Joint Detection and Tracking of Time-Varying Harmonic Components: A Flexible Bayesian Approach

Corentin Dubois and Manuel Davy

***Abstract*—This paper addresses the joint estimation and detection of time-varying harmonic components in audio signals. We follow a flexible viewpoint, where several frequency/amplitude trajectories are tracked in spectrogram using particle filtering. The core idea is that each harmonic component (composed of a fundamental partial together with several overtone partials) is considered a target. Tracking requires to define a state-space model with state transition and measurement equations. Particle filtering algorithms rely on a so-called sequential importance distribution, and we show that it can be built on previous multipitch estimation algorithms, so as to yield an even more efficient estimation procedure with established convergence properties. Moreover, as our model captures all the harmonic model information, it actually separates the harmonic sources. Simulations on synthetic and real music data show the interest of our approach.**

***Index Terms*—Audio signal analysis, Bayesian filtering, harmonic structure, multipitch estimation, particle filtering, Rao–Blackwellization, time-frequency representation, time-varying amplitude/frequency tracking.**

I. Introduction

HUMAN listeners build on their own pitch perception to understand and separate sounds from the environment. Individual sounds are mixed up when they reach the listener ears and overlap both in the time domain and in the frequency domain. Despite this sound mixing phenomenon, the human brain can recover the characteristics of each individual sound-producing event, making efficient use of the perceived pitches [1]. While the pitch separation process is unconscious for humans, such a sound analysis is hard to perform using a computer, and the design of multiple fundamental frequency (F0) estimation algorithm still sparks off research. Applications of this are computational auditory scene analysis (CASA), music information retrieval (MIR), recognition and classification of various kinds of sounds (speech, music, environmental sounds, and so on), speech/music processing and, in particular, the automatic speech/music transcription tasks (see [2] and [3] for overall discussions). Pitch tracking is inherently related to the detection and estimation of sine waves, and time-varying frequency/amplitude estimation has become a central problem in audio processing. This problem even finds applications outside the audio field, in, e.g., vibration monitoring of mechanical systems. However, audio processing has received most of the research effort.

For analysis purposes, sounds are classically split into a harmonic (tonal) part and a nonharmonic (nontonal) part. The harmonic part corresponds to a periodic or nearly periodic sound and may be modeled as the sum of several *harmonic components* (HCs), that is a *fundamental partial* together with several *overtone partials* whose frequencies are related to that of the fundamental. The nonharmonic part corresponds to nontonal sounds (for instance, percussive instruments which produce transient sounds). This paper addresses the analysis of the harmonic part, while the nonharmonic part is modeled as an additive noise. The problem of detecting and tracking time-varying HCs is made difficult by the need to jointly estimate the number of HCs together with their individual set of parameters: birth time, frequency/amplitude of each partials at each time sample, and death time.

A. Literature Review

We present here a review of the most recent works. The reader interested in other previous works may refer to [3] and [4]. Although nonexclusive, three main classes of methods can be identified: 1) those based on an auditory model; 2) those making efficient use of knowledge insertion and perceptual grouping rules (as, for instance, gestalt principles given by Bregman [5]); and 3) those using a parametric model. In the first class, approaches often have two steps: a frequency analysis is first performed to obtain a multichannel representation, then periodicity is analyzed within each channel [6] thanks to the autocorrelation function (ACF). Since the ACF generalizes poorly to the polyphonic case [7], some authors have proposed several improvements [8], [9], that generally apply iterative procedures to estimate the individual HCs [10]–[12]. Blackboard architectures [13]–[16] are typical example methods in the second class. It should be noted that prior information may be used in other ways, as proposed in [17]–[19]. In the third class of approaches, Yeh *et al.* [20] propose a multiple F0 estimation method in musical signals where the number of F0s is known in advance. Irizarry [21] proposes a method which is based on local harmonic estimation, where a parametric harmonic structure is matched framewise to the signal, using weighted least squares. The number of harmonic components is known and fixed to one. The parametric approach has also been addressed in a fully statistical Bayesian way. Goto [22], [23] proposes a method aimed at estimating the melody and bass lines in real-world musical

Manuscript received January 3, 2006; revised October 18, 2006. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. George Tzanetakis.

C. Dubois is with the IRCCyN, 44321 Nantes Cedex 03, France (e-mail: corentin.dubois@irccyn.ec-nantes.fr).

M. Davy is with the CNRS/LAGIS and the INRIA-FUTURS SequeL Team, Citi Scientifique, 59651 Villeneuve d'Ascq Cedex, France (e-mail: manuel.davy@inria.fr).

Digital Object Identifier 10.1109/TASL.2007.894522

audio signals. Raphael [24], [25] proposes to assign a "chord" label to a signal frame using a hidden Markov model (HMM) where the state parameter is the label vector and the observation is a feature vector computed from the signal frame. Davy *et al.* [26], [27] propose an offline method to analyze segments of audio where no musical note changes occur. The description of such segments is based on a polyphonic harmonic model where the number of partials, the time-varying amplitudes, the fundamental and overtone partial frequencies, and the noise variances are unknown and estimated using Markov Chain Monte Carlo (MCMC). Cemgil *et al.* [28], [29] develop a generative model for music transcription that comprises both physical (signal) and expressive descriptions of the sound generation procedure. The method consists of first defining a set of intermediate variables based on a score and a timer model, and second, a signal model. The piano-roll, parameters and hyperparameters involved in the state-space model are estimated within an approximate Bayesian procedure. Vincent and Plumbley [30] adopt a Bayesian approach in order to estimate the number of *pitched object* together with their parameters. A posterior is derived from a likelihood and local priors, and approximate maximum a posteriori estimates of these unknown parameters are computed using an iterative deterministic jump method.

These methods all make simplifying assumptions, showing the difficulty of the overall estimation problem. In many real-world audio processing tasks, it is not realistic to assume that some of the parameters are known in advance (e.g., the number of harmonic sounds). For example, consider audio signals coming from the urban sound atmosphere: they are composed of many sounds: speech, transport sounds, urban noises, etc. This composite structure precludes the design of a very precise sound model (which becomes possible whenever the analyzed sound type is known: guitar, piano, etc.).

B. Bayesian Framework

This paper proposes to define a family of Bayesian models aimed at addressing jointly the detection of HCs and the estimation of their parameters. Our models are characterized as follows: at each time t, a state vector $\boldsymbol{\mathcal{S}}_t = \{k_t, \boldsymbol{\theta}_t\}$ containing all the desired information about the harmonic audio signal has to be estimated. Here, k_t denotes the number of HCs and $\boldsymbol{\theta}_t$ denotes the vector, which size depends on k_t, containing the time-varying frequencies, amplitudes, and initial phase of each partial.

The only information available is the time-domain signal; therefore, the state is not observed directly, and we have to resort to a hidden state estimation method. Here, the observation vector $\mathbf{y}_t$ is defined as the product of the signal with a window centered at time t (i.e., $\mathbf{y}_t$ is a frame). It is related to the state $\boldsymbol{\mathcal{S}}_t = \{k_t, \boldsymbol{\theta}_t\}$ by a likelihood function

$$\mathbf{y}_t \sim p\left(\mathbf{y}_t | k_t, \boldsymbol{\theta}_t\right). \qquad (1)$$

Moreover, time evolution equations define the transitions between the state parameters $\boldsymbol{\mathcal{S}}_{t-1}$ at time $t-1$ and $\boldsymbol{\mathcal{S}}_t$ in a probabilistic way. In the Bayesian framework, this corresponds to a sequential prior, denoted

$$\{k_t, \boldsymbol{\theta}_t\} \sim p\left(k_t, \boldsymbol{\theta}_t | k_{t-1}, \boldsymbol{\theta}_{t-1}\right). \qquad (2)$$

Together with the initial distribution $p\left(k_0, \boldsymbol{\theta}_0\right)$, the likelihood function and the sequential prior define a so-called *Jump Markov System* (JMS) and the estimation of the state parameter $\{k_{0:t}, \boldsymbol{\theta}_{0:t}\}$[1] given $\mathbf{y}_{1:t}$, is performed from the posterior distribution $p\left(k_{0:t}, \boldsymbol{\theta}_{0:t} | \mathbf{y}_{1:t}\right)$.

We are interested in estimating this distribution sequentially and, in particular, one of its marginals, the so-called filtering density $p\left(k_t, \boldsymbol{\theta}_t | \mathbf{y}_{1:t}\right)$. Several reasons motivate this online approach. In some applications, data arrive sequentially and one wants to process them samplewise rather than in a fully batch way in order to reduce memory requirements and the computational burden as well as allow real-time processing.[2] Batch methods, though, provide smooth estimates. Smoothing methods in sequential context can also be implemented: for instance, one can perform the estimation with some time lag T, which comes down to estimating the density $p\left(k_t, \boldsymbol{\theta}_t | \mathbf{y}_{1:t+T}\right)$ with $T > 0$, or implement a block sampling strategy [31]. However, all these approaches may be built on a particle filter which focuses on the filtering distribution, and we have preferred to stick to this simpler framework for the sake of presentation clarity.

One of the main difficulties in this approach is that the optimal filtering problem can be solved in closed-form only in a few special cases including linear Gaussian state space models where the solution is given by the Kalman Filter. In the general case, many approaches have been proposed to approximate these distributions and particle filters (PFs) [32] have recently revolutionized this field. They allow for recursive state parameters estimation of virtually any dynamic model,[3] even if the model is nonlinear and/or non-Gaussian as is the case here. In order to explain the principle of PFs, consider the simple state-space model where $\boldsymbol{\mathcal{X}}_t$ is any state parameter defined in a continuous space[4] at time t. The idea of the basic PF algorithm is to update, at each time t, a set of N particles $\left\{\boldsymbol{\mathcal{X}}_{0:t}^{(i)}\right\}_{i=1,\ldots,N}$ distributed according to the posterior $p\left(\boldsymbol{\mathcal{X}}_{0:t} | \mathbf{y}_{1:t}\right)$. Since it is generally impossible to sample directly from the posterior, the particles are sampled from a sequential importance density and weighted by $\omega_t^{(i)}$ which is the ratio between the posterior and the importance density. Then, an estimate $\hat{E}_N\left(h_t(\boldsymbol{\mathcal{X}}_{0:t})\right)$ of the posterior expectation

$$E\left(h_t(\boldsymbol{\mathcal{X}}_{0:t})\right) = \int h_t(\boldsymbol{\mathcal{X}}'_{0:t}) p(\boldsymbol{\mathcal{X}}'_{0:t} | \mathbf{y}_{1:t}) d\boldsymbol{\mathcal{X}}'_{0:t} \qquad (3)$$

with h_t a given function such that the previous integral is well defined, is obtained by Monte Carlo integration, as follows:

$$\hat{E}_N\left(h_t(\boldsymbol{\mathcal{X}}_{0:t})\right) = \sum_{i=1}^{N} \omega_t^{*(i)} h_t\left(\boldsymbol{\mathcal{X}}_{0:t}^{(i)}\right) \qquad (4)$$

[1] For a vector $\boldsymbol{\beta}$, we adopt the notation $\boldsymbol{\beta}_{a:b} \triangleq \{\boldsymbol{\beta}_a, \boldsymbol{\beta}_{a+1}, \ldots, \boldsymbol{\beta}_b\}$.

[2] It should be outlined here that the method proposed in this paper is not yet real-time but efficient implementations of particle filters make it a reasonable short time perspective.

[3] In particular, a specific PF algorithm for JMS is described in [33].

[4] In our case, the state vector has a discrete part as well as a continuous part, and we have to design more sophisticated estimators than in the basic continuous case presented here.

where

$$\omega_t^{*(i)} = \frac{\omega_t^{(i)}}{\sum\limits_{j=1}^{N} \omega_t^{(j)}}. \tag{5}$$

The importance distribution plays a central role regarding the efficiency of the PF: its design is critical. Indeed, there is no restrictions in defining such a distribution, as long as its support includes that of the posterior. The importance distribution needs to explore the entire state space, with a special emphasis onto regions where the posterior is large, and its ability to actually do this has a direct impact on the algorithm efficiency. In our case, the importance distribution may be based on a (randomized) single frame based multiple F0 estimation algorithm, as those reviewed above. Thus, our method offers a means of combining such state-of-the-art (deterministic) algorithms with a sequential Bayesian estimation framework and, thanks to the temporal continuity constraint, compute possibly more robust estimates.

C. Paper Organization

This paper is organized as follows: Section II introduces the Bayesian sequential harmonic model. Since the main idea is to detect and track frequency/amplitude trajectories in a flexible way, we define general transition and observation distributions that can be derived into several specific models. In Section III, we describe the sequential Monte Carlo (particle filtering) algorithm aimed at estimating sequentially the state parameter. This algorithm is discussed in Section IV. Section V presents simulation results on synthetic and real signals. Finally, some conclusions and future research directions are proposed in Section VI.

II. Sequential Bayesian Harmonic Model

The information that we want to extract from audio signals is summarized in a state vector $\boldsymbol{\mathcal{S}}_t = \{k_t\boldsymbol{\theta}_t\}$, $t = 1, 2, \ldots,$, and the online estimation is performed by adopting a JMS as dynamic model. This section is devoted to the definition of the required likelihood and prior.

A. Grounds of our Approach

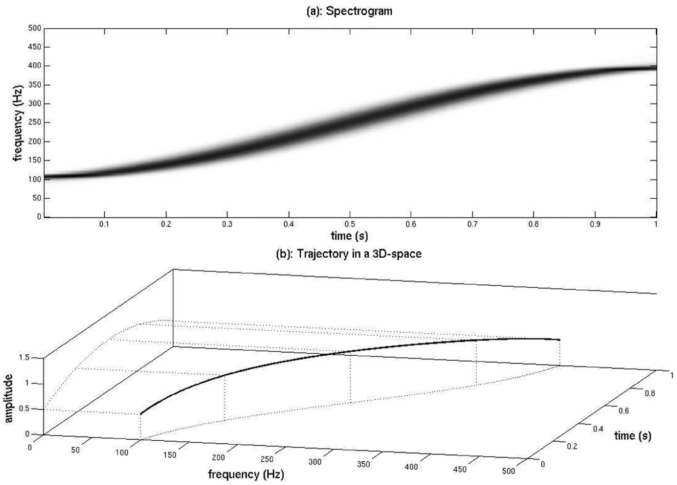

Fig. 1. (a) shows the spectrogram of a signal x. (b) shows the corresponding trajectory that we want to track over time.

Time-frequency representations (TFRs) [34], [35] provide natural signal representations suited to the joint detection and tracking of time-varying HCs task. More precisely, the short-time Fourier transform (STFT) usually separates well the individual frequency components and allows modelling of their evolution. Let $\mathbf{x} = [x_1, x_2, \ldots,]$ be a discrete time audio signal. The analysis is made local around time t by considering a window $\mathbf{w}_t$ of length $L_\mathbf{w}$ centered at time t. The STFT of the signal is

$$\mathrm{STFT}_{t,f}^{\mathbf{w}}(\mathbf{x}) = \mathrm{DFT}(\mathbf{x} \cdot \mathbf{w}_t) \tag{6}$$

where t denotes discrete time, f denotes (normalized) discrete frequency, and DFT is the discrete Fourier transform. The STFT maps the time-domain signal into the time-frequency (TF) plane from which the estimation of the state parameters $\boldsymbol{\mathcal{S}}_t$ is easier: indeed, in this space, each component is a trajectory in a 3-D-space whose coordinates are the frequency and the amplitude at time t, as it is shown in Fig. 1. However, this mapping to the TF plane is here conceptual and outlined to make the method understandable. Indeed, due to Parseval theorem, actually going to the TF plane is not necessary and the observation $\mathbf{y}_t$ finally used is the windowed signal: it is a vector of length $L_\mathbf{w}$ for all t. More precisely

$$\mathbf{y}_t = \mathbf{x} \cdot \mathbf{w}_t. \tag{7}$$

Here, the window $\mathbf{w}$ can have any shape (rectangular, Gauss, Hamming, etc.), with their own advantages and drawbacks.

Notations & Model

At each time t, we model the observation $\mathbf{y}_t$ as a sum of k_t stationary[5] HCs [26]. Each HC $j (j = 1, \ldots, k_t)$ is composed of H partials with frequencies denoted $f_{t,j,h}$ for $h = 1, \ldots, H$. The fundamental partial corresponds to $h = 1$, and H is the user-defined number of overtone partials (assumed large in general).

For perfectly harmonic signals, we have $f_{t,j,h} = hf_{t,j,1}$ for $h = 2, \ldots, H$. In that case, the frequencies of the overtone partials need not be estimated. One can choose to this model in a limited number of cases (speech for instance). Inharmonicity[6] can also be considered and the overtone partial frequencies have to be estimated. This can be done in different ways. The simplest is certainly to estimate each frequency independently of the others. A more robust solution would be to model each overtone partial frequency as $f_{t,j,h} = c_{t,j,h}hf_{t,j,1}$, for $h = 2, \ldots, H$. In that case, one has to estimate the fundamental frequency $f_{t,j,1}$ together with the inharmonicity parameters $c_{t,j,h}$ that allow for

[5] This means that the signal $\mathbf{x}$ is actually assumed to be approximatively stationary with respect to the length $L_\mathbf{w}$ of the window. This does not preclude nonstationarity over several windows.

[6] Inharmonicity is a standard phenomenon which arises when overtone partial frequencies deviate from integer multiples of the frequency of the fundamental partial.

a deviation from the overtone partial frequency $hf_{t,j,1}$. To prevent possible partials overlapping, one can hale each parameter $c_{t,j,h}$ to live in a compact, for instance $[0.8, 1.2]$. Last, in specific cases, together with these models, one can use an inharmonicity model. For example, the frequencies of overtone partials of plucked strings instruments obey [36]

$$f_{t,j,h} = hf_{t,j,1}\sqrt{1+(h^2-1)\beta}. \tag{8}$$

Fletcher and Rossing [37] propose the following piano inharmonicity model:

$$f_{t,j,h} = hf_{t,j,1}\sqrt{\frac{1+h^2\beta}{1+\beta}}. \tag{9}$$

Since the algorithm that we present in Section III is able to deal with all these frequency models and for notation convenience, we denote $\mathbf{f}_t$ the vector containing all the frequencies or frequency parameters [such as β in (8) or (9)] we need to estimate, with

$$\mathbf{f}_t = \left[f_{t,1,1}, \ldots, f_{t,j,h}, \ldots, f_{t,k_t,H}\right]^{\mathrm{T}}. \tag{10}$$

In addition to the frequency, a sine wave is classically characterized by two additional parameters, its amplitude and its initial phase, with the standard form $A\cos(2\pi f t + \varphi)$. This modelling is highly nonlinear in both f and φ, and we prefer to use the equivalent expression $a\cos(2\pi f t) + b\sin(2\pi f t)$ where the initial phase is replaced by a second (linear) amplitude parameter. As a result, one has to estimate two amplitudes denoted $a_{t,j,h}$ and $b_{t,j,h}$, $j = 1, \ldots, k_t$, and $h = 1, \ldots, H$, in addition to the frequency $f_{t,j,h}$. Once again, one can adopt several models for the amplitude estimation. The simplest one is again to not impose a parametric shape for the amplitude decay profile (from one partial to another), and the amplitudes of each partial of a given HC are estimated independently. Another solution is to estimate amplitudes dependently introducing a smooth model of the amplitude decay profile. In that case, one only needs to estimate the parameters of the parametric model.

The continuous vector to be estimated at time t is then $\boldsymbol{\theta}_t = [\mathbf{f}_t, \mathbf{a}_t, \mathbf{b}_t]$, where $\mathbf{a}_t$ and $\mathbf{b}_t$ are built as $\mathbf{f}_t$. Note that the dimension of $\boldsymbol{\theta}_t$ depends on the time and equals, at most, $3k_tH$.

B. Likelihood

Let us define the function g_t as follows:

$$\begin{aligned} g_t : \mathbb{N} \times \mathbb{R}^{3k_tH} &\longrightarrow \mathbb{R}^{L_{\mathbf{w}}} \\ (k_t, \boldsymbol{\theta}_t) &\longmapsto \mathbf{s}_t \cdot \mathbf{w} \end{aligned} \tag{11}$$

with, for $\tau = 1, \ldots, L_{\mathbf{w}}$

$$\mathbf{s}_t[\tau] = \sum_{j=1}^{k_t}\sum_{h=1}^{H} a_{t,j,h}\cos(2\pi f_{t,j,h}\tau) + b_{t,j,h}\sin(2\pi f_{t,j,h}\tau). \tag{12}$$

In practice, the actual number of overtone partials can be less than H: if the frequency of a partial is larger than the Nyquist frequency, it is not taken into account and their amplitudes are set to 0.

The observation equation is

$$\mathbf{y}_t = g_t(k_t, \boldsymbol{\theta}_t) + \mathbf{v}_t^{\mathbf{y}} \tag{13}$$

where the components of $\mathbf{v}_t^{\mathbf{y}}$ are i.i.d. zero-mean Gaussian noises of fixed variance $r^{\mathbf{y}}$. Equation (13) yields the likelihood

$$\begin{aligned} p\left(\mathbf{y}_t | k_t, \boldsymbol{\theta}_t\right) &= \mathcal{N}\left(\mathbf{y}_t; g_t(k_t, \boldsymbol{\theta}_t), r^{\mathbf{y}}\right) \\ &= \frac{1}{\left(2\pi r^{\mathbf{y}}\right)^{L_{\mathrm{w}}/2}}\exp\left(-\frac{\left\|\mathbf{y}_t - g_t(k_t, \boldsymbol{\theta}_t)\right\|^2}{2r^{\mathbf{y}}}\right). \end{aligned} \tag{14}$$

When the number of HCs equals zero, the likelihood becomes

$$p\left(\mathbf{y}_t | k_t = 0\right) = \frac{1}{\left(2\pi r^{\mathbf{y}}\right)^{L_{\mathrm{w}}/2}}\exp\left(-\frac{\left\|\mathbf{y}_t\right\|^2}{2r^{\mathbf{y}}}\right). \tag{15}$$

C. Prior

Our approach to track HCs is similar to target tracking where the state vector is classically composed of the target coordinates, and its evolution between two successive times is described by physical equations. These equations are easy to obtain when the possible moves of the target are known. In music, dynamic equations could be defined for a given instrument playing in well-controlled conditions. For example, considering a guitar, physics of the strings can describe the vibration mechanism. In the general case, it is quite difficult to define equations that would be general and tractable enough, on the evolution of k_t and $\boldsymbol{\theta}_t$. For instance, environmental sounds are composed of several kinds of sound (speech, transport sounds, urban noises, etc. …) and a model that describes well each of them is hard to define. We have adopted transition equations that are based on the heuristic argument that the frequencies and amplitudes trajectories are smooth. In other words, one considers that there is no reason for abrupt change of frequency or amplitude within a HC and that abrupt changes correspond to birth/death of HCs. Thus, at time t, the transition equations are given by, for $j = 1, \ldots, k_t$ and $h = 1, \ldots, H$

$$f_{t,j,h} = \psi_T^{\mathbf{f}}(f_{t-T+1:t-1,j,h}) + v_{t-1,j,h}^{\mathbf{f}} \tag{16}$$

$$a_{t,j,h} = \psi_T^{\mathbf{a}}(a_{t-T+1:t-1,j,h}) + v_{t-1,j,h}^{\mathbf{a}} \tag{17}$$

$$b_{t,j,h} = \psi_T^{\mathbf{b}}(b_{t-T+1:t-1,j,h}) + v_{t-1,j,h}^{\mathbf{b}} \tag{18}$$

where $\psi_T^{(\cdot)}$ is any smoothing function that proposes a value for the parameter at time t taking into account the past up to $t - T + 1$. Note that to keep the Markovian property, the state vector needs to include all the parameters from time $t - T + 1$ to time $t - 1$. The function ψ can also take into account some other features, such as the derivative of the parameters. This includes, for example, the Bayesian sequential formulation of smoothing splines [38], [39]. Once again, the flexibility of the approach allows for any choice of function ψ. Considering $T = 1$ and $\psi = \mathbf{1}$ is equivalent to consider a random walk centered on

TABLE I
EXAMPLE OF TRANSITION PROBABILITIES FOR k_t. THE PROBABILITIES b_t, e_t, AND d_t ARE DEFINED IN (25)

	$k_{t-1} = k_{min}$	$k_{min} < k_{t-1} < k_{max}$	$k_{t-1} = k_{max}$
b_t	1/10	1/10	0
e_t	9/10	8/10	9/10
d_t	0	1/10	1/10

parameter value at time $t-1$. In (16)–(18), $v^{(\cdot)}_{t-1,j,h}$ is a zero-mean white noise with Gaussian density of variance $r^{(\cdot)}_{t-1,j,h}$. We allow the variances $r^{(\cdot)}_{t-1,j,h}$ to evolve with time according to, for $j = 1, \ldots, k_t$ and $h = 1, \ldots, H$

$$\log\left(r^{\mathbf{f}}_{t,j,h}\right) = \log\left(r^{\mathbf{f}}_{t-1,j,h}\right) + \varphi_{t-1,j,h} \tag{19}$$
$$\log\left(r^{\mathbf{a}}_{t,j,h}\right) = \log\left(r^{\mathbf{a}}_{t-1,j,h}\right) + \alpha_{t-1,j,h} \tag{20}$$
$$\log\left(r^{\mathbf{b}}_{t,j,h}\right) = \log\left(r^{\mathbf{b}}_{t-1,j,h}\right) + \beta_{t-1,j,h} \tag{21}$$

with

$$\varphi_{t-1,j,h} \sim \mathcal{N}\left(0, \sigma^2_{\varphi}\right) \tag{22}$$
$$\alpha_{t-1,j,h} \sim \mathcal{N}\left(0, \sigma^2_{\alpha}\right) \tag{23}$$
$$\beta_{t-1,j,h} \sim \mathcal{N}\left(0, \sigma^2_{\beta}\right). \tag{24}$$

Implementing time-varying variances allows the model to adapt to various situations ranging from quasi-stationarity to quick frequency/amplitude changes.[7] To ensure the positivity of variances, the evolution model is defined on variance logarithms. The values of σ^2_{φ}, σ^2_{α}, and σ^2_{β} are fixed roughly and do not require fine tuning. Our purpose being to allow parameters variance to change with time, a simple and efficient choice is again a random walk, see [33] for a discussion.

The model also involves the discrete parameter k_t. It follows a Markov chain with known transition probabilities

$$k_t = k_{t-1} + \begin{cases} 1, & \text{with probability } b_t \\ 0, & \text{with probability } e_t \\ -1, & \text{with probability } d_t. \end{cases} \tag{25}$$

The precise values of the probabilities b_t, e_t and d_t do not have much importance. The aim is to allow k_t to increase or decrease, but the probability to keep it constant must be prominent. In the following, the number of components is assumed bounded; it ranges from $k_{\min}$ (possibly equals to 0) to $k_{\max}$. An example of the values of the probabilities is given in Table I.

An important remark is that limiting the evolution of parameter k_t to be ±1, is not restrictive, as is illustrated by the following example. Assume that the posteriors on the number of HCs at time $t-1$ and t are those given in Fig. 2. Moves of amplitude 1 can explain the evolution from the posterior at time $t-1$ to the one at time t, whereas the maximum *a posteriori* estimates of the number of HCs are $\hat{k}_{t-1} = 5$ and $\hat{k}_t = 2$. A similar remark would be done about the random walks used to define the state parameters evolutions models.

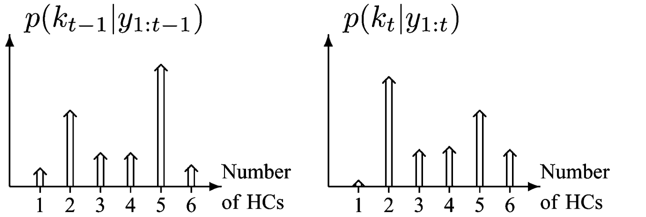

Fig. 2. Example of the posteriors on the number of HCs at time $t-1$ (on the left) and at time t (on the right).

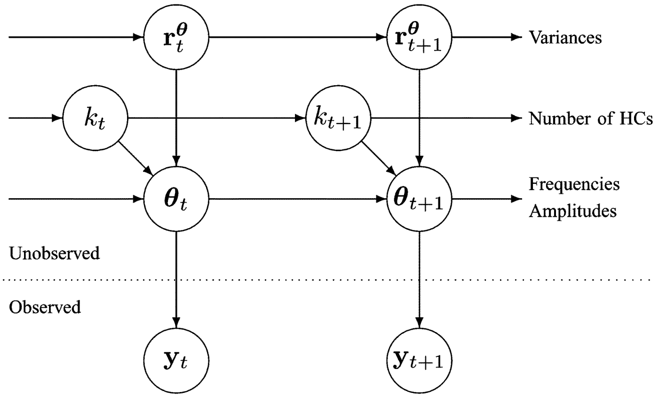

Fig. 3. Overall Bayesian sequential harmonic model. The evolution of the discrete parameter k_t and the continuous vector $\boldsymbol{\theta}_t$ are described by (26)–(28). The relation between the state vector $\boldsymbol{\mathcal{S}}_t = \{k_t, \boldsymbol{\theta}_t\}$ and the observation $\mathbf{y}_t$ is given by (29). The evolution of $\boldsymbol{\theta}_t$ is tuned by the hyperparameters vector $\mathbf{r}^{\boldsymbol{\theta}}_t$ whose evolution is described by (19)–(21).

D. Jump Markov System

The complete JMS model is described graphically in Fig. 3. It can be summarized as follows:

$$k_t \sim P(k_t|k_{t-1}) \tag{26}$$
$$\mathbf{f}_t \sim p(\mathbf{f}_t|\mathbf{f}_{t-1}) \tag{27}$$
$$\begin{bmatrix} \mathbf{a}_t \\ \mathbf{b}_t \end{bmatrix} = A(k_t, \mathbf{f}_t) \begin{bmatrix} \mathbf{a}_{t-1} \\ \mathbf{b}_{t-1} \end{bmatrix} + B(k_t, \mathbf{f}_t)\mathbf{v}'_{t-1} \tag{28}$$
$$\mathbf{y}_t = C(k_t, \mathbf{f}_t) \begin{bmatrix} \mathbf{a}_t \\ \mathbf{b}_t \end{bmatrix} + D(k_t, \mathbf{f}_t)\mathbf{v}''_{t-1} \tag{29}$$

where $\mathbf{v}'_{t-1}$ and $\mathbf{v}''_{t-1}$ are zero-mean white noises with Gaussian density of variance unity. This notation is of special interest whenever the $\psi^{(\cdot)}_T$ function in (17) and (18) is linear and if we seek to estimate amplitudes and not parameters of a parametric envelop model. The algorithm proposed in Section III exploits the attractive features of this case. As will be shown, this enables an efficient variance reduction technique known as "Rao–Blackwellization" and which makes estimation much easier. For a presentation of an algorithm applicable in the general case, see [40]. The probabilities $P(k_t|k_{t-1})$ and $p(\mathbf{f}_t|\mathbf{f}_{t-1})$ are given by (25) and (16), respectively, and matrix A, B, C, and D are completely defined (see Appendix I).

III. SEQUENTIAL MONTE CARLO ALGORITHM

In this section, we present the sequential Monte Carlo algorithm adapted to the general Bayesian framework described above. The idea of Bayesian approaches is to estimate the parameters from the state posterior density, which is proportional

[7] We understand here quick changes as rapid evolutions of the frequency/amplitude at the time scale of the spectrogram, that is about one order of magnitude higher than the window duration.

to the product between the likelihood and the prior. In the dynamic model (26)–(29), the observation equation (29), relating the state parameters at time t to the observation at time t, is not linear in k_t and $\mathbf{f}_t$. This makes the likelihood nonlinear and the problem of estimating the sequence of posterior densities $p(k_{0:t}, \boldsymbol{\theta}_{0:t}|\mathbf{y}_{1:t})$ cannot be solved analytically with the Kalman filter. To overcome these constraints, the state-of-the-art solution is sequential Monte Carlo (SMC). SMC methods allow online parameter estimation by combining the powerful Monte Carlo sampling methods with Bayesian inference [41]. The main idea is the use of a random and adaptive grid of N weighted particles to approximate the posterior density [42].

However, in our case, the observation (29) is nonlinear only in $(k_t, \mathbf{f}_t)$. Actually, conditional on k_t and $\mathbf{f}_t$, the model is linear and Gaussian. Factorizing the posterior density, one can emphasize this as follows:

$$p\left(k_{0:t}, \mathbf{f}_{0:t}, \mathbf{a}_{0:t}, \mathbf{b}_{0:t}|\mathbf{y}_{1:t}\right) = \underbrace{p\left(\mathbf{a}_{0:t}, \mathbf{b}_{0:t}|\mathbf{y}_{1:t}, k_{0:t}, \mathbf{f}_{0:t}\right)}_{(30)^a} \underbrace{p\left(k_{0:t}, \mathbf{f}_{0:t}|\mathbf{y}_{1:t}\right)}_{(30)^b}. \qquad (30)$$

The conditional posterior density $(30)^a$ is Gaussian and its first two moments can be computed analytically whenever $k_{0:t}$ and $\mathbf{f}_{0:t}$ are known. Thus, the estimation of the sequence of posterior densities is performed by combining SMC methods (that is a PF) to approximate $(30)^b$ with Kalman filtering to compute analytically $(30)^a$ [43]. This combination is known as Rao–Blackwellized particle filter (RBPF).

A. RBPF

The RBPF provides, at each time t, a set of N weighted particles

$$\left\{\left(k_{0:t}^{(i)}, \mathbf{f}_{0:t}^{(i)}, \boldsymbol{\mu}_t^{(i)}, \boldsymbol{\Sigma}_t^{(i)}\right), \omega_t^{(i)}\right\}_{i=1,\ldots,N} \qquad (31)$$

where $\omega_t^{(i)}$ is the weight, $\boldsymbol{\mu}_t^{(i)}$ the mean and $\boldsymbol{\Sigma}_t^{(i)}$ the covariance of the Gaussian density $p\left(\mathbf{a}_{0:t}, \mathbf{b}_{0:t}|\mathbf{y}_{1:t}, k_{0:t}^{(i)}, \mathbf{f}_{0:t}^{(i)}\right)$. Then, the marginal posterior density is approximated as follows:

$$\hat{p}(k_{0:t}, \mathbf{f}_{0:t}|\mathbf{y}_{1:t}) = \sum_{i=1}^{N} \omega_t^{(i)} \delta_{\left(k_{0:t}^{(i)}, \mathbf{f}_{0:t}^{(i)}\right)}(k_{0:t}, \mathbf{f}_{0:t}) \qquad (32)$$

where δ denotes the Dirac function. The conditional density of $[\mathbf{a}_{0:t} \quad \mathbf{b}_{0:t}]^{\mathrm{T}}$ is a Gaussian mixture

$$\begin{aligned}\hat{p}(\mathbf{a}_{0:t}, \mathbf{b}_{0:t}|\mathbf{y}_{1:t}) &= \sum_{i=1}^{N} \omega_t^{(i)} p\left(\mathbf{a}_{0:t}, \mathbf{b}_{0:t}|\mathbf{y}_{1:t}, k_{0:t}^{(i)}, \mathbf{f}_{0:t}^{(i)}\right) \\ &= \sum_{i=1}^{N} \omega_t^{(i)} \mathcal{N}\left(\mathbf{a}_{0:t}, \mathbf{b}_{0:t}; \boldsymbol{\mu}_t^{(i)}, \boldsymbol{\Sigma}_t^{(i)}\right). \qquad (33)\end{aligned}$$

The PF is sequential and it updates the set of particles from time t to time $t+1$. Thus, the densities (32) and (33) are updated at time $t+1$. The basic PF algorithm consists of two steps: first sequential importance sampling and, second, resampling. The RBPF is similar to the PF but, for each sample $\left(k_t^{(i)}, \mathbf{f}_t^{(i)}\right)$, $i = 1, \ldots, N$, we update the mean and the covariance matrix of the amplitudes using exact computations. The outline of the RBPF algorithm is presented in Table II. Below, we give detailed explanations about the RBPF algorithm.

TABLE II
OUTLINE OF THE RBPF ALGORITHM AT TIME t

```
At time $t = 0$
    Initialization
    For $i = 1 \ldots N$
        Sample $\left(k_t^{(i)}, \mathbf{f}_t^{(i)}\right)$ from $p(k_0, \mathbf{f}_0)$
        Set $\boldsymbol{\mu}_t^{(i)} = \boldsymbol{\mu}_0^{(i)}$ and $\boldsymbol{\Sigma}_t^{(i)} = \boldsymbol{\Sigma}_0^{(i)}$
        Set $\mathbf{r}_t^{(i)} = \mathbf{r}_0^{(i)}$
At time $t \geq 1$
    Sequential Importance Sampling
    For $i = 1 \ldots N$
        Update the particles
            Sample $k_t^{(i)} \sim P(k_t|k_{t-1})$
            Sample $\mathbf{f}_t^{(i)} \sim q_t(\mathbf{f}_t^{(i)}|\mathbf{f}_{t-1}^{(i)})$. See Section III-B.
            Compute $\boldsymbol{\mu}_t^{(i)}$ and $\boldsymbol{\Sigma}_t^{(i)}$ from $\mathbf{y}_t$, $k_t^{(i)}$, $\mathbf{f}_t^{(i)}$, $\boldsymbol{\mu}_{t-1}^{(i)}$ and $\boldsymbol{\Sigma}_{t-1}^{(i)}$
            with the equations of the Kalman filter.
        Update the hyperparameters
            Sample $\mathbf{r}_t^{(i)} \sim p(\mathbf{r}_t|\mathbf{r}_{t-1}, \mathbf{I}_t)$ as described in Section III-C.
        Compute the weights
            Use (37).
    Normalize the weights, such that $\sum_{i=1}^{N} \omega_t^{(i)} = 1$.
    Resampling: duplicate particles with high weights, suppress
    particles with low weights and set $\omega_t^{(i)} = \frac{1}{N}$.
    Output Estimates $\hat{k}_t$, $\hat{\mathbf{f}}_t$ and $[\hat{\mathbf{a}}_t \ \hat{\mathbf{b}}_t]^{\mathrm{T}}$ are computed as explained
    in (40)-(42).
```

1) Sequential Importance Sampling: This procedure has three main steps. First, one extends each particle $\left(k_{0:t}^{(i)}, \mathbf{f}_{0:t}^{(i)}\right)$ at time $t+1$. Since we cannot sample from the marginal posterior $p\left(k_{0:t+1}, \mathbf{f}_{0:t+1}|\mathbf{y}_{1:t+1}\right)$ directly, we introduce an appropriate importance density $\pi_{t+1}\left(k_{0:t+1}, \mathbf{f}_{0:t+1}\right)$ from which $\left(k_{0:t+1}^{(i)}, \mathbf{f}_{0:t+1}^{(i)}\right)$ are sampled. More precisely, the importance density is chosen so as to be written sequentially as

$$\pi_{t+1}\left(k_{0:t+1}, \mathbf{f}_{0:t+1}\right) = \pi_t\left(k_{0:t}, \mathbf{f}_{0:t}\right) q_{t+1}\left(k_{t+1}, \mathbf{f}_{t+1}|k_t, \mathbf{f}_t\right). \qquad (34)$$

The particles at time t are extended with a new state vector $\left(k_{t+1}^{(i)}, \mathbf{f}_{t+1}^{(i)}\right)$, sampled from q_{t+1}. The second step is the update of $\boldsymbol{\mu}_t^{(i)}$ and $\boldsymbol{\Sigma}_t^{(i)}$. The mean $\boldsymbol{\mu}_{t+1}^{(i)}$ and the variance $\boldsymbol{\Sigma}_{t+1}^{(i)}$ are computed from $\mathbf{y}_{t+1}$, $k_{t+1}^{(i)}$, $\mathbf{f}_{t+1}^{(i)}$, $\boldsymbol{\mu}_t^{(i)}$, and $\boldsymbol{\Sigma}_t^{(i)}$ with the equations of the Kalman filter given in Appendix II. Finally, the weights are computed: the particles $\left(k_{0:t+1}^{(i)}, \mathbf{f}_{0:t+1}^{(i)}\right)$ are sampled from π_{t+1}, and the density one wants to approximate is $p\left(k_{0:t+1}, \mathbf{f}_{0:t+1}|\mathbf{y}_{1:t+1}\right)$. The weights correct this discrepancy

$$\omega_{t+1}^{(i)} = \frac{p\left(k_{0:t+1}^{(i)}, \mathbf{f}_{0:t+1}^{(i)}|\mathbf{y}_{1:t+1}\right)}{\pi_{t+1}\left(k_{0:t+1}^{(i)}, \mathbf{f}_{0:t+1}^{(i)}\right)}. \qquad (35)$$

Thanks to the factorized shape of π_{t+1} in (34), and to the following recursive formula:

$$\begin{aligned}p\left(k_{0:t+1}, \mathbf{f}_{0:t+1}|\mathbf{y}_{1:t+1}\right) &= p\left(k_{0:t}, \mathbf{f}_{0:t}|\mathbf{y}_{1:t}\right) \\ &\times \frac{p\left(\mathbf{y}_{t+1}|k_{0:t+1}, \mathbf{f}_{0:t+1}, \mathbf{y}_{0:t}\right) p\left(k_{t+1}, \mathbf{f}_{t+1}|k_t, \mathbf{f}_t\right)}{p\left(\mathbf{y}_{t+1}|\mathbf{y}_{1:t}\right)} \qquad (36)\end{aligned}$$

the weights are computed sequentially as

$$\omega_{t+1}^{(i)} \propto \omega_t^{(i)} p\left(\mathbf{y}_{t+1}|k_{0:t+1}^{(i)}, \mathbf{f}_{0:t+1}^{(i)}, \mathbf{y}_{0:t}\right) \times \frac{p\left(k_{t+1}^{(i)}, \mathbf{f}_{t+1}^{(i)}|k_t^{(i)}, \mathbf{f}_t^{(i)}\right)}{q_{t+1}\left(k_{t+1}^{(i)}, \mathbf{f}_{t+1}^{(i)}|k_t^{(i)}, \mathbf{f}_t^{(i)}\right)} \tag{37}$$

where $\propto$ denotes "proportional to." The likelihood term is a Gaussian function

$$p\left(\mathbf{y}_{t+1}|k_{0:t+1}^{(i)}, \mathbf{f}_{0:t+1}^{(i)}, \mathbf{y}_{0:t}\right) = \mathcal{N}\left(\mathbf{y}_{t+1}; \mathbf{y}_{t|t-1}^{(i)}, S_t^{(i)}\right) \tag{38}$$

where $\mathbf{y}_{t|t-1}^{(i)}$ and $S_t^{(i)}$ are computed by Kalman filter equations given in Appendix II. Note that this likelihood does not simplify to $p\left(\mathbf{y}_{t+1}|k_{t+1}^{(i)}, \mathbf{f}_{t+1}^{(i)}\right)$ because there is a dependency on past values through $\mathbf{a}_{0:t+1}$ and $\mathbf{b}_{0:t+1}$. The weights are computed up to a normalizing constant, and normalization ensures that the sum equals 1.

2) Resampling: After a few iterations of the previous step, it happens that one particle emerges and has a weight close to 1, and that all the other particles have a weight close to 0. To limit this unavoidable weights degeneracy and reduce this loss of variability, a selection scheme associates to each particle i a number of N_i offsprings, with $\sum_{i=1}^{N} N_i = N$. N_i is randomly related to $\omega_{t+1}^{(i)}$ so that particles with high weights are duplicated and particles with low weights are suppressed.

3) Output: The estimates $\hat{k}_t$ and $\hat{\mathbf{f}}_t$ are computed from the approximation of the posterior $\hat{p}(k_{0:t}, \mathbf{f}_{0:t}|\mathbf{y}_{1:t})$ given in (32) and $[\,\hat{\mathbf{a}}_t \quad \hat{\mathbf{b}}_t\,]^{\mathrm{T}}$ are computed from the approximation of the conditional density $\hat{p}(\mathbf{a}_{0:t}, \mathbf{b}_{0:t}|\mathbf{y}_{1:t})$ given in (33). More precisely, the estimation of the number of HCs at time t is computed by marginal maximum a *posteriori*: it corresponds to the value of k_t that is most represented in the particles

$$\hat{k}_t = \underset{k_{\min} \le k \le k_{\max}}{\arg\max} \sum_{i \in \left\{ i' | 1 \le i' \le N, k_t^{(i')} = k \right\}} \omega_t^{(i)}. \tag{39}$$

Thus, after resampling

$$\hat{k}_t = \underset{k_{\min} \le k \le k_{\max}}{\arg\max} \; N_{t,k} \tag{40}$$

where $N_{t,k}$ is the number of particles with $k_t = k$. The estimates $\hat{\mathbf{f}}_t$ and $[\,\hat{\mathbf{a}}_t \quad \hat{\mathbf{b}}_t\,]^{\mathrm{T}}$ are computed by marginal MMSE as follows, for $j = 1, \ldots, \hat{k}_t$:

$$\hat{\mathbf{f}}_{t,j} = \frac{1}{N_j} \sum_{i=1}^{N_j} \mathbf{f}_{t,j}^{(i)} \tag{41}$$

$$[\,\hat{\mathbf{a}}_{t,j} \quad \hat{\mathbf{b}}_{t,j}\,]^{\mathrm{T}} = \frac{1}{N_j} \sum_{i=1}^{N_j} \boldsymbol{\mu}_{t,j}^{(i)}. \tag{42}$$

Here, some remarks are necessary to better understand how estimates are computed. Once the estimated number of HCs $\hat{k}_t$ is computed, one may use the $N_{t,\hat{k}_t}$ particles such that $k_t = \hat{k}_t$, to compute $\hat{\mathbf{f}}_t$ and $[\,\hat{\mathbf{a}}_t \quad \hat{\mathbf{b}}_t\,]^{\mathrm{T}}$. The frequencies of these particles (and the corresponding amplitudes) can be sorted in the same order as in a common reference, say $\mathbf{f}_t^{\mathbf{y}}$ and/or $\hat{\mathbf{f}}_{t-1}$, and thus, the average over all these $N_{t,\hat{k}_t}$ particles makes sense. However, to improve the estimation, one can also consider, for a given HC $j (j = 1, \ldots, \hat{k}_t)$, the N_j particles that contain this frequency, even if $k_t \neq \hat{k}_t$ for these particles. Indeed, particles with less than $\hat{k}_t$ HCs are very likely to contain only frequencies that are also contained in particles with $k_t = \hat{k}_t$ (otherwise, they are unlikely to survive to the resampling process). Similarly, particles with more than $\hat{k}_t$ HCs are also likely to contain at least the $\hat{k}_t$ HCs of interest.

The design of the importance distribution q is of paramount importance in sequential importance sampling algorithms, and Section III-B addresses this issue. Section III-C develops a specific part of this algorithm: the hyperparameters update procedure.

B. Frequencies Update

A strategy aimed at limiting weights degeneracy consists of selecting the importance density that minimizes the variance of the importance weights, conditional upon the simulated trajectory $\left\{k_{0:t-1}^{(i)}, \boldsymbol{\theta}_{0:t-1}^{(i)}\right\}$ and on the observations $\mathbf{y}_{1:t}$ [32]. It has been shown [41] that this optimal importance density is $\pi_t\left(k_{0:t}, \mathbf{f}_{0:t}\right) = p\left(k_t, \mathbf{f}_t | k_{0:t-1}, \mathbf{f}_{0:t-1}, \mathbf{y}_{1:t}\right)$. In practice, this optimal importance density is difficult to sample from, and one seeks to design densities as close as possible to the optimal one and from which it is easy to sample. A very basic choice is to select the transition density as importance density, i.e., $q_t\left(\mathbf{f}_t|\mathbf{f}_{t-1}\right) = p\left(\mathbf{f}_t|\mathbf{f}_{t-1}\right)$. Even if it provides simple algorithms (usually named *Bootstrap Filter* or *Condensation Filter*), it does not use the new observation, and one will prefer to it more "complicated" importance densities. Indeed, the importance sampling framework allows the design of cleverer proposals, making use of the most recent information available at time t. To realize this combination between the information given by the state vector at time $t-1$ and the one given by the observation at time t, we adopt a very simple solution. One extracts a vector $\mathbf{f}_t^{\mathbf{y}}$, whose dimension is the same as $\mathbf{f}_t^{(i)}$, from the observation $\mathbf{y}_t$. Thus, the proposal density is a Gaussian whose mean is a linear combination of $\mathbf{f}_t^{\mathbf{y}}$ and $\mathbf{f}_{t-1}^{(i)}$. In the case of the creation of a new HC, that is when $k_t^{(i)} = k_{t-1}^{(i)} + 1$, the frequencies are sampled directly from $\mathbf{f}_t^{\mathbf{y}}$.

The vector $\mathbf{f}_t^{\mathbf{y}}$ can be extracted from the observation by an existing single frame-based multiple F0 estimation algorithm. In our case, one has chosen to build a simple version of such algorithms, based on a simple heuristics as follows: the discrete Fourier transform (DFT) of the observation $\mathbf{y}_t$ is computed. Then, for each peak of DFT$(\mathbf{y}_t)$, we evaluate the possibility that it corresponds to a fundamental partial frequency. More precisely, we associate to this peak a coefficient equals to the sum of the amplitudes, read in DFT$(\mathbf{y}_t)$, of all its possible overtone partials. Finally, the frequency with the largest coefficient is considered as the fundamental frequency of the first HC. Thanks to (12), an HC without inharmonicity is built with H partials, based on this fundamental frequency. This HC is removed from $\mathbf{y}_t$, and the procedure is iterated until $k_{\max}$ HCs are extracted. This proposal is quite simple and has some limitations: it gives limited results in the case of an HC whose fundamental partial frequency corresponds to the overtone partial frequency of another HC, i.e., when two fundamental frequencies are in octave relation. To overcome this, HCs with fundamental frequencies equal to integer multiples of the fundamental frequencies extracted by this procedure are added to the list of HCs candidates. The goal of our simple proposal is, above all, to validate our flexible approach and the possibility of our algorithm to improve results given by the proposal, notably selecting the HCs really present in the signal and refining the estimation of the corresponding state parameters. An important thing to note here is, as mentioned in the Introduction, that other heuristics could be used here and, especially, existent deterministic algorithms.

C. Hyperparameters Update

A specific characteristics of our model is the possibility for the variances of the evolution noises to evolve with time. These variances are grouped into a vector $\mathbf{r}_t$. For $j = 1, \ldots, k_t$ and $h = 1, \ldots, H$

$$\mathbf{r}_t = \left[, \ldots, r^{\mathbf{f}}_{t,j,h}, \ldots, r^{\mathbf{a}}_{t,j,h}, \ldots, r^{\mathbf{b}}_{t,j,h}, \ldots, \right]. \quad (43)$$

One can consider that $\mathbf{r}_t$ is hierarchically above $\boldsymbol{\theta}_t$ and then is a vector of hyperparameters. Its estimation is managed by the RBPF in the same way as the state vector is and, in the same way as $\mathbf{y}_t$ is the observation for the state parameters, $\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-1}$ is the observation for the hyperparameters.

The optimal density is $p(\mathbf{r}_t|\mathbf{r}_{t-1}, \mathbf{I}_t)$, where $\mathbf{I}_t = \boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-1}$. Bayes formula gives

$$p(\mathbf{r}_t|\mathbf{r}_{t-1}, \mathbf{I}_t) = \frac{p(\mathbf{I}_t|\mathbf{r}_t)p(\mathbf{r}_t|\mathbf{r}_{t-1})}{p(\mathbf{I}_t|\mathbf{r}_{t-1})}. \quad (44)$$

The evolution models defined by (19), (20), and (21) concern the logarithm of the variance. Thus, the hyperparameters vector $\mathbf{r}_t$ is, in fact, defined as the vector of the logarithm of all the variances. The three densities involved in the Bayes formula in (44) are Gaussian and, r_t and I_t being any components of vectors $\mathbf{r}_t$ and $\mathbf{I}_t$, respectively, we have

$$\begin{aligned} p(r_t|r_{t-1}) &= \frac{1}{\sqrt{2\pi}\sigma_r} \exp\left(-\frac{1}{2}\frac{(r_t - r_{t-1})^2}{\sigma_r^2}\right) \\ p(I_t|r_t) &= \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left[I_t^2 e^{-r_t} + r_t\right]\right) \\ p(I_t|r_{t-1}) &= \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left[I_t^2 e^{-r_{t-1}} + r_{t-1}\right]\right) \end{aligned}$$

where σ_r denotes alternatively σ_φ, σ_α, or σ_β. Note that $p(\mathbf{r}_t|\mathbf{r}_{t-1}, \mathbf{I}_t)$ is not Gaussian, because $p(\mathbf{I}_t|\mathbf{r}_t)$ is not Gaussian w.r.t. $\mathbf{r}_t$. This optimal density has good properties making it possible to sample from it by using the accept/reject algorithm [44]. This algorithm allows to simulate from a target distribution F with density f (here the optimal proposal density is $p(\mathbf{r}_t|\mathbf{r}_{t-1}, \mathbf{I}_t)$) by thinning out samples from a candidate distribution G with density g (here chosen as a Gaussian density) which support contains the support of F. The algorithm is as follows.

1) Generate $x \sim G(x)$.

2) Generate $u \sim \mathcal{U}[0, 1]$.

3) Accept x if $u \leq (f(x)/Cg(x))$.

4) Goto 1).

The accepted values all have distribution F and the number of candidates to obtain one target is geometric with success probability $1/C$. The constant C is selected so that

$$C = \sup_x \frac{f(x)}{g(x)} < \infty.$$

In our case, the density g is $g(x) = \mathcal{N}\left(x; r_{t-1}, \sigma_r^2\right)$.

IV. Discussion

In this Section, we discuss several features of our method. Since the goal of this paper is to propose a flexible Bayesian approach to perform the task of detection and tracking of HCs, each choice made was motivated by simplicity and generality. The consequence of this is twofold: 1) Our method is virtually applicable to a large number of real-world cases; and 2) it offers a great flexibility, that is, it allows for most existing methods to be embedded into our framework and the model can be refined to adapt to specific signals (known instrument in music, for instance).

A. About the Dynamic Model

Several remarks can be made about the dynamic model. First, the evolution (25) selected for k_t, favors the case where the number of HCs is kept constant while allowing it to increase or to decrease in the same proportion. This corresponds to a simple and quite natural heuristics. However, these increase/decrease probabilities could be changed, e.g., by setting $b_t < d_t$ to avoid overfitting.

An important point concerns the observations. In our previous works [40], [45], the observation at time t was the corresponding column of the spectrogram, that is, the square magnitude of the DFT of the windowed signal. The spectrogram maps the data into the TF plane from which the estimation of the number k_t of HCs, the time-varying partial frequencies and amplitudes at time t is easier and physically sound. The fallout of this choice is the loss of phase information which cannot be estimated in the spectrogram. Since the estimation results were promising, we tried to take into account the phase [46]. The results showed that the estimations were globally better than in [40] and [45] except for k_t. The right solution is then to combine both: let the windowed signal be the observation, and let the spectrogram be used to design the frequency importance distribution; this is the solution followed here.

B. About the Frequencies Importance Distribution

It is well known that, in PF algorithms, the importance density must be selected carefully. In our case, it is especially true.

Actually, the importance density must be as close as possible to the optimal one [42] so that most of the particles it proposes must be in good "regions" of the state space, that is, where the posterior is high. This is to ensure that most of the particles have a significantly large weight and contribute to the posterior MC approximation. A basic condition for this is that the importance density combines the information coming from the state vector at the previous time and the current observation. The importance density proposed in Section III-B is then a good solution. However, in our method, others could be designed. Our importance density extracts recursively the most predominant HC from the signal frame and is inspired by different methods. Instead of this, one can directly use one of the numerous algorithms proposed in previous works (quickly presented in Introduction). Indeed, an importance density can be derived from most of these methods, yielding a stochastic algorithm with well controlled convergence properties, and that may overperform the method used to build the importance density. In Section III, we described this efficient framework with a simple importance density, but a key remark is that our method casts the problem of joint detection and tracking of time-varying HCs in a general framework which is not linked to a particular importance density.

For instance, the method developed by Yeh *et al.* [20] would allow the design of an efficient importance density. They propose a multiple F0 estimation method in musical signals when the number of F0s is known in advance. From an observed spectrum and a generative quasi-harmonic spectral model, the method generates an F0 candidate list and assigns to each F0 a hypothetical partial sequence (HPS) taking into account low inharmonicity. Then, each HPS is evaluated by a score function constructed on three principles: harmonicity, spectral smoothness, and synchronous amplitude evolution within a single source. This algorithm is certainly a more sophisticated version of our simple importance density, and it can easily be turned into an importance density.

V. Results

In this section, we present some results aimed at validating our method. The importance density is designed from the simple heuristics proposed in Section III-B, and two ways of modeling the inharmonicity are tested. We present simulation results both on synthetic and real data. More precisely, we first present some simulations on a synthetic signal in order to work in a well-controlled context. Then, we work on a MIDI signal. Here, estimated values can also be compared to theoretical ones, and the signal is closed to a real one. Last, a real music signal is studied.

A. Synthetic Signals

In this section, we apply our method to a synthetic signal. This study is split into two parts. First, we test the efficiency of our method without additive noise on the simulated signal. Then, we study the evolution of the results when the variance of the additive noise increases.

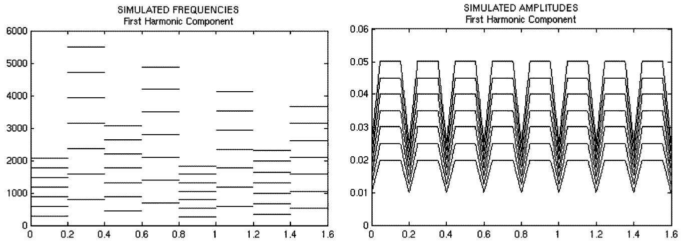

Fig. 4. Simulated frequencies (left) and amplitudes (right).

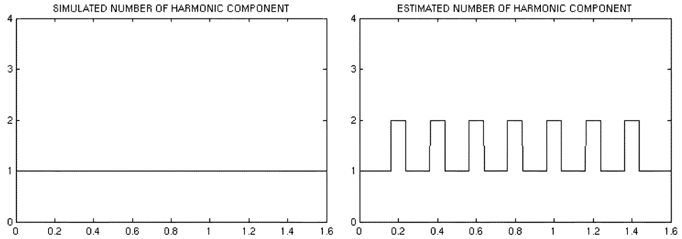

Fig. 5. Simulated (left) and estimated (right) number of HCs over time. Estimates are computed by (40).

The simulated signal is a sequence of eight notes (or HCs). Each note has seven partials of decreasing amplitude from 0.05 for the fundamental frequency to 0.02 for the seventh partial. The duration of the signal is 1.6 s (0.2 s for each note) and its sampling rate is 11 025 Hz. The simulated parameters are shown in Fig. 4.

In these simulations, each frequency is estimated independently of the others, and the evolution equation is described by a simple random walk. The algorithm parameters are as follows:

- window's shape: Gauss;
- window's length: $L_{\mathbf{w}} = 1024$ points ($\approx$ 93 ms);
- window step: 16 points ($\approx$ 1.5 ms) the overlap is 98.4%;
- maximum number of HCs: $k_{\max} = 3$;
- minimum number of HCs: $k_{\min} = 1$;
- number of partials: $H = 10$;
- number of particles: $N = 500$.

1) Estimation Without Additive Noise: The simulation results on the synthetic signal without additive noise are presented in Figs. 5 and 6. Estimates are computed as indicated in Section III-A. An optional post-processing step can be added to remove partials with too low amplitude with respect to others, within a HC (here, the threshold is 0.0045). In Fig. 5, the estimated number of HCs toggles between 1 and 2, whereas there is only one simulated HC. Actually, this is due to the window. Indeed, the length of the window is such that it can cover both the end of an HC and the beginning of the following one. This is confirmed in Fig. 6, where the frequencies of the second HC match the ones of the emerging HC and then the ones of the dying HC. Taking into account this windowing effect, one can note that both times where changes occur and state parameters are quite well estimated. In the simulated signal, each note has seven partials, and our algorithm wants to estimate $H = 10$ partials. One verifies that the three last partials actually have zero amplitudes. However, when a new component appears, the algorithm first activates an additional partial and then increase the number of HCs. This explains why, in Fig. 6, an eighth transient partial appears at times.

2) Estimation With Additive Noise: The previous study is repeated, using the same simulated signal. Here, however, noises with increasing variance is added. We also compare the results

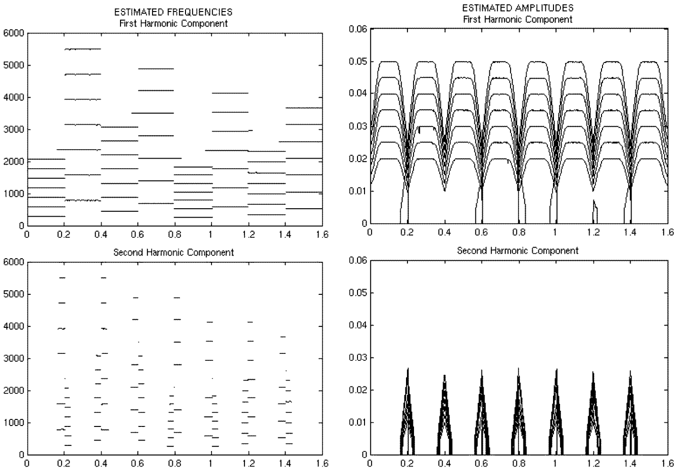

Fig. 6. Estimated frequencies (left) and amplitudes (right) of the first two HCs (there is no third HC), over time. Estimates are computed by (41) and (42).

TABLE III
EVOLUTION OF THE RMS ERROR WITH THE SNR

SNR (dB)	∞	64.42	50.60	42.58	36.96	23.77	16.71
rms^a (10^{-3})	0.24	0.36	0.50	0.52	1.27	1.57	3.38
rms^h (10^{-3})	0.48	0.69	0.80	0.85	1.37	1.94	3.43

obtained by our algorithm to those obtained by using the simple heuristics used to design the frequency proposal distribution (see Section III-B). In each case, we compute the mean of the root mean square (rms) error over time

$$\text{rms} = \frac{1}{T}\sum_{t=0}^{T} \text{rms}_t \tag{45}$$

where rms_t is given by

$$\text{rms}_t = \sqrt{\frac{\sum_{i=1}^{N} \omega_t^{(i)} \left\| \mathbf{y}_t - g_t\left(k_t^{(i)}, \boldsymbol{\theta}_t^{(i)}\right)\right\|^2}{L_{\mathbf{w}}}}. \tag{46}$$

The values of the rms errors functions of the signal-to-noise ratio (SNR) are given in Table III, where rms^a corresponds to estimates given by our algorithm, and rms^h corresponds to those given by the heuristics.. We note that, with an SNR lower than 20 dB, the estimates become meaningless. This study shows that, even in a noisy case, our algorithm improves the performance of the proposal distribution. Then, our method can virtually improve results given by existing methods through the design of the proposal distribution.

B. MIDI Signal

In this section, we apply our method to a MIDI music signal. The original signal is a flute/piano duet playing the famous French little song "Le roi Dagobert," whose the score is given in Fig. 7. It is sampled at a rate of 11 025 Hz and has a duration of 25 s. This signal is quite interesting because it is composed in turn of 0, 1, or 2 notes, and notes are sometimes in octave relation.

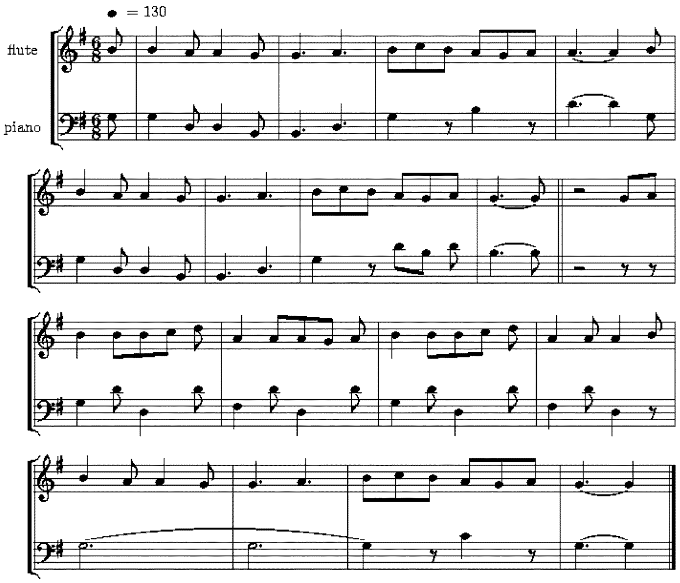

Fig. 7. Score of the famous French little song "Le roi Dagobert."

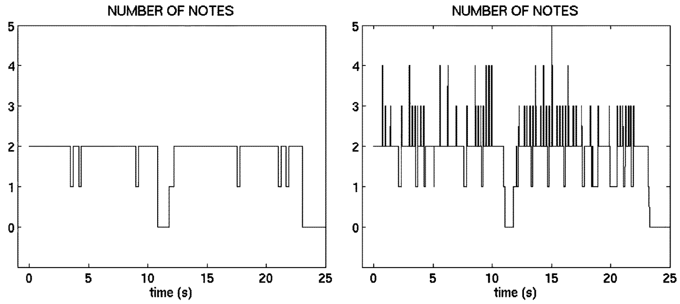

Fig. 8. Theoretical (left) and estimated (right) number of played notes over time for the MIDI signal.

Here, we have chosen to consider the following inharmonicity model, for $h = 2, \ldots, H$

$$f_h = hf_1\sqrt{1+\beta h^2} \tag{47}$$

where f_h is the frequency of the overtone partial h, f_1 is the frequency of the fundamental partial, and β is the inharmonicity parameter. Thus, one only needs to estimate these two last parameters together with all the amplitudes. The algorithm parameters are as follows:

- window's shape: Gauss;
- window's length: $L_{\mathbf{w}} = 1024$ points ($\approx$ 93 ms);
- window step: 50 points ($\approx$ 4.5 ms) the overlap is 95.1%;
- maximum number of HCs: $k_{\max} = 5$;
- minimum number of HCs: $k_{\min} = 0$;
- number of partials: $H = 40$;
- number of particles: $N = 500$.

The simulation results are given in Figs. 8–10. The computation time is about 12 h. In Fig. 9, vertical lines indicate times when notes start that is their attack. As a general comment, the estimation results are quite satisfactory. Moreover, from the estimates, an audio signal can be reconstructed, and the perceptive difference with the original one is small, excluding perhaps an unspecified estimation of the phase that introduces a kind of

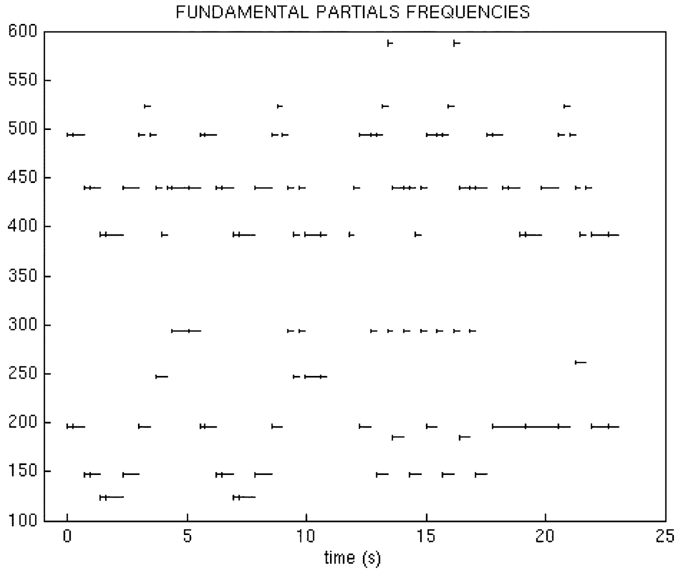

Fig. 9. Theoretical fundamental partial frequencies over time for the MIDI signal. They are computed from the score given in Fig. 7.

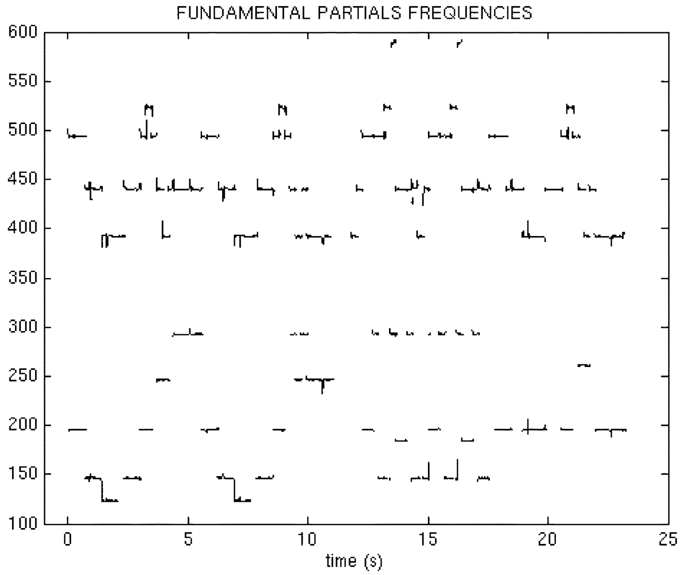

Fig. 10. Estimated fundamental partial frequencies over time for the MIDI signal.

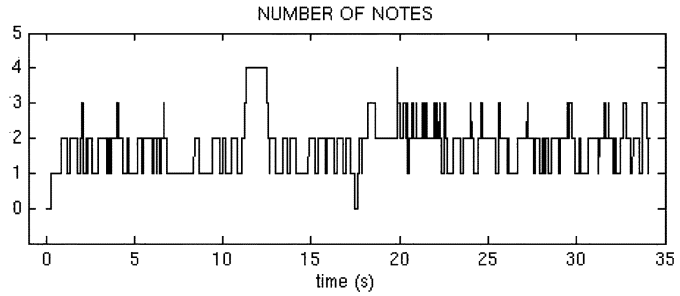

Fig. 11. Estimated number of notes over time for the real signal.

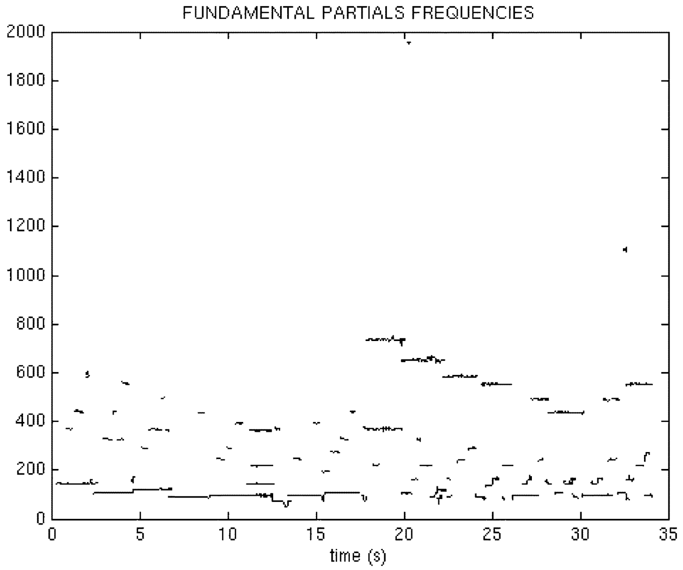

Fig. 12. Estimated fundamental partial frequencies over time for the real signal.

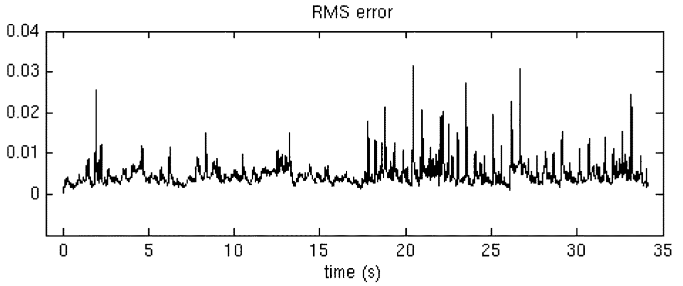

Fig. 13. RMS error over time for the real signal. It is computed by (45).

little background noise. However, it is important to remark that the estimated number of notes, in Fig. 8, does not always corresponds to the theoretical one. It is both due to the windowing effect and to echo. Indeed, when the instrument stops playing the note according to the score, it can be heard after some time. Another reason which could be invoked is that the attack process is not harmonic, and during this transient step, the algorithm can resort to supplementary HCs.

C. Real Signal

In this section, we apply our method to an extract of the Johann Pachelbel's Canon in D. It is sampled at a rate of 22 050 Hz and has a duration of 34 s. The same approach as previously is adopted and the algorithm parameters are as follows:

- window's shape: Gauss;
- window's length: $L_{\mathbf{w}} = 2048$ points (≈ 93 ms);
- window step: 100 points ($\approx$4.5 ms) the overlap is 95.1%;
- maximum number of HCs: $k_{\max} = 5$;
- minimum number of HCs: $k_{\min} = 0$;
- number of partials: $H = 40$;
- number of particles: $N = 500$.

The simulation results are given in Figs. 11–13. The computation time is about 61 h.

VI. Conclusion and Future Work

In this paper, we have presented a rigorous Bayesian framework to address the joint detection and tracking of HCs problem. By using a jump Markov system and a state-of-the-art Monte

Carlo algorithm, we have shown that, for a given audio signal, it is possible to estimate the features of the harmonic part. More precisely, the number of harmonic components together with their frequency structure are sequentially estimated. Our model includes time-varying amplitudes and inharmonicity. We have also shown that this formalism allows for existing methods to be used through the design of the importance density and thus to be improved. We obtain good estimation results that validate our method. There remains some work to be done, notably to decrease the computation time, which is one of the main drawback of our algorithm. However, an important remark is that our method has been developed directly in a sequential context and optimizations of the code can be performed to come close the real-time.

APPENDIX I
JMS MATRIX DEFINITION

Matrix A and B, of size $2k_tH \times 2k_tH$, are defined by (28), that summarizes(17) and (18)

$$\begin{aligned} A &= \mathbf{I}_{2k_tH\times 2k_tH} \\ B &= \mathrm{diag}\left(\left[\sqrt{r^{\mathbf{a}}_{t-1,j,h}}, \ldots, \sqrt{r^{\mathbf{b}}_{t-1,j,h}}\right]\right). \end{aligned}$$

Equation (29) summarizes the observation equation (13) and allows the definition of matrix C, of size $L_\mathbf{w} \times 2k_tH$, and matrix D as

$$C = \left[\left(C^{(c)}_{j,h}\right) \cdots \left(C^{(s)}_{j,h}\right)\right] \qquad D = \sqrt{r^{\mathbf{y}}}\mathbf{I}_{L_\mathbf{w}\times L_\mathbf{w}}$$

with

$$C^{(c)}_{j,h} = \begin{bmatrix} \cos(2\pi f_{t,j,h}1)\mathbf{w}[1] \\ \vdots \\ \cos(2\pi f_{t,j,h}\tau)\mathbf{w}[\tau] \\ \vdots \\ \cos(2\pi f_{t,j,h}L_\mathbf{w})\mathbf{w}[L_\mathbf{w}] \end{bmatrix}$$

$$C^{(s)}_{j,h} = \begin{bmatrix} \sin(2\pi f_{t,j,h}1)\mathbf{w}[1] \\ \vdots \\ \sin(2\pi f_{t,j,h}\tau)\mathbf{w}[\tau] \\ \vdots \\ \sin(2\pi f_{t,j,h}L_\mathbf{w})\mathbf{w}[L_\mathbf{w}] \end{bmatrix}$$

APPENDIX II
KALMAN FILTER EQUATIONS

From $k^{(i)}_{t+1}$ and $\mathbf{f}^{(i)}_{t+1}$, the matrices $A^{(i)}$, $B^{(i)}$, $C^{(i)}$, and D can be computed as indicated in Appendix I. Then, $\boldsymbol{\mu}^{(i)}_{t+1}$ and $\boldsymbol{\Sigma}^{(i)}_{t+1}$ are computed with the Kalman filter equations

$$\begin{aligned} \boldsymbol{\mu}^{(i)}_{t|t-1} &= A^{(i)}\boldsymbol{\mu}^{(i)}_{t-1} \\ \boldsymbol{\Sigma}^{(i)}_{t|t-1} &= A^{(i)}\boldsymbol{\Sigma}^{(i)}_{t-1}A^{(i)\mathrm{T}} + B^{(i)}B^{(i)\mathrm{T}} \\ S^{(i)}_t &= C^{(i)}\boldsymbol{\Sigma}^{(i)}_{t|t-1}C^{(i)\mathrm{T}} + DD^{\mathrm{T}} \\ \mathbf{y}^{(i)}_{t|t-1} &= C^{(i)}\boldsymbol{\mu}^{(i)}_{t|t-1} \\ \boldsymbol{\mu}^{(i)}_t &= \boldsymbol{\mu}^{(i)}_{t|t-1} + \boldsymbol{\Sigma}^{(i)}_{t|t-1}C^{(i)\mathrm{T}}S^{-1(i)}_t\left(\mathbf{y}_t - \mathbf{y}^{(i)}_{t|t-1}\right) \\ \boldsymbol{\Sigma}^{(i)}_t &= \boldsymbol{\Sigma}^{(i)}_{t|t-1} - \boldsymbol{\Sigma}^{(i)}_{t|t-1}C^{(i)\mathrm{T}}S^{-1(i)}_tC^{(i)}\boldsymbol{\Sigma}^{(i)}_{t|t-1}. \end{aligned}$$

REFERENCES

[1] D. F. Rosenthal and H. G. Okuno, *Computational Auditory Scene Analysis*. Mahwah, NJ: Lawrence Erlbaum, 1998.

[2] S. W. Hainsworth, “Techniques for the automated analysis of musical audio,” Ph.D. dissertation, Signal Processing Group, Dept. Eng., Univ. Cambridge, Cambridge, U.K., 2003.

[3] A. P. Klapuri and M. Davy, *Signal Processing Techniques for Music Transcription*. New York: Springer, 2006.

[4] A. P. Klapuri, “Signal processing methods for the automatic transcription of music,” Ph.D. dissertation, Tampere Univ. Technol., Tampere, Finland, 2004.

[5] A. S. Bregman, *Auditory Scene Analysis: The Perceptual Organization of Sound*. Cambridge, MA: MIT Press, 1990.

[6] R. Meddis and M. J. Hewitt, “Virtual pitch and phase sensitivity of a computer model of the auditory periphery. I: Pitch identification,” *J. Acoust. Soc. Amer. (JASA)*, vol. 89, pp. 2866–2882, Jun. 1991.

[7] A. P. Klapuri, “Automatic transcription music,” M.S. thesis, Tampere University of Technology, Tampere, Finland, 1998.

[8] T. Tolonen and M. Karjalainen, “A computationally efficient multipitch analysis model,” *IEEE Trans. Speech Audio Process.*, vol. 8, no. 6, pp. 708–716, Nov. 2000.

[9] M. Wu, D. Wang, and G. J. Brown, “A multipitch tracking algorithm for noisy speech,” *IEEE Trans. Speech Audio Process.*, vol. 11, no. 3, pp. 229–241, May 2003.

[10] R. Meddis and M. J. Hewitt, “Modeling the identification of concurrent vowels with different fundamental frequencies,” *J. Acoust. Soc. Amer. (JASA)*, vol. 91, pp. 233–245, Jan. 1992.

[11] A. de Cheveigné, “Concurrent vowel identification. III. A neural model of harmonic interference cancellation,” *J. Acoust. Soc. Amer. (JASA)*, vol. 101, pp. 2857–2865, May 1997.

[12] A. P. Klapuri, “A perceptually motivated multiple-F0 estimation method,” in *IEEE WASPAA*, New Paltz, NY, Oct. 2005, pp. 291–294.

[13] K. D. Martin, “A blackboard system for automatic transcription of simple polyphonic music,” MIT Media Laboratory Perceptual Computing Section, Cambridge, MA, 1996, Tech. Rep. 385.

[14] ——, “Automatic transcription of simple polyphonic music: Robust front end processing,” MIT Media Laboratory Perceptual Computing Section, Cambridge, MA, 1996, Tech. Rep. 399.

[15] D. Godsmark and G. J. Brown, “A blackboard architecture for computational auditory scene analysis,” in *Speech Commun.*, Apr. 1999, vol. 27, pp. 351–366.

[16] J. P. Bello and M. Sandler, “Blackboard system and top-down processing for the transcription of simple polyphonic music,” in *Proc. COST G-6 Conf. Digital Audio Effects*, Verona, Italy, Dec. 2000, pp. 135–140.

[17] K. Kashino, K. Nakadia, T. Kinoshita, and H. Tanaka, “Application of Bayesian probability network to music scene analysis,” in *Proc. IJCAI, CASA workshop*, Montréal, QC, Canada, Aug. 1995, pp. 52–59.

[18] D. K. Mellinger, “Event Formation and Separation in Musical Sound,” Ph.D., Dept. Comput. Sci., Stanford Univ., Stanford, CA, 1991.

[19] G. J. Brown and M. Cooke, “Perceptual grouping of musical sounds: A computational model,” *J. New Music Res.*, vol. 23, pp. 107–132, 1994.

[20] C. Yeh, A. Röbel, and X. Rodet, “Multiple fundamental frequency estimation of polyphonic music signals,” in *Proc. IEEE ICASSP*, Philadelphia, PA, Mar. 2005, vol. 3, pp. 225–228.

[21] R. A. Irizarry, “Local harmonic estimation in musical sound signals,” *J. Amer. Statist. Assoc.*, vol. 96, pp. 357–367, Jun. 2001.

[22] M. Goto, “A robust predominant-F0 estimation method for real-time detection of melody and bass lines in CD recordings,” in *Proc. IEEE ICASSP*, Istanbul, Turkey, Jun. 2000, vol. 2, pp. 757–760.

[23] ——, “A predominent-F0 estimation method for CD recordings: MAP estimation using EM algorithm for adaptative tone models,” in *Proc. IEEE ICASSP*, Salt Lake City, UT, May 2001, vol. 5, pp. 3365–3368.

[24] C. Raphael, “Automatic transcription of piano music,” in *Proc. ISMIR*, Paris, France, Oct. 2002.

[25] C. Raphael and J. Stoddard, “Harmonic analysis with probabilistic graphical models,” in *Proc. ISMIR*, Baltimore, MD, Oct. 2003.

[26] M. Davy and S. J. Godsill, “Bayesian harmonic models for musical signal analysis,” in *Bayesian Statistics 7*, Valencia, Spain, Jun. 2002, pp. 105–124.

[27] M. Davy, S. J. Godsill, and J. Idier, “Bayesian analysis of polyphonic Western tonal music,” *J. Acoust. Soc. Amer. (JASA)*, vol. 119, pp. 2498–2517, Apr. 2006.

[28] A. T. Cemgil, B. Kappen, and D. Barber, “Generative model based polyphonic music transcription,” in *IEEE WASPAA*, New Paltz, NY, Oct. 2003, pp. 181–184.

[29] A. T. Cemgil, H. J. Kappen, and D. Barber, "A generative model for music transcription," *IEEE Trans. Audio, Speech Language Process.*, vol. 14, no. 2, pp. 679–694, Mar. 2006.

[30] E. Vincent and M. D. Plumbley, "A prototype system for object coding of musical audio," in *Proc. IEEE WASPAA*, New Paltz, NY, Oct. 2005, vol. 15, pp. 239–242.

[31] A. Doucet, M. Briers, and S. Sénécal, "Efficient block sampling strategies for sequential Monte Carlo methods," *J. Comput. Graph. Statist.*, vol. 15, pp. 1–19, 2006.

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

[33] C. Andrieu, M. Davy, and A. Doucet, "Efficient particle filtering for jump Markov systems. Application to time-varying autoregressions," *IEEE Trans. Signal Process.*, vol. 51, no. 7, pp. 1762–1770, Jul. 2003.

[34] F. Hlawatsch and G. F. Boudreaux-Bartels, "Linear and quadratic time-frequency signal representations," *IEEE Signal Process. Mag.*, vol. 9, no. 2, pp. 21–67, Apr. 1992.

[35] P. Flandrin, *Temps-Fréquence*. Paris, France: Hermès, 1993.

[36] A. P. Klapuri, "Multiple fundamental frequency estimation based on harmonicity and spectral smoothness," *IEEE Trans. Speech Audio Process.*, vol. 11, no. 6, pp. 804–816, Nov. 2003.

[37] N. H. Fletcher and T. D. Rossing, *The Physics of Musical Instruments*. New York: Springer, 1998.

[38] G. Wahba, "Improper priors, spline smoothing and the problem of guarding against model errors in regression," *J. R. Statist. Soc. Series B*, vol. 40, pp. 364–372, 1978.

[39] ——, "Bayesian confidence intervals for the cross-validated smoothing spline," *J. R. Statist. Soc. Series B*, vol. 45, pp. 133–150, 1983.

[40] C. Dubois and M. Davy, "Harmonic tracking using sequential Monte Carlo," in *Proc. SSP Workshop*, Bordeaux, France, Jul. 2005, pp. 1292–1297.

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

[42] A. Doucet, "On sequential simulation-based methods for Bayesian filtering," Signal Process. Group, Dept. Eng., Univ. Cambridge, Cambridge, U.K., 1998, Tech. Rep.

[43] N. de Freitas, "Rao–Blackwellised particle filtering for fault diagnosis," in *Proc. IEEE Aerospace Conf.*, Big Sky, MT, Mar. 2002, vol. 4, pp. 1767–1772.

[44] G. Casella and C. P. Robert, *Monte Carlo Statistical Methods*. New York: Springer, 2005.

[45] C. Dubois, M. Davy, and J. Idier, "Tracking of time-frequency components using particle filtering," in *Proc. IEEE ICASSP*, Philadelphia, PA, Mar. 2005, vol. 4, pp. 9–12.

[46] C. Dubois and M. Davy, "Suivi de trajectoires temps-fréquence par filtrage particulaire," in *Proc. GRETSI*, Louvain-la-Neuve, Belgium, Sep. 2005, vol. 2, pp. 985–988.

Corentin Dubois was born in Clermont Ferrand, France, in 1980. He received the Engineer degree in electrical engineering and signal processing from the Ecole Centrale de Nantes, Nantes, France, in 2003 and the Ph.D. degree in signal processing from the University of Nantes in November 2006.

He is currently with IRCCyN, Nantes. His research interests include audio signal processing and sequential Monte Carlo techniques (particle filters).

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

From 2000 to 2002, he was a Research Associate at the Signal Processing Group, University of Cambridge, Cambridge, U.K. He is currently a Chargé de Recherches at the Centre National de la Recherche Scientifique (CNRS), Lille, France, and member of the INRIA-FUTURS "SequeL" Team. Since 1998, he has published a book, about 15 journal papers, and 40 conference papers. In 2003 and 2004, he co-organized the Machine Learning Summer School. From 2002 to 2006, he was Principal Investigator in four industrial research contracts. His current research activities are centered around machine learning, kernel algorithms, Bayesian/Monte Carlo methods for signal processing (Markov chain Monte Carlo, particle filtering, sequential Monte Carlo), and reinforcement learning. His preferred application fields are audio and multisensor data fusion.