetilde{\mathcal{F}}^{\mathbf{x}^*,\mathbf{t}}_{n+1} \\ \widetilde{\mathcal{F}}^{\mathbf{y},\mathbf{t}}_{n+1} \end{bmatrix} = \begin{bmatrix} \widetilde{\mathcal{F}}^{\mathbf{x}^*,\mathbf{x}^*}_{n+1} & \widetilde{\mathcal{F}}^{\mathbf{x}^*,\mathbf{y}}_{n+1} \\ \widetilde{\mathcal{F}}^{\mathbf{y},\mathbf{x}^*}_{n+1} & \widetilde{\mathcal{F}}^{\mathbf{y},\mathbf{y}}_{n+1} \end{bmatrix} \quad (29)$$
$$\widetilde{\mathcal{Q}}_{n+1} = \mathbf{P}^{\mathbf{t}}_n - \widetilde{\mathcal{F}}_{n+1} \mathbf{P}^{\mathbf{t}}_{n+1} \widetilde{\mathcal{F}}^T_{n+1} = \begin{bmatrix} \widetilde{\mathcal{Q}}^{\mathbf{x}^*,\mathbf{x}^*}_{n+1} & \widetilde{\mathcal{Q}}^{\mathbf{x}^*,\mathbf{y}}_{n+1} \\ \widetilde{\mathcal{Q}}^{\mathbf{y},\mathbf{x}^*}_{n+1} & \widetilde{\mathcal{Q}}^{\mathbf{y},\mathbf{y}}_{n+1} \end{bmatrix}. \quad (30)$$

Then we have

$$\begin{aligned}
\widehat{\mathbf{x}}^*_{n|n:N} &= \widetilde{\mathcal{F}}^{\mathbf{x}^*,\mathbf{x}^*}_{n+1} [\widehat{\mathbf{x}}^*_{n+1|n:N} - \widehat{\mathbf{x}}^*_{n+1}] + \widetilde{\mathcal{F}}^{\mathbf{x}^*,\mathbf{y}}_{n+1} [\mathbf{y}_n - \widehat{\mathbf{y}}_n] \\
&\quad + \widehat{\mathbf{x}}^*_n, && (31) \\
\widehat{\mathbf{y}}_{n-1|n:N} &= \widetilde{\mathcal{F}}^{\mathbf{y},\mathbf{x}^*}_{n+1} [\widehat{\mathbf{x}}^*_{n+1|n:N} - \widehat{\mathbf{x}}^*_{n+1}] + \widetilde{\mathcal{F}}^{\mathbf{y},\mathbf{y}}_{n+1} [\mathbf{y}_n - \widehat{\mathbf{y}}_n] \\
&\quad + \widehat{\mathbf{y}}_{n-1}, && (32) \\
\mathbf{P}^{\mathbf{x}^*,\mathbf{x}^*}_{n|n:N} &= \widetilde{\mathcal{Q}}^{\mathbf{x}^*,\mathbf{x}^*}_{n+1} + \widetilde{\mathcal{F}}^{\mathbf{x}^*,\mathbf{x}^*}_{n+1} \mathbf{P}^{\mathbf{x}^*,\mathbf{x}^*}_{n+1|n:N} (\widetilde{\mathcal{F}}^{\mathbf{x}^*,\mathbf{x}^*}_{n+1})^T, && (33) \\
\mathbf{K}^*_{n|n-1:N} &= [\widetilde{\mathcal{Q}}^{\mathbf{x}^*,\mathbf{y}}_{n+1} + \widetilde{\mathcal{F}}^{\mathbf{x}^*,\mathbf{x}^*}_{n+1} \mathbf{P}^{\mathbf{x}^*,\mathbf{x}^*}_{n+1|n:N} (\widetilde{\mathcal{F}}^{\mathbf{y},\mathbf{x}^*}_{n+1})^T] \\
&\quad \times [\widetilde{\mathcal{Q}}^{\mathbf{y},\mathbf{y}}_{n+1} + \widetilde{\mathcal{F}}^{\mathbf{y},\mathbf{x}^*}_{n+1} \mathbf{P}^{\mathbf{x}^*,\mathbf{x}^*}_{n+1|n:N} (\widetilde{\mathcal{F}}^{\mathbf{y},\mathbf{x}^*}_{n+1})^T]^{-1}, && (34) \\
\widehat{\mathbf{x}}^*_{n|n-1:N} &= \widehat{\mathbf{x}}^*_{n|n:N} + \mathbf{K}^*_{n|n-1:N} (\mathbf{y}_{n-1} - \widehat{\mathbf{y}}_{n-1|n:N}), && (35) \\
\mathbf{P}^{\mathbf{x}^*,\mathbf{x}^*}_{n|n-1:N} &= \mathbf{P}^{\mathbf{x}^*,\mathbf{x}^*}_{n|n:N} - \mathbf{K}^*_{n|n-1:N} [\widetilde{\mathcal{Q}}^{\mathbf{y},\mathbf{y}}_{n+1} \\
&\quad + \widetilde{\mathcal{F}}^{\mathbf{y},\mathbf{x}^*}_{n+1} \mathbf{P}^{\mathbf{x}^*,\mathbf{x}^*}_{n+1|n:N} (\widetilde{\mathcal{F}}^{\mathbf{y},\mathbf{x}^*}_{n+1})^T] \mathbf{K}^{*^T}_{n|n-1:N}. && (36)
\end{aligned}$$

The algorithm is initialized at $n = N+1$ by

$$\widehat{\mathbf{x}}^*_{N+1|N} = \widehat{\mathbf{x}}^*_{N+1} + \mathbf{P}^{\mathbf{x}^*,\mathbf{y}}_{N+1} (\mathbf{P}^{\mathbf{y},\mathbf{y}}_N)^{-1} [\mathbf{y}_N - \widehat{\mathbf{y}}_N], \quad (37)$$
$$\mathbf{P}^{\mathbf{x}^*,\mathbf{x}^*}_{N+1|N} = \mathbf{P}^{\mathbf{x}^*,\mathbf{x}^*}_{N+1} - \mathbf{P}^{\mathbf{x}^*,\mathbf{y}}_{N+1} (\mathbf{P}^{\mathbf{y},\mathbf{y}}_N)^{-1} \mathbf{P}^{\mathbf{y},\mathbf{x}^*}_{N+1}. \quad (38)$$

Proof 5 *Using (24), (19) and Propositions 7 and 6, we get*

$$p(\mathbf{t}_n|\mathbf{t}_{n+1}) \sim \mathcal{N}(\widetilde{\mathcal{F}}_{n+1}\mathbf{t}_{n+1} + (\mathbf{I} - \widetilde{\mathcal{F}}_{n+1}\mathcal{F}_n)\widehat{\mathbf{t}}_n, \widetilde{\mathcal{Q}}_{n+1}), \quad (39)$$

in which $\widetilde{\mathcal{F}}_{n+1}$ and $\widetilde{\mathcal{Q}}_{n+1}$ are given respectively by (29) and (30).

Next, from (15) we have

$$p(\mathbf{x}^*_{n+1}|\mathbf{y}_{n:N}) \sim \mathcal{N}(\widehat{\mathbf{x}}^*_{n+1|n:N}, \mathbf{P}^{\mathbf{x}^*,\mathbf{x}^*}_{n+1|n:N}). \quad (40)$$

Using (39), (40) and Proposition 7, and marginalizing w.r.t. $\mathbf{x}^*_{n+1}$, we get

$$p(\mathbf{t}_n|\mathbf{y}_{n:N}) \sim \mathcal{N}(\begin{bmatrix} \widehat{\mathbf{x}}^*_{n|n:N} \\ \widehat{\mathbf{y}}_{n-1|n:N} \end{bmatrix}, \mathbf{P}^{\mathbf{t}}_{n|n:N}),$$

from which we deduce (31), (32) and (33). Using Proposition 6, we get (34), (35) and (36). ■

5. SOME PROPERTIES OF GAUSSIAN R.V.

The derivations in this paper rely on the following results, which are recalled for convenience of the reader :

Proposition 6 *Let*

$$(\mathbf{u}_1, \mathbf{u}_2) \sim \mathcal{N}(\begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \begin{bmatrix} \mathbf{\Sigma}_{1,1} & \mathbf{\Sigma}_{1,2} \\ \mathbf{\Sigma}_{2,1} & \mathbf{\Sigma}_{2,2} \end{bmatrix}).$$

Then conditionally on $\mathbf{u}_2$, $\mathbf{u}_1 \sim \mathcal{N}(\mu_1 + \mathbf{\Sigma}_{1,2}\mathbf{\Sigma}^{-1}_{2,2}(\mathbf{u}_2 - \mu_2), \mathbf{\Sigma}_{1,1} - \mathbf{\Sigma}_{1,2}\mathbf{\Sigma}^{-1}_{2,2}\mathbf{\Sigma}_{2,1})$.

Proposition 7 *Let $\mathbf{u}_1 \sim \mathcal{N}(\mu_1, \mathbf{\Sigma}_1)$ and conditionally on $\mathbf{u}_1$, let $\mathbf{u}_2 \sim \mathcal{N}(\mathbf{A}\mathbf{u}_1 + \mathbf{b}, \mathbf{\Sigma}_{2|1})$. Then*

$$(\mathbf{u}_1, \mathbf{u}_2) \sim \mathcal{N}(\begin{bmatrix} \mu_1 \\ \mathbf{A}\mu_1 + \mathbf{b} \end{bmatrix}, \begin{bmatrix} \mathbf{\Sigma}_1 & \mathbf{\Sigma}_1\mathbf{A}^T \\ \mathbf{A}\mathbf{\Sigma}_1 & \mathbf{\Sigma}_{2|1} + \mathbf{A}\mathbf{\Sigma}_1\mathbf{A}^T \end{bmatrix}).$$

6. REFERENCES

[1] Y. Ephraim and N. Merhav, "Hidden Markov processes," *IEEE Transactions on Information Theory*, vol. 48, no. 6, pp. 1518–69, June 2002.

[2] W. Pieczynski, "Pairwise Markov chains," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 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 International Symposium on Remote Sensing*, Crete, Grece, September 22-27 2002.

[4] W. Pieczynski and F. Desbouvries, "Kalman filtering using pairwise Gaussian models," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP 03)*, Hong-Kong, 2003.

[5] B. Ait-El-Fquih and F. Desbouvries, "Kalman filtering for triplet Markov chains : Applications and extensions," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP 05)*, Philadelphia, USA, March 19-23 2005.

[6] 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.

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

[8] D. C. Fraser and J. E. Potter, "The optimum linear smoother as a combination of two optimum linear filters," *IEEE Transactions on Automatic Control*, vol. 7, pp. 387–90, August 1969.

[9] R. E. Kalman, "A new approach to linear filtering and prediction problems," *J. Basic Eng., Trans. ASME, Series D*, vol. 82, no. 1, pp. 35–45, 1960.

[10] Y. C. Ho and R. C. K. Lee, "A Bayesian approach to problems in stochastic estimation and control," *IEEE Transactions on Automatic Control*, vol. 9, pp. 333–339, October 1964.

[11] B. D. O. Anderson and J. B. Moore, *Optimal Filtering*, Prentice Hall, Englewood Cliffs, New Jersey, 1979.

[12] T. Kailath, A. H. Sayed, and B. Hassibi, *Linear estimation*, Prentice Hall Information and System Sciences Series. Prentice Hall, Upper Saddle River, New Jersey, 2000.

[13] 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.

[14] R. S. Lipster and A. N. Shiryaev, *Statistics of Random Processes, Vol. 2 : Applications*, chapter 13 : "Conditionally Gaussian Sequences : Filtering and Related Problems", Springer Verlag, Berlin, 2001.

[15] S. Derrode and W. Pieczynski, "Signal and image segmentation using pairwise Markov chains," *IEEE Transactions on Signal Processing*, vol. 52, no. 9, pp. 2477–89, 2004.

[16] P. Lanchantin and W. Pieczynski, "Unsupervised non stationary image segmentation using triplet Markov chains," in *Advanced Concepts for Intelligent Vision Systems (ACVIS 04)*, Brussels, Belgium, August 31 - September 3 2004.

[17] W. Pieczynski and F. Desbouvries, "On Triplet Markov chains," in *Proceedings of the International Symposium on Applied Stochastic Models and Data Analysis (ASMDA 2005)*, Brest, France, May 17-20 2005.

[18] M. Mariton, *Jump linear systems in automatic control*, Marcel Dekker, New York, 1990.

[19] A. Doucet, N. J. Gordon, and V. Krishnamurthy, "Particle filters for state estimation of jump Markov linear systems," *IEEE Transactions on Signal Processing*, vol. 49, no. 3, pp. 613–24, March 2001.

[20] H. W. Sorenson, "Kalman filtering techniques," in *Advances in Control Systems Theory and Appl.*, C. T. Leondes, Ed., vol. 3, pp. 219–92. Acad. Press, 1966.

[21] C.K. Chui and G. Chen, *Kalman Filtering with Real-Time Applications*, Berlin, DE: Springer, 1999.

[22] L. E. Baum and T. Petrie, "Statistical inference for probabilistic functions of finite state Markov chains," *Ann. Math. Stat.*, vol. 37, pp. 1554–63, 1966.

[23] L. E. Baum and J. A. Eagon, "An inequality with applications to statistical estimation for probabilistic functions of a Markov process and to a model for ecology," *Bull. Amer. Meteorol. Soc.*, vol. 73, pp. 360–63, 1967.

[24] G. D. Forney, "The Viterbi algorithm," *Proceedings of the IEEE*, vol. 61, pp. 268–78, March 1973.

[25] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimal decoding algorithm," *IEEE Transactions on Information Theory*, vol. 13, pp. 260–69, April 1967.

[26] H. L. Weinert, *Fixed interval smoothing for state-space models*, The Kluwer international series in engineering and computer science. Kluwer Academic Publishers, Boston, 2001.

[27] B. Delyon, "Remarks on linear and nonlinear filtering," *IEEE Transactions on Information Theory*, vol. 41, no. 1, pp. 317–22, January 1995.

[28] P. L. Ainsleigh, N. Kehtarnavaz, and R. L. Streit, "Hidden Gauss-Markov models for signal classification," *IEEE Transactions on Signal Processing*, vol. 50, no. 6, pp. 1355–1367, June 2002.