EFFICIENT PARTICLE FILTERING FOR JUMP MARKOV SYSTEMS

C. Andrieu, M. Davy, A. Doucet

CUED/F-INFENG/TR.427

February 2002

Signal Processing Group

Cambridge University Engineering Department

Trumpington Street

Cambridge CB2 1PZ, UK

E-mail: C.Andrieu@bristol.ac.uk, md283@eng.cam.ac.uk, a.doucet@ee.muz.oz.au

Contents

Abstract

In this paper we present an efficient particle filtering method to perform optimal estimation in Jump Markov (Nonlinear) Systems. This original method relies on the combination of techniques presented recently in the filtering literature, namely the Auxiliary Particle Filter [15] and the Unscented Transform [10]. This algorithm is applied to the complex problem of time-varying autoregressive estimation with an unknown time-varying model order. Simulations demonstrate the performance of our method compared to standard particle filtering techniques.

1 Introduction

1.1 Background

Many estimation problems arising in signal processing and related fields can be cast in the following form. A dynamic model represents the *signal* of interest $\{x_t\}$ ($t \geq 0$ and $x_t \in \mathbb{R}^{n_x}$), which is not observed but statistically related to the *observations* $\{y_t\}$ ($t \geq 1$ and $y_t \in \mathbb{R}^{n_y}$). More formally the process is typically described with a Markov transition density

$$p(x_t|x_{0:t-1}) = f(x_t|x_{t-1}), \tag{1}$$

where, for a set of variables l_t, we denote $l_{a:b} \triangleq \{l_a, l_{a+1}, \ldots, l_b\}$. The observations $y_{1:t}$ are usually assumed to be independent conditional upon the signal process $\{x_t\}$, and marginally distributed according to

$$p(y_t|x_{0:t}) = g(y_t|x_t). \tag{2}$$

One is then interested in estimating the sequence of posterior densities $p(x_{0:t}|y_{1:t})$ and typically their marginals $p(x_t|y_{1:t})$.

This so-called optimal filtering problem usually does not admit a closed-form solution and many approaches have been proposed in the last 35 years in order to approximate these distributions. Particle filtering techniques are simulation-based methods that have recently revolutionized this field. They allow for recursive state estimation of virtually any dynamic model, even when elements of non-linearity and non-Gaussianity are present. Spectacular results can be found for real applications in [6], and the field is currently growing at a steady pace. One of the great interest of particle filtering techniques, and more generally of Monte Carlo methods, is that they allow for heuristic approaches previously developed to be embedded in a rigorous framework.

The problem addressed in this paper is that of the development of efficient particle filtering techniques to perform on-line detection and estimation for a very important class of dynamic models, named Jump Markov Systems (JMS) or Markov Switching State Space Models. JMS are a class of models appearing in signal processing, target tracking and econometrics among others [2], [13]. This class extends significantly the class of models (1)-(2) and specific particle filtering methods have to be developed to perform optimal estimation efficiently.

1.2 Jump Markov Systems

1.2.1 Statistical Model

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

$$\pi_{ij} \triangleq \Pr\{r_{t+1} = j | r_t = i\}, \quad (i, j \in S). \tag{3}$$

We define s 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,

$$p(x_t | x_{0:t-1}, r_{1:t}) = f_{r_{t-1} r_t}(x_t | x_{t-1}). \tag{4}$$

The initial state x_0 is distributed according to a distribution h. 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

$$p(y_t | x_{0:t}, r_{1:t}, y_{1:t-1}) = g_{r_t}(y_t | y_{1:t-1}, x_t), \tag{5}$$

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.

The class of processes under study here is a generalization of the Jump Markov Linear Models (JMLS) considered in [5], [13]. 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

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

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. However, it is of great practical interest to relax these linearity and Gaussianity assumptions which are unrealistic in many real-world applications, as illustrated by the following target tracking examples; see Section 3 for an application to Time-Varying Autoregressive models (TVAR).

Example 1. *Bearings-Only Tracking for a Maneuvering Source* [2]. Let us consider the following standard tracking model. A single target evolves in a 2-D plan $x - y$. Conditional upon a maneuver

r_t of the target ($\{r_t\}$ is modeled as a Markov chain) the state $x_t \triangleq (l_{x,t}, l_{y,t}, s_{x,t}, s_{y,t})$ consists of the location $(l_{x,t}, l_{y,t})$ and velocity $(s_{x,t}, s_{y,t})$, which evolve according to a standard linear Gaussian model

$$x_t = A(r_t)\, x_{t-1} + B(r_t)\, v_t, \tag{6}$$

We observe

$$y_t = \tan^{-1}\left(\frac{l_{y,t}}{l_{x,t}}\right) + \varepsilon_t. \tag{7}$$

The nonlinearity appearing in the observation equation (7) precludes the use of standard suboptimal filtering techniques developed for JMLS.

Example 2. *Multitarget tracking in clutter noise* [2]. Assume one observes r_t targets in clutter noise; r_t is the unknown time-varying number of targets modelled as a Markov chain. Conditional upon r_t, the state $x_t = (x_{1,t}, \ldots, x_{r_t,t})$ consists of the aggregation of each state target $x_{i,t}$, $i = 1, \ldots, r_t$. Typically, one assumes that the targets evolve independently of each other and are statistically similar, i.e.

$$f_{r_{t-1}r_t}(x_t|\, x_{t-1}) = \prod_{i=1}^{r_t} k_{r_{t-1}r_t}(x_{i,t}|\, x_{i,t-1}),$$

for a family of conditional densities $\{k_{ij}\}$. We observe at each time a random number of observations composed of noisy nonlinear observations of the targets plus some clutter noise. The time-varying dimension of the state x_t and the nonlinearity in the observations preclude the use of standard suboptimal filtering schemes.

1.2.2 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 $p(x_{0:t}, r_{1:t}|\, y_{1:t})$:

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

where

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

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.

1.3 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,

$$p\,(dx_{0:t}, r_{1:t}|\,y_{1:t}) \approx \sum_{i=1}^{N} w_t^{(i)} \delta_{x_{0:t}^{(i)}}\,(dx_{0:t})\,\mathbb{I}_{\left\{r_{1:t}^{(i)}\right\}}\,(r_{1:t})\,,\ \sum_{i=1}^{N} w_t^{(i)} = 1,\ w_t^{(i)} \geq 0. \tag{8}$$

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 $\delta_{x,r}\,(dx, r) = \mathbb{I}_{\{r\}}\,(r)\,\delta_x\,(dx)$. 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.

The art of particle filtering consists mainly of the way the particles are updated and propagated through time. In particular it is extremely important to guide the exploration of these particles through the series of state spaces using any available information (*e.g.* the observations $y_{1:t}$) or salient feature of the underlying process (*e.g.* conditional linearity). It is thus no surprise if most efforts in the field have been devoted to these latter aspects. 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 combination of several methods that have been recently proposed in the literature, mainly the Auxiliary Particle Filter (APF) [15] and a suboptimal

deterministic filtering method, the Unscented Kalman Filter (UKF), which is a particular instance of the Unscented Transform (UT) [10], [11].

We apply our methodology to nonstationary signal detection and estimation, using flexible time-varying autoregressive representation (TVAR). We adopt here a pole representation of the autoregressive process which allows for the specification of intuitive priors (*e.g.* smoothness, abrupt changes) and natural interpretations. This choice of parameterization introduces many non-linearities 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 2 we briefly review the basic principles of particle filtering techniques and detail our generic algorithm adapted to JMS. In Section 3 we introduce the TVAR problem and performance of the procedure is demonstrated on synthetic signals. A discussion is given in Section 4.

2 Particle Filtering for JMS

2.1 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 $p\left(x_{0:t}, r_{1:t} | \, y_{1:t}\right)$; see [6] for further details.

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

$$\widehat{p}_N\left(dx_{0:t}, r_{1:t} | \, y_{1:t}\right) = \sum_{i=1}^{N} w_t^{(i)} \delta_{\widetilde{x}_{0:t}^{(i)}, \widetilde{r}_{1:t}^{(i)}}\left(dx_{0:t}, r_{1:t}\right), \tag{9}$$

where the importance weights satisfy

$$\begin{aligned} w_t^{(i)} &\propto w_{t-1}^{(i)} \frac{p\left(\widetilde{x}_{0:t}^{(i)}, \widetilde{r}_{1:t}^{(i)} \middle| y_{1:t}\right)}{p\left(\widetilde{x}_{0:t-1}^{(i)}, \widetilde{r}_{1:t-1}^{(i)} \middle| y_{1:t-1}\right) q\left(\widetilde{x}_t^{(i)}, \widetilde{r}_t^{(i)} \middle| \widetilde{x}_{0:t-1}^{(i)}, \widetilde{r}_{1:t-1}^{(i)}, y_{1:t}\right)} \\ &\propto w_{t-1}^{(i)} \frac{g_{\widetilde{r}_t^{(i)}}\left(y_t \middle| y_{1:t-1}, \widetilde{x}_t^{(i)}\right) f_{\widetilde{r}_{t-1}^{(i)} \widetilde{r}_t^{(i)}}\left(\widetilde{x}_t^{(i)} \middle| \widetilde{x}_{t-1}^{(i)}\right) \pi_{\widetilde{r}_{t-1}^{(i)} \widetilde{r}_t^{(i)}}}{q\left(\widetilde{x}_t^{(i)}, \widetilde{r}_t^{(i)} \middle| \widetilde{x}_{0:t-1}^{(i)}, \widetilde{r}_{1:t-1}^{(i)}, y_{1:t}\right)}. \end{aligned} \tag{10}$$

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

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

and the associated importance weight is proportional to the predictive likelihood

$$w_t \propto w_{t-1} p\left(y_t \middle| y_{1:t-1}, x_{t-1}, r_{t-1}\right), \tag{11}$$

where

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

This scenario is referred to as "full adaption" in [15].

Finally, one obtains N particles $\left\{x_{0:t}^{(i)}, r_{1:t}^{(i)}\right\}$ approximately distributed according to $p\left(x_{0:t}, r_{1:t} \middle| y_{1:t}\right)$ by resampling/selection from the weighted empirical distribution given in Eq. (9). There are several resampling procedures available in the literature. We adopt here the stratified sampling scheme described in [14].

This "optimal" importance sampling case deserves special attention. Indeed, the importance weights w_t given by Eq. (11) 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 – Eq. (11) – do not admit a closed form expression. However this scenario motivates an alternative particle filtering method known as APF [15], see Subsection 2.2.1, where one analytically approximates the predictive likelihoods, or its behavior, whenever necessary.

2.2 Strategies for Efficient Particle Filtering

2.2.1 Auxiliary Particle Filter

The idea behind APF is, at time t, to extend existing trajectories $\left\{x_{0:t-1}^{(i)}, r_{1:t-1}^{(i)}\right\}$ that are the most promising, in the sense that their predictive likelihoods $p\left(y_t \middle| y_{1:t-1}, x_{t-1}^{(i)}, r_{t-1}^{(i)}\right)$ are large.

However the analytical computation of these predictive likelihoods might prove to be intractable and approximation is needed. Recall that

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

In [15], the authors propose (in the non switching case) simple approximations where

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

where $\zeta\left(x_{t-1}^{(i)}, r_t\right)$ is the mode or mean of $f_{r_{t-1}^{(i)} r_t}\left(x_t|\, x_{t-1}^{(i)}\right)$. In many applications, especially if $g_{r_t}\left(y_t|\, y_{1:t-1}, x_t\right)$ varies significantly over the significant regions of $f_{r_{t-1}^{(i)} r_t}\left(x_t|\, \widetilde{x}_{t-1}^{(i)}\right)$, 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 towards uninteresting regions. It is thus fundamental to be able to approximate properly 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. We propose here to approximate this integral with

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

where $\psi_{r_t}\left(x_{t-1}^{(i)}, r_{t-1}^{(i)}\right)$ is a deterministic mapping/integration technique described in the next subsection (we omit the observations $y_{1:t}$ in $\psi_{r_t}(\cdot)$ to simplify notation). Then, an estimate of the desired quantity is

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

The SISR extends each particle $x_{0:t-1}^{(i)}, r_{1:t-1}^{(i)}$ with weight $w_{t-1}^{(i)}$ by sampling $\widetilde{x}_t^{(i)}, \widetilde{r}_t^{(i)}$ according to a conditional distribution $q\left(x_t, r_t|\, x_{0:t-1}^{(i)}, r_{1:t-1}^{(i)}, y_{1:t}\right)$ to obtain $\left(x_{0:t-1}^{(i)}, \widetilde{x}_t^{(i)}\right)$ and $\left(r_{1:t-1}^{(i)}, \widetilde{r}_t^{(i)}\right)$, then the associated weight satisfies Eq. (10), and can be rewritten as follows

$$\begin{aligned} w_t^{(i)} \quad \propto \quad & w_{t-1}^{(i)} \widehat{p}\left(y_t|\, y_{1:t-1}, x_{t-1}^{(i)}, r_{t-1}^{(i)}\right) \\ & \times \frac{g_{\widetilde{r}_t^{(i)}}\left(y_t|\, y_{1:t-1}, \widetilde{x}_t^{(i)}\right) f_{r_{t-1}^{(i)} \widetilde{r}_t^{(i)}}\left(\widetilde{x}_t^{(i)} \middle|\, x_{t-1}^{(i)}\right) \pi_{r_{t-1}^{(i)} \widetilde{r}_t^{(i)}}}{\widehat{p}\left(y_t|\, y_{1:t-1}, x_{t-1}^{(i)}, r_{t-1}^{(i)}\right) q\left(\widetilde{x}_t^{(i)}, \widetilde{r}_t^{(i)} \middle|\, x_{0:t-1}^{(i)}, r_{1:t-1}^{(i)}, y_{1:t}\right)} \end{aligned} \tag{12}$$

The interest of this decomposition is that the term $w_{t-1}^{(i)} \widehat{p}\left(y_t|\, y_{1:t-1}, x_{t-1}^{(i)}, r_{t-1}^{(i)}\right)$, which is independent of $\widetilde{r}_t^{(i)}, \widetilde{x}_t^{(i)}$, mimics the weight of the "full adaption" scenario described earlier. It therefore suggests the possibility of resampling $\left\{x_{0:t-1}^{(i)}, r_{1:t-1}^{(i)}\right\}$ according to the weights $\lambda_t^{(i)} \propto$

$w_{t-1}^{(i)} \widehat{p}\left(y_t | y_{1:t-1}, x_{t-1}^{(i)}, r_{t-1}^{(i)}\right)$ in order to obtain N particles $\left\{\widetilde{x}_{0:t-1}^{(i)}, \widetilde{r}_{1:t-1}^{(i)}\right\}$ which are then approximately sampled from a distribution "close" to $p\left(x_{0:t-1}, r_{1:t-1} | y_{1:t}\right)$. We then extend each particle by sampling $\widetilde{x}_t^{(i)}, \widetilde{r}_t^{(i)}$ according to a conditional distribution $q\left(x_t, r_t | \widetilde{x}_{0:t-1}^{(i)}, \widetilde{r}_{1:t-1}^{(i)}, y_{1:t}\right)$. Contrary to the full adaption case, it is however necessary to re-weight the particles by

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

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

2.2.2 The 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 equations (1)-(2) can alternatively be represented in the following manner

$$x_t = \varphi\left(x_{t-1}, v_t\right) \tag{13}$$

$$y_t = \gamma\left(x_t, \varepsilon_t\right), \tag{14}$$

where $\{v_t\}$ and $\{\varepsilon_t\}$ are typically mutually independent zero-mean i.i.d. sequences, φ and γ are non-linearities (similarly we will introduce φ_{ij} and γ_{ij} for JMS). Both the EKF and UKF rely on approximations of the system defined in Eq.'s (13)-(14), but are of different nature. Nevertheless, for both scenarios the result of such approximations is that the series of predictive and filtering densities $\{p\left(x_t | y_{1:t-1}\right)\}$ and $\{p\left(x_t | y_{1:t}\right)\}$ are replaced with series of Gaussian distributions $\left\{\mathcal{N}\left(x_t; m_{t|t-1}, P_{t|t-1}\right)\right\}$ and $\left\{\mathcal{N}\left(x_t; m_{t|t}, P_{t|t}\right)\right\}$. Such approximations allow, in principle, for the application of the Kalman filter recursions in order to compute $m_{t|t-1}$, $P_{t|t-1}$ from $m_{t-1|t-1}$, $P_{t-1|t-1}$ (this makes use of the evolution equation) and $m_{t|t}$, $P_{t|t}$ from $m_{t|t-1}$, $P_{t|t-1}$ and the observation y_t (this makes use of the observation equation).

The EKF relies on linearizations of the evolution and observation equations (13)-(14), 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 $\left\{\overline{x}_{t-1}^{(i)}\right\}$, the "sigma points" [11], possess the correct mean equal to $x_{t-1|t-1}$ and covariance $P_{t-1|t-1}$. Then the

mean and sample autocovariance of the set $\left\{\varphi\left(\overline{x}_{t-1}^{(i)},0\right)\right\}$ should be a good approximation of $x_{t|t-1}$ and $P_{t|t-1}$ respectively. Similarly the mean and autocovariance of $\left\{\gamma\left(\varphi\left(\overline{x}_{t-1}^{(i)},0\right),0\right)\right\}$ should lead to reasonable approximations of $x_{t|t-1} \triangleq \mathbb{E}(\gamma(\varphi(x_{t-1},v_t),\varepsilon_t)|\, y_{1:t-1})$ 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 $x_{t|t}$ and $P_{t|t}$ with the Kalman filter. Given values for $x_{t|t}$ and $P_{t|t}$, various methods have been proposed in order to generate a new set $\left\{\overline{x}_t^{(i)}\right\}$ [10], [12].

2.3 Algorithm

Based on the elements presented above, it is possible to propose the following generic particle filtering algorithm for JMS.

Particle Filter for JMS

At time $t=0$, **Step 0**: *Initialization*

- For $i=1,...,N$ sample $x_0^{(i)} \sim p(x_0)$ and set the weights $w_0^{(i)} = 1/N$.

At time $t \geq 1$,

Step 1: *Auxiliary variable resampling step*

- For $i=1,...,N$, compute $\lambda_t^{(i)}$ as[1]

$$\lambda_t^{(i)} \propto w_{t-1}^{(i)} \sum_{r_t \in S} \pi_{r_{t-1}^{(i)} r_t} \psi_{r_t}\left(x_{t-1}^{(i)}, r_{t-1}^{(i)}\right), \quad \sum_{i=1}^{N} \lambda_t^{(i)} = 1, \tag{15}$$

where $\psi_{r_t}\left(x_{t-1}^{(i)}, r_{t-1}^{(i)}\right)$ are computed using an unscented approximation:

$$\psi_{r_t}\left(x_{t-1}^{(i)}, r_{t-1}^{(i)}\right) = \frac{1}{n}\sum_{i=1}^{n} g_{r_t}\left(y_t|y_{1:t-1}, \varphi_{r_{t-1}^{(i)} r_t}(\overline{x}_{t-1}^{(i)})\right)$$

The same sigma points $\{\overline{x}_{t-1}^{(i)}\}$ are used to compute $m_{t|t}(r_t), P_{t|t}(r_t)$ for all r_t in S.

- Multiply/Discard particles $\left\{x_{0:t-1}^{(i)}, r_{1:t-1}^{(i)}\right\}$ and the associated statistics $\left\{\psi_{r_t}\left(x_{t-1}^{(i)}, r_{t-1}^{(i)}\right), m_{t|t}^{(i)}(r_t), P_{t|t}^{(i)}(r_t); r_t \in S\right\}$ with respect to high/low importance weights $\lambda_t^{(i)}$ to obtain N particles $\left\{\widetilde{x}_{0:t-1}^{(i)}, \widetilde{r}_{1:t-1}^{(i)}\right\}$ and the associated statistics $\left\{\psi_{r_t}\left(\widetilde{x}_{t-1}^{(i)}, \widetilde{r}_{t-1}^{(i)}\right), \widetilde{m}_{t|t}^{(i)}(r_t), \widetilde{P}_{t|t}^{(i)}(r_t); r_t \in S\right\}$.

Step 2: *Importance sampling step*

- For $i=1,...,N$, sample $\widetilde{r}_t^{(i)} \sim q\left(r_t|\, y_{1:t}, \widetilde{x}_{0:t-1}^{(i)}, \widetilde{r}_{0:t-1}^{(i)}\right) \propto \pi_{r_{t-1}^{(i)} r_t} \psi_{r_t}\left(\widetilde{x}_{t-1}^{(i)}, \widetilde{r}_{t-1}^{(i)}\right)$.
- For $i=1,...,N$, extend the trajectories with $\widetilde{x}_t^{(i)} \sim \mathcal{N}\left(x; \widetilde{m}_{t|t}^{(i)}\left(\widetilde{r}_t^{(i)}\right), \widetilde{P}_{t|t}^{(i)}\left(\widetilde{r}_t^{(i)}\right)\right)$.[2]

[1] For t=1, π_{r_{t-1},r_t} in Eq. (15) should be replaced with the stationary distribution of the discrete Markov chain.

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

- Compute the importance weights as

$$w_t^{(i)} \propto \frac{g_{\widetilde{r}_t^{(i)}}\left(y_t|\widetilde{x}_t^{(i)}\right) f_{\widetilde{r}_{t-1}^{(i)}\widetilde{r}_t^{(i)}}\left(\widetilde{x}_t^{(i)}\middle|\widetilde{x}_{t-1}^{(i)}\right)}{\psi_{\widetilde{r}_t^{(i)}}\left(\widetilde{x}_{t-1}^{(i)},\widetilde{r}_{t-1}^{(i)}\right)\mathcal{N}\left(\widetilde{x}_t^{(i)};\widetilde{m}_{t|t}^{(i)}\left(\widetilde{r}_t^{(i)}\right),\widetilde{P}_{t|t}^{(i)}\left(\widetilde{r}_t^{(i)}\right)\right)}.$$

- Rename the particles $\{\widetilde{x}_t^{(i)},\widetilde{r}_{t-1}^{(i)}\}$ into $\{x_t^{(i)},r_{t-1}^{(i)}\}$

2.4 Discussion and Extension

Discussion. The combination of the APF together with the UKF ensures good statistical properties of the proposed algorithm, as it tends to approximate the "full adaption" scenario of [15]. However we would like here to stress on its computational efficiency, as some quantities are computed once, but used twice in **Step 1** and **Step 2**. More precisely, the evaluation of $\psi_{r_t}\left(x_{t-1}^{(i)},r_{t-1}^{(i)}\right)$ requires both the building of sigma points, the evaluation of $\psi_{r_t}\left(x_{t-1}^{(i)},r_{t-1}^{(i)}\right)$ for $r_t \in S$ such that $\pi_{\widetilde{r}_{t-1}^{(i)}r_t} > 0$ and the sum $\sum_{r_t\in S}\pi_{r_{t-1}^{(i)}r_t}\psi_{r_t}\left(x_{t-1}^{(i)},r_{t-1}^{(i)}\right)$. The quantities $\psi_{r_t}\left(\widetilde{x}_{t-1}^{(i)},\widetilde{r}_{t-1}^{(i)}\right)$ and $\sum_{r_t\in S}\pi_{r_{t-1}^{(i)}r_t}\psi_{r_t}\left(\widetilde{x}_{t-1}^{(i)},\widetilde{r}_{t-1}^{(i)}\right)$ are used in **Step 2** to draw $\widetilde{r}_t^{(i)}$, and the sigma points are re-used in **Step 2** in order to compute the Kalman filter quantities, which themselves are used to build a good proposal density for $\widetilde{x}_t$. In addition, the discrete state $\widetilde{r}_t^{(i)}$ is sampled according to an estimate of the predicted likelihood of each state $r_t \in S$, which yields better efficiency than sampling from *e.g.* the prior distribution $\pi_{\widetilde{r}_{t-1}^{(i)}r_t}$

Extension. There are several potential extensions to this algorithm. In particular, it is possible for example to use Markov Chain Monte Carlo (MCMC) steps so as to rejuvenate the path of the particles [8] (see [16] for an introduction to MCMC methods). It has been shown to significantly improve the quality of fixed-lag smoothing estimates, *i.e.* estimates based on $p\left(x_{t-L+1},r_{t-L+1}|\,y_{1:t}\right)$ where $L>0$ [6]. At time t, any standard MCMC algorithm, such as the Metropolis-Hastings (MH) algorithm or Gibbs sampler, of invariant density $p\left(x_{0:t},r_{1:t}|\,y_{1:t}\right)$ may be used. In the context of JMS, dependent upon $r_{1:t}$, the state x_t may lie in subspaces of different dimensions. In this case, it is possible to combine our particle filtering algorithm with reversible jump MCMC so as to allow for the particles to move from one subspace to another [1].

3 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 on-line and introduce interferences, leading to major tracking difficulties. On the other hand, representations such as the spectrogram can be computed on-line 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 analysed is summarised 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 shall 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 to alternative techniques such as the spectrogram.

3.1 Problem description

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

$$y_t = a_{1,t} y_{t-1} + a_{2,t} y_{t-2} + \ldots + a_{K_t,t} y_{t-K_t} + v_t, \tag{16}$$

where v_t is a centered Gaussian noise. Eq. (16) can be written in vector form as:

$$y_t = a_{1:K_t,t}^\mathsf{T}\, y_{t-1:t-K_t} + v_t. \tag{17}$$

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,

$$\chi_t(x) = 1 - a_{1,t}x - a_{2,t}x^2 - \ldots - a_{K_t,t}x^{K_t}. \tag{18}$$

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 $\eta_{i,t}$, $i = 1, \ldots, k_t^\eta$ denote the real-valued poles. The complex

poles can equivalently be expressed in terms of moduli and "instantaneous" frequencies, as

$$z_{i,t} = \rho_{i,t} e^{j2\pi\nu_{i,t}} \ , \ i = 1, \ldots, k_t^z. \tag{19}$$

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 on-line the TVAR poles, the parameters k_t^η and k_t^z and hyperparameters. In [9], a simpler model was presented with a fixed number of complex poles and fixed hyperparameters.

3.2 Model and state space representation

Here the state vector x_t contains the moduli and instantaneous frequencies of the poles, but also possible hyperparameters α_t. 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 non-linear transformation from the pole representation to the coefficients of the polynomial $\chi_t(x)$ combined with past observations, more precisely

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

3.3 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

$$\begin{aligned}
\nu_{i,t} &= \frac{1}{M}\left(\nu_{i,t-1} + \nu_{i,t-2} + \ldots + \nu_{i,t-M}\right) + u_{\nu_i,t} \\
\rho_{i,t} &= \frac{1}{M}\left(\rho_{i,t-1} + \rho_{i,t-2} + \ldots + \rho_{i,t-M}\right) + u_{\rho_i,t} \\
\eta_{i,t} &= \frac{1}{M}\left(\eta_{i,t-1} + \eta_{i,t-2} + \ldots + \eta_{i,t-M}\right) + u_{\eta_i,t},
\end{aligned} \tag{20}$$

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 non-stationarities 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 $\lambda \in \{\nu, \rho, \eta\}$,

$$\alpha^{\lambda}_{i,t} = \alpha^{\lambda}_{i,t-1} + w_{\lambda_i,t}, \tag{21}$$

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), $\pi_{(i,j),(i,j-1)} = 5/22$ (remove a real pole), see Subsection 3.2 for the "boundary cases". Of course, many other prior probabilities are possible, but different settings do not much influence the results. Here we have favoured 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 Eq. (20).
- **Removal of a pole**: draw uniformly at random a pole to be deleted and simply remove it.

3.4 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: $M = 10$, 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.

For the first two experiments (Fig. 2-3) the dimension of the model was kept fixed with $k^z = 4$ and $k^\eta = 0$, while 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 (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 (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 the tone[3]. After this point, the filter required 300 iterations before converging again towards 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.

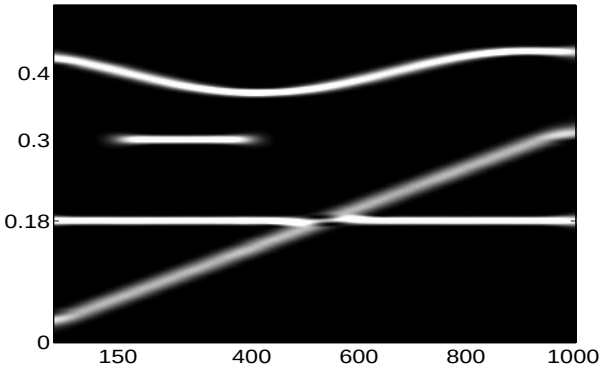

Fig. 1. Spectrogram of the data (1000 time samples long) computed with a Hamming window (171 time samples long).

[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.

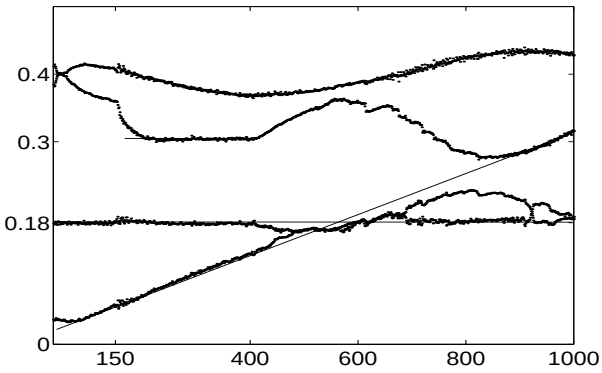

Fig. 2. Estimation with fixed hyperparameters and dimension. Solid lines indicate the actual location of the chirp and tone components.

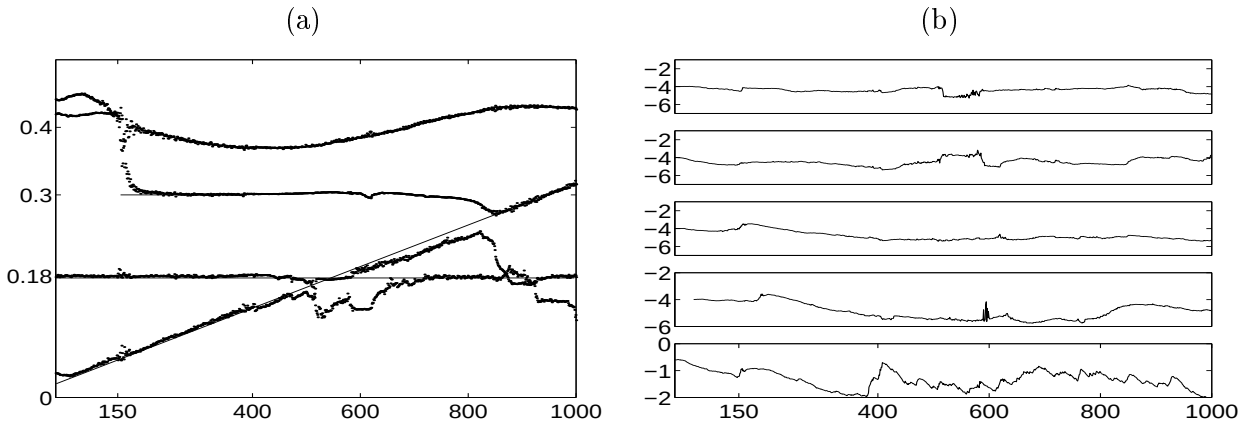

Fig. 3. Frequency estimation with fixed dimension (a), but the hyperparameters are estimated. The right plot (b) represents from top to bottom: $\log_{10}(\delta^2_{\nu,t})$ for the chirp, the stationary tone, the transient tone, the sine-modulated component, and $\log_{10}(\delta^2_{y,t})$.

In order to compare the efficiency of our improved particle filter to 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 on-line. Fig. 6 displays five simulations results for the three following cases: our

[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 Eq.(20).

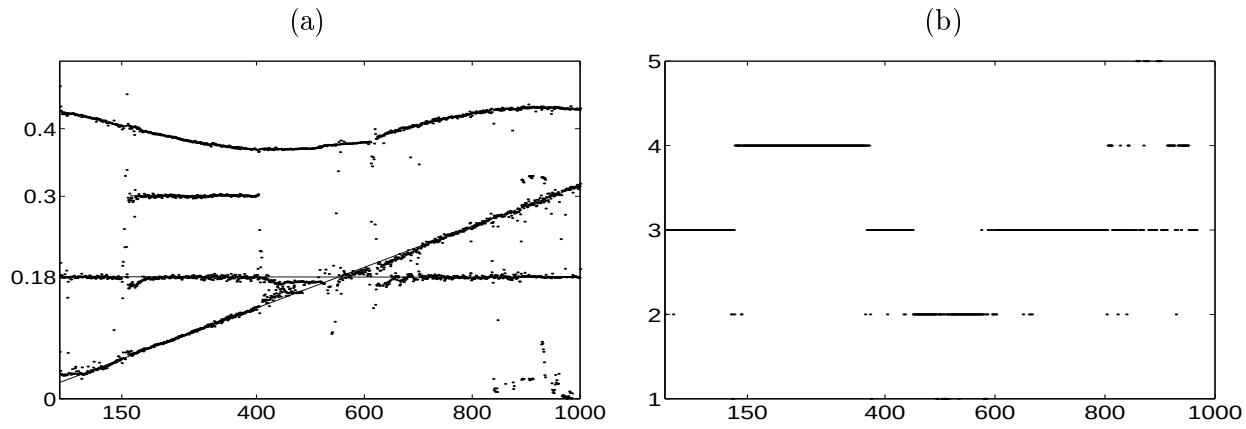

Fig. 4. Joint on-line estimation and detection of spectral components. (a) Estimated frequencies. (b) Estimated number of spectral components. The hyperparameters $\delta^2_{\lambda,t} = 0.01$, $\lambda \in \{\nu, \rho, \eta, y\}$ are also estimated but not plotted since they are very similar to those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

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.'s 6 (a) and (b), our filter is more robust. The estimation accuracy was better with our particle filter compared to 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 SIR algorithm with $N_{\text{SIR algo.}} = 1800$, 24 simulations failed, including 17 major failures, whereas 74 simulations failed (including 53 major failures) when $N_{\text{SIR algo.}} = 500$. These results demonstrate the superiority, expected by the careful design of the algorithm, in terms of robustness: it is always capable of converging again towards the trajectories after losing them, which is not the case for the SIR algorithm (even with more than 3 times as many particles).

Finally, simulations were run with $N_{\text{Improved algo.}} = 10000$. In this case, our algorithm has a deterministic behavior since all the simulations provided the same results.

[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.

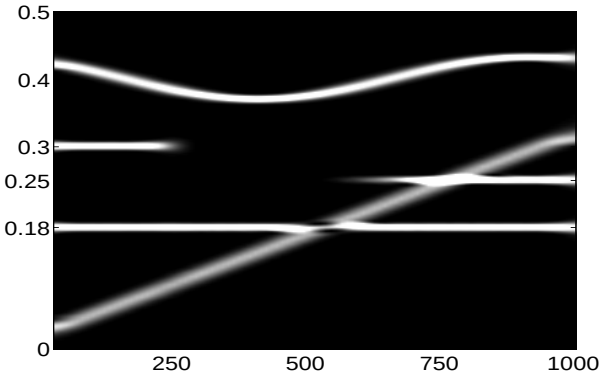

Fig. 5. Spectrogram of the second data (1000 time sample long) computed with a Hamming window (171 time sample long).

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.

4 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 multitarget tracking scenarios is currently being investigated.

References

[1] Andrieu C., N. De Freitas and Doucet, A., "Sequential MCMC for Bayesian Model Selection", Proc. Work. IEEE HOS 1999.

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

[3] Davy M., Leprettre B., Doncarli C. and Martin N., "Tracking of spectral lines in an ARCAP time-frequency representation", EUSIPCO-98, Rhodes Island, Greece, 1998.

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

[5] Doucet, A., Gordon, N.J. and Krishnamurthy, V., "Particle filters for state estimation of jump Markov linear systems," *IEEE Trans. Signal Processing*, vol. 49, no.3, pp. 613-624, 2001.

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

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

[8] Gilks W.R. and Berzuini C., "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] Ikoma N. and Maeda H., "Nonstationary spectral peak estimation by Monte Carlo filter", Proc. Conf. Adaptative Systems for Signal Processing, Canada, 2000.

[10] Julier, S.J. and Uhlmann, J.K., "A general method for approximating nonlinear transformations of probabilitity distributions", *Technical Report*, Engineering Dept., University of Oxford.

[11] Julier, S.J. and Uhlmann, J.K., "A new extension of the Kalman filter to nonlinear systems", Proceedings of AeroSense, Orlando, 1997.

[12] Julier, S.J., "The scaled unscented transformation", to appear in *Automatica*, 2002.

[13] Chang-Jin Kim and Charles R. Nelson, *State-Space Models with Regime Switching*, MIT Press, 1999.

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

[15] Pitt M.K. and Shephard N., "Filtering via simulation: auxiliary particle filter," *J. Am. Statist. Ass.*, vol. 94, pp. 590-599, 1999.

[16] Robert, C.P. and Casella, G., *Monte Carlo Statistical Methods*, Springer-Verlag, 1999.

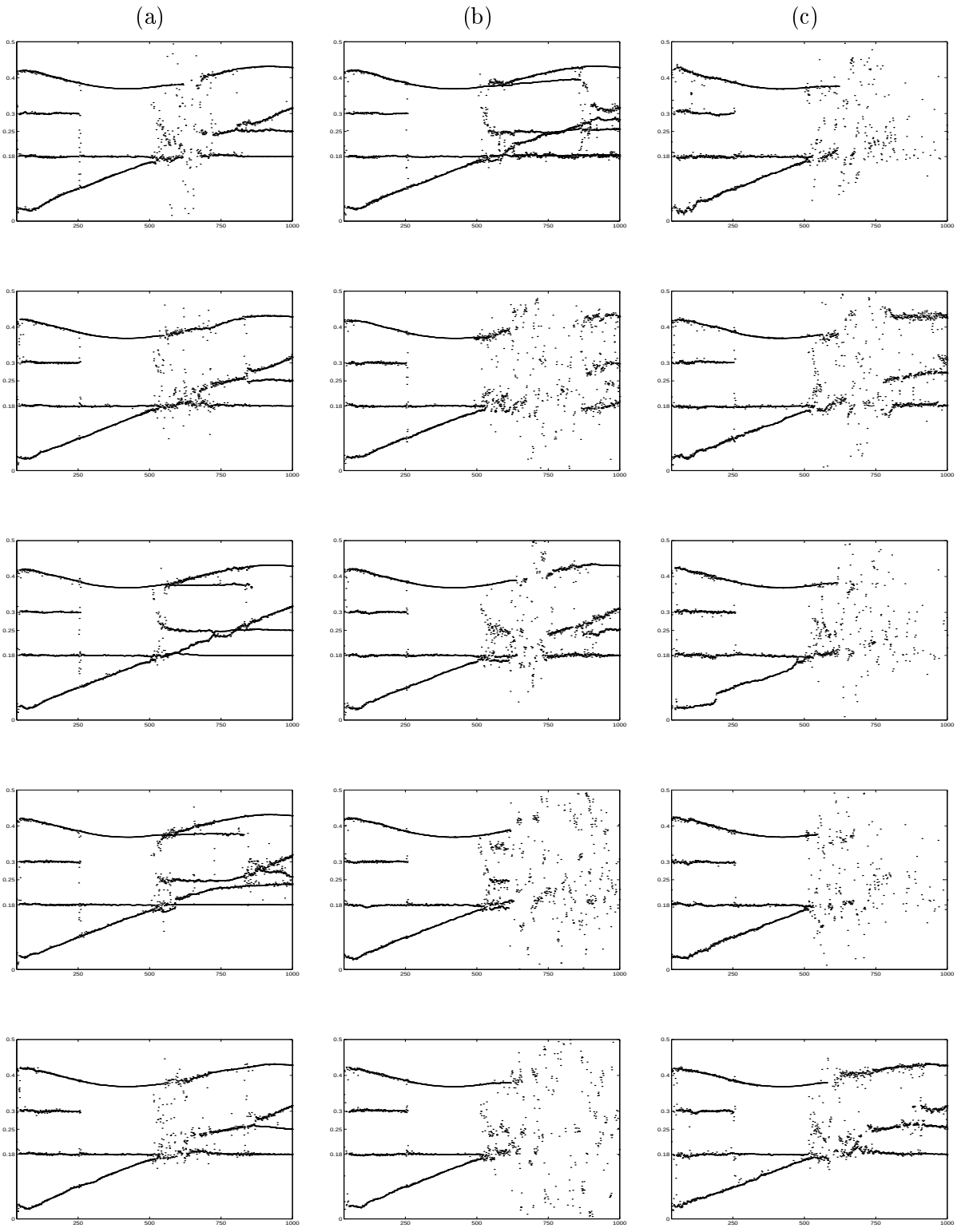

Fig. 6. Comparison of our particle filter with the standard SIR algorithm. Column (a): five frequency estimations with our algorithm ($N_{\text{Improved algo.}} = 500$ particles). Column (b): five frequency estimations with the SIR algorithm ($N_{\text{SIR algo.}} = 1800$ particles). This number of particles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