Iterative Algorithms for State Estimation of Jump Markov Linear Systems

Arnaud Doucet and Christophe Andrieu

***Abstract*—Jump Markov linear systems (JMLSs) are linear systems whose parameters evolve with time according to a finite state Markov chain. Given a set of observations, our aim is to estimate the states of the finite state Markov chain and the continuous (in space) states of the linear system.**

In this paper, we present original deterministic and stochastic iterative algorithms for optimal state estimation of JMLSs. The first stochastic algorithm yields minimum mean square error (MMSE) estimates of the finite state space Markov chain and of the continuous state of the JMLS. A deterministic and a stochastic algorithm are given to obtain the marginal maximum *a posteriori* (MMAP) sequence estimate of the finite state Markov chain. Finally, a deterministic and a stochastic algorithm are derived to obtain the MMAP sequence estimate of the continuous state of the JMLS.

Computer simulations are carried out to evaluate the performance of the proposed algorithms. The problem of deconvolution of Bernoulli-Gaussian (BG) processes and the problem of tracking a maneuvering target are addressed.

***Index Terms*—Jump Markov linear systems, MCMC, simulated annealing, switching state-space models.**

I. Introduction

JUMP Markov linear systems are linear systems whose parameters evolve with time according to a finite state Markov chain. These models are widely used in several fields of signal processing and include as particular cases common models in impulsive deconvolution [7], [10], [16], [19], [21], [26], [30], digital communications [20], [23], and target tracking [2], [27]. Given a set of observations, our aim in this paper is to estimate the states of the finite state Markov chain and the continuous (in space) states of the linear system. More precisely, our aim is to estimate, respectively, the MMSE estimates of the states and the MMAP sequence estimates of the finite state Markov chain and of the continuous state of the JMLS.

Under assumptions detailed later on, it is well known that exact computation of these three estimates for JMLS involves a prohibitive computational cost exponential in the number, say T, of observations [32]. Thus, it is necessary to consider in practice suboptimal estimation algorithms. A variety of such suboptimal algorithms has already been proposed in the literature to solve these estimation problems [2], [11], [15], [23], [31], [32].

In this paper, we present original iterative stochastic and deterministic algorithms to perform MMSE and MMAP estimation of JMLS. The stochastic algorithms developed to estimate the MMSE and MMAP estimates are based, respectively, on homogeneous and nonhomogeneous Markov chain Monte Carlo (MCMC) methods [29]. The developed nonhomogeneous MCMC methods correspond to simulated annealing (SA) algorithms. The deterministic algorithms developed to estimate the MMAP estimates are based on coordinate ascent optimization methods. These latter algorithms are easily deduced from their stochastic counterparts and *vice versa*. The key point of our algorithms is that they have a computational cost $O(T)$ per iteration. They have also several other advantages compared with current methods.

- They neither require the state covariance matrix to be strictly positive nor the transition matrix of the state-space model to be regular, contrary to [6], [15], and [18].
- Contrary to [11], [21], [23], and [30], they avoid the introduction of any missing data set. Consequently, they can deal, for example, with a dynamic noise modeled as a BG process when inference on the finite Markov chain is required. This is of major interest in impulsive deconvolution [26].
- They allow us to consider autoregressive moving-average (ARMA) models, whereas [4], [7], and [16] are restricted to MA models.

Moreover, it can be theoretically established that they are more efficient in various aspects. We discuss in detail these issues in Section VI.

We now list the main results and the organization of this paper. Section II formally presents the signal model and estimation objectives. In Section III, an MCMC algorithm based on the Gibbs sampler is proposed to compute the MMSE estimates. In Section IV, a deterministic coordinate ascent optimization method and a SA version of the Gibbs sampler developed in Section III are presented to obtain the MMAP estimate of the finite state Markov chain of the JMLS. In Section V, a deterministic coordinate ascent optimization method and a SA algorithm are presented to obtain the MMAP estimate of continuous states of the JMLS. In Section VI, a discussion of the previous work on related problems and on the algorithms developed here is given. The computational and theoretical advantages of our algorithms over previous algorithms are explained. In Section VII, we demonstrate the performance of the proposed algorithms for deconvolution of BG processes on simulated and real data. We also address the problem of tracking of a maneuvering

Manuscript received January 25, 1999; revised February 20, 2001. The associate editor coordinating the review of this paper and approving it for publication was Dr. Joseph M. Francos.

A. Doucet was with the Signal Processing Group, Department of Engineering, University of Cambridge, Cambridge, U.K. He is now with the Department of Electrical and Electronic Engineering, University of Melbourne, Parkville, Australia (e-mail: doucet@ee.mu.oz.au).

C. Andrieu was with the Signal Processing Group, Department of Engineering, University of Cambridge, Cambridge, U.K. He is now with the Department of Mathematics, Statistics Group, University of Bristol, Bristol U.K. (e-mail: C.Andrieu@bristol.ac.uk).

Publisher Item Identifier S 1053-587X(01)03891-0.

target. Appendix B contains all the notation used in the paper. Finally, the proofs of lemmas and propositions are grouped in Appendix C.

II. Problem Formulation

A. Signal Model

Let r_t denote a discrete-time, time-homogeneous, s-state, first-order Markov chain with transition probabilities $p_{ij} \triangleq \Pr\{r_{t+1} = j \,|\, r_t = i\}$ for any $i, j \in S$, where $S \triangleq \{1, 2, \ldots, s\}$. The transition probability matrix $[p_{ij}]$ is, thus, an $s \times s$ matrix, with elements satisfying $p_{ij} \geq 0$ and $\sum_{j=1}^{s} p_{ij} = 1$, for each $i \in S$. Denote the initial probability distribution as $p_i \triangleq \Pr\{r_1 = i\}$ for $i \in S$ such that $p_i \geq 0, \forall i \in S$ and $\sum_{i=1}^{s} p_i = 1$. Consider the following JMLS:

$$x_{t+1} = A(r_{t+1})x_t + B(r_{t+1})v_{t+1} + F(r_{t+1})u_{t+1} \quad (1)$$

$$y_t = C(r_t)x_t + D(r_t)w_t + G(r_t)u_t \quad (2)$$

where

$x_t \in \mathbb{R}^{n_x}$	system state;
$y_t \in \mathbb{R}^{n_y}$	observation at time;
$t, u_t \in \mathbb{R}^{n_u}$	known deterministic input;
$v_t \in \mathbb{R}^{n_v}$	zero-mean white Gaussian noise sequence with covariance;
$I_{n_v}, w_t \in \mathbb{R}^{n_w}$	zero-mean white Gaussian noise sequence with covariance I_{n_w} and $D(i)D^{\mathrm{T}}(i) > 0$ $(\forall i \in S)$.

The matrices $A(\cdot), B(\cdot), C(\cdot), D(\cdot), F(\cdot)$ and $G(\cdot)$ are functions of the Markov chain state r_t, i.e., $(A(\cdot), B(\cdot), C(\cdot), D(\cdot), F(\cdot), G(\cdot)) \in \{(A(i), B(i), C(i), D(i), F(i), G(i)); i \in S\}$. They evolve according to the realization of the finite state Markov chain r_t. We assume that $x_0 \sim \mathcal{N}(\hat{x}_0, P_0)$ and let x_0, v_t and w_t be mutually independent for all t. The model parameters λ are assumed known, where $\lambda \triangleq \{p_i, p_{ij}, A(i), B(i), C(i), D(i), F(i), G(i), \hat{x}_0, P_0; i, j \in S\}$. Finally, let R denote the set of paths of the finite Markov chain r_t of non-null prior probability.

B. Estimation Objectives

Given $\mathbf{y}_{1:T}$, assuming that the model parameters λ are exactly known, all Bayesian inference for JMLS relies on the joint posterior distribution $p(\mathbf{r}_{1:T}, \mathbf{x}_{0:T} \,|\, \mathbf{y}_{1:T})$. In this paper, we consider the three following optimal estimation problems:

1) *MMSE estimates of* $\mathbf{x}_{0:T}$ *and* $\mathbf{r}_{1:T}$: Compute optimal (in a mean square sense) estimates of $\mathbf{x}_{0:T}$ and $\mathbf{r}_{1:T}$ given by $\mathbb{E}\{\mathbf{r}_{1:T} \,|\, \mathbf{y}_{1:T}\}$ and $\mathbb{E}\{\mathbf{x}_{0:T} \,|\, \mathbf{y}_{1:T}\}$.
2) *MMAP sequence estimate of* $\mathbf{r}_{1:T}$: Compute optimal (in a MAP sense) state estimates of $\mathbf{r}_{1:T}$ by maximizing $p(\mathbf{r}_{1:T} \,|\, \mathbf{y}_{1:T})$, i.e., $\hat{\mathbf{r}}_{1:T} \triangleq \arg\max_{\mathbf{r}_{1:T}} p(\mathbf{r}_{1:T} \,|\, \mathbf{y}_{1:T})$.
3) *MMAP sequence estimate of* $\mathbf{x}_{0:T}$: Compute optimal (in a MAP sense) state estimates of $\mathbf{x}_{0:T}$ by maximizing $p(\mathbf{x}_{0:T} \,|\, \mathbf{y}_{1:T})$, i.e., $\hat{\mathbf{x}}_{0:T} \triangleq \arg\max_{\mathbf{x}_{0:T}} p(\mathbf{x}_{0:T} \,|\, \mathbf{y}_{1:T})$.

III. Minimum Mean Square Error Estimation

The MMSE estimates $\mathbb{E}\{\mathbf{r}_{1:T} \,|\, \mathbf{y}_{1:T}\}$ and $\mathbb{E}\{\mathbf{x}_{0:T} \,|\, \mathbf{y}_{1:T}\}$ are obtained by integration with respect to the joint posterior distribution. If we were able to obtain N (for large N) independent and identically distributed (i.i.d.) samples $\{\mathbf{r}_{1:T}^{(k)}; k = 1, \ldots, N\}$ distributed according to the distribution $p(\mathbf{r}_{1:T} \,|\, \mathbf{y}_{1:T})$, then, using the law of large numbers, MMSE estimates could be computed by averaging, thus solving the state estimation problem. Unfortunately, obtaining such i.i.d. samples from the posterior distribution $p(\mathbf{r}_{1:T} \,|\, \mathbf{y}_{1:T})$ is not straightforward. Thus, alternative sampling schemes must be investigated.

A. Markov Chain Monte Carlo Method

We compute samples from the posterior distribution $p(\mathbf{r}_{1:T} \,|\, \mathbf{y}_{1:T})$ using an MCMC method [29]. The key idea of MCMC methods is to run an ergodic Markov chain whose invariant distribution is the distribution of interest. The obtained samples are then used to compute MMSE estimates of the states $\mathbf{r}_{1:T}$ and $\mathbf{x}_{0:T}$. The proposed algorithm proceeds as follows.

MCMC algorithm to obtain the MMSE estimates

1) Initialization. Set randomly $\mathbf{r}_{1:T}^{(0)} \in R$.
2) Iteration $k, k \geq 1$
   - For $t = 1, \ldots, T$, sample $r_t^{(k)} \sim p(r_t \,|\, \mathbf{y}_{1:T}, \mathbf{r}_{-t}^{(k)})$.
   - Optional Step. Compute $\mathbf{x}_{0:T}^{(k)} = \mathbb{E}[\mathbf{x}_{0:T} \,|\, \mathbf{y}_{1:T}, \mathbf{r}_{1:T}^{(k)}]$,

where $\mathbf{r}_{-t}^{(k)} \triangleq (r_1^{(k)}, \ldots, r_{t-1}^{(k)}, r_{t+1}^{(k-1)}, \ldots, r_T^{(k-1)})$.

Once the algorithm has been iterated N times, the MMSE estimates of $\mathbf{r}_{1:T}$ and $\mathbf{x}_{0:T}$ are computed using

$$\bar{\mathbf{r}}_{1:T}(N) \triangleq \frac{1}{N} \sum_{k=0}^{N-1} \mathbf{r}_{1:T}^{(k)}, \quad \bar{\mathbf{x}}_{0:T}(N) \triangleq \frac{1}{N} \sum_{k=0}^{N-1} \mathbf{x}_{0:T}^{(k)}. \quad (3)$$

The different steps of this algorithm are detailed in the following subsections. In order to simplify notation, we drop the superscript $\cdot^{(k)}$ from all variables at iteration k when it is unnecessary.

B. Implementation and Convergence Issues

1) Implementation Issues: This algorithm requires sampling from $p(r_t \,|\, \mathbf{y}_{1:T}, \mathbf{r}_{-t})$ and computing $\mathbb{E}\{\mathbf{x}_{0:T} \,|\, \mathbf{y}_{1:T}, \mathbf{r}_{1:T}\}$. Given the sequence $\mathbf{r}_{1:T}$, the system (1) and (2) is linear Gaussian; therefore, estimating $\mathbf{x}_{0:T} = \mathbb{E}[\mathbf{x}_{0:T} \,|\, \mathbf{y}_{1:T}, \mathbf{r}_{1:T}]$ can be straightforwardly done using a Kalman smoother [1] in $O(T)$ operations. To sample from $p(r_t \,|\, \mathbf{y}_{1:T}, \mathbf{r}_{-t})$, a direct solution would consist of evaluating for $i \in S$ the distribution $p(r_t = i \,|\, \mathbf{y}_{1:T}, \mathbf{r}_{-t}) \propto p(\mathbf{y}_{1:T} \,|\, r_t = i, \mathbf{r}_{-t}) p(r_t = i \,|\, \mathbf{r}_{-t})$ using s times a Kalman filter to compute the s likelihood terms $p(\mathbf{y}_{1:T} \,|\, r_t = i, \mathbf{r}_{-t})$ for $i = 1, \ldots, s$. As we need to sample from $p(r_t \,|\, \mathbf{y}_{1:T}, \mathbf{r}_{-t})$ for $t = 1, \ldots, T$, this would result in an algorithm of computational complexity $O(T^2)$. We develop here an algorithm of complexity $O(T)$ that relies on the following key decomposition of the

likelihood $p(\mathbf{y}_{1:T} \,|\, \mathbf{r}_{1:T})$ that allows for the efficient computation of $p(r_t \,|\, \mathbf{y}_{1:T}, \mathbf{r}_{-t})$ for $t = 1, \ldots, T$. Indeed, for any $t = 2, \ldots, T-1$ (the modifications needed to handle the case of boundaries are straightforward and omitted here), we have

$$p(\mathbf{y}_{1:T} | \, \mathbf{r}_{1:T}) = p(\mathbf{y}_{1:t-1} \,|\, \mathbf{r}_{1:t-1}) p(y_t \,|\, \mathbf{y}_{1:t-1}, \mathbf{r}_{1:t}) \times \int p(\mathbf{y}_{t+1:T} \,|\, \mathbf{r}_{t+1:T}, x_t) p(x_t \,|\, \mathbf{y}_{1:t}, \mathbf{r}_{1:t}) \, dx_t \quad (4)$$

where

$$p(\mathbf{y}_{t:T} \,|\, \mathbf{r}_{t:T}, x_{t-1}) = \int p(\mathbf{y}_{t+1:T} \,|\, \mathbf{r}_{t+1:T}, x_t) p(y_t, x_t \,|\, r_t, x_{t-1}) \, dx_t. \quad (5)$$

The two first terms on the right-hand side of (4) can be computed using a forward recursion based on the Kalman filter [1]. It appears that it is possible to evaluate the third term using a backward recursion given by (5). The following lemma gives a useful expression for $p(\mathbf{y}_{t:T} \,|\, \mathbf{r}_{t:T}, x_{t-1})$.

Lemma 1: For any $t = 1, \ldots, T$, $p(\mathbf{y}_{t:T} \,|\, \mathbf{r}_{t:T}, x_{t-1})$ is a Gaussian distribution with mean $M_t(\mathbf{r}_{t:T}) x_{t-1} + \mathbb{E}\{[N_t(\mathbf{r}_{t:T})]\}$ and covariance $\mathrm{cov}\{N_t(\mathbf{r}_{t:T})\} > 0$, where $M_t(\mathbf{r}_{t:T})$ and $N_t(\mathbf{r}_{t:T})$ are defined in Appendix B.A. Let us define $L_t(\mathbf{r}_{t:T}) \triangleq N_t(\mathbf{r}_{t:T}) N_t^{\mathrm{T}}(\mathbf{r}_{t:T})$ and, subsequently, the following quantities:

$$P'^{-1}_{t-1\,|\,t}(\mathbf{r}_{t:T}) \triangleq M_t^{\mathrm{T}}(\mathbf{r}_{t:T}) L_t^{-1}(\mathbf{r}_{t:T}) M_t(\mathbf{r}_{t:T}),$$
$$P'^{-1}_{t-1\,|\,t}(\mathbf{r}_{t:T}) m'_{t-1\,|\,t}(\mathbf{r}_{t:T}) \triangleq M_t^{\mathrm{T}}(\mathbf{r}_{t:T}) L_t^{-1}(\mathbf{r}_{t:T}) \mathbf{y}_{t:T}.$$

If $p(\mathbf{y}_{t:T} \,|\, \mathbf{r}_{t:T}, x_{t-1})$ is integrable in x_{t-1}, then $(\int p(\mathbf{y}_{t:T} \,|\, \mathbf{r}_{t:T}, x_{t-1}) \, dx_{t-1})^{-1} p(\mathbf{y}_{t:T} \,|\, \mathbf{r}_{t:T}, x_{t-1})$ is a Gaussian distribution of argument x_{t-1}, mean $m'_{t-1\,|\,t}(\mathbf{r}_{t:T})$, and covariance $P'_{t-1\,|\,t}(\mathbf{r}_{t:T})$, hence the introduction of this notation. In the general case, $p(\mathbf{y}_{t:T} \,|\, \mathbf{r}_{t:T}, x_{t-1})$ is not integrable in x_{t-1}, but $P'^{-1}_{t-1\,|\,t}(\mathbf{r}_{t:T})$ and $P'^{-1}_{t-1\,|\,t}(\mathbf{r}_{t:T}) m'_{t-1\,|\,t}(\mathbf{r}_{t:T})$ always satisfy the following backward information filter recursion.

1) *Initialization*

$$P'^{-1}_{T\,|\,T}(r_T) = C^{\mathrm{T}}(r_T) (D(r_T) D^{\mathrm{T}}(r_T))^{-1} C(r_T)$$
$$P'^{-1}_{T\,|\,T}(r_T) m'_{T\,|\,T}(r_T) = C^{\mathrm{T}}(r_T) (D(r_T) D^{\mathrm{T}}(r_T))^{-1} (y_T - G(r_T) u_T). \quad (6)$$

2) *Backward recursion. For* $t = T-1, \ldots, 1$,

$$\mathbf{\Delta}_{t+1} = \left[I_{n_v} + B^{\mathrm{T}}(r_{t+1}) P'^{-1}_{t+1\,|\,t+1}(\mathbf{r}_{t+1:T}) B(r_{t+1}) \right]^{-1}$$
$$P'^{-1}_{t\,|\,t+1}(\mathbf{r}_{t+1:T}) = A^{\mathrm{T}}(r_{t+1}) P'^{-1}_{t+1\,|\,t+1}(\mathbf{r}_{t+1:T}) \times \Big(I_{n_x} - B(r_{t+1}) \mathbf{\Delta}_{t+1}(\mathbf{r}_{t+1:T}) \times B^{\mathrm{T}}(r_{t+1}) P'^{-1}_{t+1\,|\,t+1}(\mathbf{r}_{t+1:T}) \Big) A(r_{t+1})$$
$$P'^{-1}_{t\,|\,t+1}(\mathbf{r}_{t+1:T}) m'_{t\,|\,t+1}(\mathbf{r}_{t+1:T}) = A^{\mathrm{T}}(r_{t+1}) \times \Big(I_{n_x} - P'^{-1}_{t+1\,|\,t+1}(\mathbf{r}_{t+1:T}) \times B(r_{t+1}) \mathbf{\Delta}_{t+1}(\mathbf{r}_{t+1:T}) B^{\mathrm{T}}(r_{t+1}) \Big) \times P'^{-1}_{t+1\,|\,t+1}(\mathbf{r}_{t+1:T}) (m'_{t+1\,|\,t+1}(\mathbf{r}_{t+1:T}) - F(r_{t+1}) u_{t+1})$$
$$P'^{-1}_{t\,|\,t}(\mathbf{r}_{t:T}) = P'^{-1}_{t\,|\,t+1}(\mathbf{r}_{t+1:T}) + C^{\mathrm{T}}(r_t) (D(r_t) D^{\mathrm{T}}(r_t))^{-1} C(r_t)$$
$$P'^{-1}_{t\,|\,t}(\mathbf{r}_{t:T}) m'_{t\,|\,t}(\mathbf{r}_{t:T}) = P'^{-1}_{t\,|\,t+1}(\mathbf{r}_{t+1:T}) m'_{t\,|\,t+1}(\mathbf{r}_{t+1:T}) + C^{\mathrm{T}}(r_t) (D(r_t) D^{\mathrm{T}}(r_t))^{-1} (y_t - G(r_t) u_t). \quad (7)$$

Proof: See Appendix B.A.

Now, combining (4) and the previous lemma, one obtains an expression for $p(r_t \,|\, \mathbf{y}_{1:T}, \mathbf{r}_{-t})$. ■

Proposition 1: For any $t = 2, \ldots, T-1$, we have, if $P_{t\,|\,t}(\mathbf{r}_{1:t}) = 0_{n_x \times n_x}$

$$p(r_t \,|\, \mathbf{y}_{1:T}, \mathbf{r}_{-t}) \propto p_{r_{t-1} r_t} p_{r_t r_{t+1}} N(\tilde{y}_{t\,|\,t-1}(\mathbf{r}_{1:t}), S_t(\mathbf{r}_{1:t})) \times \exp\left(-\frac{1}{2} m_{t\,|\,t}^{\mathrm{T}}(\mathbf{r}_{1:t}) P'^{-1}_{t\,|\,t+1}(\mathbf{r}_{t+1:T}) m_{t\,|\,t}(\mathbf{r}_{1:t}) - 2 m_{t\,|\,t}^{\mathrm{T}}(\mathbf{r}_{1:t}) P'^{-1}_{t\,|\,t+1}(\mathbf{r}_{t+1:T}) m'_{t\,|\,t+1}(\mathbf{r}_{t+1:T}) \right). \quad (8)$$

If $P_{t\,|\,t}(\mathbf{r}_{1:t}) \neq 0_{n_x \times n_x}$, then it exists that $\tilde{\Pi}_{t\,|\,t}(\mathbf{r}_{1:t})$, and $\tilde{Q}_{t\,|\,t}(\mathbf{r}_{1:t})$ such that $P_{t\,|\,t}(\mathbf{r}_{1:t}) = \tilde{Q}_{t\,|\,t}(\mathbf{r}_{1:t}) \tilde{\Pi}_{t\,|\,t}(\mathbf{r}_{1:t}) \tilde{Q}^{\mathrm{T}}_{t\,|\,t}(\mathbf{r}_{1:t})$. The matrices $\tilde{Q}_{t\,|\,t}(\mathbf{r}_{1:t})$ and $\tilde{\Pi}_{t\,|\,t}(\mathbf{r}_{1:t})$ are straightforwardly obtained using the singular value decomposition of $P_{t\,|\,t}(\mathbf{r}_{1:t})$. Matrix $\tilde{\Pi}_{t\,|\,t}(\mathbf{r}_{1:t})$ is a $n_t \times n_t, 1 \leq n_t \leq n_x$ diagonal matrix with the nonzero eigenvalues of $P_{t\,|\,t}(\mathbf{r}_{1:t})$ as elements. Then, one has

$$p(r_t | \mathbf{y}_{1:T}, \mathbf{r}_{-t}) \propto p_{r_{t-1} r_t} p_{r_t r_{t+1}} N(\tilde{y}_{t|t-1}(\mathbf{r}_{1:t}), S_t(\mathbf{r}_{1:t})) \times \left| \tilde{\Pi}_{t|t}(\mathbf{r}_{1:t}) \tilde{Q}^{\mathrm{T}}_{t|t}(\mathbf{r}_{1:t}) P'^{-1}_{t|t+1}(\mathbf{r}_{t+1:T}) \tilde{Q}_{t|t}(\mathbf{r}_{1:t}) + I_{n_t} \right|^{-1/2} \times \exp\left(-\frac{1}{2} \left[m_{t|t}^{\mathrm{T}}(\mathbf{r}_{1:t}) P'^{-1}_{t|t+1}(\mathbf{r}_{t+1:T}) m_{t|t}(\mathbf{r}_{1:t}) - 2 m_{t|t}^{\mathrm{T}}(\mathbf{r}_{1:t}) P'^{-1}_{t|t+1}(\mathbf{r}_{t+1:T}) m'_{t|t+1}(\mathbf{r}_{t+1:T}) - (m'_{t|t+1}(\mathbf{r}_{t+1:T}) - m_{t|t}(\mathbf{r}_{1:t}))^{\mathrm{T}} \times P'^{-1}_{t|t+1}(\mathbf{r}_{t+1:T}) R_{t|t}(\mathbf{r}_{1:t}) \times P'^{-1}_{t|t+1}(\mathbf{r}_{t+1:T}) (m'_{t|t+1}(\mathbf{r}_{t+1:T}) - m_{t|t}(\mathbf{r}_{1:t})) \right) \quad (9)$$

where

$$R_{t\,|\,t}(\mathbf{r}_{1:t}) = \tilde{Q}_{t\,|\,t}(\mathbf{r}_{1:t}) \left[\tilde{\Pi}^{-1}_{t\,|\,t}(\mathbf{r}_{1:t}) + \tilde{Q}^{\mathrm{T}}_{t\,|\,t}(\mathbf{r}_{1:t}) \times P'^{-1}_{t\,|\,t+1}(\mathbf{r}_{t+1:T}) \tilde{Q}_{t\,|\,t}(\mathbf{r}_{1:t}) \right]^{-1} \tilde{Q}^{\mathrm{T}}_{t\,|\,t}(\mathbf{r}_{1:t}). \quad (10)$$

The quantities $m_{t|t-1}(\mathbf{r}_{1:t}), P_{t|t-1}(\mathbf{r}_{1:t}), m_{t|t}(\mathbf{r}_{1:t}), P_{t|t}(\mathbf{r}_{1:t})$, $\tilde{y}_{t|t-1}(\mathbf{r}_{1:t})$, and $S_t(\mathbf{r}_{1:t})$ are, respectively, the one-step ahead prediction and covariance of x_t, the filtered estimate and covariance of x_t, the innovation at time t, and the covariance of this innovation. These quantities are given by the Kalman filter, the system (1) and (2) being linear Gaussian until t conditional upon $\mathbf{r}_{1:t}$.

Proof: See Appendix B.B.

To sum up, the algorithm to sample from $p(r_t \,|\, \mathbf{y}_{1:T}, \mathbf{r}_{-T})$ for $t = 1, \ldots, T$ requires first the computation of the backward information filter, second, the evaluatation of $p(r_t \,|\, \mathbf{y}_{1:T}, \mathbf{r}_{-T})$ combining the information and Kalman filters, and finally, sampling from $p(r_t \,|\, \mathbf{y}_{1:T}, \mathbf{r}_{-T})$ and storing accordingly the

updated set of sufficient statistics $m_{t\,|\,t}(\mathbf{r}_{1:t-1}^{(k)}, r_t^{(k)})$ and $P_{t\,|\,t}(\mathbf{r}_{1:t-1}^{(k)}, r_t^{(k)})$. It proceeds as follows at iteration k.

Backward–Forward procedure

1) For $t = T, \ldots, 1$ compute and store $P'^{-1}_{t\,|\,t+1}(\mathbf{r}_{t+1:T}^{(k-1)})$ and $P'^{-1}_{t\,|\,t+1}(\mathbf{r}_{t+1:T}^{(k-1)}) m'_{t\,|\,t+1}(\mathbf{r}_{t+1:T}^{(k-1)})$ using (6) and (7).

2) For $t = 1, \ldots, T$

- For $i = \ldots, s$, run the one step-ahead Kalman filter with $r_t = i$, store $m_{t\,|\,t}(\mathbf{r}_{1:t-1}^{(k)}, r_t = i)$, and $P_{t\,|\,t}(\mathbf{r}_{1:t-1}^{(k)}, r_t = i)$, then compute up to a normalizing constant $p(r_t = i \,|\, \mathbf{y}_{1:T}, \mathbf{r}_{-t}^{(k)})$ using (9).
- Sample $r_t^{(k)} \sim p(r_t \,|\, \mathbf{y}_{1:T}, \mathbf{r}_{-t}^{(k)})$, store $m_{t\,|\,t}(\mathbf{r}_{1:t-1}^{(k)}, r_t^{(k)})$ and $P_{t\,|\,t}(\mathbf{r}_{1:t-1}^{(k)}, r_t^{(k)})$.

End For ■.

The complexity of this algorithm at each iteration is thus $O(T)$.

2) Convergence Issues: It is easy to check that the simulated sequence $\{\mathbf{r}_{1:T}^{(k)}; k \in \mathbb{N}\}$ is a finite state-space irreducible and aperiodic Markov chain on R so that uniform geometric convergence of the Markov chain toward its invariant distribution $p(r_{1:T} \,|\, \mathbf{y}_{1:T})$ holds; see, for example, [29]. Consequently, $\bar{\mathbf{r}}_{1:T}(N)$ and $\bar{\mathbf{x}}_{0:T}(N)$, given by (3), converge almost surely toward the MMSE estimates and satisfy a central limit theorem.

IV. Marginal Maximum *a Posteriori* State Sequence Estimation of $\mathbf{r}_{1:T}$

Obtaining a global maximum of $p(\mathbf{r}_{1:T} \,|\, \mathbf{y}_{1:T})$ requires the solution of a nondeterministic polynomial (NP) combinatorial optimization problem [31]. We propose a suboptimal deterministic algorithm and a stochastic globally convergent algorithm to solve it.

A. Algorithms

The proposed deterministic algorithm is a coordinate ascent method that uses the fact that one can maximize iteratively and successively $p(r_t \,|\, \mathbf{y}_{1:T}, \mathbf{r}_{-t})$ over r_t for $t = 1, \ldots, T$. The stochastic algorithm is a SA algorithm that is just a nonhomogeneous version of the algorithm proposed in the previous section, which can also be interpreted as a randomized version of the deterministic algorithm. It is a nonhomogeneous Markov chain whose transition kernel at iteration k depends on a so-called cooling schedule $\{\gamma_k; k \in \mathbb{N}\}$ verifying $\gamma_{k+1} \geq \gamma_k, \lim_{k\to+\infty} \gamma_k = +\infty$ and $\gamma_0 > 0$ [29]. The proposed algorithms proceed as follows.

Deterministic/Stochastic Algorithms to Estimate the MMAP Sequence of $\mathbf{r}_{1:T}$

1) Initialization. Set randomly $\mathbf{r}_{1:T}^{(0)} \in R$.

2) Iteration k, $k \geq 1$

- **Deterministic algorithm**: For $t = 1, \ldots, T, r_t^{(k)} = \operatorname{argmax}_{\{1,\ldots,s\}} p(r_t \,|\, \mathbf{y}_{1:T}, \mathbf{r}_{-t}^{(k)})$.
- **Stochastic algorithm:** For $t = 1, \ldots, T$, sample $r_t^{(k)} \sim \bar{p}^{\gamma_k}(r_t \,|\, \mathbf{y}_{1:T}, \mathbf{r}_{-t}^{(k)})$. ■

$\bar{p}^{\gamma_k}(r_t \,|\, \mathbf{y}_{1:T}, \mathbf{r}_{-t}^{(k)}) \propto [p(r_t \,|\, \mathbf{y}_{1:T}, \mathbf{r}_{-t}^{(k)})]^{\gamma_k}$, and $\mathbf{r}_{-t}^{(k)} \triangleq (r_1^{(k)}, \ldots, r_{t-1}^{(k)}, r_{t+1}^{(k-1)}, \ldots, r_T^{(k-1)})$.

B. Implementation and Convergence Issues

1) Implementations Issues: In Section III, we proposed a backward-forward procedure to estimate and sample $p(r_t \,|\, \mathbf{y}_{1:T}, \mathbf{r}_{-t})$ for $t = 1, \ldots, T$. We can use the same procedure to estimate $p(r_t \,|\, \mathbf{y}_{1:T}, \mathbf{r}_{-t})$ to maximize it or to sample from $\bar{p}^{\gamma_k}(r_t \,|\, \mathbf{y}_{1:T}, \mathbf{r}_{-t}^{(k)})$ as $p(r_t \,|\, \mathbf{y}_{1:T}, \mathbf{r}_{-t})$ is a discrete distribution. Thus, the computational complexity of these algorithms is $O(T)$.

2) Convergence Issues: The deterministic algorithm is a simple coordinate ascent method where the components are maximized one at a time. By construction, the sequence $(\mathbf{r}_{1:T}^{(k)})$ defined by the deterministic algorithm satisfies $p(\mathbf{r}_{1:T}^{(k)} \,|\, \mathbf{y}_{1:T}) \geq p(\mathbf{r}_{1:T}^{(k-1)} \,|\, \mathbf{y}_{1:T})$. It converges in a finite number of iterations toward a local maximum of $p(\mathbf{r}_{1:T} \,|\, \mathbf{y}_{1:T})$.[1] The convergence of the SA algorithm toward the set of global maxima of $p(\mathbf{r}_{1:T} \,|\, \mathbf{y}_{1:T})$ follows from standard arguments [14] for a logarithmic cooling schedule $\gamma_k = C \ln(k)$, where $C > 0$.

Remark 1: C is unknown in practice. Moreover, a logarithmic cooling schedule goes too slowly to infinity to be implemented. "Faster" linear growing cooling schedules are used in applications; see Section VII.

V. Marginal Maximum *a Posteriori* State Sequence Estimation of $\mathbf{x}_{0:T}$

Obtaining the MMAP sequence of $\mathbf{x}_{0:T}$ requires the solution of a complex global optimization problem. We propose a suboptimal deterministic algorithm and a stochastic algorithm to solve it.

A. Algorithms

The proposed suboptimal deterministic algorithm is a coordinate ascent method that maximizes successively and iteratively $p(x_t \,|\, \mathbf{y}_{1:T}, \mathbf{x}_{-t})$ over x_t for $t = 0, \ldots, T$. The proposed stochastic algorithm is a SA algorithm, which is a randomized version of the deterministic algorithm. It is a nonhomogeneous Markov chain whose transition kernel at iteration k depends on a so-called cooling schedule $\{\gamma_k; k \in \mathbb{N}\}$, verifying $\gamma_{k+1} \geq \gamma_k, \lim_{k\to+\infty} \gamma_k = +\infty$ and $\gamma_0 > 0$. To work, these two algorithms require the following assumption.

Assumption: $B(i)B^{\mathrm{T}}(i) > 0$ for all $i \in \{1, \ldots, s\}$.[2]

The proposed algorithms proceed as follows.

Deterministic/Stochastic Algorithms to Estimate the MMAP Sequence of $\mathbf{x}_{0:T}$

1) Initialization. Set randomly $\mathbf{x}_{0:T}^{(0)}$.

2) Iteration k, $k \geq 1$

- **Deterministic algorithm**: For $t = 0, \ldots, T$, $x_t^{(k)} = \operatorname{argmax}_{x_t} p(x_t \,|\, \mathbf{y}_{1:T}, \mathbf{x}_{-t}^{(k)})$.

[1] In a finite space, the notion of local maximum is induced by the updating scheme of the algorithm used to perform maximization.

[2] If $B(i)B^{\mathrm{T}}(i) \neq 0_{n_x \times n_x}$, then the jump Markov linear system can be transformed to a new system where the noise covariance matrix is positive definite; see [17, Sec. 3.9] for details.

• **Stochastic algorithm:** For $t = 0, \ldots, T$, sample $x_t^{(k)} \sim \bar{p}^{\gamma_k}(x_t \,|\, \mathbf{y}_{1:T}, \mathbf{x}_{-t}^{(k)})$. ■

$\bar{p}^{\gamma_k}(x_t \,|\, \mathbf{y}_{1:T}, \mathbf{x}_{-t}^{(k)}) \propto [p(x_t \,|\, \mathbf{y}_{1:T}, \mathbf{x}_{-t}^{(k)})]^{\gamma_k}$, and $\mathbf{x}_{-t}^{(k)} \triangleq (x_0^{(k)}, \ldots, x_{t-1}^{(k)}, x_{t+1}^{(k-1)}, \ldots, x_T^{(k-1)})$.

B. Implementation and Convergence Issues

1) Implementation Issues: This algorithm requires the estimation of $p(x_t \,|\, \mathbf{y}_{1:T}, \mathbf{x}_{-t}) \propto p(\mathbf{y}_{1:T}, \mathbf{x}_{0:T})$ to maximize it or to sample from $\bar{p}^{\gamma_k}(x_t \,|\, \mathbf{y}_{1:T}, \mathbf{x}_{-t}^{(k)})$. A direct evaluation of this distribution would have a complexity $O(T^2)$. We develop here an algorithm whose complexity is $O(T)$. Conditional upon $\mathbf{x}_{0:T}$, the system (1) and (2) is a finite state-space HMM [28]. Thus, the likelihood $p(\mathbf{y}_{1:T}, \mathbf{x}_{0:T})$ can be computed by the forward-backward recursion adapted to this particular case. (The modifications needed to handle the case of boundaries are straightforward and omitted here.) For any $t = 2, \ldots, T-1$

$$\begin{aligned} &p(\mathbf{y}_{1:T}, \mathbf{x}_{0:T}) \\ &\quad = p(\mathbf{y}_{1:t-1}, \mathbf{x}_{0:t-1}) p(y_t, x_t \,|\, \mathbf{y}_{1:t-1}, \mathbf{x}_{0:t-1}) \\ &\qquad \times \sum_{i=1}^{s} p(\mathbf{y}_{t+1:T}, \mathbf{x}_{t+1:T} \,|\, r_t = i, x_t) p(r_t = i \,|\, \mathbf{y}_{1:t}, \mathbf{x}_{0:t}) \end{aligned} \tag{11}$$

where

$$\begin{aligned} &p(\mathbf{y}_{t:T}, \mathbf{x}_{t:T} \,|\, r_{t-1} = i, x_{t-1}) \\ &\quad = \sum_{j=1}^{s} p(\mathbf{y}_{t+1:T}, \mathbf{x}_{t+1:T} \,|\, r_t = j, x_t) \\ &\qquad \times p(y_t, x_t \,|\, r_t = j, x_{t-1}) p_{ij}. \end{aligned} \tag{12}$$

The two first terms on the right-hand side of (11) can be computed using a forward recursion based on the HMM filter [1], [28]. It is possible to evaluate the third term using a backward recursion given by (12). Combining these results, we get the following expression for $p(x_t \,|\, \mathbf{y}_{1:T}, \mathbf{x}_{-t})$.

Proposition 2: For any $t = 2, \ldots, T-2$, we have

$$\begin{aligned} &p(x_t \,|\, \mathbf{y}_{1:T}, \mathbf{x}_{-t}) \\ &\quad = \sum_{i=1}^{s} \sum_{j=1}^{s} \alpha_t(i,j) N(x_t - m_t(i,j), P_t(i,j)) \end{aligned} \tag{13}$$

where

$$\begin{aligned} P_t^{-1}(i,j) &= (B(i)B^{\mathrm{T}}(i))^{-1} + C^{\mathrm{T}}(i)(D(i)D^{\mathrm{T}}(i))^{-1}C(i) \\ &\quad + A^{\mathrm{T}}(j)(B(j)B^{\mathrm{T}}(j))^{-1}A(j) \end{aligned} \tag{14}$$

$$\begin{aligned} m_t(i,j) = P_t(i,j)[&(B(i)B^{\mathrm{T}}(i))^{-1}(A(i)x_{t-1} + F(i)u_t) \\ &+ C^{\mathrm{T}}(i)(D(i)D^{\mathrm{T}}(i))^{-1}(y_t - G(i)u_t) \\ &+ A^{\mathrm{T}}(j)(B(j)B^{\mathrm{T}}(j))^{-1}(x_{t+1} - F(j)u_{t+1})] \end{aligned} \tag{15}$$

and

$$\alpha_t(i,j) = \left[\sum_{i,j} \tilde{\alpha}_t(i,j) \right]^{-1} \tilde{\alpha}_t(i,j) \tag{16}$$

where $\tilde{\alpha}_t(i,j)$ is given by

$$\begin{aligned} \tilde{\alpha}_t(i,j) = &p(\mathbf{y}_{t+2:T}, \mathbf{x}_{t+2:T} \,|\, r_{t+1} = j, x_{t+1}) \\ &\times p_{ij} p(r_t = i \,|\, \mathbf{y}_{1:t-1}, \mathbf{x}_{0:t-1}) \\ &\times N^{-1}(m_t(i,j), P_t(i,j)) \\ &\times N(y_t - G(i)u_t, D(i)D^{\mathrm{T}}(i)) \\ &\times N(A(i)x_{t-1} + F(i)u_t, B(i)B^{\mathrm{T}}(i)) \\ &\times N(x_{t+1} - F(j)u_{t+1}, B(j)B^{\mathrm{T}}(j)) \\ &\times N(y_{t+1} - C(j)x_{t+1} - G(j)u_{t+1}, D(j)D^{\mathrm{T}}(j)). \end{aligned} \tag{17}$$

Proof: See Appendix B.C.

To maximize $p(x_t \,|\, \mathbf{y}_{1:T}, \mathbf{x}_{-t})$ or to sample from $\bar{p}^{\gamma_k}(x_t \,|\, \mathbf{y}_{1:T}, \mathbf{x}_{-t})$ for $t = 0, \ldots, T$, the algorithm proceeds thus as follows at iteration k.

Backward–Forward procedure

1) For $t = T, \ldots, 1$ compute and store $p(\mathbf{y}_{t:T}, \mathbf{x}_{t:T}^{(k-1)} \,|\, r_{t-1}, x_{t-1}^{(k-1)})$ for any $r_{t-1} = 1, \ldots, s$ using (12).

2) For $t = 0, \ldots, T$

• For $i = 1, \ldots, s$, run one step-ahead the HMM one-step ahead predictor $p(r_t = i \,|\, \mathbf{y}_{1:t-1}, \mathbf{x}_{0:t-1}^{(k)})$, and then compute $p(x_t \,|\, \mathbf{y}_{1:T}, \mathbf{x}_{-t}^{(k)})$ using (13) to (17).

• **Deterministic version**: $x_t^{(k)} = \arg\max_{x_t} p(x_t \,|\, \mathbf{y}_{1:T}, \mathbf{x}_{-t}^{(k)})$.

• **Stochastic version**: Sample $x_t^{(k)} \sim \bar{p}^{\gamma_k}(x_t \,|\, \mathbf{y}_{1:T}, \mathbf{x}_{-t}^{(k)})$. ■

This algorithm has a complexity $O(T)$ per iteration. However, there are additional problems as maximizing $p(x_t \,|\, \mathbf{y}_{1:T}, \mathbf{x}_{-t}^{(k)})$ does not admit any analytical solution, and sampling from $\bar{p}^{\gamma_k}(x_t \,|\, \mathbf{y}_{1:T}, \mathbf{x}_{-t}^{(k)})$ is not obvious. To maximize $p(x_t \,|\, \mathbf{y}_{1:T}, \mathbf{x}_{-t}^{(k)})$, we propose to use a gradient-based method initialized at the previous estimate $x_t^{(k-1)}$. To sample from $\bar{p}^{\gamma_k}(x_t \,|\, \mathbf{y}_{1:T}, \mathbf{x}_{-t}^{(k)})$, we propose to use a Metropolis–Hastings (M–H) algorithm coupled with an accept/reject procedure [29, p. 270], i.e., we use an M–H algorithm with a proposal distribution proportional to $\min(p^{\gamma_k}(x_t \,|\, \mathbf{y}_{1:T}, \mathbf{x}_{-t}^{(k)}), C_k p_k^*(x_t \,|\, \mathbf{y}_{1:T}, \mathbf{x}_{-t}^{(k)}))$, where $p_k^*(x_t \,|\, \mathbf{y}_{1:T}, \mathbf{x}_{-t}^{(k)})$ is defined later.

MH—accept/reject procedure to sample from $\bar{p}^{\gamma_k}(x_t \,|\, \mathbf{y}_{1:T}, \mathbf{x}_{-t}^{(k)})$

1) Repeat

• Sample a candidate $x_t^* \sim p_k^*(x_t \,|\, \mathbf{y}_{1:T}, \mathbf{x}_{-t}^{(k)})$, and sample $u \sim \mathcal{U}_{[0,1]}$.

Until $u \leq (p^{\gamma_k}(x_t^* \,|\, \mathbf{y}_{1:T}, \mathbf{x}_{-t}^{(k)}) / C_k p_k^*(x_t^* \,|\, \mathbf{y}_{1:T}, \mathbf{x}_{-t}^{(k)}))$.

2) Sample $u \sim \mathcal{U}_{[0,1]}$. Set $x_t^{(k)} = x_t^*$ with probability

$$\min\left\{1, \frac{p^{\gamma_k}\left(x_t^* \,|\, \mathbf{y}_{1:T}, \mathbf{x}_{-t}^{(k)}\right) p_k^*\left(x_t^{(k-1)} \,|\, \mathbf{y}_{1:T}, \mathbf{x}_{-t}^{(k)}\right)}{p^{\gamma_k}\left(x_t^{(k-1)} \,|\, \mathbf{y}_{1:T}, \mathbf{x}_{-t}^{(k)}\right) p_k^*\left(x_t^* \,|\, \mathbf{y}_{1:T}, \mathbf{x}_{-t}^{(k)}\right)}\right\}$$

$\times$ if $p^{\gamma_k}\left(x_t^* \,|\, \mathbf{y}_{1:T}, \mathbf{x}_{-t}^{(k)}\right) > C_k p_k^*\left(x_t^* \,|\, \mathbf{y}_{1:T}, \mathbf{x}_{-t}^{(k)}\right)$

$$\min\left\{1, \frac{C_k p_k^*\left(x_t^{(k-1)} \,|\, \mathbf{y}_{1:T}, \mathbf{x}_{-t}^{(k)}\right)}{p^{\gamma_k}\left(x_t^{(k-1)} \,|\, \mathbf{y}_{1:T}, \mathbf{x}_{-t}^{(k)}\right)}\right\}$$ otherwise

otherwise set $x_t^{(k)} = x_t^{(k-1)}$. ■

The proposal distribution $p_k^*(x_t \,|\, \mathbf{y}_{1:T}, \mathbf{x}_{-t}^{(k)})$ is a mixture of Gaussians given by

$$p_k^*\left(x_t \,|\, \mathbf{y}_{1:T}, \mathbf{x}_{-t}^{(k)}\right) = \sum_{i=1}^{s}\sum_{j=1}^{s} \alpha_t^*(i,j) N(x_t - m_t(i,j), \gamma_k P_t(i,j))$$

with

$$\alpha_t^*(i,j) = \frac{\alpha_t(i,j)|P_t(i,j)|^{(1-\gamma_k)/2}}{\sum_{l=1}^{s}\sum_{m=1}^{s} \alpha_t(l,m)|P_t(l,m)|^{(1-\gamma_k)/2}}$$

and

$$C_k = \left(\sum_{l=1}^{s}\sum_{m=1}^{s} \alpha_t(l,m)|P_t(l,m)|^{(1-\gamma_k)/2}\right) \times (2\pi)^{\frac{n_x}{2}(1-\gamma_k)}\gamma_k^{-\frac{n_x}{2}}.$$

2) Convergence Issues: By construction, the sequence $(\mathbf{x}_{0:T}^{(k)})$ defined by the deterministic algorithm satisfies $p(\mathbf{x}_{0:T}^{(k)} \,|\, \mathbf{y}_{1:T}) \geq p(\mathbf{x}_{0:T}^{(k-1)} \,|\, \mathbf{y}_{1:T})$. Actually, the gradient-based algorithm does not ensure that $x_t^{(k)} = \arg\max_{x_t} p(x_t \,|\, \mathbf{y}_{1:T}, \mathbf{x}_{-t}^{(k)})$ as it can be trapped in local maxima, but it ensures that $x_t^{(k)}$ is chosen such that $p(x_t^{(k)} \,|\, \mathbf{y}_{1:T}, \mathbf{x}_{-t}^{(k)}) \geq p(x_t^{(k-1)} \,|\, \mathbf{y}_{1:T}, \mathbf{x}_{-t}^{(k)})$ so that $p(\mathbf{x}_{0:T}^{(k)} \,|\, \mathbf{y}_{1:T})$ still increases monotonically.

As $\mathbf{x}_{0:T}$ does not lie in a finite or in a compact space, it appears much more difficult to prove convergence of the sequence $(\mathbf{x}_{0:T}^{(k)})$ toward the set of global maxima. We have not established such a result. However, it is easy to see that the assumption on $B(i)B^{\mathrm{T}}(i)$ ensures that the associated homogeneous Markov chains $\gamma_k = \gamma > 0$ for any $k \in \mathbb{N}$ is ergodic of limiting distribution $\bar{p}^{\gamma}(\mathbf{x}_{0:T} \,|\, \mathbf{y}_{1:T})$.

VI. Discussion

Numerous methods have been proposed earlier in the literature to address these problems. We have already discussed in the introduction the interest of our algorithms. We detail them here.

1) MMSE Estimation/MMAP Estimation of $\mathbf{r}_{1:T}$: To obtain MMSE estimates, previous algorithms are mainly based on a fixed-interval smoothing extension of deterministic and heuristic finite Gaussian mixtures approximations used in filtering such as the popular interactive multiple models (IMM) algorithm [2]; see, for example, [3], [15], and [18] for related methods. These algorithms are noniterative and computationally cheaper than the one we present. However, it is very difficult to quantify the approximations that are done. Moreover, these algorithms make the assumption that $p(\mathbf{y}_{t:T} \,|\, \mathbf{r}_{t:T}, x_{t-1})$ is integrable in x_{t-1}; it implies, for example, that $C^{\mathrm{T}}(r_T)(D(r_T)D^{\mathrm{T}}(r_T))^{-1}C(r_T)$ has to be regular, which is rarely true in practice, as outlined in [15, p. 1848]. This conservative assumption is relaxed here. In [5], an alternative Monte Carlo integration algorithm is proposed based on data augmentation [29], which samples iteratively and successively from $p(\mathbf{x}_{0:T} \,|\, \mathbf{y}_{1:T}, \mathbf{r}_{1:T})$ and $p(\mathbf{r}_{1:T} \,|\, \mathbf{y}_{1:T}, \mathbf{x}_{0:T})$; see [11] for convergence proofs. It requires the assumption $B(i)B^{\mathrm{T}}(i) > 0$; see also [10], [21], [23], and [30] for a similar assumption resulting from the introduction of the missing data set $\mathbf{x}_{0:T}$. The proposed algorithm does not perform data augmentation as it is based on a recursion that evaluates at each iteration $p(r_t \,|\, \mathbf{y}_{1:T}, \mathbf{r}_{-t})$ for $t = 1, \ldots, T$, the continuous states $\mathbf{x}_{0:T}$ being integrated out. The proposed stochastic and deterministic algorithms to maximize $p(\mathbf{r}_{1:T} \,|\, \mathbf{y}_{t:T})$ are based on the same recursion. A related strategy has been developed earlier in [19] and [26] to obtain the MMAP estimate of $\mathbf{r}_{1:T}$, but the proposed popular single most likely replacement (SMLR) algorithm has a computational complexity $O(T^2)$. This prohibitive computational complexity has motivated alternative approaches in the context of impulsive deconvolution [16]. An algorithm for MA models excited by BG processes is presented in [7], but its complexity depends explicitly on the square number of occurrences of the Bernoulli process. However, in a state-space framework [6], an algorithm of complexity $O(T)$ has already recently been proposed. Nevertheless, this algorithm is based on an approximate initialization of a backward recursion and assumes that $A(i)$ is regular for any i. Our recursion has a similar complexity at each iteration, but it does not rely on any approximation and makes no assumption on $A(i)$. The resulting algorithms thus have a wider range of applicability.

Moreover, even if $B(i)B^{\mathrm{T}}(i) > 0$, the proposed deterministic algorithm to obtain the MMAP of $\mathbf{r}_{1:T}$ is ensured to have a better asymptotic convergence rate than the expectation maximization (EM) algorithm in [23]. Indeed, it is a simple coordinate ascent method that avoids the introduction of missing data [24]. Finally, in the case where r_t is an independent sequence, the proposed stochastic algorithm is ensured to have a lower maximum correlation (see [22] for a definition) than the algorithm described in [5] and [11] according to ([22, Th. 5.1]).

2) MMAP Estimation of $\mathbf{x}_{0:T}$: To obtain the MMAP sequence estimate of $\mathbf{x}_{0:T}$, an EM algorithm has been recently proposed in [23]. It introduces the set $\mathbf{r}_{1:T}$ of missing data. Our deterministic algorithm is a simple coordinate ascent method that does not introduce any missing data. It is thus ensured to have a better asymptotic convergence rate than the EM algorithm in [23] according to [24]. Although we have not obtained any theoretical convergence on the proposed stochastic algorithm, the latter appears less sensitive to initialization in practice than its deterministic counterpart, as demonstrated in simulations presented in Section VII. It is thus of practical interest.

VII. Simulations

In simulations, the deterministic algorithms are iterated until convergence. Convergence occurs after no more than eight iterations in our experiments. Theoretically, the three stochastic algorithms require an infinite number of iterations to give the exact values of the MMSE and MMAP estimates. For all our simulations, we discard the first N_0 iterations to compute the MMSE estimates using the MCMC sampler. These first N_0 iterations correspond to the so-called burn-in period of the Markov chain.[3] As in [9], the MCMC sampler algorithm is then iterated until the computed values of the ergodic averages are no longer modified. To ensure convergence toward the set of global

[3]Methods for determining the burn-in period N_0 are beyond the scope of this paper; see [29] for an overview of such methods.

maxima, the SA algorithm presented in Section IV requires a logarithmic cooling schedule. Such a schedule is too slow to be implemented in practice. As it is usually done in practice, we implement N iterations of the SA algorithms with a linear cooling schedules, i.e., $\gamma_k = ak + b$ [29], satisfying $\gamma_0 = 0.1$ and $\gamma_N = 10$. Then, we apply the deterministic algorithms.

Computer simulations were carried out to evaluate the performance of our algorithms. All the algorithms were coded using Matlab©, and the simulations were performed on a Pentium II®. Section VII-A considers the problem of estimating a sparse signal based on a set of noisy data. We applied our algorithms to both simulated signals and a real data set. Section VII-B considers the problem of tracking a maneuvering target.

A. Deconvolution of Bernoulli–Gaussian Processes

In several problems related to seismic signal processing and nuclear science [7], [16], [19], [21], [26], [30], the signal of interest can be modeled as the output of a linear filter excited by a BG process and observed in white Gaussian noise. ARMA models allow for a parsimonious representation of the impulse response of the system and enjoy much popularity [26]. The signal of interest can be modeled as the output of an ARMA model filter excited by a BG process and observed in white Gaussian noise. For an ARMA(p,q) model, we have

$$s_t = \sum_{i=1}^{p} a_i s_{t-i} + v'_t + \sum_{i=1}^{q} b_i v'_{t-i} \tag{18}$$

$$y_t = s_t + w'_t \tag{19}$$

where

$$v'_t \sim \lambda \mathcal{N}\left(0, \sigma_v^2\right) + (1-\lambda)\delta_0, \quad w'_t \sim \mathcal{N}\left(0, \sigma_w^2\right) \tag{20}$$

for $0 < \lambda < 1$, and δ_0 is the delta-Dirac measure in 0. v'_t is assumed to be a white noise sequence. Note that it could also be modeled as a first-order Markov sequence to take into account the dead time of the sensor [21]. It is convenient from an algorithmic point of view to introduce the latent Bernoulli process $r_t \in \{1,2\}$ such that $\Pr(r_t = 1) = \lambda$ and

$$v'_t \,|\, r_t = 1 \sim \mathcal{N}\left(0, \sigma_v^2\right), \quad v'_t \,|\, r_t = 2 \sim \delta_0. \tag{21}$$

We can define an i.i.d. Gaussian sequence $v_t \sim \mathcal{N}(0,1)$ such that, conditional upon $r_t = 1$, $v'_t = \sigma_v v_t$. If we introduce the state vector x_t, such that $n_x = \max(p, q+1)$, and extend the ARMA coefficients, i.e., $a_i = 0$ for $i > p$ and $b_i = 0$ for $i > q$, then it is standard to re-express this model as a JMLS (1) and (2); see, for example, [33, p. 298].

We address here an application from nuclear science [10]. The aim is to deconvolve the output of a digital spectrometer. The transfer function is modeled by an ARMA$(3,2)$ whose impulse response is displayed in Fig. 1. The other parameters are equal to $\lambda = 0.07, \sigma_v = 0.37$ and $\sigma_w = 0.41$ (these parameters correspond to the real data set discussed below). In this application, these parameters are estimated in a preliminary *calibration* step. We apply our algorithms to some simulated signals.

We start with the MCMC algorithm presented in Section III to compute the MMSE estimates $\mathbb{E}\{\mathbf{x}_{1:T} \,|\, \mathbf{y}_{1:T}\}$. To compute the MMAP sequence $\mathbf{r}_{1:T}$, we implement both the SA algorithm and the deterministic algorithm. In this case, it is not possible to apply the algorithms presented in Section V as $B(2)\, B^{\mathrm{T}}(2) = 0$. The algorithm that computes the MMSE estimate is compared with the fixed-interval smoothers developed in [15] and [18]. The algorithms that compute the MMAP sequence $\mathbf{r}_{1:T}$ are compared with the algorithm presented in [19]. The MCMC and SA algorithms were run for $N = 100$ and $N = 1000$ iterations. All the simulations were run on $T = 400$ points and averaged over $M = 100$ independent runs with the same random initialization. The performance measure for the MMSE algorithms is the root mean square (RMS) position error computed as in (22), shown at the bottom of the next page, following from the MMSE estimates with respect to the true simulated trajectories, where $\hat{x}_t^{\mathrm{MMSE}}(m)$ is the MMSE estimate of x_t of the mth Monte Carlo simulation. For the MMAP, the performance measure is of course the penalized log-likelihood of the MMAP estimate $\hat{\mathbf{r}}_{1:T}^{\mathrm{MMAP}}$, i.e., $\log p(\hat{\mathbf{r}}_{1:T}^{\mathrm{MMAP}} \,|\, \mathbf{y}_{1:T})$. The results are presented in Tables I and II.

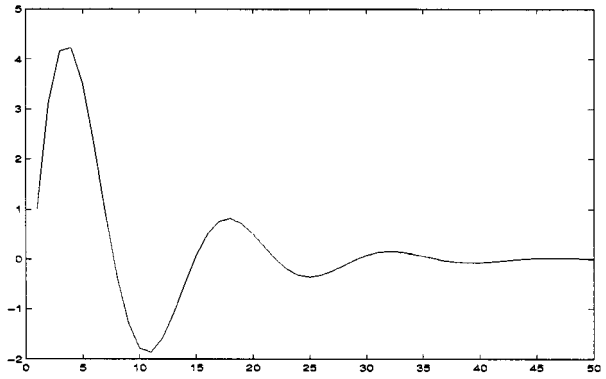

Fig. 1. Impulse response of the $\mathrm{ARMA}(3,2)$ model.

TABLE I
PERFORMANCE MEASURE FOR MMSE ESTIMATION

Algorithm	RMS	Computational time (in secs.)
MMSE estimate [18]	43.3	3.6
MMSE estimate [15]	45.4	2.9
MMSE estimate, $N = 100$	38.4	168.4
MMSE estimate, $N = 1000$	38.4	1686.0

It appears that our iterative algorithm that computes the MMSE with $N = 100$, although about roughly 50 times slower than the algorithms in [15] and [18], clearly outperforms them. Moreover, increasing the number of iterations of this algorithm by a factor 10 does not appear to modify the results. This suggests that the MCMC algorithm has converged toward the MMSE estimate. In terms of MMAP sequence estimation, our algorithms also outperform the current method applicable to this problem. The deterministic algorithm we propose has performance comparable with the SMLR algorithm in [19] while being computationally much cheaper (this improvement increases as T increases). The SA algorithms outperform the deterministic algorithms. Similarly to the MMSE case, it does

TABLE II
PERFORMANCE MEASURE FOR MMAP ESTIMATION OF $\mathbf{r}_{1:T}$

Algorithm	$\log p\left(\widehat{\mathbf{r}}_{1:T}^{MMAP} \mid \mathbf{y}_{1:T}\right)$	Computational time (in secs.)
MMAP estimate [19]	-234.7	692.9
D-MMAP $\mathbf{r}_{1:T}$	-234.7	10.7
S-MMAP $\mathbf{r}_{1:T}$, $N = 100$	-232.3	172.7
S-MMAP $\mathbf{r}_{1:T}$,, $N = 1000$	-232.2	1726.2

not appear to be really useful to use a large number of iterations. It is difficult to specify another objective criterion to compare these algorithms. However, by visual inspection, the estimated sequences obtained using the SA algorithm appear often better than the ones obtained using the deterministic algorithm.

B. Tracking of a Maneuvering Target

We address the problem of tracking a maneuvering target in noise. The difficulty in this problem arises from the uncertainty in the maneuvering command driving the target. The state of the target at time t is denoted as $x_t \triangleq (p_{x,t}, v_{x,t}, p_{y,t}, v_{y,t})^{\mathrm{T}}$, where $p_{x,t}$ $(p_{y,t})$ and $v_{x,t}$ $(v_{y,t})$ represent the position and velocity of the target in the x (respectively, in the y) direction. It evolves according to a JMLS model of parameters [2]

$$A = \begin{pmatrix} 1 & \rho & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \rho \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad B = 0.1 I_4$$
$$C = I_4, \quad G = 0_{4 \times n_u} \tag{23}$$

and $D = \sqrt{3}\,\mathrm{diag}(20, 1, 20, 1)$. The switching term is $F(r_t)u_t$, where r_t is a three-state Markov chain corresponding to the three possible maneuver commands: straight, left turn, and right turn. It has the following transition probabilities: $p_{m,m} = 0.9$ and $p_{m,n} = 0.05$ for $m \neq n$. We have for any t

$$\begin{aligned} F(1)u_t &= (0, 0, 0, 0)^{\mathrm{T}} \\ F(2)u_t &= (-1.225, -0.35, 1.225, 0.35)^{\mathrm{T}} \\ F(3)u_t &= (1.225, 0.35, -1.225, -0.35)^{\mathrm{T}}. \end{aligned} \tag{24}$$

We apply the algorithms developed in this paper to compute the MMSE estimate $\mathbb{E}\{\mathbf{x}_{1:T} \mid \mathbf{y}_{1:T}\}$, the MMAP sequence $\mathbf{r}_{1:T}$, and $\mathbf{x}_{1:T}$. The MCMC algorithm that computes the MMSE estimate is compared with the smoothers developed in [11], [15], and [18]. The algorithms that compute the MMAP sequences $\mathbf{r}_{1:T}$ and $\mathbf{x}_{1:T}$ are compared with the algorithms presented in [11] and [23]. The MCMC and SA algorithms were run for $N = 100$ and $N = 1000$ iterations. All the simulations were run on $T = 400$ points and averaged over $M = 100$ independent runs with the same random initialization. The performance measure for the MMSE algorithms is the root mean square (RMS) position error computed, shown in (25) at the bottom of the page, follows from the MMSE estimates with respect to the true simulated trajectories, where $\hat{p}_{x,t}^{\mathrm{MMSE}}(m)$ [respectively, $\hat{p}_{y,t}^{\mathrm{MMSE}}(m)$] is the MMSE target position estimate in the x (resp. y) direction at time t of the mth Monte Carlo simulation. The performance measure for the MMAP algorithms is the penalized log-likelihood of the MMAP estimates. We present, in Tables III-V, the performance of the different algorithms.

Our conclusions are very similar to those of the previous example. Our iterative algorithm that computes the MMSE with $N = 100$, although about roughly 50 times slower than the other deterministic algorithms, outperforms them, and increasing the number of iterations by a factor 10 does not modify the results. It also outperforms the alternative MCMC algorithm described in [11] for a small number of iterations. This is consistent with the theoretical results in [22], which suggest that our MCMC algorithm converges faster as $\mathbf{x}_{0:T}$ is integrated out. In terms of MMAP sequence estimation of $\mathbf{r}_{1:T}$, our deterministic algorithm has a computational complexity that is much cheaper than

$$\mathrm{RMS} = \sqrt{\frac{1}{MT} \sum_{m=1}^{M} \sum_{t=1}^{T} \left(x_t - \hat{x}_t^{\mathrm{MMSE}}(m)\right)^{\mathrm{T}} \left(x_t - \hat{x}_t^{\mathrm{MMSE}}(m)\right)} \tag{22}$$

$$\mathrm{RMS} = \sqrt{\frac{1}{MT} \sum_{m=1}^{M} \sum_{t=1}^{T} \left[\left(p_{x,t} - \hat{p}_{x,t}^{\mathrm{MMSE}}(m)\right)^2 + \left(p_{y,t} - \hat{p}_{y,t}^{\mathrm{MMSE}}(m)\right)^2\right]} \tag{25}$$

TABLE III
PERFORMANCE MEASURE FOR MMSE ESTIMATION

Algorithm	RMS	Computational time (in secs.)
MMSE estimate [18]	24.31	2.6
MMSE estimate [15]	25.67	1.8
MMSE estimate [11], $N = 100$	20.46	129.4
MMSE estimate [11], $N = 1000$	20.37	1319.6
MMSE estimate, $N = 100$	20.38	140.3
MMSE estimate, $N = 1000$	20.37	1405.7

the SMLR algorithm [19]. It is also cheaper than the EM algorithm developed in [23] and performs better. The SA algorithms outperform the deterministic algorithm and the algorithm presented in [11]. It does not appear to be really useful to use a large number of iterations for the SA. The same conclusion holds for $\mathbf{x}_{0:T}$. Note that as the variance of the observation noise and/or the dynamic noise decreases, our algorithms become increasingly more efficient than the EM-based algorithms, which, in the limit case, where one of these variances is equal to zero, do not even converge.

VIII. CONCLUSION

In this paper, we presented iterative deterministic and stochastic algorithms to compute MMSE and MMAP estimates for JMLS. These algorithms have a wider range of applicability than the current methods. Moreover the computational cost of an iteration and the memory requirements are linear in the data length. The deterministic algorithms proposed to estimate the MMAP state sequence estimates are coordinate ascent methods that compare favorably both theoretically and practically to EM-based algorithms. However, as any deterministic optimization method, they are sensitive to initialization. The stochastic algorithms based on homogeneous and nonhomogeneous MCMC methods are ensured to converge asymptotically toward the required estimates. In practice, they appear less sensitive to initialization. Two applications were presented to illustrate the performance of these algorithms for deconvolution of BG processes and tracking of a maneuvering target. Although we addressed here the case where the parameters of the JMLS were known, our algorithms can straightforwardly be used as part of more complex MCMC algorithms to perform parameter estimation [10], [29]. Finally, these algorithms can be combined with particle filtering methods [12] to perform online state estimation; see [13] for details.

APPENDIX A
NOTATION

- n_z: Dimension of an arbitrary vector z.
- $t \in \{1, 2, \ldots\}$: Discrete time.
- k: Iteration number of the various iterative algorithms.
- For $p < q$, , $\mathbf{z}_{p:q} \triangleq (z_p, z_{p+1}, \ldots, z_q)$.
- $N(m, \Sigma) = |2\pi\Sigma|^{-1/2} \exp(-1/2 m^{\mathrm{T}} \Sigma^{-1/2} m)$.
- $\mathcal{N}(m, \Sigma)$: Gaussian distribution of mean m and covariance Σ.
- $\mathcal{U}_{[0,1]}$: Uniform distribution on $[0, 1]$.
- $z \sim p(z)$: z distributed according to $p(z)$.
- $z \,|\, y \sim p(z)$: z conditional upon y distributed according to $p(z)$.
- For φ, $S^T \to \mathbb{R}$, $\int \varphi(\mathbf{r}) d\mathbf{r} \triangleq \sum_{\mathbf{r}_{1:T}(i) \in S^T} \varphi(\mathbf{r}_{1:T}(i))$.
- I_n: Identity matrix of dimensions $n \times n$.
- $[A]_{i:j,k:l}$: Submatrix including the ith to jth rows and the kth to lth columns of matrix A.
- A^{T}: Transpose matrix.

APPENDIX B
PROOFS OF LEMMAS AND PROPOSITIONS

A. Proof of Lemma 1

The observations can be expressed as follows: $\mathbf{y}_{t:T} = M_t(\mathbf{r}_{t:T}) x_{t-1} + N_t(\mathbf{r}_{t:T})$ where $M_t(\mathbf{r}_{t:T})$ is a $n_y(T-t+1) \times n_x$ matrix, and $N_t(\mathbf{r}_{t:T})$ is a $n_y(T-t+1) \times 1$ matrix such that

$$[M_t(\mathbf{r}_{t:T})]_{n_y(i-1)+1:n_y i, 1:nx} = C(r_{t+i-1}) \prod_{j=t+i-1}^{t} A(r_j)$$

$$[N_t(\mathbf{r}_{t:T})]_{n_y(i-1)+1:n_y i, 1:1} = D(r_{t+i-1}) w_{t+i-1} + \text{remaining terms}$$

for any $i \in \{1, 2, \ldots, T-t+1\}$. Thus, the distribution $p(\mathbf{y}_{t:T} \,|\, \mathbf{r}_{t:T}, x_{t-1})$ is Gaussian with mean $M_t(\mathbf{r}_{t:T}) x_{t-1} + \mathbb{E}[N_t(\mathbf{r}_{t:T})]$ and covariance $\mathrm{cov}[N_t(\mathbf{r}_{t:T})] > 0$, where the positiveness comes from the assumption $D(i) D^{\mathrm{T}}(i) > 0$ for $i \in S$. We define $L_t(\mathbf{r}_{t:T}) \triangleq N_t(\mathbf{r}_{t:T}) N_t^{\mathrm{T}}(\mathbf{r}_{t:T})$. To compute the parameters of the fixed-interval distribution $p(x_t \,|\, \mathbf{y}_{1:T})$, Mayne has established the algorithm to compute recursively in time $P'^{-1}_{t-1\,|\,t} \triangleq M_t^{\mathrm{T}}(\mathbf{r}_{t:T}) L_t^{-1}(\mathbf{r}_{t:T}) M_t(\mathbf{r}_{t:T})$ and $P'^{-1}_{t-1\,|\,t}$ $m'_{t-1\,|\,t} \triangleq M_t^{\mathrm{T}}(\mathbf{r}_{t:T}) L_t^{-1}(\mathbf{r}_{t:T}) \mathbf{y}_{t:T}$ [25]. This recursion is a backward information filter [1].

B. Proof of Proposition 1

$$\begin{aligned} p(r_t \,|\, \mathbf{y}_{1:T}, \mathbf{r}_{-t}) &\propto p(r_t \,|\, \mathbf{r}_{-t}) p(\mathbf{y}_{1:T} \,|\, \mathbf{r}_{-t}, r_t) \\ &\propto p(r_t \,|\, r_{t-1}, r_{t+1}) p(y_t \,|\, \mathbf{y}_{1:t-1}, \mathbf{r}_{1:t}) \\ &\quad \times \int p(\mathbf{y}_{t+1:T} \,|\, \mathbf{r}_{t+1:T}, x_t) p(x_t \,|\, \mathbf{y}_{1:t}, \mathbf{r}_{1:t})\, dx_t. \end{aligned}$$

The two first terms are easy to evaluate as $p(r_t \,|\, r_{t-1}, r_{t+1})$ is given by the transition matrix of the Markov chain and $p(y_t \,|\, \mathbf{y}_{1:t-1}, \mathbf{r}_{1:t}) = N(\tilde{y}_{t\,|\,t-1}(\mathbf{r}_{1:t}), S_t(\mathbf{r}_{1:t})), S_t(\mathbf{r}_{1:t})$, where the innovation $\tilde{y}_{t\,|\,t-1}(\mathbf{r}_{1:t})$ is computed using the Kalman filter. Now, the last term is equal to

$$\begin{aligned} &\int p(\mathbf{y}_{t+1:T} \,|\, \mathbf{r}_{t+1:T}, x_t) p(x_t \,|\, \mathbf{r}_{1:t}, \mathbf{y}_{1:t})\, dx_t \\ &= N(\mathbf{y}_{t+1:T} - M_{t+1}(\mathbf{r}_{t+1:T}) m_{t\,|\,t}(\mathbf{r}_{1:t}), L_{t+1}(\mathbf{r}_{t+1:T}) \\ &\quad + M_{t+1}(\mathbf{r}_{t+1:T}) P_{t\,|\,t}(\mathbf{r}_{1:t}) M_{t+1}^{\mathrm{T}}(\mathbf{r}_{t+1:T})) \end{aligned}$$

TABLE IV
PERFORMANCE MEASURE FOR MMAP ESTIMATION OF $\mathbf{r}_{1:T}$

Algorithm	$\log p\left(\widehat{\mathbf{r}}_{1:T}^{MMAP} \middle\vert \mathbf{y}_{1:T}\right)$	Computational time (in secs.)
MMAP estimate [19]	50.1	582.3
MMAP-EM algorithm [23]	49.9	15.4
MMAP algorithm [11], $N = 100$	50.2	131.3
MMAP algorithm [11], $N = 1000$	50.0	1323.2
Deterministic-MMAP $\mathbf{r}_{1:T}$	50.1	8.5
SA-MMAP $\mathbf{r}_{1:T}$, $N = 100$	51.3	142.9
SA-MMAP $\mathbf{r}_{1:T}$, $N = 1000$	51.4	1432.0

TABLE V
PERFORMANCE MEASURE FOR MMAP ESTIMATION OF $\mathbf{x}_{0:T}$

Algorithm	$\log p\left(\widehat{\mathbf{x}}_{0:T}^{MMAP} \middle\vert \mathbf{y}_{1:T}\right)$	Computational time (in secs.)
MMAP-EM algorithm [23]	-183.4	18.9
Deterministic-MMAP $\mathbf{x}_{0:T}$	-183.0	12.4
SA-MMAP $\mathbf{x}_{0:T}$, $N = 100$	-182.3	177.8
SA-MMAP $\mathbf{x}_{0:T}$, $N = 1000$	-182.2	1782.1

by straightforward calculations. When $P_{t\,|\,t}(\mathbf{r}_{1:t}) = 0_{n_x \times n_x}$, (8) follows immediately. If $P_{t\,|\,t}(\mathbf{r}_{1:t}) \neq 0_{n_x \times n_x}$ and $P_{t\,|\,t}(\mathbf{r}_{1:t})$ is symmetric, we have $\Pi_{t\,|\,t}(\mathbf{r}_{1:t})$, which is a $n_t \times n_t$ diagonal matrix with $1 \leq n_t \leq n_x$ first nonzero diagonal terms and $Q_{t\,|\,t}^{\mathrm{T}}(\mathbf{r}_{1:t}) Q_{t\,|\,t}(\mathbf{r}_{1:t}) = I_{n_x}$, such that

$$P_{t\,|\,t}(\mathbf{r}_{1:t}) = \tilde{Q}_{t\,|\,t}(\mathbf{r}_{1:t}) \tilde{\Pi}_{t\,|\,t}(\mathbf{r}_{1:t}) \tilde{Q}_{t\,|\,t}^{\mathrm{T}}(\mathbf{r}_{1:t}).$$

Then

$$\begin{aligned} &\left| L_{t+1}(\mathbf{r}_{t+1:T}) + M_{t+1}(\mathbf{r}_{t+1:T}) P_{t\,|\,t}(\mathbf{r}_{1:t}) M_{t+1}^{\mathrm{T}}(\mathbf{r}_{t+1:T}) \right| \\ &\quad = \left| L_{t+1}(\mathbf{r}_{t+1:T}) \right| \left| \tilde{\Pi}_{t\,|\,t}(\mathbf{r}_{1:t}) \tilde{Q}_{t\,|\,t}^{\mathrm{T}}(\mathbf{r}_{1:t}) \right. \\ &\qquad \left. \times P_{t\,|\,t+1}'^{-1}(\mathbf{r}_{t+1:T}) \tilde{Q}_{t\,|\,t}(\mathbf{r}_{1:t}) + I_{n_t} \right| \end{aligned}$$

and

$$\begin{aligned} &\Big[L_{t+1}(\mathbf{r}_{t+1:T}) + M_{t+1}(\mathbf{r}_{t+1:T}) \tilde{Q}_{t\,|\,t}(\mathbf{r}_{1:t}) \tilde{\Pi}_{t\,|\,t}(\mathbf{r}_{1:t}) \\ &\quad \times \tilde{Q}_{t\,|\,t}^{\mathrm{T}}(\mathbf{r}_{1:t}) M_{t+1}^{\mathrm{T}}(\mathbf{r}_{t+1:T}) \Big]^{-1} \\ &\qquad = L_{t+1}^{-1}(\mathbf{r}_{t+1:T}) - L_{t+1}^{-1}(\mathbf{r}_{t+1:T}) M_{t+1}(\mathbf{r}_{t+1:T}) \\ &\qquad\quad \times R_{t\,|\,t}(\mathbf{r}_{1:t}) M_{t+1}^{\mathrm{T}}(\mathbf{r}_{t+1:T}) L_{t+1}^{-1}(\mathbf{r}_{t+1:T}) \end{aligned}$$

where $R_{t\,|\,t}(\mathbf{r}_{1:t})$ is given by (10). Therefore

$$\begin{aligned} &(\mathbf{y}_{t+1:T} - M_{t+1}(\mathbf{r}_{t+1:T}) m_{t\,|\,t}(\mathbf{r}_{1:t}))^{\mathrm{T}} \Big[L_{t+1}(\mathbf{r}_{t+1:T}) \\ &\quad + M_{t+1}(\mathbf{r}_{t+1:T}) P_{t\,|\,t}(\mathbf{r}_{1:t}) M_{t+1}^{\mathrm{T}}(\mathbf{r}_{t+1:T}) \Big]^{-1} \\ &\quad \times (\mathbf{y}_{t+1:T} - M_{t+1}(\mathbf{r}_{t+1:T}) m_{t\,|\,t}(\mathbf{r}_{1:t})) \\ &= \mathbf{y}_{t+1:T}^{\mathrm{T}} L_{t+1}^{-1}(\mathbf{r}_{t+1:T}) \mathbf{y}_{t+1:T} + m_{t\,|\,t}^{\mathrm{T}}(\mathbf{r}_{1:t}) \\ &\quad \times P_{t\,|\,t+1}'^{-1}(\mathbf{r}_{t+1:T}) m_{t\,|\,t}(\mathbf{r}_{1:t}) \\ &\quad - 2 m_{t\,|\,t}^{\mathrm{T}}(\mathbf{r}_{1:t}) P_{t\,|\,t+1}'^{-1}(\mathbf{r}_{t+1:T}) m_{t\,|\,t+1}'(\mathbf{r}_{t+1:T}) \\ &\quad - (m_{t\,|\,t+1}'(\mathbf{r}_{t+1:T}) - m_{t\,|\,t}(\mathbf{r}_{1:t}))^{\mathrm{T}} P_{t\,|\,t+1}'^{-1}(\mathbf{r}_{t+1:T}) \\ &\quad \times R_{t\,|\,t}(\mathbf{r}_{1:t}) P_{t\,|\,t+1}'^{-1}(\mathbf{r}_{t+1:T}) \\ &\quad \times (m_{t\,|\,t+1}'(\mathbf{r}_{t+1:T}) - m_{t\,|\,t}(\mathbf{r}_{1:t})). \end{aligned}$$

Equation (9) follows as $\mathbf{y}_{t+1:T}^{\mathrm{T}} L_t^{-1}(\mathbf{r}_{t+1:T}) \mathbf{y}_{t+1:T}$ and $|L_t(\mathbf{r}_{t+1:T})|$ do not depend on r_t.

C. Proof of Proposition 2

We have $p(x_t \,|\, \mathbf{y}_{1:T}, \mathbf{x}_{-t}) \propto p(\mathbf{y}_{1:T}, \mathbf{x}_{-t}, x_t)$ where $p(\mathbf{y}_{1:T}, \mathbf{x}_{-t}, x_t)$ is given by (11) and

$$\begin{aligned} &p(\mathbf{y}_{t+1:T}, \mathbf{x}_{t+1:T} \,|\, r_t = i, x_t) \\ &\quad = \sum_{j=1}^{s} p_{ij} p(\mathbf{y}_{t+2:T}, \mathbf{x}_{t+2:T} \,|\, r_{t+1} = j, x_{t+1}) \\ &\qquad \times p(y_{t+1}, x_{t+1} \,|\, r_{t+1} = j, x_t) \end{aligned}$$

Therefore, we have the equation at the top of the next page,

$$
\begin{aligned}
p(\mathbf{y}_{1:T}, \mathbf{x}_{0:T}) &= p(\mathbf{y}_{1:t-1}, \mathbf{x}_{0:t-1}) p(y_t, x_t \,|\, \mathbf{y}_{1:t-1}, \mathbf{x}_{0:t-1}) \\
&\quad \times \sum_{i=1}^{s} \left(\sum_{j=1}^{s} p_{ij} p(\mathbf{y}_{t+2:T}, \mathbf{x}_{t+2:T} \,|\, r_{t+1} = j, x_{t+1}) p(y_{t+1}, x_{t+1} \,|\, r_{t+1} = j, x_{t+1}) \right) p(r_t = i \,|\, \mathbf{y}_{1:t}, \mathbf{x}_{0:t}) \\
&\propto \sum_{i=1}^{s} \sum_{j=1}^{s} p(r_t = i \,|\, \mathbf{y}_{1:t-1}, \mathbf{x}_{0:t-1}) p_{ij} \\
&\quad \times p(\mathbf{y}_{t+2:T}, \mathbf{x}_{t+2:T} \,|\, r_{t+1} = j, x_{t+1}) p(y_t, x_t \,|\, r_t = i, x_{t-1}) \\
&\quad \times p(y_{t+1}, x_{t+1} \,|\, r_{t+1} = j, x_t)
\end{aligned}
$$

---

where

$$
\begin{aligned}
&p(y_t, x_t \,|\, r_t = i, x_{t-1}) p(y_{t+1}, x_{t+1} \,|\, r_{t+1} = j, x_t) \\
&\quad = p(y_t \,|\, r_t = i, x_t) p(x_t \,|\, r_t = i, x_{t-1}) \\
&\qquad \times p(y_{t+1} \,|\, r_{t+1} = j, x_{t+1}) \\
&\qquad \times p(x_{t+1} \,|\, r_{t+1} = j, x_t) \\
&\quad \propto N(y_t - C(i)x_t - G(i)u_t, D(i)D^{\mathrm{T}}(i)) \\
&\qquad \times N(x_t - A(i)x_{t-1} - F(i)u_t, B(i)B^{\mathrm{T}}(i)) \\
&\qquad \times N(y_{t+1} - C(j)x_{t+1} - G(j)u_{t+1}, D(j)D^{\mathrm{T}}(j)) \\
&\qquad \times N(x_{t+1} - A(j)x_t - F(j)u_{t+1}, B(j)B^{\mathrm{T}}(j)).
\end{aligned}
$$

By identifying a quadratic form of argument x_t, we obtain (17) after a few calculations.

ACKNOWLEDGMENT

The authors are grateful to A. Ahmed for pointing out and solving a problem in a preliminary version of this paper.

REFERENCES

[1] B. D. O. Anderson and J. B. Moore, *Optimal Filtering*. Englewood Cliffs, NJ: Prentice-Hall, 1979.

[2] Y. Bar-Shalom and X. R. Li, *Multitarget-Multisensor Tracking: Principles and Techniques*. Storrs, CT: Univ. of Connecticut Press, 1995.

[3] V. A. Bukhalev, "Optimal smoothing in systems with random jump structure," *Automat. Remote Contr.*, no. 6, pp. 46–56, 1992.

[4] O. Cappé, A. Doucet, É. Moulines, and M. Lavielle, "Simulation-based methods for blind maximum-likelihood filter identification," *Signal Process.*, vol. 73, no. 1, pp. 3–25, 1999.

[5] C. K. Carter and R. Kohn, "On Gibbs sampling for state space models," *Biometrika*, vol. 81, no. 3, pp. 541–553, 1994.

[6] ——, "Markov chain Monte Carlo methods in conditionally Gaussian state space models," *Biometrika*, vol. 83, pp. 589–601, 1996.

[7] F. Champagnat, Y. Goussard, and J. Idier, "Unsupervised deconvolution of sparse spike trains using stochastic approximation," *IEEE Trans. Signal Processing*, vol. 44, pp. 2988–2998, Dec. 1996.

[8] P. DeJong and N. Shephard, "The simulation smoother for time series models," *Biometrika*, vol. 82, no. 2, pp. 339–350, 1995.

[9] J. Diebolt and C. P. Robert, "Estimation of finite mixture distributions through Bayesian sampling," *J. R. Statist. Soc. B*, vol. 56, pp. 363–375, 1994.

[10] A. Doucet and P. Duvaut, "Bayesian estimation of state space models applied to deconvolution of Bernoulli-Gaussian processes," *Signal Process.*, vol. 57, pp. 147–161, 1997.

[11] A. Doucet, A. Logothetis, and V. Krishnamurthy, "Stochastic sampling algorithms for state estimation of jump Markov linear systems," *IEEE Trans. Automat. Contr.*, vol. 45, pp. 188–201, Feb. 2000.

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

[13] A. Doucet, N. J. Gordon, and V. Krishnamurthy, Particle filters for state estimation of jump Markov linear systems, in *IEEE Trans. Signal Processing*, vol. 49, pp. 613–624, Mar. 2001.

[14] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distribution and the Bayesian restoration of images," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-6, pp. 721–741, July 1984.

[15] R. E. Helmick, W. D. Blair, and S. A. Hoffman, "Fixed-interval smoothing for Markovian switching systems," *IEEE Trans. Inform. Theory*, vol. 41, pp. 1845–1855, Nov. 1995.

[16] K. F. Kaaresen, "Deconvolution of sparse spike trains by iterated window maximization," *IEEE Trans. Signal Processing*, vol. 45, pp. 1173–1183, May 1997.

[17] A. H. Jazwinski, *Stochastic Processes and Filtering Theory*. New York: Academic, 1970.

[18] G. Kitagawa, "The two-filter formula for smoothing and an implementation of the Gaussian-sum smoother," *Ann. Inst. Stat. Math.*, vol. 46, pp. 605–623, 1994.

[19] J. Kormylo and J. M. Mendel, "Maximum-likelihood detection and estimation of Bernoulli-Gaussian processes," *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 482–488, 1982.

[20] V. Krishnamurthy and A. Logothetis, "Adaptive nonlinear filters for narrowband interference suppression in spread spectrum CDMA," *IEEE Trans. Commun.*, vol. 47, pp. 742–753, May 1999.

[21] M. Lavielle, "Bayesian deconvolution of Bernoulli-Gaussian processes," *Signal Process.*, vol. 33, pp. 67–79, 1993.

[22] J. S. Liu, W. H. Wong, and A. Kong, "Covariance structure of the Gibbs sampler with applications to the comparisons of estimators and augmentation schemes," *Biometrika*, vol. 81, pp. 27–40, 1994.

[23] A. Logothetis and V. Krishnamurthy, "A Bayesian expectation-maximization framework for estimating jump Markov linear systems," *IEEE Trans. Signal Processing*, vol. 47, pp. 2139–2156, Aug. 1999.

[24] G. J. McLachlan and T. Krishnan, *The EM Algorithm and Extensions*. New York: Wiley, 1996.

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

[26] J. M. Mendel, *Maximum-Likelihood Deconvolution: A Journey into Model-Based Signal Processing*. New York: Springer-Verlag, 1990.

[27] G. W. Pulford and R. J. Evans, "Probabilistic data association for systems with multiple simultaneous measurements," *Automatica*, vol. 32, no. 9, pp. 1311–1316, 1996.

[28] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, pp. 257–285, Feb. 1989.

[29] C. P. Robert and G. Cassela, *Monte Carlo Statistical Methods*. New York: Springer-Verlag, 1999.

[30] I. Santamaria-Caballero, C. J. Pantaleón-Prieto, and A. Artés-Rodriguez, "Sparse deconvolution using adaptive mixed—Gaussian models," *Signal Process.*, vol. 54, pp. 161–172, 1996.

[31] D. M. Titterington, A. F. M. Smith, and U. E. Makov, *Statistical Analysis of Finite Mixture Distributions*. New York: Wiley, 1985.

[32] J. K. Tugnait, "Adaptive estimation and identification for discrete systems with Markov jump parameters," *IEEE Trans. Automat. Contr.*, vol. AC-25, pp. 1054–1065, 1982.

[33] M. West and J. Harrison, *Bayesian Forecasting and Dynamic Models*, 2nd ed. New York: Springer, 1997.

Arnaud Doucet was born in Melle, France, on November 2, 1970. He received the M.S. degree from Institut National des Télécommunications, Paris, France, in 1993 and the Ph.D. degree from University of Paris XI, Orsay, France, in 1997.

In 1998, he was a visiting scholar with the Signal Processing Group, Cambridge University, Cambridge, U.K., supported by a postdoctoral grant from the French Ministry of Defence. From 1999 to February 2001, he was a Research Associate with the same group. He is now a Lecturer with the Department of Electrical and Electronic Engineering, Melbourne University, Parkville, Australia. His research interests include Bayesian estimation, Markov chain Monte Carlo methods, and sequential Monte Carlo methods. He has co-edited, with J. F. G. de Freitas and N. J. Gordon, *Sequential Monte Carlo Methods in Practice* (New York: Springer-Verlag).

Christophe Andrieu was born in France in 1968. He received the M.S. degree from Institut National des Télécommunications, Paris, France, in 1993 and the D.E.A. degree in 1994 and the Ph.D. degree in 1998 from the University of Paris XV. From 1998 to September 2000, he was a research associate with the Signal Processing Group, Cambridge University, Cambridge, U.K. He is now a lecturer with the Department of Mathematics, Bristol University, Bristols, U.K. His research interests include spectral analysis, source separation, Markov chain Monte Carlo methods, and stochastic optimization.