Efficient Particle Filtering for Jump Markov Systems. Application to Time-Varying Autoregressions

Christophe Andrieu, Manuel Davy, and Arnaud Doucet

Abstract—**In this paper, we present an efficient particle filtering method to perform optimal estimation in jump Markov (nonlinear) systems (JMSs). Such processes consist of a mixture of heterogeneous models and possess a natural hierarchical structure. We take advantage of these specificities in order to develop a generic filtering methodology for these models. The method relies on an original and nontrivial combination of techniques that have been presented recently in the filtering literature, namely, the auxiliary particle filter and the unscented transform. This algorithm is applied to the complex problem of time-varying autoregressive estimation with an unknown time-varying model order. More precisely, we develop an attractive and original probabilistic model that relies on a flexible pole representation that easily lends itself to interpretations. We show that this problem can be formulated as a JMS and that the associated filtering problem can be efficiently addressed using the generic methodology developed in this paper. Simulations demonstrate the performance of our method compared to standard particle filtering techniques.**

I. Introduction

JUMP Markov systems (JMSs) are a very important class of models appearing in signal processing, target tracking, and econometrics, among others [2], [14]. Different from standard (continuous state space) hidden Markov models, JMSs combine hierarchically discrete/continuous state spaces in the following way. Let $\{r_t\}$ $(t \geq 1)$ be a stationary, finite, discrete, first-order homogeneous Markov chain taking its values in a set S, with transition probabilities

$$\pi_{ij} \triangleq \Pr\{r_{t+1} = j | r_t = i\}, \qquad (i, j \in S). \tag{1}$$

We define s as the finite number of elements of S. Now, consider a family of s^2 densities $\{f_{ij}(x'|x)\}$, where $x \in \mathbb{R}^{n_x}$ and $x' \in \mathbb{R}^{n_{x'}}$, and define the state transition conditional densities

$$p(x_t | x_{0:t-1}, r_{1:t}) = f_{r_{t-1}r_t}(x_t | x_{t-1}) \tag{2}$$

where, for a set of variables l_t, we denote $l_{a:b} \triangleq \{l_a, l_{a+1}, \ldots, l_b\}$. The initial state x_0 is distributed according to a distribution p_0. Note that *the dimension, or nature, of x_t might be a function of the sequence $\{r_t\}$*, but we do not make this dependence explicit in order to alleviate notation. Neither the process $\{r_t\}$ nor $\{x_t\}$ are observed. Instead, we observe $\{y_t\}$ $(t \geq 1)$, where

$$p(y_t | x_{0:t}, r_{1:t}, y_{1:t-1}) = g_{r_t}(y_t | y_{1:t-1}, x_t) \tag{3}$$

with $y_t \in \mathbb{R}^{n_{y_t}}$ (the number of observations can vary over time). It is possible to add exogenous variables $\{u_t\}$ in the equations, i.e., $\{f_{ij}\}$ and $\{g_j\}$ can also depend on u_t, but we omit them to simplify notation. Note that the above model could be written as a standard hidden Markov model with mixed continuous/discrete state (x_t, r_t), but we prefer this presentation as it emphasizes the hierarchical structure of the model and motivates our methodology.

The class of processes under study here is a generalization of the jump Markov linear models (JMLS) considered in [5] and [14]. In the JMLS case, $f_{r_{t-1}r_t}(x_t|x_{t-1})$ [resp. $g_{r_t}(y_t|x_t, y_{1:t-1})$] are Gaussian and linear in x_{t-1} (resp. in x_t), i.e., one has

$$\begin{aligned} x_t &= A(r_t)x_{t-1} + B(r_t)v_t, \qquad x_0 \sim \mathcal{N}(m_0, P_0) \\ y_t &= C(r_t)x_t + D(r_t)\varepsilon_t \end{aligned}$$

where $v_t \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, I_{n_v})$ and $\varepsilon_t \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, I_{n_\varepsilon})$ are mutually independent sequences of independent and identically distributed (i.i.d.) Gaussian random variables. In this context, Monte Carlo methods combined with Kalman filtering techniques can be used [5]. However, it is of great practical interest to relax these linearity and Gaussianity assumptions, which are unrealistic in many real-world applications. As we will see in Section III, time-varying autoregressions (TVARs) with time-varying model order enter the class of JMS, but other examples include *bearings-only tracking for a maneuvering source* [2], where the state x_t consists of the location and velocity of the target and where $\{r_t\}$ is a given maneuver and *multitarget tracking in clutter noise* [2], where r_t is the time-varying number of targets, and the state $x_t = (x_{1,t}, \ldots, x_{r_t,t})$ consists of the aggregation of each state target $x_{i,t}$, $i = 1, \ldots, r_t$.

A. Estimation Objectives

The aim of optimal filtering is to estimate *sequentially in time* the unknown "hidden" states $\{x_t, r_t\}$ and, more precisely, the series of posterior distributions $p(x_{0:t}, r_{1:t}|y_{1:t})$. Their marginals, and in particular the filtering densities $p(x_t, r_t|y_{1:t})$, are of interest in practice. A simple application of Bayes' rule allows for an easy formulation of the recursion that updates $p(x_{0:t-1}, r_{1:t-1}|y_{1:t-1})$ to $p(x_{0:t}, r_{1:t}|y_{1:t})$:

$$p(x_{0:t}, r_{1:t}|y_{1:t}) = p(x_{0:t-1}, r_{1:t-1}|y_{1:t-1}) \times \frac{g_{r_t}(y_t|y_{1:t-1}, x_t) f_{r_{t-1}r_t}(x_t|x_{t-1}) \pi_{r_{t-1}r_t}}{p(y_t|y_{1:t-1})}$$

Manuscript received March 27, 2002; revised November 7, 2002. M. Davy was supported by the European Project MOUMIR (http://www.moumir.org). The associate editor coordinating the review of this paper and approving it for publication was Dr. Naofal M. W. Al-Dhahir.

C. Andrieu is with the Statistics Group, Department of Mathematics, University of Bristol, Bristol, U.K. (e-mail: c.andrieu@bris.ac.uk).

M. Davy is with IRCCyN/CNRS, École Centrale de Nantes, Nantes, France, and with the Signal Processing Group, Cambridge University Engineering Department, Cambridge, U.K. (e-mail: Manuel.Davy@irccyn.ec-nantes.fr).

A. Doucet is with the Cambridge University Engineering Department, Signal Processing Group, Cambridge, U.K. (e-mail: ad2@eng.cam.ac.uk).

Digital Object Identifier 10.1109/TSP.2003.810284

where

$$p(y_t|y_{1:t-1}) = \sum_{r_{t-1},\, r_t \in S} \pi_{r_{t-1}r_t} \int g_{r_t}(y_t|y_{1:t-1},\, x_t) \times f_{r_{t-1}r_t}(x_t|x_{t-1}) p(x_{t-1},\, r_{t-1}|y_{1:t-1})\, dx_{t-1:t}.$$

There is no closed-form solution to this recursion and for state estimates of the form

$$\mathbb{E}[\phi(x_{0:t},\, r_{1:t})|y_{1:t}] \triangleq \int \sum_{r_{1:t}} \phi(x_{0:t},\, r_{1:t}) p(x_{0:t},\, r_{1:t}|y_{1:t})\, dx_{0:t}$$

which include the minimum mean square estimate (MMSE) of the state $\mathbb{E}[x_t|y_{1:t}]$ and its covariance, for example. To simplify notation, finite sums will be replaced further on by integrals whenever it is convenient.

B. Resolution and Organization of the Paper

We propose here to approximate $p(x_{0:t},\, r_{1:t}|y_{1:t})$ using particle filtering methods. The key idea of particle filtering is to use an adaptive stochastic grid approximation of the posterior distribution of the state vector with $N \gg 1$ weighted particles (values of the grid) evolving randomly in time according to a simulation-based rule, that is, the density is approximated by a weighted mixture of points

$$p\left(dx_{0:t},\, r_{1:t}|y_{1:t}\right) \approx \sum_{i=1}^{N} w_t^{(i)}\, \delta_{x_{0:t}^{(i)}}(dx_{0:t}) \mathbb{I}_{\{r_{1:t}^{(i)}\}}(r_{1:t})$$

$$\sum_{i=1}^{N} w_t^{(i)} = 1, \qquad w_t^{(i)} \geq 0 \tag{4}$$

so that, for example

$$\mathbb{E}[\phi(x_{0:t},\, r_{1:t})|y_{1:t}] \approx \sum_{i=1}^{N} w_t^{(i)} \phi\left(x_{0:t}^{(i)},\, r_{1:t}^{(i)}\right).$$

We will further denote $\delta_{x_0,\, r_0}(dx,\, r) = \mathbb{I}_{\{r_0\}}(r)\, \delta_{x_0}(dx)$, where $\mathbb{I}_{r_0}(r)$ is the indicator function such that $\mathbb{I}_{r_0}(r) = 1$ if $r = r_0$ and $\mathbb{I}_{r_0}(r) = 0$ otherwise. The adaptive algorithm is designed such that the concentration of particles in a given region of the state space, say A, represents the probability of A under the posterior distribution, i.e., $\int_A p(x_{0:t},\, r_{1:t}|y_{1:t})\, dx_{0:t}\, dr_{1:t}$. Therefore, computational efforts focus on different zones of the state space according to their importance, resulting in efficient algorithms. The particles evolve with time in a series of growing spaces and can either give birth to offspring particles or die, depending on their ability to represent the different characteristics of interest of the posterior distributions, which are dictated by the observation process and the dynamics of the underlying system.

We propose here to develop a generic approach in order to design efficient particle filtering techniques adapted to the class of JMS described earlier. Our approach is an original and nontrivial combination of several methods that have been recently proposed in the literature, mainly, the auxiliary particle filter (APF) [16] and a suboptimal deterministic filtering method known as the unscented Kalman filter (UKF), which is a particular instance of the unscented transform (UT) [11], [12]. Our work essentially differs from other works on APF in that we propose an original implementation of the APF idea that relies on the UKF. As opposed to previous works, our combination enables both the derivation of an efficient importance sampling density and an accurate estimation of the auxiliary weights without additional computational load.

We apply our methodology to nonstationary signal detection and estimation, using flexible time-varying autoregressions (TVAR). We adopt here a pole representation of the autoregressive process that allows for the specification of intuitive priors (e.g., smoothness or abrupt changes of spectral components) and natural interpretations. Furthermore, a direct and naive update of the number of AR coefficients can cause unpredictable and unwanted changes on the existing spectral components. Our approach, on the contrary, allows for additions or removals of spectral components that have limited effect on existing components, as opposed to the models considered in, e.g., [17] and [19]. However, this choice of parameterization introduces many nonlinearities and is made complex by the fact that the number of poles is unknown and might evolve with time.

The paper is organized as follows. In Section II, we briefly review the basic principles of particle filtering techniques, which are essential for understanding the further developments that lead to our generic algorithm that take advantage of the JMS structure. In Section III, we introduce the TVAR problem, and performance of the procedure is demonstrated on synthetic signals. A discussion is given in Section IV.

II. Particle Filtering for JMS

A. Sequential Importance Sampling and Resampling

We briefly describe here how to apply the sequential importance sampling resampling (SISR) method in order to approximately sample from $p(x_{0:t},\, r_{1:t}|y_{1:t})$; see [6] for further details.

At time $t-1$, assume we have, say, N weighted particles $\{x_{0:t-1}^{(i)},\, r_{1:t-1}^{(i)}\}$ associated with the weights $\{w_{t-1}^{(i)}\}$ such that

$$p\left(dx_{0:t-1},\, r_{1:t-1}|y_{1:t-1}\right) \approx \sum_{i=1}^{N} w_{t-1}^{(i)}\, \delta_{x_{0:t-1}^{(i)},\, r_{1:t-1}^{(i)}}\left(dx_{0:t-1},\, r_{1:t-1}\right).$$

We want to obtain N particles $\{x_{0:t}^{(i)},\, r_{1:t}^{(i)}\}$ distributed according to $p(x_{0:t},\, r_{1:t}|y_{1:t})$. At time t, we extend each particle $x_{0:t-1}^{(i)},\, r_{1:t-1}^{(i)}$ by sampling $\tilde{x}_t^{(i)},\, \tilde{r}_t^{(i)}$ according to a conditional distribution $q(x_t,\, r_t|x_{0:t-1}^{(i)},\, r_{1:t-1}^{(i)},\, y_{1:t})$ to obtain N new particles $\{\tilde{x}_{0:t}^{(i)},\, \tilde{r}_{1:t}^{(i)}\}$, where $\tilde{x}_{0:t}^{(i)} \triangleq (x_{0:t-1}^{(i)},\, \tilde{x}_t^{(i)})$, and $\tilde{r}_{1:t}^{(i)} \triangleq (r_{1:t-1}^{(i)},\, \tilde{r}_t^{(i)})$. To correct for the discrepancy between the distribution of the particles $\tilde{x}_{0:t}^{(i)},\, \tilde{r}_{1:t}^{(i)}$ and $p(x_{0:t},\, r_{1:t}|y_{1:t})$, we use importance sampling so that $p(x_{0:t},\, r_{1:t}|y_{1:t})$ is approximated by the empirical distribution

$$\hat{p}_N\left(dx_{0:t},\, r_{1:t}|y_{1:t}\right) = \sum_{i=1}^{N} w_t^{(i)}\, \delta_{\tilde{x}_{0:t}^{(i)},\, \tilde{r}_{1:t}^{(i)}}\left(dx_{0:t},\, r_{1:t}\right) \tag{5}$$

where the importance weights satisfy

$$w_t^{(i)} \propto w_{t-1}^{(i)} \times \frac{p\left(\tilde{x}_{0:t}^{(i)}, \tilde{r}_{1:t}^{(i)} \middle| y_{1:t}\right)}{p\left(\tilde{x}_{0:t-1}^{(i)}, \tilde{r}_{1:t-1}^{(i)} \middle| y_{1:t-1}\right) q\left(\tilde{x}_t^{(i)}, \tilde{r}_t^{(i)} \middle| \tilde{x}_{0:t-1}^{(i)}, \tilde{r}_{1:t-1}^{(i)}, y_{1:t}\right)}$$
$$\propto w_{t-1}^{(i)} \frac{g_{\tilde{r}_t^{(i)}}\left(y_t \middle| y_{1:t-1}, \tilde{x}_t^{(i)}\right) f_{\tilde{r}_{t-1}^{(i)} \tilde{r}_t^{(i)}}\left(\tilde{x}_t^{(i)} \middle| \tilde{x}_{t-1}^{(i)}\right) \pi_{\tilde{r}_{t-1}^{(i)} \tilde{r}_t^{(i)}}}{q\left(\tilde{x}_t^{(i)}, \tilde{r}_t^{(i)} \middle| \tilde{x}_{0:t-1}^{(i)}, \tilde{r}_{1:t-1}^{(i)}, y_{1:t}\right)}. \tag{6}$$

The performance of the algorithm depends on the importance density $q(x_t, r_t | x_{0:t-1}, r_{1:t-1}, y_{1:t})$. The "optimal" importance density, that is, the density minimizing the conditional variance of the weights conditional upon $y_{1:t-1}$, is [4]

$$p(x_t, r_t | x_{0:t-1}, r_{1:t-1}, y_{1:t}) \propto g_{r_t}(y_t | y_{1:t-1}, x_t) f_{r_{t-1} r_t}(x_t | x_{t-1}) \pi_{r_{t-1} r_t}$$

and the associated importance weight is proportional to the predictive likelihood

$$w_t \propto w_{t-1} p(y_t | y_{1:t-1}, x_{t-1}, r_{t-1}) \tag{7}$$

where

$$p(y_t | y_{1:t-1}, x_{t-1}, r_{t-1}) = \sum_{r_t \in S} \pi_{r_{t-1} r_t} \int g_{r_t}(y_t | y_{1:t-1}, x) f_{r_{t-1} r_t}(x_t | x_{t-1}) \, dx_t.$$

This scenario is referred to as "full adaption" in [16].

Finally, one obtains N particles $\{x_{0:t}^{(i)}, r_{1:t}^{(i)}\}$ approximately distributed according to $p(x_{0:t}, r_{1:t} | y_{1:t})$ by resampling/selection from the weighted empirical distribution given in (5). There are several resampling procedures available in the literature. We adopt here the stratified sampling scheme described in [15].

This "optimal" importance sampling case deserves special attention. Indeed, the importance weights w_t given by (7) do not actually depend on (x_t, r_t). This means that resampling/selection can be performed before extending trajectories, thus selecting the most promising trajectories before extension. However, in most practical cases, it is impossible to use the "optimal" importance sampling density as the predictive likelihoods of particles [see (7)] do not admit a closed-form expression. However, this scenario motivates an alternative particle filtering method known as APF [16] (see Section II-B1), where one analytically approximates the predictive likelihoods, or its behavior, whenever necessary.

B. Strategies for Efficient Particle Filtering

1) Auxiliary Particle Filter: The idea behind APF is, at time t, to extend existing trajectories $\{x_{0:t-1}^{(i)}, r_{1:t-1}^{(i)}\}$ that are the most promising in the sense that their predictive likelihoods $p(y_t | y_{1:t-1}, x_{t-1}^{(i)}, r_{t-1}^{(i)})$ are large. However, the analytical computation of these predictive likelihoods might prove to be intractable, and approximation is needed. Recall that

$$p\left(y_t \middle| y_{1:t-1}, x_{t-1}^{(i)}, r_{t-1}^{(i)}\right) = \sum_{r_t \in S} \pi_{r_{t-1}^{(i)} r_t} \int g_{r_t}(y_t | y_{1:t-1}, x_t) f_{r_{t-1}^{(i)} r_t}\left(x_t \middle| x_{t-1}^{(i)}\right) dx_t.$$

In [16], the authors propose (in the nonswitching case) simple approximations where

$$\int g_{r_t}(y_t | x_t, y_{1:t-1}) f_{r_{t-1}^{(i)} r_t}\left(x_t \middle| x_{t-1}^{(i)}\right) dx_t \approx g_{r_t}\left(y_t \middle| y_{1:t-1}, \zeta\left(x_{t-1}^{(i)}, r_t\right)\right)$$

where $\zeta(x_{t-1}^{(i)}, r_t)$ is the mode or mean of $f_{r_{t-1}^{(i)} r_t}(x_t | x_{t-1}^{(i)})$. In many applications, especially if $g_{r_t}(y_t | y_{1:t-1}, x_t)$ varies significantly over the significant regions of $f_{r_{t-1}^{(i)} r_t}(x_t | \tilde{x}_{t-1}^{(i)})$, then the approximation of the predictive likelihood can be very poor and lead to performance far below that of the SISR algorithm. Indeed, one ends up biasing the exploration of the space toward uninteresting regions. It is thus fundamental to be able to properly approximate the predictive likelihood. An obvious solution would consist of using a second-stage Monte Carlo method for each particle. It is, however, too computationally intensive and introduces further Monte Carlo variation. Both the inaccuracy and computational problems have been overlooked in previous studies involving APF. In order to overcome these problems, we propose here an efficient solution based on a deterministic approximation of the predictive likelihood integral with

$$\int g_{r_t}(y_t | y_{1:t-1}, x_t) f_{r_{t-1}^{(i)} r_t}\left(x_t \middle| x_{t-1}^{(i)}\right) dx_t \approx \psi_{r_t}\left(y_{1:t}, x_{t-1}^{(i)}, r_{t-1}^{(i)}\right)$$

where $\psi_{r_t}(x_{t-1}^{(i)}, r_{t-1}^{(i)})$ is a deterministic mapping/integration technique (we omit the observations $y_{1:t}$ in $\psi_{r_t}(\cdot)$ to simplify notation). The interest of our approach is that it does not require additional computations since it will also be used to form the importance distribution $q(x_t, r_t | x_{0:t-1}^{(i)}, r_{1:t-1}^{(i)}, y_{1:t})$, as explained in the next subsection. We propose to estimate the desired quantity as

$$\hat{p}\left(y_t \middle| y_{1:t-1}, x_{t-1}^{(i)}, r_{t-1}^{(i)}\right) = \sum_{r_t \in S} \pi_{r_{t-1}^{(i)} r_t} \psi_{r_t}\left(x_{t-1}^{(i)}, r_{t-1}^{(i)}\right). \tag{8}$$

The SISR extends each particle $x_{0:t-1}^{(i)}, r_{1:t-1}^{(i)}$ with weight $w_{t-1}^{(i)}$ by sampling $\tilde{x}_t^{(i)}, \tilde{r}_t^{(i)}$ according to a conditional distribution $q(x_t, r_t | x_{0:t-1}^{(i)}, r_{1:t-1}^{(i)}, y_{1:t})$ to obtain $(x_{0:t-1}^{(i)}, \tilde{x}_t^{(i)})$ and $(r_{1:t-1}^{(i)}, \tilde{r}_t^{(i)})$; then, the associated weight satisfies (6) and can be rewritten as

$$w_t^{(i)} \propto w_{t-1}^{(i)} \hat{p}\left(y_t \middle| y_{1:t-1}, x_{t-1}^{(i)}, r_{t-1}^{(i)}\right) \times \frac{g_{\tilde{r}_t^{(i)}}\left(y_t \middle| y_{1:t-1}, \tilde{x}_t^{(i)}\right) f_{r_{t-1}^{(i)} \tilde{r}_t^{(i)}}\left(\tilde{x}_t^{(i)} \middle| x_{t-1}^{(i)}\right) \pi_{r_{t-1}^{(i)} \tilde{r}_t^{(i)}}}{\hat{p}\left(y_t \middle| y_{1:t-1}, x_{t-1}^{(i)}, r_{t-1}^{(i)}\right) q\left(\tilde{x}_t^{(i)}, \tilde{r}_t^{(i)} \middle| x_{0:t-1}^{(i)}, r_{1:t-1}^{(i)}, y_{1:t}\right)}. \tag{9}$$

The interest of this decomposition is that the term $w^{(i)}_{t-1}\hat{p}(y_t|y_{1:t-1}, x^{(i)}_{t-1}, r^{(i)}_{t-1})$, which is independent of $\tilde{r}^{(i)}_t$, $\tilde{x}^{(i)}_t$, mimics the weight of the "full adaption" scenario described earlier. It therefore suggests the possibility of resampling $\{x^{(i)}_{0:t-1}, r^{(i)}_{1:t-1}\}$ according to the weights $\lambda^{(i)}_t \propto w^{(i)}_{t-1}\hat{p}(y_t|y_{1:t-1}, x^{(i)}_{t-1}, r^{(i)}_{t-1})$ in order to obtain N particles $\{\tilde{x}^{(i)}_{0:t-1}, \tilde{r}^{(i)}_{1:t-1}\}$, which are then approximately sampled from a distribution "close" to $p(x_{0:t-1}, r_{1:t-1}|y_{1:t})$. We then extend each particle by sampling $\tilde{x}^{(i)}_t$, $\tilde{r}^{(i)}_t$ according to a conditional distribution $q(x_t, r_t|\tilde{x}^{(i)}_{0:t-1}, \tilde{r}^{(i)}_{1:t-1}, y_{1:t})$. Contrary to the full adaption case, it is, however, necessary to re-weight the particles by

$$w^{(i)}_t \propto \frac{g_{\tilde{r}^{(i)}_t}\left(y_t \middle| y_{1:t-1}, \tilde{x}^{(i)}_t\right) f_{\tilde{r}^{(i)}_{t-1}\tilde{r}^{(i)}_t}\left(\tilde{x}^{(i)}_t \middle| \tilde{x}^{(i)}_{t-1}\right) \pi_{\tilde{r}^{(i)}_{t-1}\tilde{r}^{(i)}_t}}{\hat{p}\left(y_t \middle| y_{1:t-1}, \tilde{x}^{(i)}_{t-1}, \tilde{r}^{(i)}_{t-1}\right) q\left(\tilde{x}^{(i)}_t, \tilde{r}^{(i)}_t \middle| \tilde{x}^{(i)}_{0:t-1}, \tilde{r}^{(i)}_{1:t-1}, y_{1:t}\right)}$$

as the samples $\{\tilde{x}^{(i)}_{0:t-1}, \tilde{r}^{(i)}_{1:t-1}\}$ are no longer distributed according to $p(x_{0:t-1}, r_{1:t-1}|y_{1:t})$ (even as $N \to \infty$).

The problem of constructing an efficient deterministic mapping ψ_r for our problem is the subject of the following subsection.

2) Unscented Transform: The unscented Kalman filter (UKF) is an alternative to the extended Kalman filter (EKF), which possesses many advantages. Both approaches are motivated by the fact that in most cases, a single dynamic model defined by standard (nonjump) Markov systems equations can alternatively be represented in the following manner:

$$x_t = \varphi(x_{t-1}, v_t) \qquad y_t = \gamma(x_t, \varepsilon_t) \tag{10}$$

where $\{v_t\}$ and $\{\varepsilon_t\}$ are typically mutually independent zero-mean i.i.d. sequences, and φ and γ are nonlinearities (similarly, we will introduce φ_{ij} and γ_{ij} for JMS). Both the EKF and UKF rely on approximations of the system defined in (10) but are of a different nature. Nevertheless, for both scenarios, the result of such approximations is that the series of predictive and filtering densities $\{p(x_t|y_{1:t-1})\}$ and $\{p(x_t|y_{1:t})\}$ are replaced with series of Gaussian distributions $\{\mathcal{N}(x_t; m_{t|t-1}, P_{t|t-1})\}$ and $\{\mathcal{N}(x_t; m_{t|t}, P_{t|t})\}$.

The EKF relies on linearizations of the evolution and observation (10), followed with a direct application of the Kalman recursions on the first- and second-order moments. The solution adopted by the UKF is a second-order truncation of the statistics of the posterior distributions at hand, followed by the Kalman recursions. More precisely, assume that a set of n points $\{\overline{x}^{(i)}_{t-1}\}$, known as the "sigma points" [12], possess the correct mean equal to $m_{t-1|t-1}$ and covariance $P_{t-1|t-1}$. Then, the sample mean and autocovariance of the set $\{\varphi(\overline{x}^{(i)}_{t-1}, 0)\}$ should be a good approximation of $m_{t|t-1}$ and $P_{t|t-1}$, respectively. Similarly, the sample mean and autocovariance of $\{\gamma(\varphi(\overline{x}^{(i)}_{t-1}, 0), 0)\}$ should lead to reasonable approximations of $y_{t|t-1} \triangleq \mathbb{E}(\gamma(\varphi(x_{t-1}, v_t), \varepsilon_t)|y_{1:t-1})$ and $cov(y_t|y_{1:t-1})$, which are required for the Kalman filter recursion. Following the same principle, the crosscovariance $cov(x_t, y_t|y_{1:t-1})$ can also be computed. Given these quantities, it is then possible to take into account the new observation y_t and calculate $m_{t|t}$ and $P_{t|t}$ with the Kalman filter. Given values for $m_{t|t}$ and $P_{t|t}$, various methods have been proposed in order to generate a new set $\{\overline{x}^{(i)}_t\}$ [11], [13].

In the context of JMS, the quantity $\psi_{r_t}(x^{(i)}_{t-1}, r^{(i)}_{t-1})$ used to approximate the predictive likelihood for the APF step (see Section II-B1) is computed for every possible new state r_t using the unscented approximation:

$$\psi_{r_t}\left(x^{(i)}_{t-1}, r^{(i)}_{t-1}\right) = \frac{1}{n}\sum_{i=1}^{n} g_{r_t}\left(y_t \middle| y_{1:t-1}, \varphi_{r^{(i)}_{t-1}r_t}\left(\overline{x}^{(i)}_{t-1}, 0\right)\right). \tag{11}$$

In addition to yielding accurate auxiliary weights, the approximation of the predictive likelihood of (8) and (11) also provides an efficient importance distribution for r_t, namely

$$q\left(r_t \middle| y_{1:t}, \tilde{x}^{(i)}_{0:t-1}, \tilde{r}^{(i)}_{0:t-1}\right) \propto \pi_{\tilde{r}^{(i)}_{t-1}r_t}\psi_{r_t}\left(\tilde{x}^{(i)}_{t-1}, \tilde{r}^{(i)}_{t-1}\right).$$

With this importance distribution, states r_t with high predictive distribution (independently on the future state x_t) are proposed with high probability, whereas unlikely states r_t are rarely selected.

C. Algorithm

Based on the elements presented above, it is possible to propose the following generic particle filtering algorithm for JMS, where the notations $\tilde{m}^{(i)}_{t|t}(r_t)$, $\tilde{P}^{(i)}_{t|t}(r_t)$; $r_t \in S$) are introduced to emphasize that these quantities are computed for each possible transition of r_t.

Particle Filter for JMS

```
At time $t = 0$, Step 0: Initialization
• For $i = 1, \ldots, N$ sample $x^{(i)}_0 \sim p(x_0)$ and set the weights $w^{(i)}_0 = 1/N$.
At time $t \geq 1$,
  Step 1: Auxiliary variable resampling step
  • For $i = 1, \ldots, N$, compute $\lambda^{(i)}_t$ as[1]
      $$\lambda^{(i)}_t \propto w^{(i)}_{t-1}\sum_{r_t \in S}\pi_{r^{(i)}_{t-1}r_t}\psi_{r_t}\left(x^{(i)}_{t-1}, r^{(i)}_{t-1}\right), \quad \sum_{i=1}^{N}\lambda^{(i)}_t = 1 \tag{12}$$
    where $\psi_{r_t}(x^{(i)}_{t-1}, r^{(i)}_{t-1})$ is computed as in (11). The same sigma points $\{\overline{x}^{(i)}_{t-1}\}$ are used to compute $m_{t|t}(r_t), P_{t|t}(r_t)$ for all $r_t$ in $S$.
  • Multiply/Discard particles $\{x^{(i)}_{0:t-1}, r^{(i)}_{1:t-1}\}$ and the associated statistics $\{\psi_{r_t}(x^{(i)}_{t-1}, r^{(i)}_{t-1}), m^{(i)}_{t|t}(r_t), P^{(i)}_{t|t}(r_t); r_t \in S\}$ with respect to high/low importance weights $\lambda^{(i)}_t$ to obtain $N$ particles $\{\tilde{x}^{(i)}_{0:t-1}, \tilde{r}^{(i)}_{1:t-1}\}$ and the associated statistics $\{\psi_{r_t}(\tilde{x}^{(i)}_{t-1}, \tilde{r}^{(i)}_{t-1}), \tilde{m}^{(i)}_{t|t}(r_t), \tilde{P}^{(i)}_{t|t}(r_t); r_t \in S\}$.
  Step 2: Importance sampling step
  • For $i = 1, \ldots, N$, sample $\tilde{r}^{(i)}_t \sim q(r_t|y_{1:t}, \tilde{x}^{(i)}_{0:t-1}, \tilde{r}^{(i)}_{1:t-1}) \propto \pi_{\tilde{r}^{(i)}_{t-1}r_t}\psi_{r_t}(\tilde{x}^{(i)}_{t-1}, \tilde{r}^{(i)}_{t-1})$.
```

[1] For $t = 1$, π_{r_{t-1}, r_t} in (12) should be replaced with the stationary distribution of the discrete Markov chain.

```
• For $i = 1, \ldots, N$, extend the trajectories with $\tilde{x}_t^{(i)} \sim \mathcal{N}(x; \tilde{m}_{t|t}^{(i)}(\tilde{r}_t^{(i)}), \tilde{P}_{t|t}^{(i)}(\tilde{r}_t^{(i)}))$.[2]
• Compute the importance weights as
    $w_t^{(i)} \propto \frac{g_{\tilde{r}_t^{(i)}}\left(y_t \middle| \tilde{x}_t^{(i)}\right) f_{\tilde{r}_{t-1}^{(i)} \tilde{r}_t^{(i)}}\left(\tilde{x}_t^{(i)} \middle| \tilde{x}_{t-1}^{(i)}\right)}{\psi_{\tilde{r}_t^{(i)}}\left(\tilde{x}_{t-1}^{(i)}, \tilde{r}_{t-1}^{(i)}\right) \mathcal{N}\left(\tilde{x}_t^{(i)}; \tilde{m}_{t|t}^{(i)}\left(\tilde{r}_t^{(i)}\right), \tilde{P}_{t|t}^{(i)}\left(\tilde{r}_t^{(i)}\right)\right)}.$
• Rename the particles $\{\tilde{x}_{0:t}^{(i)}, \tilde{r}_{1:t}^{(i)}\}$ into $\{x_{0:t}^{(i)}, r_{1:t}^{(i)}\}$.
```

Note that this algorithm could be extended by introducing Markov chain Monte Carlo (MCMC) steps in order to rejuvenate the path of the particles [8]. In the context of JMS, dependent on $r_{1:t}$, the state x_t may lie in subspaces of different dimensions, and thus, reversible jump MCMC should be considered [1], [9].

III. Application to TVAR Estimation

Many applications such as satellite vibration monitoring, gearbox fault detection, or music processing require the development of tools for the analysis of spectral trajectories in the time-frequency plane. Generally, the number of such spectral trajectories is unknown and evolves in time. More importantly, no precise model for the evolution of such trajectories is available. Standard time-frequency techniques [7] yield comprehensive representations of the signal energy content but have several drawbacks that preclude their use for spectral trajectory tracking. On the one hand, representations such as the Wigner–Ville transform yield very good time-frequency resolution but cannot be computed online and introduce interferences, leading to major tracking difficulties. On the other hand, representations such as the spectrogram can be computed online and introduce few cross-terms but are known for their poor resolution. Moreover, it should be added that in either case, an additional postprocessing of the representation is required in order to identify the spectral trajectories and their features (start time, initial frequency, *etc.*). This extra step generally leads to both conceptual (i.e., definition of what a trajectory or a spectral components are) and practical difficulties (i.e., complex algorithms).

Time-varying autoregressive (TVAR) frequency estimation is useful in the present context since it is both flexible and parsimonious. Indeed, the relevant information about the spectral content of the signal being analyzed is summarized with a reduced number of frequency parameters at each time instant as opposed to classical time-frequency representations, which compute energy levels in the entire time-frequency plane. As we will see, this general modeling allows for rather natural and intuitive definitions of trajectories in the time-frequency plane and leads, in practice, to excellent estimation accuracy compared with alternative techniques such as the spectrogram.

A. Problem Description

We are interested in TVAR models, with online estimation of both the model order (denoted K_t) at time t and the TVAR coefficients (denoted $a_{K_t,t} = (a_{1,t}, a_{2,t}, \ldots, a_{K_t,t})^{\mathrm{T}}$ at time t). The observed signal evolution is then described as

$$y_t = a_{1,t} y_{t-1} + a_{2,t} y_{t-2} + \cdots + a_{K_t,t} y_{t-K_t} + v_t \qquad (13)$$

where v_t is a centered Gaussian noise. Equation (13) can be written in vector form as

$$y_t = a_{1:K_t,t}^{\mathrm{T}} y_{t-1:t-K_t} + v_t. \qquad (14)$$

For modeling and practical reasons, we prefer to parameterize the TVAR in terms of its "instantaneous" poles, which are the zeros of the polynomial associated with the autoregressive process at time t

$$\chi_t(x) = 1 - a_{1,t} x - a_{2,t} x^2 - \cdots - a_{K_t,t} x^{K_t}. \qquad (15)$$

The poles can either be complex-conjugate or real-valued, and we need to distinguish between them. We denote $z_{i,t}$, $i = 1, \ldots, k_t^z$ the complex-valued poles (k_t^z is the number of distinct pairs of complex poles conjugate of $z_{i,t}$), and $\eta_{i,t}$ and $i = 1, \ldots, k_t^\eta$ denote the real-valued poles. The complex poles can equivalently be expressed in terms of moduli and "instantaneous" frequencies as

$$z_{i,t} = \rho_{i,t} e^{j 2\pi \nu_{i,t}}, \qquad i = 1, \ldots, k_t^z. \qquad (16)$$

It will be useful to introduce the transformation from the set of poles of χ_t to its coefficients, which we will denote Φ. Finally, the TVAR model is completely described at time t by the distribution of v_t and $\{k_t^\eta, \eta_{1:k_t^\eta,t}, k_t^z, \rho_{1:k_t^z,t}, \nu_{1:k_t^z,t}\}$. We consider the problem of estimating online the TVAR poles, the parameters k_t^η, and k_t^z and hyperparameters. In [10], a simpler model was presented with a fixed number of complex poles and fixed hyperparameters.

B. Model and State Space Representation

Here, the state vector x_t contains the moduli and instantaneous frequencies of the poles but possible hyperparameters α_t as well. The dimension of x_t is therefore time-varying as the number of poles is allowed to evolve with time. The Markov chain $\{r_t\}$ will represent here the number of real and imaginary poles and, therefore, takes its values in $S = \{(i, j); i \in \{0, \ldots, k_{\max}^z\}, j \in \{0, \ldots, k_{\max}^\eta\}\}$. We assume that the transition matrix of this Markov chain is such that $\pi_{(i,j),(i',j')} = 0$ if $|i - i'| > 1$ or $|j - j'| > 1$. This means that the process is allowed to add/remove no more than one real pole or pair of conjugate poles at each time instant. Some further restrictions are needed for the "boundary" values of i and j: When either number of poles is zero, then no such pole can be removed, and similarly, when either maximum number of poles is reached, then no such poles can be added. There is an infinite number of possible specifications for the transition probabilities $\{f_{ij}(x'|x)\}$ that depend on the application at hand. Here, we will assume that the processes are linear Gaussian and aim at introducing some smoothness prior. The observation equation consists of the nonlinear transformation from the pole representation to the coefficients of the polynomial $\chi_t(x)$ combined with past observations; more precisely

$$y_t = \Phi(r_t, x_t) y_{t-1:t-K_t} + v_t.$$

[2] Any other distribution, such as a heavy tailed distribution (e.g., a t-distribution) could also be used.

C. Example

When the dimension of the AR process is fixed, we assume linear models for the evolution of the different poles and, more precisely, order M, moving average (MA) processes

$$\begin{aligned}\nu_{i,t} &= \frac{1}{M}\left(\nu_{i,t-1}+\nu_{i,t-2}+\cdots+\nu_{i,t-M}\right)+u_{\nu_i,t}\\ \rho_{i,t} &= \frac{1}{M}\left(\rho_{i,t-1}+\rho_{i,t-2}+\cdots+\rho_{i,t-M}\right)+u_{\rho_i,t}\\ \eta_{i,t} &= \frac{1}{M}\left(\eta_{i,t-1}+\eta_{i,t-2}+\cdots+\eta_{i,t-M}\right)+u_{\eta_i,t} \qquad (17)\end{aligned}$$

which ensure smooth trajectories for these parameters. For $\lambda \in \{\nu, \rho, \eta\}$, we assume that $u_{\lambda_i,t}$ are centered white Gaussian noises, independent for different i and different λ, with variances $\sigma^2_{\lambda_i,t}$. These variances are generally unknown, and we therefore include them as part of the inference problem. In order to accommodate for possible nonstationarities, we assume the following evolution model for these hyperparameters. For $\lambda \in \{\nu, \rho, \eta\}$, we introduce the notation $\alpha^{\lambda}_{i,t} = \log(\sigma^2_{\lambda_i,t-1})$, and similarly, $\alpha^{y}_{i,t} = \log(\sigma^2_{y,t-1})$. Now, the evolution models for these variances are, for $\lambda \in \{\nu, \rho, \eta\}$

$$\alpha^{\lambda}_{i,t} = \alpha^{\lambda}_{i,t-1} + w_{\lambda_i,t} \qquad (18)$$

where the $w_{\lambda_i,t}$ are centered white Gaussian noises, independent for different i and λ, with variances $\delta^2_{\lambda,t}$. Similarly, we assume that the variance of the observation noise follows the following law:

$$\alpha^{y}_{i,t} = \alpha^{y}_{i,t-1} + w_{y,t}$$

where $w_{y,t}$ is a centered white Gaussian noise with variance $\delta^2_{y,t}$, independent of all other dynamic noises. The four hyperparameters $\delta^2_{\nu,t}$, $\delta^2_{\rho,t}$, $\delta^2_{\eta,t}$, and $\delta^2_{y,t}$ are fixed.

The hyperparameters k^z_t and k^η_t may be unknown and vary over time. We assume the probabilities $\pi_{(i,j),(i,j)} = 10/22$ (no dimension change), $\pi_{(i,j),(i+1,j)} = 1/22$ (add a complex pole), $\pi_{(i,j),(i-1,j)} = 5/22$ (remove a complex pole), $\pi_{(i,j),(i,j+1)} = 1/22$ (add a real pole), and $\pi_{(i,j),(i,j-1)} = 5/22$ (remove a real pole); see Section III-B for the "boundary cases." Of course, many other prior probabilities are possible, but different settings do not influence the results much. Here, we have favored parsimonious models and, thus, limited the addition of poles. Now, we describe the model on the different transitions. The mechanism for adding a real pole differs from that of the complex pole case only by the fact that the frequency is set to zero. We therefore focus on the complex case.

- **Addition of a pole**: A new frequency is drawn uniformly in $(0, \pi)$, and a modulus is proposed from the uniform distribution on (0, 1). The other existing pole characteristics are updated according to (17).
- **Removal of a pole**: Draw uniformly at random a pole to be deleted and simply remove it.

D. Simulations

We present here some simulation results obtained with synthetic data, whose spectrogram is presented in Fig. 1. This signal is composed of one stationary tone at normalized frequency 0.18, a transient tone at frequency 0.3 between time samples 150 and 400, a sine-modulated component with mean normalized frequency 0.4, and a linear chirp. Note that the number of spectral components is time-varying. A Gaussian white noise is added to the data such that the SNR is about 20 dB. The following set of parameters was chosen: $M = 10$, and the number of particles was set to $N = 1000$. In order to demonstrate the interest of our approach, we have run our experiment on the same set of data but with different variants of our model.

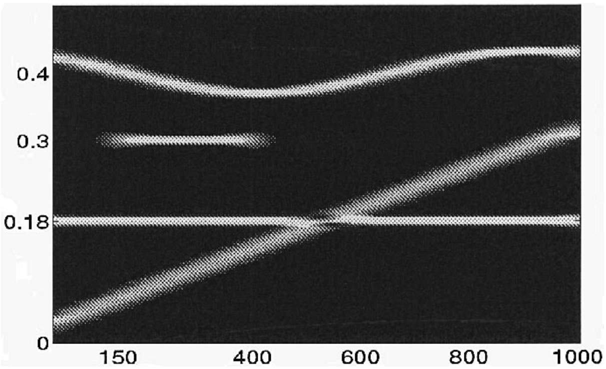


Fig. 1. Spectrogram of the data (1000 time samples long) computed with a Hamming window (171 time samples long).

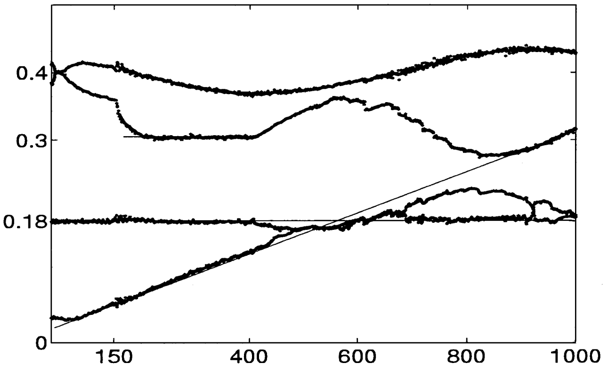


Fig. 2. Estimation with fixed hyperparameters and dimension. Solid lines indicate the actual location of the chirp and tone components.

For the first two experiments (see Figs. 2 and 3), the dimension of the model was kept fixed with $k^z = 4$ and $k^\eta = 0$, whereas the hyperparameters $\delta^2_{\lambda,t}$, $\lambda \in \{\nu, \rho, \eta, y\}$ were estimated for the experiment corresponding to Fig. 3. Finally, both the hyperparameters and the dimension k^z_t of the problem were estimated (see Fig. 4). The parameters are estimated by MMSE. In the most complex case, where the dimensions are also sampled, a mixed MMAP/MMSE estimator is implemented: At time t, the estimate $(\widehat{k^z_t, k^\eta_t})$ is the most represented (k^z_t, k^η_t) in terms of the cumulated weights among the set of particles; the other parameters and hyperparameters are estimated by MMSE on the set $\mathcal{A}$ of particles whose dimension $(k^{z,(i)}_t, k^{\eta,(i)}_t)$ verifies $(k^{z,(i)}_t, k^{\eta,(i)}_t) = (\widehat{k^z_t, k^\eta_t})$ (this second step requires the weights of particles in $\mathcal{A}$ to be normalized such that their sum is one).

The results of the first simulation (see Fig. 2) were obtained with hyperparameter values of $\delta^2_{\lambda,t} = 0.01$ for $\lambda \in \{\nu, \rho, \eta, y\}$. As can be seen from the plots, the frequencies were correctly estimated until the chirp crossed

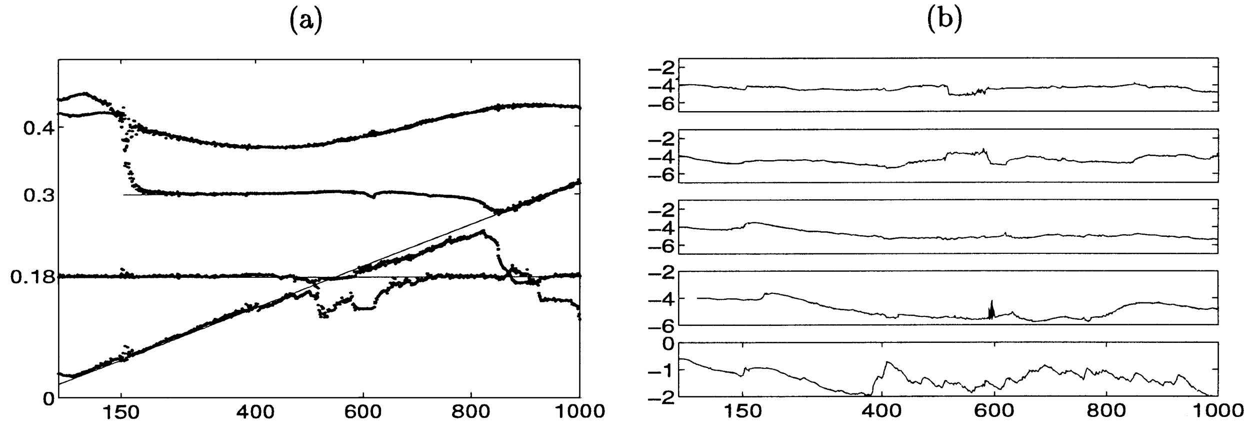


Fig. 3. (a) Frequency estimation with fixed dimension, but the hyperparameters are estimated. (b) represents from top to bottom: $\log_{10}(\delta^2_{\nu,t})$ for the chirp, the stationary tone, the transient tone, the sine-modulated component, and $\log_{10}(\delta^2_{y,t})$.

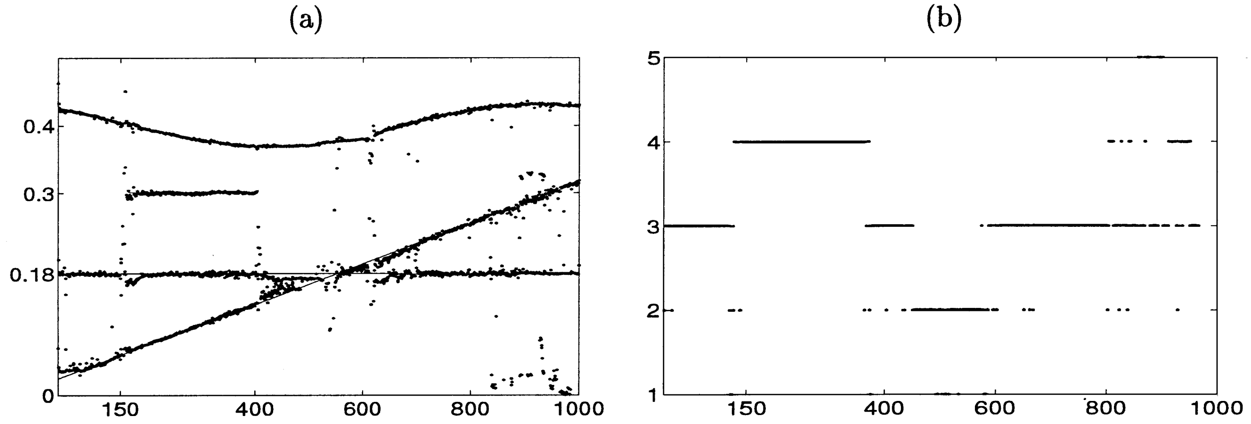


Fig. 4. Joint on-line estimation and detection of spectral components. (a) Estimated frequencies. (b) Estimated number of spectral components. The hyperparameters $\delta^2_{\lambda,t} = 0.01, \lambda \in \{\nu, \rho, \eta, y\}$ are also estimated but not plotted since they are very similar to those in Fig. 3(b).

the tone.[3] After this point, the filter required 300 iterations before converging again toward the true frequencies with good accuracy. In Fig. 3, the hyperparameters $\delta^2_{\lambda,t}, \lambda \in \{\nu, \rho, \eta, y\}$ were estimated on-line (with $k^z = 4$ and $k^\eta = 0$). This resulted in better estimation accuracy and more robustness as the hyperparameters tuning the "reactivity" of the filter adapted automatically to local situations. The best estimation results, however, were obtained in the third experiment, where k^z_t was estimated in addition to the hyperparameters $\delta^2_{\lambda,t}$, $\lambda \in \{\nu, \rho, \eta, y\}$. In Fig. 4, both the number of components and the frequency trajectories were estimated with good accuracy.

In order to compare the efficiency of our improved particle filter with a standard algorithm, we ran a second set of simulations. The spectrogram of the data is plotted in Fig. 5. Our algorithm is compared to the standard SIR algorithm[4] in the case where k^z_t, k^η_t and $\delta^2_{\lambda,t}, \lambda \in \{\nu, \rho, \eta, y\}$ were estimated online. Fig. 6 displays five simulations results for the three following cases: our algorithm with $N_{\text{Improved algo.}} = 500$ [column (a)], SIR algorithm with $N_{\text{SIR algo.}} = 1800$ [column (b)]—the computation time is then similar to that of our algorithm with $N_{\text{Improved algo.}} = 500$—and SIR algorithm with $N_{\text{SIR algo.}} = 500$ [column (c)]. When the number of particles used in each algorithm is the same ($N_{\text{Improved algo.}} = N_{\text{SIR algo.}} = 500$) [columns (a) and (c)], our algorithm is more precise and more robust. When the computation time is the same [in order to obtain a fair comparison, both filters have been programmed in C language and share as many common subroutines as possible; see Fig. 6(a) and (b)], our filter is more robust. The estimation

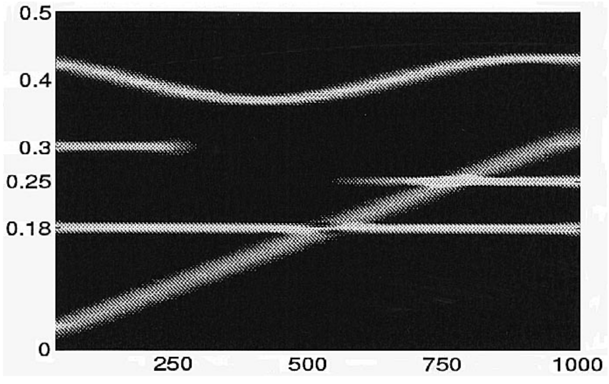


Fig. 5. Spectrogram of the second data (1000 time sample long) computed with a Hamming window (171 time sample long).

[3] For all the simulations, the amplitudes of the poles are not plotted since their behavior is fairly standard and less instructive than that of the frequencies.

[4] In the SIR algorithm, the importance distribution for the number of poles is the prior defining the moves *update, add, remove* described earlier, and the importance distribution used to sample the parameters is the prior distribution defined in (17).

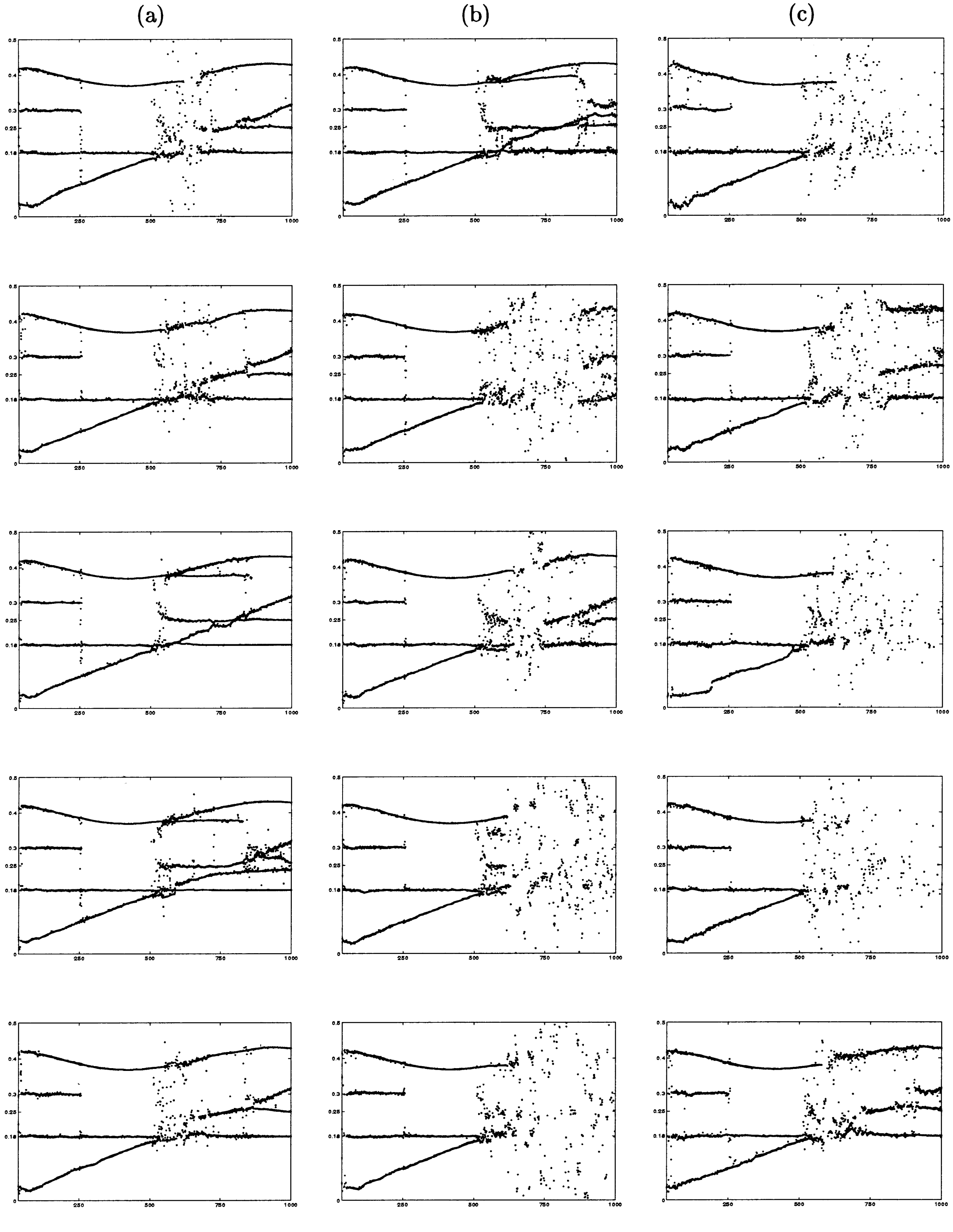


Fig. 6. Comparison of our particle filter with the standard SIR algorithm. (a) Five frequency estimations with our algorithm ($N_{\text{Improved algo.}} = 500$ particles). (b) Five frequency estimations with the SIR algorithm ($N_{\text{SIR algo.}} = 1800$ particles). This number of particles yields the same computation time as in (a). (c) Five frequency estimations with the SIR algorithm ($N_{\text{SIR algo.}} = 500$ particles).

accuracy was better with our particle filter compared with the SIR algorithm, whenever $N_{\text{SIR algo.}} = 500$ or $N_{\text{SIR algo.}} = 1800$.

Moreover, 100 simulations were run for the three scenarios $N_{\text{Improved algo.}} = 500$, $N_{\text{SIR algo.}} = 1800$ and $N_{\text{SIR algo.}} = 500$. Out of 100 simulations, our filter, with $N_{\text{Improved algo.}} = 500$, failed[5] eight times, with no case of major failure. For the

[5] We consider that a simulation fails whenever the trajectories are lost for more than 100 time samples. Major failures occur when the trajectories are lost and never found again.

SIR algorithm with $N_{\mathrm{SIR\ algo.}} = 1800$, 24 simulations failed, including 17 major failures, whereas 74 simulations failed (including 53 major failures) when $N_{\mathrm{SIR\ algo.}} = 500$. These results demonstrate the superiority, expected by the careful design of the algorithm, in terms of robustness. It is always capable of converging again toward the trajectories after losing them, which is not the case for the SIR algorithm (even with more than three times as many particles).

Finally, simulations were run with $N_{\mathrm{Improved\ algo.}} = 10\,000$. In this case, our algorithm has a deterministic behavior since all the simulations provided the same results.

These simulations show that it is possible to track an unknown number of spectral trajectories in the time-frequency plane. Our simulations were much successful than, e.g., those presented in [3] in a similar context, and many real applications are possible, such as music transcription, which will shortly be investigated.

IV. Conclusion

In this paper, we develop efficient particle filtering techniques especially tailored for Jump Markov systems. We apply our strategy to the estimation of time-varying autoregressive processes in the scenarios where the number of poles is unknown and evolves with time. Application of our algorithm to synthetic signals demonstrates the interest of our modeling and the superior efficiency of the algorithm over standard particle filtering techniques. Application of our methodology to complex multi-target tracking scenarios is currently being investigated.

References

[1] C. Andrieu, N. De Freitas, and A. Doucet, “Sequential MCMC for Bayesian model selection,” in *Proc. Workshop IEEE HOS*, 1999.

[2] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation, Tracking and Navigation: Theory, Algorithms and Software*. New York: Wiley, 2001.

[3] M. Davy, B. Leprettre, C. Doncarli, and N. Martin, “Tracking of spectral lines in an ARCAP time-frequency representation,” in *Proc. EUSIPCO*, Rhodes Island, Greece, 1998.

[4] A. Doucet, S. J. Godsill, and C. Andrieu, “On sequential Monte Carlo sampling methods for Bayesian filtering,” *Statist. Comput.*, vol. 10, pp. 197–208, 2000.

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

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

[7] P. Flandrin, *Time-Frequency/Time-Scale Analysis*. San Diego, CA: Academic, 1999.

[8] W. R. Gilks and C. Berzuini, “Following a moving target—Monte Carlo inference for dynamic Bayesian models,” *J. R. Statist. Soc. B*, vol. 63, no. 1, pp. 127–146, 2001.

[9] P. J. Green, “Reversible jump Markov chain Monte Carlo computation and Bayesian model determination,” *Biometrika*, vol. 82, pp. 711–732, 1995.

[10] N. Ikoma and H. Maeda, “Nonstationary spectral peak estimation by Monte Carlo filter,” in *Proc. Conf. Adaptative Syst. Signal Process.*, 2000.

[11] S. J. Julier and J. K. Uhlmann, “A general method for approximating nonlinear transformations of probability distributions,” Eng. Dept., Univ. Oxford, Oxford, U.K., Tech. Rep..

[12] S. J. Julier and J. K. Uhlmann, “A new extension of the Kalman filter to nonlinear systems,” in *Proc. AeroSense*, Orlando, FL, 1997.

[13] S. J. Julier, “The scaled unscented transformation,” in *Proc. Amer. Contr. Conf.*, 2002.

[14] C.-J. Kim and C. R. Nelson, *State-Space Models With Regime Switching*. Cambridge, U.K.: MIT Press, 1999.

[15] G. Kitagawa, “Monte Carlo filter and smoother for non-Gaussian nonlinear state space models,” *J. Comput. Graph. Statist.*, vol. 5, pp. 1–25, 1996.

[16] M. K. Pitt and N. Shephard, “Filtering via simulation: Auxiliary particle filter,” *J. Am. Statist. Assoc.*, vol. 94, pp. 590–599, 1999.

[17] R. Prado and G. Huerta, “Time-varying autoregressions with model order uncertainty,” J. Time Series Anal., 2002, to be published.

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

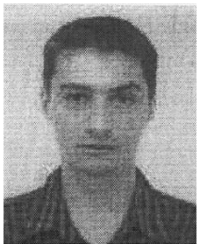
[19] J. Vermaak, C. Andrieu, A. Doucet, and S. J. Godsill, “Particle methods for Bayesian modeling and enhancement of speech signals,” *IEEE Trans. Speech Audio Processing*, vol. 10, pp. 173–185, Mar. 2002.

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, Bristol, U.K. His research interests include spectral analysis, source separation, Markov chain Monte Carlo methods, sequential Monte Carlo methods, stochastic optimization, and stochastic approximation and their applications.

Manuel Davy was born in Caen, France, in 1972. He received the ingénieur degree in electrical engineering in 1996 from Ecole Centrale de Nantes, Nantes, France, and the Ph.D. degree from the University of Nantes in 2000.

He is currently a chargé de recherches CNRS at IRCCyN, Nantes, where his research activity is centered around parametric and nonparametric methods for audio processing.



Arnaud Doucet was born in France on November 2, 1970. He graduated from Institut National des Telecommunications in June 1993 and received the Ph.D. degree from Universite Paris-Sud, Orsay, France, in December 1997.

From January 1998 to February 2001, he was a research associate with Cambridge University, Cambridge, U.K. From March 2001 to August 2002, he was a Senior Lecturer with the Department of Electrical Engineering, Melbourne University, Parkville, Australia. Since September 2002, he has been a University Lecturer with the Engineering Department, Cambridge University. His research interests include sequential Monte Carlo methods, Markov chain Monte Carlo methods, optimal filtering, and control and reinforcement learning.