Bayesian analysis of polyphonic western tonal music

Manuel Davy, Simon Godsill and Jérôme Idier

Citation: The Journal of the Acoustical Society of America **119**, 2498 (2006); doi: 10.1121/1.2168548
View online: https://doi.org/10.1121/1.2168548
View Table of Contents: https://asa.scitation.org/toc/jas/119/4
Published by the Acoustical Society of America

Bayesian analysis of polyphonic western tonal music

Manuel Davy[a)]
LAGIS/CNRS, Cité scientifique, BP 48, 59651 Villeneuve d'Ascq, France

Simon Godsill[b)]
Signal Processing Group, Department of Engineering, University of Cambridge Trumpington Street, Cambridge CB2 1PZ, United Kingdom

Jérôme Idier[c)]
IRCCyN/CNRS, 1 rue de la Noë, BP92101, 44321 Nantes cedex 3, France

(Received 30 August 2004; revised 4 January 2006; accepted 5 January 2006)

This paper deals with the computational analysis of musical audio from recorded audio waveforms. This general problem includes, as subtasks, music transcription, extraction of musical pitch, dynamics, timbre, instrument identity, and source separation. Analysis of real musical signals is a highly ill-posed task which is made complicated by the presence of transient sounds, background interference, or the complex structure of musical pitches in the time-frequency domain. This paper focuses on models and algorithms for computer transcription of multiple musical pitches in audio, elaborated from previous work by two of the authors. The audio data are supposedly presegmented into fixed pitch regimes such as individual chords. The models presented apply to pitched (tonal) music and are formulated via a Gabor representation of nonstationary signals. A Bayesian probabilistic structure is employed for representation of prior information about the parameters of the notes. This paper introduces a numerical Bayesian inference strategy for estimation of the pitches and other parameters of the waveform. The improved algorithm is much quicker and makes the approach feasible in realistic situations. Results are presented for estimation of a known number of notes present in randomly generated note clusters from a real musical instrument database.
 [DOI: 10.1121/1.2168548]

PACS number(s): 43.75.-z, 43.60.-c, 43.60.Pt, 43.60.Uv [ADP] Pages: 2498–2517

I. INTRODUCTION

Polyphonic music modeling is a challenging problem, which includes many possible subtasks, such as simultaneous multiple fundamental frequency estimation, time-varying amplitude/frequency tracking, modeling of inharmonicity, source separation, and inference about instrument-specific structures. Among the numerous approaches proposed in the literature, Bayesian approaches have been surprisingly rare, considering the large quantities of prior information available about musical signals. Musical signals are highly structured, both in the time domain and in the frequency domain (see, for example, Ref. 1 or 2). In the time domain, note transitions and percussive sounds occur at times related to the tempo and beat positions in the music. In the frequency domain, two levels of structure can be considered for harmonic notes. First, each note is composed of a fundamental partial, whose frequency is related to the "pitch" of the note, and overtone partials, whose relative amplitudes and frequencies determine the timbre of the note. This frequency domain description can be regarded as an empirical approximation to the true process, which is in reality a complex nonlinear time-domain system, as pointed out by McIntyre *et al.*[3] and Fletcher and Rossing.[2] Second, several notes played at the same time form chords or polyphony. Finally, higher levels of structure are present in sequences of chords, melodic shape, etc. This structural information can be incorporated into a probabilistic framework, so as to yield a statistical model, termed a *polyphonic Bayesian harmonic model* in this paper.

Numerous fundamental frequency estimation and analysis techniques can be found in the literature. Most apply only to monophonic (single voice) recordings and rely on non-parametric signal analysis tools (local autocorrelation function, spectrogram, etc.). Certain authors have adopted methods with statistical modeling elements, often using iterative procedures to estimate the individual components of a musical signal (see, for example, works by de Cheveigné *et al.*,[4,5] Virtanen and Klapuri,[6,7] Irizarry,[8,9] or Kameoka.[10]). Several existing Bayesian approaches include the work by Kashino *et al.*,[11–13] Raphael,[14,15] and Tabrikian *et al.*,[16] who adopt Bayesian hierarchical models for high level features in music such as chords, notes, timbre, etc., and Sterian *et al.*,[17] who adopt Bayesian tracking ideas for modeling of time-varying frequency partials. In a related vein, Parra and Jain[18] present Kalman filtering estimators for the harmonic plus noise model. A recent development is that of Cemgil *et al.*[19,20] who develop state-space models of musical harmonics and estimate them within an approximate Bayesian procedure.

The complete polyphonic transcription task is a great technical challenge. In this paper, we consider the subtask of multiple fundamental frequency estimation in segments of

[a)]Electronic mail: manuel.davy@ec-lille.fr
[b)]Electronic mail: sjg@eng.cam.ac.uk
[c)]Electronic mail: jerome.idier@irccyn.ec-nantes.fr

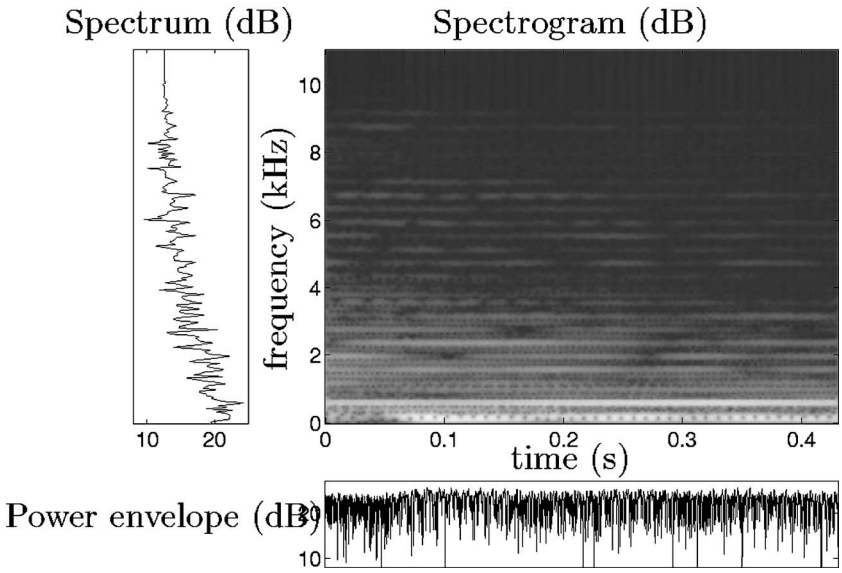

FIG. 1. Spectrogram, spectrum, and power envelope of a musical harmonic signal: a guitar alone plays two notes simultaneously. The spectrogram is computed with a Hamming window with length 11 ms. The harmonic structure, as well as the nonstationarity of partial amplitudes, appear clearly.

audio where it is assumed that no musical note changes occur. Such segments can be obtained efficiently using segmentation algorithms based on time-frequency representations,[21] support vector machines,[22,23] generalized likelihood ratio,[24] or other more music-specific methods.[25,26] We propose to describe these segments using a polyphonic music model where the number of partials, the time-varying amplitudes, the fundamental and overtone partial frequencies, and the noise variances are unknown. This new model is slightly modified compared to the model presented in our previous work (see Refs. 27–30). In particular, the noise is now assumed white (as opposed to colored in the previous Davy and Godsill papers), and the inharmonicity parameter takes a multiplicative form. These modifications were found to make parameter estimation more robust. The main contribution of this paper consists of the parameter estimation algorithm—here a Markov chain Monte Carlo (MCMC) method—which has been completely redesigned. Compared to the previous technique,[29,30] the computational load as been reduced dramatically: the computation time can be up to 10 000 times shorter. This enables us to present systematic tests involving various instruments arranged randomly into "chords," which was feasible only on a smaller scale with the previous approach.

The paper is organized as follows: in Sec. II we introduce the polyphonic Bayesian harmonic model—this involves modifications over our previous models both in terms of the inharmonicity models applied and in the prior structures adopted. In Sec. III, we describe the fast MCMC algorithm used to estimate the model parameters. In Sec. IV, the algorithm convergence is demonstrated, and results obtained with real instruments are reported. Results involving up to four instruments are presented, and a statistical analysis of estimation performance is proposed. Section V gives some conclusions and future work directions.

II. A BAYESIAN MODEL OF HARMONIC MUSIC

In this section, we present a flexible mathematical model of harmonic music. This model is then embedded in a probabilistic framework.

A. General model

Classically, acoustic signals produced by harmonic instruments are almost periodic and can be described as a sum of sinusoids.[2] Each individual note composing the signal is supposed to feature M partials, one of which is a *fundamental partial*, with angular frequency denoted ω_1, and the remaining $M-1$ are *overtone partials* with angular frequencies denoted ω_m, $m=2,\ldots,M$. Note that the fundamental frequency considered here does not coincide exactly with the common definition of fundamental frequency as the inverse of the period of the time domain signal. The overtone partial frequencies are approximately related to ω_1 by $\omega_m \approx m\omega_1$. Various instrument-specific models describe the relation between fundamental and overtone partial frequencies (see for example the piano model in Appendix A). In Fig. 1, the spectrogram, spectrum, and power envelope of a classical guitar record are plotted. The harmonic structure of guitar music appears clearly on the spectrum as well as on the spectrogram, where partials are the horizontal components with regular spacing. In the following, it is assumed that the music signal $y[t]$, $t=1,2,\ldots,N$, includes K notes having M_k partials with frequencies $\omega_{k,m}$ ($m=1,\ldots,M_k$ and $k=1,\ldots,K$).

Musical signals produced by harmonic instruments have the frequency structure described above; however, the partials amplitudes are generally nonconstant (see Fig. 1). An efficient modeling strategy is inspired by Gabor time-frequency representations[31] in which a signal $y[t]$, $t=1,2,\ldots,N$, is projected on a basis of Gabor atoms well localized in time and frequency. Our polyphonic harmonic model relies on a family of real-valued and nonzero phased Gabor atoms. Let $g[t]$ be one of these Gabor atoms with length $2N_g+1$, localized at time t_0 and frequency ω_0. It has two components:

$$g_c[t] = \phi[t]\cos(\omega_0 t) \text{ for } t = t_0 - N_g, \ldots, t_0 + N_g,$$
$$\text{and } g_c[t] = 0 \text{ otherwise}, \quad (1)$$

$$g_s[t] = \phi[t]\sin(\omega_0 t) \text{ for } t = t_0 - N_g, \ldots, t_0 + N_g,$$
$$\text{and } g_s[t] = 0 \text{ otherwise}, \tag{2}$$

where $\phi[t]$ is a parametrized envelope with, e.g., Gaussian shape, Hamming shape, etc. In standard discrete Gabor analysis, atom times $\{t_0\}$ and frequencies $\{\omega_0\}$ are positioned on a regular time-frequency lattice—see Ref. 32 for a fully Bayesian analysis in this setting; see also Ref. 33 for Gabor atom-based analysis of musical audio within a projection pursuit setting. Here, a regular time lattice is kept, but the atom frequencies correspond to the partial frequencies $\omega_{k,m}$, and these are allowed to vary continuously across a predefined range of frequencies. Let $\phi[t]$ be an atom envelope function, let I be a nonzero positive integer, and let $\Delta_t = (N-1)/I$. The Gabor atom-based model used in this paper is written (where ω_s is the sampling frequency)

$$y[t] = \sum_{k=1}^{K} \sum_{m=1}^{M_k} \sum_{i=0}^{I} \phi[t - i\Delta_t] \left\{ a_{k,m,i} \cos\left(\frac{\omega_{k,m}}{\omega_s} t\right) + b_{k,m,i} \sin\left(\frac{\omega_{k,m}}{\omega_s} t\right) \right\} + v[t]. \tag{3}$$

The atom lengths N_g and number of atom time locations $I+1$ have to be set to predefined values so as to capture important fluctuations in the musical signals, such as variations around note attacks, note decays, amplitude variations due to vibrato, and their variability between different instruments. The model presented in Eq. (3) includes relatively few parameters and is computationally tractable, while being sufficiently regularized that ambiguities between low-frequency partials and genuine amplitude modulations are unlikely to occur.

Another interpretation of Eq. (3) is as follows: each basis function $\phi[t - i\Delta_t]$, $i = 0, \ldots, I$, defines a frame, and a harmonic model is defined on each frame. In traditional signal processing approaches, the frames are processed almost independently. Here, all frames are processed together via the model in Eq. (3) which states that the frequencies are the same in each frame; this potentially makes the estimation of unknown parameters more accurate.

Matrix and vector notation for the above models will be used from now on, as follows. Let $\boldsymbol{\beta}$ denote the vector of amplitude parameters with length $2R(I+1)$, where $R = \sum_{k=1}^{K} M_k$ is the overall number of partials (or *model order*),

$$\beta_{2R_k(I+1)+2(M_k i+m)-1} = a_{k,m,i},$$
$$\beta_{2R_k(I+1)+2(M_k i+m)} = b_{k,m,i}, \tag{4}$$

for $i = 0, \ldots, I$, $m = 1, \ldots, M_k$, and $k = 1, \ldots, K$, with $R_k = \sum_{j=1}^{k-1} M_j$ (with the convention that $R_1 = 0$). Similarly, the matrix $\mathbf{D}$ of size $N \times 2R(I+1)$, whose columns are the Gabor atoms located at specified time instants $i\Delta_t$ and partial frequencies $\omega_{k,m}$, has entries defined by

$$\mathbf{D}_{t,2R_k(I+1)+2(M_k i+m)-1} = \phi(t - i\Delta_t)\cos\left(\frac{\omega_{k,m}}{\omega_s} t\right),$$
$$\mathbf{D}_{t,2R_k(I+1)+2(M_k i+m)} = \phi(t - i\Delta_t)\sin\left(\frac{\omega_{k,m}}{\omega_s} t\right), \tag{5}$$

for $i = 0, \ldots, I$, $m = 1, \ldots, M_k$, and $k = 1, \ldots, K$. Let $\mathbf{y} = [y[1], \ldots, y[N]]^t$ and $\mathbf{v} = [v[1], \ldots, v[N]]^t$ denote the vectors of signal samples and noise samples, respectively, where $\cdot^t$ denotes matrix transpose. Then Eq. (3) can be written more compactly as

$$\mathbf{y} = \mathbf{D}\boldsymbol{\beta} + \mathbf{v}. \tag{6}$$

The unknown parameters defining this model are the amplitudes $\boldsymbol{\beta}$, the number of partials for each note $\mathbf{M} = [M_1, \ldots, M_K]$, the partial frequencies $\boldsymbol{\omega} = [\omega_{1,1}, \ldots, \omega_{1,M_1}, \ldots, \omega_{K,1}, \ldots, \omega_{K,M_K}]$, and the variance of the noise vector $\mathbf{v}$, σ_v^2. The total number of unknown parameters is $R(2I+3)+2$. Considering a music signal with $N = 10\,000$ time samples, $K = 2$ notes, and $R = 50$ partials, the number of parameters is 1651 for $I = 15$ (this corresponds to atoms with length $2N_g + 1 = 30$ ms and 50% overlap).

B. Partial frequency models

In music produced by real instruments, inharmonicity will often be significant, i.e., the simple relation $\omega_{k,m} = m\omega_{k,1}$ (for $k = 1, \ldots, K$) is not satisfied. A well-known example is for struck or plucked strings[2] for which it is possible to develop a specific frequency model. However, given that we are here interested in analyzing generic harmonic instrumental music, in which the instruments playing may be unknown *a priori*, we adopt a more flexible form, which can adapt to less predictable inharmonicities found in real instruments:

$$\omega_{k,m} = m\omega_{k,1}(1 + \delta_{k,m}) \quad \text{where } \delta_{k,m} \ll 1 \text{ is to be estimated.} \tag{7}$$

The δ parameters are thus treated as unknown random variables within the proposed estimation scheme. Note that this new inharmonicity model is different from the one proposed in previous works.[29,30] The new model, being multiplicative rather than additive, allows more flexibility for high-frequency partials to deviate from their nominal "ideal" frequency $m\omega_{k,1}$. Note also that for a given k, the $\omega_{k,1}$ and $\delta_{k,m}$ terms uniquely specify the partial frequencies $\omega_{k,m}$. Thus we can work with either parametrization interchangeably, and we will usually refer to the $\omega_{k,m}$ terms directly in the following.

We now embed the polyphonic harmonic model defined in Eq. (3) in a probabilistic framework so as to allow efficient and accurate estimation of the model parameters, given an acoustic signal $\mathbf{y}$.

C. Bayesian harmonic music model

Assume the noise samples $\mathbf{v}$ in Eq. (6) are independent and identically distributed random variables, with zero-mean

Gaussian distribution (whose variance is denoted σ_v^2). Then the likelihood function of the model parameters is

$$p(\mathbf{y}|\boldsymbol{\beta},\sigma_v^2,\boldsymbol{\omega},\mathbf{M},K) = (2\pi\sigma_v^2)^{-N/2}\exp\left[-\frac{1}{2\sigma_v^2}\|\mathbf{y}-\mathbf{D}\boldsymbol{\beta}\|^2\right]. \tag{8}$$

Remark: Here, the noise is assumed white Gaussian. It is also possible to implement a correlated Gaussian (or even non-Gaussian) noise model by appropriate modification of the likelihood Eq. (8). An example of the use of autoregressive coloured noise can be found in our previous work[29] (see also printed comments from discussants). Here we do not explore this option, as simulations showed that autoregressive colored noise could capture some of the partials, thus making computation complicated and degrading convergence of the algorithms.

Since the objective is to estimate the note parameters, one could simply try to find the maximum likelihood (ML) parameter set $(\hat{\boldsymbol{\beta}}_{\mathrm{ML}},\hat{\sigma}^2_{v\mathrm{ML}},\hat{\boldsymbol{\omega}}_{\mathrm{ML}},\hat{\mathbf{M}}_{\mathrm{ML}},\hat{K}_{\mathrm{ML}})$ that maximizes the likelihood function. The ML parameter estimate $(\hat{\boldsymbol{\beta}}_{\mathrm{ML}},\hat{\sigma}^2_{v\mathrm{ML}},\hat{\boldsymbol{\omega}}_{\mathrm{ML}},\hat{\mathbf{M}}_{\mathrm{ML}},\hat{K}_{\mathrm{ML}})$ minimizes the energy of the residual $\mathbf{r}_{\mathrm{ML}}=\mathbf{y}-\hat{\mathbf{D}}_{\mathrm{ML}}\hat{\boldsymbol{\beta}}_{\mathrm{ML}}$, where $\hat{\mathbf{D}}_{\mathrm{ML}}$ is computed with the parameters $(\hat{\boldsymbol{\beta}}_{\mathrm{ML}},\hat{\sigma}^2_{v\mathrm{ML}},\hat{\boldsymbol{\omega}}_{\mathrm{ML}},\hat{\mathbf{M}}_{\mathrm{ML}},\hat{K}_{\mathrm{ML}})$ [see Eq. (8)]. This approach is, however, not satisfactory in practice as it tends to favor solutions with too many partials M_k and notes K (see Refs. 34 and 10). This is a well-known over-fitting problem of ML procedures which can be dealt with in principle using over-fitting penalization terms and methods such as AIC[35] and BIC[36] to determine an appropriate model order.

Here we choose to penalize over-fitting of the data, both by the parameters $\boldsymbol{\beta},\sigma_v^2,\boldsymbol{\omega}$ and the overall model order R in a fully Bayesian framework. This is achieved by specifying *prior distributions* that encode our prior knowledge or belief about parameter values and their probable variability between different notes, instruments, recording setup, etc. A joint prior probability distribution $p(\boldsymbol{\beta},\sigma_v^2,\boldsymbol{\omega},\mathbf{M},K)$ is specified, leading via Bayes' theorem to a *posterior distribution*

$$p(\boldsymbol{\beta},\sigma_v^2,\boldsymbol{\omega},\mathbf{M},K|\mathbf{y}) \propto p(\mathbf{y}|\boldsymbol{\beta},\sigma_v^2,\boldsymbol{\omega},\mathbf{M},K) \times p(\boldsymbol{\beta},\sigma_v^2,\boldsymbol{\omega},\mathbf{M},K). \tag{9}$$

A possible estimate of the model order parameters is then given by maximum *a posteriori* (MAP), as follows

$$(\hat{\mathbf{M}},\hat{K}) = \arg\max_{(\mathbf{M},K)} p(\mathbf{M},K|\mathbf{y})$$

where

$$p(\mathbf{M},K|\mathbf{y}) = \int p(\boldsymbol{\beta},\sigma_v^2,\boldsymbol{\omega},\mathbf{M},K|\mathbf{y})d\boldsymbol{\beta}d\sigma_v^2d\boldsymbol{\omega}, \tag{10}$$

while the parameters for a particular model could be estimated for example by the minimum mean squared error (MMSE) estimator (written below) of the amplitudes parameter:

$$\hat{\boldsymbol{\beta}} = \int \boldsymbol{\beta}p(\boldsymbol{\beta},\sigma_v^2,\boldsymbol{\omega}|\mathbf{y},\hat{\mathbf{M}},\hat{K})d\boldsymbol{\beta}d\sigma_v^2d\boldsymbol{\omega}. \tag{11}$$

In fact, this approach gives a somewhat simplified viewpoint, since the posterior expectations above may not yield an appropriate estimator of $\boldsymbol{\beta}$ in our case (the ordering of individual notes in $\boldsymbol{\beta}$, σ_v^2, $\boldsymbol{\omega}$, and $\mathbf{M}$ is nonunique, this is the *label switching problem*). Bayesian estimation in label switching contexts is quite difficult. Stephens[37] proposes a relabelling algorithm, and Jasra[38] proposes a review of possible relabelling methods. The technique employed here is simpler—further discussion is left until the MCMC algorithms are presented.

The posterior distribution and any Bayesian estimates made are influenced through Eq. (9) by both the likelihood function, which evaluates the model fit to the data $\mathbf{y}$, and the priors that encode the prior knowledge. The constant of proportionality in Eq. (9) will not be required as it depends only upon the (fixed) data $\mathbf{y}$—see Sec. III.

1. Prior distributions

A hierarchical structure is selected, which has the advantage of expressing $p(\boldsymbol{\beta},\sigma_v^2,\boldsymbol{\omega},\mathbf{M},K)$, whose structure is complex, as a product of simpler elementary priors:

$$p(\boldsymbol{\beta},\sigma_v^2,\boldsymbol{\omega},\mathbf{M},K) = p(\boldsymbol{\beta}|\sigma_v^2,\boldsymbol{\omega},\mathbf{M},K)p(\boldsymbol{\omega}|\mathbf{M},K) \times p(\mathbf{M}|K)p(K)p(\sigma_v^2). \tag{12}$$

The individual prior terms above are now detailed. For some discussion of alternative priors, see our previous work.[29]

(*i*) The prior distribution $p(\boldsymbol{\beta}|\sigma_v^2,\boldsymbol{\omega},\mathbf{M},K)$ for the $2R(I+1)$-dimensional amplitude parameter $\boldsymbol{\beta}$ is selected as zero-mean Gaussian, with covariance matrix $(\sigma_v^2/\xi^2)\mathbf{I}$, where $\mathbf{I}$ is an identity matrix and ξ^2 is a scaling parameter that can be interpreted as a signal-to-noise ratio, as pointed out by Andrieu and Doucet.[39] As explained in previous work,[40] selecting a diagonal covariance matrix permits the fast MCMC implementation presented below. Moreover, this diagonal covariance matrix does not degrade noticeably the overall algorithm performance, while enabling dramatic computational savings. Comparing the results presented in our previous work[29] to those presented below shows that the parameter estimation is accurate in both cases, while the computation time is significantly shorter with the new algorithm. The *hyperparameter* ξ^2 used to define the prior of $\boldsymbol{\beta}$ is treated as an additional unknown parameter in our framework, so that the full parameter set is actually augmented to $[\boldsymbol{\beta},\sigma_v^2,\boldsymbol{\omega},\xi^2,\mathbf{M},K]$. In the following, we do not explicitly include ξ^2 in the unknown parameter set for the sake of notational simplicity. The prior distribution for ξ^2 is chosen as inverted gamma with small parameters, e.g., $\alpha_\xi=10^{-4}$ and $\beta_\xi=10^{-4}$:

$$p(\xi^2) = \mathcal{IG}(\alpha_\xi,\beta_\xi) \propto \frac{e^{-\beta_\xi/\xi^2}}{\xi^{2(\alpha_\xi+1)}}. \tag{13}$$

(*ii*) The prior distribution for frequencies $p(\boldsymbol{\omega}|\mathbf{M},K)$. In the general model of Eq. (7), the frequency prior structure is

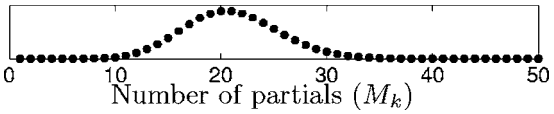

FIG. 2. Prior distribution of the number of partials $p(M_k)$ for any $k=1,\ldots,K$. The Λ_k hyperparameter is set to 20 in this case. This prior favors values of M_k between approximately 10 and 30.

$$p(\boldsymbol{\omega}|\mathbf{M},K)=\prod_{k=1}^{K}\left[p(\omega_{k,1}|M_k)\prod_{m=1}^{M_k}p(\delta_{k,m})\right], \tag{14}$$

where each $p(\delta_{k,m})$ is zero-mean Gaussian with variance σ_δ^2, and, for generality, $p(\omega_{k,1}|M_k)$ is uniform over the whole frequency range $[0,\omega_s/2M_k]$ for each note $k=1,\ldots,K$. Note that $p(\delta_{k,m})$ must be restricted to $[-1/2m,1/2m]$ so that the partial frequencies cannot switch. In practice, σ_δ^2 is a user-defined parameter set to the same very small value for each partial (we used $\sigma_\delta^2=3\times10^{-8}$ in simulations) and the restriction to $[-1/2m,1/2m]$ is numerically true without explicitly truncating $p(\delta_{k,m})$. The maximum possible frequency is $\omega_s/2M_k$ in order to avoid aliasing problems. Other more instrument- or genre-specific frequency priors could easily be envisaged (see Appendix A for a piano example).

(*iii*) The parameter $\mathbf{M}$ tunes the number of partials comprising the notes. Accurate estimation of M_k $(k=1,\ldots,K)$ is critical for correct frequency estimation. On the one hand, too many partials (M too large) generally result in underestimation of the fundamental frequency $\omega_{k,1}$: the estimated $\omega_{k,1}$ is generally one or two octaves below the correct frequency, and many partials with small amplitude appear in between the actual partials. On the other hand, not enough partials (M too small) does not capture enough information, and this may be a problem for signal reconstruction. It is thus crucial to define a prior which penalizes large numbers of partials, while allowing possibly many partials when needed. Following Andrieu and Doucet,[39] we have found that a balanced solution consists of selecting a *Poisson distribution* for each M_k,

$$p(M_k=m|\Lambda_k)=e^{-\Lambda_k}\frac{\Lambda_k^m}{m!}\quad\text{for } k=1,\ldots,K, \tag{15}$$

with $p(\mathbf{M}|K)=\Pi_{k=1}^{K}p(M_k|\Lambda_k)$, where Λ_k is left random with prior $p(\Lambda_k)$. The prior $p(\Lambda_k)$ is a gamma distribution with the same user-defined parameters α_Λ and β_Λ for each note,

$$p(\Lambda_k)=\mathcal{G}(\alpha_\Lambda,\beta_\Lambda)\propto\Lambda_k^{\alpha_\Lambda-1}e^{-\beta_\Lambda\Lambda_k}, \tag{16}$$

where the parameters are selected as $\alpha_\Lambda=1$ and $\beta_\Lambda=2$ to ensure this prior being vague, with infinite mean and variance.

For a given Λ_k, the shape of $p(M_k=m)$ is plotted in Fig. 2, with $\Lambda_k=20$. Again, we do not include it explicitly in the parameter set. In practice, M_k is limited to the range $[M_{\min},\ldots,M_{\max}]$, and $p(M_k=m)=(\Lambda_k^m/m!)/(\Sigma_{m'=M_{\min}}^{M_{\max}}\Lambda_k^{m'}/m'!)$.

This two-level prior is actually equivalent to the one-level prior

$$\begin{aligned}p(M_k=m)&=\int p(M_k=m|\Lambda_k)p(\Lambda_k)d\Lambda_k\\&\propto\int e^{-\Lambda_k}\frac{\Lambda_k^m}{m!}\Lambda_k^{\alpha_k-1}e^{-\beta_\Lambda\Lambda_k}\,d\Lambda_k\\&\propto\frac{\Gamma(m+\alpha_\Lambda)}{\Gamma(\alpha_\Lambda)m!}(\beta_\Lambda+1)^{-m},\end{aligned} \tag{17}$$

where $\Gamma(\cdot)$ is the Gamma function which coincides with the factorial operator $\Gamma(a+1)=a!$ for integer a. In practice, α_λ is set to one, yielding $p(M_k=m)=(\beta_\Lambda+1)^{-m}$, which is a monotonically decreasing function with decreasing rate set by β_Λ. This prior modeling enables a flexible framework: changing the parameters α_Λ and β_Λ defines various families of priors $p(M_k=m)$, with different decreasing profiles.

(*iv*) The number of notes K has a similar two-level prior as M_k, $k=1,\ldots,K$, with hyperparameter denoted Λ'.

(*v*) The noise amplitude parameter prior $p(\sigma_v^2)$ is selected as an inverted Gamma distribution

$$p(\sigma_v^2)\propto(\sigma_v^2)^{-\gamma_0/2-1}e^{-2/\nu_0\sigma_v^2}, \tag{18}$$

where ν_0 and γ_0 have small user-defined values, since this will favor solutions having small residual energy[41]

Using Eq. (9), one can compute the posterior distribution $p(\boldsymbol{\beta},\sigma_v^2,\boldsymbol{\omega},\mathbf{M},K|\mathbf{y})$. Moreover, the integral

$$p(\boldsymbol{\omega},\mathbf{M},K|\mathbf{y})=\int p(\boldsymbol{\beta},\sigma_v^2,\boldsymbol{\omega},\mathbf{M},K|\mathbf{y})d\boldsymbol{\beta}d\sigma_v^2 \tag{19}$$

can be calculated analytically in our case. In addition, $p(\sigma_v^2|\boldsymbol{\omega},\mathbf{M},K,\mathbf{y})$ and $p(\boldsymbol{\beta}|\sigma_v^2,\boldsymbol{\omega},\mathbf{M},K,\mathbf{y})$ can also be computed analytically. Standard computations lead to

$$\begin{aligned}p(\boldsymbol{\omega},\mathbf{M},K|\mathbf{y})\propto{}&(\gamma_0+\mathbf{y}^t\mathbf{P}\mathbf{y})^{(-N+\nu0)/2}\det(\mathbf{S})^{1/2}p(\boldsymbol{\omega}|\mathbf{M},K)\\&\times\exp\left(-\sum_{k=1}^{K}\Lambda_k\right)\frac{\Lambda'^K}{K!}\prod_{k=1}^{K}\frac{\Lambda_k^{M_k}}{M_k!},\end{aligned} \tag{20}$$

where $\mathbf{P}=\mathbf{I}-\mathbf{D}\mathbf{S}\mathbf{D}^t$ is an N-dimensional square matrix whose computation requires the inversion of the $2R(I+1)$-dimensional square matrix $\mathbf{S}=[\mathbf{D}^t\mathbf{D}+(1/\xi^2)\mathbf{I}]^{-1}$. The posterior distribution of σ_v^2, conditional on the other parameters, is an inverse gamma distribution with parameters $(N+\nu_0)/2$ and $(\gamma_0+\mathbf{y}^t\mathbf{P}\mathbf{y})/2$,

$$\begin{aligned}p(\sigma_v^2|\boldsymbol{\omega},\mathbf{M},\mathbf{y})&=\mathcal{IG}\left(\frac{N+\nu_0}{2},\frac{\gamma_0+\mathbf{y}^t\mathbf{P}\mathbf{y}}{2}\right)\\&\propto\exp\left(-\frac{\gamma_0+\mathbf{y}^t\mathbf{P}\mathbf{y}}{2\sigma_v^2}\right)\sigma_v^{-N-\nu_0-2},\end{aligned} \tag{21}$$

and posterior distribution of $\boldsymbol{\beta}$ conditional on the other parameters is a Gaussian distribution in dimension $2R(I+1)$,

$$p(\boldsymbol{\beta}|\sigma_v^2,\boldsymbol{\omega},\mathbf{M},\mathbf{y})=\mathcal{N}(\boldsymbol{\mu},\sigma_v^2\mathbf{S}), \tag{22}$$

where the mean is $\boldsymbol{\mu}=\mathbf{S}\mathbf{D}^t\mathbf{y}$ and the covariance matrix is $\sigma_v^2\mathbf{S}$. Efficient strategies to compute these quantities are presented in Ref. 40.

III. ALGORITHM FOR PARAMETER ESTIMATION

As stated in the text around Eq. (11), parameter estimation of, for example, the frequency parameter $\boldsymbol{\omega}$ requires the evaluation of integrals which cannot be calculated analytically. It is thus necessary to compute such integrals using numerical techniques. Grid-based and other simplistic approaches are not feasible owing to the high dimensionality of the problem and the high accuracy required in the frequency parameters in order to fit the data well.

The approach we adopt instead is a Monte Carlo integration. Assume we are able to generate random samples $(\tilde{\boldsymbol{\beta}}^{(l)}, \tilde{\sigma}_v^{2(l)}, \tilde{\boldsymbol{\omega}}^{(l)}, \tilde{\mathbf{M}}^{(l)}, \tilde{K}^{(l)})$, called Monte Carlo samples, distributed jointly according to $p(\boldsymbol{\beta}, \sigma_v^2, \boldsymbol{\omega}, \mathbf{M}, K|\mathbf{y})$ for some large number of samples $l=1,\ldots,L$. Then these samples can be used to form Monte Carlo estimates of any of the unknown parameters, as explained in Sec. III B. The actual generation of the Monte Carlo samples is performed using a Markov chain technique presented in Sec. III A.

A. Markov chain Monte Carlo (MCMC)

In order to estimate the unknown parameters, it is necessary to generate samples $(\tilde{\boldsymbol{\beta}}^{(l)}, \tilde{\sigma}_v^{2(l)}, \tilde{\boldsymbol{\omega}}^{(l)}, \tilde{\mathbf{M}}^{(l)}, \tilde{K}^{(l)})$ distributed according to $p(\boldsymbol{\beta}, \sigma_v^2, \boldsymbol{\omega}, \mathbf{M}, K|\mathbf{y})$ that will be used in the Monte Carlo estimation procedures described in Sec. III B. A general technique consists of generating a *Markov chain* using an iterative algorithm. Monte Carlo estimation together with Markov chain samplers forms the class of MCMC methods.

Generally speaking, Markov chain algorithms produce a series of samples whose distribution asymptotically approaches a given so-called *target distribution* $\pi(\boldsymbol{\theta})$, where $\boldsymbol{\theta}$ denotes a parameter vector to be estimated. In this paper, $\boldsymbol{\theta}$ is composed of $\boldsymbol{\beta}$, σ_v^2, $\boldsymbol{\omega}$, $\mathbf{M}$, K, and $\pi(\boldsymbol{\theta}) = p(\boldsymbol{\beta}, \sigma_v^2, \boldsymbol{\omega}, \mathbf{M}, K|\mathbf{y})$. Two important algorithms are the *Gibbs sampler* and the *Metropolis-Hastings sampler*.[42,43] The MCMC algorithm have been widely used in signal processing applications.[27,28,39,44–46] In this paper, we implement a specially constructed version of Metropolis-Hastings (MH) sampling, whose principle is described in Algorithm 1 for a vector $\boldsymbol{\theta}$ with a fixed number of dimensions N_θ and for the target distribution $\pi(\boldsymbol{\theta})$. Another presentation of the MCMC in the context of acoustics can be found in Ref. 47.

Algorithm 1 consists of sampling the components of $\boldsymbol{\theta}$ one at a time, according to a distribution $q(\boldsymbol{\theta})$. This distribution is called a *proposal*, because it is used to form the proposed candidate parameter value, denoted $\boldsymbol{\theta}^\star$. This candidate is randomly accepted or rejected according to the ratio (23). After some (large number of) iterations, the samples produced $\{\ldots, \tilde{\boldsymbol{\theta}}_j^{(l)}, \tilde{\boldsymbol{\theta}}_j^{(l+1)}, \tilde{\boldsymbol{\theta}}_j^{(l+2)}, \ldots\}$ are distributed according to the distribution $\pi(\boldsymbol{\theta})$, as required.

Algorithm 1: Generic one-at-a-time Metropolis-Hastings algorithm

1. Initialization.

- Set $l \leftarrow 1$

% Step 1.1 The parameter vector $\boldsymbol{\theta}$ is initialized

- For $j=1,\ldots,N_\theta$, do
  - Sample $\tilde{\boldsymbol{\theta}}^{(1)}$ according to some initial proposal distribution $q_1(\boldsymbol{\theta})$

2. While $l<L$, do

%Step 1.2 The parameter vector $\boldsymbol{\theta}$ is sampled one component at a time

- For $j=1,\ldots,N_\theta$, do (where j is the component index in $\boldsymbol{\theta}$)
  - Form the candidate by first setting $\boldsymbol{\theta}^\star \leftarrow \tilde{\boldsymbol{\theta}}^{(l-1)}$, then changing its component #j (denoted $\theta_j^\star$) by sampling it according to the proposal distribution $q(\theta_j|\tilde{\theta}_j^{(l-1)})$
  - Compute the MH ratio defined as follows

$$g(\boldsymbol{\theta}^\star, \tilde{\boldsymbol{\theta}}^{(l-1)}) = \frac{\pi(\boldsymbol{\theta}^\star)}{\pi(\tilde{\boldsymbol{\theta}}^{(l-1)})} \frac{q(\tilde{\theta}_j^{(l-1)}|\theta_j^\star)}{q(\theta_j^\star|\tilde{\theta}_j^{(l-1)})} \tag{23}$$

  - With probability $\alpha=\min[1, g(\boldsymbol{\theta}^\star, \tilde{\boldsymbol{\theta}}^{(l-1)})]$, accept the candidate $\boldsymbol{\theta}^\star$, i.e., set $\tilde{\boldsymbol{\theta}}^{(l)}=\boldsymbol{\theta}^\star$ or, with probability $1-\alpha$, reject the candidate, i.e., set $\tilde{\boldsymbol{\theta}}^{(l)} = \tilde{\boldsymbol{\theta}}^{(l-1)}$
- Set $l \leftarrow l+1$

A key element in MH algorithms is the *proposal distribution* q. This distribution must be simple enough to enable direct sampling, for example a Gaussian or uniform distribution. In principle, MCMC algorithms explore the space of possible values of $\boldsymbol{\theta}$ and focus on regions where the probability mass of $\pi(\boldsymbol{\theta})$ is large. By construction, it is able to switch from one region of the parameter space to another region where $\pi(\boldsymbol{\theta})$ has large probability mass. The computation of the accept/reject ratio involves the ratio $\pi(\boldsymbol{\theta}^\star)/\pi(\tilde{\boldsymbol{\theta}}^{(l-1)})$, which shows that π needs only be known up to a multiplicative constant, as any proportionality constant cancels out. Algorithm 1 is designed for variables $\boldsymbol{\theta}$ with fixed dimension N_θ. This is not the case in our problem since the dimension of $\boldsymbol{\theta}$ depends on $\mathbf{M}$ and K. Special extensions of MH algorithms, known as reversible jump MCMC, are designed to address cases where N_θ is unknown,[43,45,48] and we adopt these in this paper, following previous work.[27–30,39]

B. Computation of parameter estimates

We now assume that a Markov chain of samples has been generated and that only the samples after convergence are kept. They are denoted $(\tilde{\boldsymbol{\beta}}^{(l)}, \tilde{\sigma}_v^{2(l)}, \tilde{\boldsymbol{\omega}}^{(l)}, \tilde{\mathbf{M}}^{(l)}, \tilde{K}^{(l)})$ for $l=1,\ldots,L$. As discussed earlier, estimation of parameters using Bayesian methods can be carried out in many differing ways, such as maximizing the posterior distribution or using posterior mean parameter estimates. Here, however, none of these approaches is suitable. Owing to the label switching problem, there exist many sets of parameters which achieve the same posterior probability: they are all related through a different ordering of the individual notes in the parameter vectors $\boldsymbol{\beta}$, $\boldsymbol{\omega}$, and $\mathbf{M}$ (label switching problem). It is thus risky to compute averages over the Monte Carlo samples,

since we could average together fundamental frequencies of, e.g., the note with true $\omega_{k,1}=440$ Hz with the note with true $\omega_{k',1}=660$ Hz. Moreover, the posterior admits also local maxima corresponding to octave ambiguities, fifth ambiguities, etc., which would also be mixed in the posterior mean estimate. In order to limit this problem, we adopt the following scheme: First, the number of notes K is estimated by maximizing the marginal posterior distribution $p(K|\mathbf{y})$ (*marginal MAP estimation scheme*). This requires the integration of $\boldsymbol{\beta}, \sigma_v^2, \boldsymbol{\omega}, \mathbf{M}$ in the full posterior $p(\boldsymbol{\beta}, \sigma_v^2, \boldsymbol{\omega}, \mathbf{M}, K|\mathbf{y})$. This is computed very simply from the Monte Carlo samples by only considering the $\tilde{K}^{(l)}$ component and setting $\hat{K}$ as the most represented number of notes among the samples $\tilde{K}^{(l)}$. Second, $\widehat{\mathbf{M}}$ is estimated. In theory, the marginal MAP procedure cannot be used to estimate $\mathbf{M}$ because of the label switching problem. However, this solution is practical insofar as the Markov chains generated in simulations do not switch the labels whenever convergence to one of the maxima of $p(\boldsymbol{\beta}, \sigma_v^2, \boldsymbol{\omega}, \mathbf{M}, K|\mathbf{y})$ has occurred. In simulations, the chain converges to a posterior local maximum and keeps exploring around by adding or removing partials and changing $\boldsymbol{\omega}$ and $\boldsymbol{\beta}$, but without jumping to a completely different posterior maximum where the notes would be ordered differently (this happens if we run millions of iterations, but it does not improve the practical parameter estimation).

The Markov chain's practical difficulty in jumping from one local maximum to another enables, paradoxically, the best possible Bayesian estimates of $\boldsymbol{\beta}$, σ_v^2, $\boldsymbol{\omega}$, and $\mathbf{M}$: the discrete parameter $\widehat{\mathbf{M}}$ is computed by marginal MAP, which consists here of counting the number of Monte Carlo samples such that $\tilde{K}^{(l)}=\hat{K}$ and $\tilde{\mathbf{M}}^{(l)}=\mathbf{M}$ for each possible value of $\mathbf{M}$, and selecting the most represented value as the estimate $\widehat{\mathbf{M}}$. The continuous parameters $\boldsymbol{\beta}, \sigma_v^2, \boldsymbol{\omega}$ are computed by marginal MMSE estimation, that is, by averaging over the Monte Carlo samples that satisfy jointly $\tilde{K}^{(l)}=\hat{K}$ and $\tilde{\mathbf{M}}^{(l)}=\widehat{\mathbf{M}}$. For example, the fundamental partial frequency of note k is estimated by

$$\hat{\omega}_{k,1}=\frac{1}{L'}\sum_{l'=1}^{L'}\omega_{k,1}^{(l')}, \tag{24}$$

where the notation l' and L' is used to emphasize the restriction to Monte Carlo samples that satisfy jointly the condition $\tilde{K}^{(l)}=\hat{K}$ and $\tilde{\mathbf{M}}^{(l)}=\widehat{\mathbf{M}}$. It is also possible to evaluate our uncertainty about the value of this parameter via the empirical variance estimate $(1/L')\Sigma_{l'=1}^{L'}[\omega_{k,1}^{(l')}-\hat{\omega}_{k,1}]^2$.

The above Monte Carlo parameter estimation technique is based on the law of large numbers which states that the expectation of a random variable, Eq. (11), can be approximated by the average of numerous samples of this random variable. Note that a suitably defined Monte Carlo integration may only require a few hundred samples $(\tilde{\boldsymbol{\beta}}^{(l)}, \tilde{\sigma}_v^{2(l)}, \tilde{\boldsymbol{\omega}}^{(l)}, \tilde{\mathbf{M}}^{(l)})$, as opposed to the huge number of grid points in Riemann integration for equivalent precision.[43]

C. A fast MCMC algorithm for polyphonic harmonic models

There are many possible options for creating a MCMC algorithm for a complex model such as this. The choices include which parameters to group together in each MH sampling step, which parameters to integrate out or marginalise (known as *Rao-Blackwellization*[43]), which proposal distributions to use, and how to design the most efficient implementations. Here we choose a scheme that has been optimized for computational speed while retaining good convergence properties.

The algorithm we implement in order to generate samples from the target distribution $p(\boldsymbol{\theta}, \mathbf{M}, K|\mathbf{y})$ is in essence a MH algorithm with several reversible jump steps, since the dimensions of $\boldsymbol{\theta}$, $\mathbf{M}$, and K vary together. The algorithm can be described hierarchically:

(i) At the highest level, we either sample the number of notes K or the parameters of the notes and we sample the noise variance σ_v^2 (see Algorithm 2).
(ii) At the level of notes, k is fixed and we want to sample the number of partials M_k (see Algorithm 5 below and Algorithms 7–10 in Appendix C).
(iii) At the level of individual partials, k and m are fixed and we want to sample the frequency $\omega_{k,m}$ and amplitudes $a_{k,m,i}$ and $b_{k,m,i}$, $i=0,\ldots,I$ (see Algorithm 6).

1. General algorithm structure

We now describe the algorithms from the note level to the frequency level. In these algorithms, every MH step has the same general structure as in Algorithm 1 and consists of (1) generating a candidate, (2) computing the MH ratio, and (3) performing the accept/reject test with respect to the target distribution. In order to simplify notation we omit the symbols $\tilde{\cdot}^{(l)}$ and $\cdot^{\star}$ wherever it is clear from the context that we are dealing with Markov chain samples and with candidates.

Algorithm 2: Overall Metropolis-Hastings algorithm for music

1. Initialization.

- Set $l \leftarrow 1$

% Step 2.1 Initialize the parameters $\boldsymbol{\omega}$, σ_v^2, $\boldsymbol{\beta}$, $\mathbf{M}$ and K

- Sample $\tilde{K}_1$ according to some initial distribution $q_{\text{init}}(K)$
- For $k=1,\ldots,\tilde{K}_1$ sample $\tilde{M}_k^{(1)}$ according to its prior distribution Eq. (15)
- Sample $\tilde{\omega}^{(1)}$ according to $q_{\text{init}}(\omega|\mathbf{y})$ where $q_{\text{init}}(\omega|\mathbf{y})$ is the probability distribution proportional to the Fourier spectrum of $\mathbf{y}$ (see Ref. 39 for a similar implementation).
- Sample the noise variance parameter $\tilde{\sigma}_v^{2(1)}$ according to $p(\sigma_v^2|\tilde{\boldsymbol{\omega}}^{(1)}, \tilde{\mathbf{M}}^{(1)}, \mathbf{y})$ given in Eq. (21)
- Sample the amplitudes $\tilde{\boldsymbol{\beta}}^{(1)}$ according to $p(\boldsymbol{\beta}|\tilde{\sigma}_v^{2(1)}, \tilde{\boldsymbol{\omega}}^{(1)}, \tilde{\mathbf{M}}^{(1)}\mathbf{y})$ given in Eq. (22)

2. While $l<L$, do

% Step 2.2 Sample the note parameters $\widetilde{K}^{(l)}$, $\widetilde{\mathbf{M}}^{(l)}$, $\widetilde{\boldsymbol{\omega}}^{(l)}$ and $\widetilde{\boldsymbol{\beta}}^{(l)}$

- with probability μ_K, try to add a note using the *birth move* (see Algorithm 3).
- Otherwise, with probability ν_K, try to remove a note using the *death move* (see Algorithm 4).
- Otherwise, with probability $1-\mu_K-\nu_K$, try the *update move* as follows
  - Set $\widetilde{K}^{(l)} \leftarrow \widetilde{K}^{(l-1)}$
  - For $k=1,\ldots,\widetilde{K}^{(l)}$, update the parameters of note #k (see Algorithm 5).
- For $k=1,\ldots,K$, update the parameters of note #k (Algorithm 5).

% Step 2.3 Sample the noise variance parameter σ_v^2 from $p(\sigma_v^2|\widetilde{\boldsymbol{\omega}}^{(l)},\widetilde{\mathbf{M}}^{(l)},\mathbf{y})$ given in Eq. (21)

- Set $l \leftarrow l+1$,

In Algorithm 2, $\widetilde{\mathbf{D}}^{(l)}$ denotes the matrix $\mathbf{D}$ computed with the parameters $\widetilde{\boldsymbol{\omega}}^{(l)}$, $\widetilde{\mathbf{M}}^{(l)}$, and $\widetilde{K}^{(l)}$ [see Eq. (5)]. The birth and death moves are aimed at adding/removing notes and include a reversible jump accept/reject test, similar in principle to the test in Algorithm 1: a candidate is generated, then it is tested with respect to the sample of iteration $l-1$ via a *reversible jump* MH accept/reject test. The probabilities to try the birth move (μ_K) or the death move (ν_K) are computed at each iteration based on the prior distribution of K, and the current value $\widetilde{K}^{(l-1)}$ (see Appendix B for details). The birth move is presented below, and the proposal distribution q_{note} is presented in Sec. III E.

Algorithm 3: Note birth

- Compute the residual $\mathbf{r}_0=\mathbf{y}-\widetilde{\boldsymbol{\beta}}^{(l-1)}\widetilde{\mathbf{D}}^{(l-1)}$
- Generate a candidate by increasing K, namely $K^{\bigstar} \leftarrow \widetilde{K}^{(l-1)}+1$
- Sample $M_{K+1}^{\bigstar}$ using the prior distribution $p(M_k)$ and set $\mathbf{M}^{\bigstar} \leftarrow [\widetilde{\mathbf{M}}^{(l-1)}, M_{K+1}^{\bigstar}]$
- Sample $\boldsymbol{\omega}_{K+1}^{\bigstar}=[\omega_{K+1,1}^{\bigstar},\ldots,\omega_{K+1,M_{K+1}}^{\bigstar}]$ using the proposal distribution $q_{\text{note}}(\boldsymbol{\omega}|\mathbf{r}_0,M_{K+1})$ and set $\boldsymbol{\omega}^{\bigstar} \leftarrow [\widetilde{\boldsymbol{\omega}}^{(l-1)}, \boldsymbol{\omega}_{K+1}^{\bigstar}]$
- Let $\mathbf{P}^{\bigstar}$ and $\mathbf{S}^{\bigstar}$ be the $\mathbf{P}$ and $\mathbf{S}$ matrices related to candidate note featuring $M_{K+1}^{\bigstar}$ partials with frequencies $\boldsymbol{\omega}_{K+1}^{\bigstar}$. Compute

$$g_{\text{birth}}=\left[\frac{\gamma_0+\|\mathbf{r}_0\|^2}{\gamma_0+\mathbf{r}_0^{\text{t}}\mathbf{P}^{\bigstar}\mathbf{r}_0}\right]^{(N+\nu_0)/2}\frac{p(\boldsymbol{\omega}_{K+1}^{\bigstar},M_{K+1}^{\bigstar},K^{\bigstar})}{p(\widetilde{K}^{(l-1)})}\frac{\nu_{K+1}}{\mu_K}\times\frac{\det(\mathbf{S}^{\bigstar})^{1/2}}{K^{\bigstar}q_{\text{note}}(\boldsymbol{\omega}_{K+1}^{\bigstar}|\mathbf{r}_0,M_{K+1}^{\bigstar})} \tag{25}$$

- With probability $\min(1,g_{\text{birth}})$, accept the candidate: Set $\widetilde{K}^{(l)} \leftarrow K^{\bigstar}$, $\widetilde{\mathbf{M}}^{(l)} \leftarrow \mathbf{M}^{\bigstar}$, $\widetilde{\boldsymbol{\omega}}^{(l)} \leftarrow \boldsymbol{\omega}^{\bigstar}$ and sample $\widetilde{\boldsymbol{\beta}}_{K+1}^{(l)}$ according to $p(\boldsymbol{\beta}_{K+1}|\widetilde{\sigma}_v^{2(l-1)},\widetilde{\boldsymbol{\omega}}_{K+1}^{(l)},\widetilde{M}_{K+1}^{(l)},K=1,\widetilde{I}^{(l)},\mathbf{r}_0)$ given in Eq. (22).

The opposite move is the death move, as detailed in the following algorithm.

Algorithm 4: Note death

- Generate a candidate by decreasing K, namely $K^{\bigstar} \leftarrow \widetilde{K}^{(l-1)}-1$,
- Sample a note index j in $\{1,\ldots,\widetilde{K}^{(l-1)}\}$ with equal probabilities
- Remove the parameters of note #j from $\widetilde{\mathbf{M}}^{(l-1)}$ and $\boldsymbol{\omega}^{(l-1)}$. This yields the candidates $\mathbf{M}^{\bigstar}$ and $\boldsymbol{\omega}^{\bigstar}$.
- Compute the residual $\mathbf{r}_j=\mathbf{y}-\boldsymbol{\beta}_{-j}\mathbf{D}_{-j}$ where $\boldsymbol{\beta}_{-j}$ (resp. $\mathbf{D}_{-j}$) is the vector of amplitudes $\widetilde{\boldsymbol{\beta}}^{(l-1)}$ (resp. the matrix of Gabor atoms $\widetilde{\mathbf{D}}^{(l-1)}$) computed with parameters $\widetilde{K}^{(l)}$, $\widetilde{\mathbf{M}}^{(l)}$, $\widetilde{\boldsymbol{\omega}}^{(l)}$ and $\widetilde{I}^{(l)}$ where the entries related to note #j have been removed.
- Let $\mathbf{P}_j$ and $\mathbf{S}_j$ be the $\mathbf{P}$ and $\mathbf{S}$ matrices related to note #j. Compute

$$g_{\text{death}}=\left[\frac{\gamma_0+\mathbf{r}_j^{\text{t}}\mathbf{P}_j\mathbf{r}_j}{\gamma_0+\|\mathbf{r}_j\|^2}\right]^{(N+\nu_0)/2}\frac{p(K^{\bigstar})}{p(\boldsymbol{\omega}_j,M_j,\widetilde{K}^{(l-1)})}\frac{\mu_{K-1}}{\nu_K}\widetilde{K}^{(l-1)}\times\frac{q_{\text{note}}(\boldsymbol{\omega}_j|\mathbf{r}_j,M_j)}{\det(\mathbf{S}_j)^{1/2}} \tag{26}$$

- With probability $\min(1,g_{\text{death}})$, accept the candidate: Set $\widetilde{K}^{(l)} \leftarrow K^{\bigstar}$, $\widetilde{\mathbf{M}}^{(l)} \leftarrow \mathbf{M}^{\bigstar}$, $\widetilde{\boldsymbol{\omega}}^{(l)} \leftarrow \boldsymbol{\omega}^{\bigstar}$ and remove the components corresponding to note #j from $\widetilde{\boldsymbol{\beta}}^{(l-1)}$ so as to obtain $\widetilde{\boldsymbol{\beta}}^{(l)}$

Algorithm 5, presented below, is aimed at sampling the parameters $\mathbf{M}$, $\boldsymbol{\omega}$, and $\boldsymbol{\beta}$ using a reversible jump MH scheme with $p(\boldsymbol{\beta},\boldsymbol{\omega},\mathbf{M}|\mathbf{y},\sigma_v^2,K)$ as target distribution. The steps involve proposing changes to the numbers of partials in each note, and also the frequencies in each note.

Algorithm 5: Metropolis-Hastings algorithm for note parameter updates

% Step 5.1 Compute the residual $\mathbf{r}_k$ that contains the note #k

- Compute the residual $\mathbf{r}_k=\mathbf{y}-\mathbf{D}_{-k}\boldsymbol{\beta}_{-k}$ where $\boldsymbol{\beta}_{-k}$ (resp. $\mathbf{D}_{-k}$) is the vector of amplitudes $\widetilde{\boldsymbol{\beta}}^{(l-1)}$ (resp. the matrix of Gabor atoms $\widetilde{\mathbf{D}}^{(l-1)}$) computed with parameters $\widetilde{\mathbf{M}}^{(l)}$ and $\widetilde{\boldsymbol{\omega}}^{(l)}$ where the entries related to note #k have been removed.

% Step 5.2 The number of partials M_k is sampled

- Sample a positive integer n uniformly in $[0,\ldots,\max(M_{\max}-\widetilde{M}_k^{(l-1)},\widetilde{M}_k^{(l-1)}-M_{\min})]$ and sample an integer e uniformly in $[-1,0,1]$
- With probability λb_{M_k}, try the *n-increase move*, see Algorithm 7 in Appendix C.
- Otherwise, with probability λd_{M_k}, try the *n-decrease move*, see Algorithm 8 in Appendix C.
- Otherwise, with probability $(1-\lambda)/2$, try the *divide move*, see Algorithm 9 in Appendix C.
- Otherwise, with probability $(1-\lambda)/2$, try the *multiply move*, see Algorithm 10 in Appendix C.
- Otherwise, with probability $\lambda(1-b_{M_k}-d_{M_k})$, try the *note update move* as follows
  - Set $\widetilde{M}_k^{(l)} \leftarrow \widetilde{M}_k^{(l-1)}$
  - Update the partial frequencies $\boldsymbol{\omega}_k$ (see Algorithm 6)

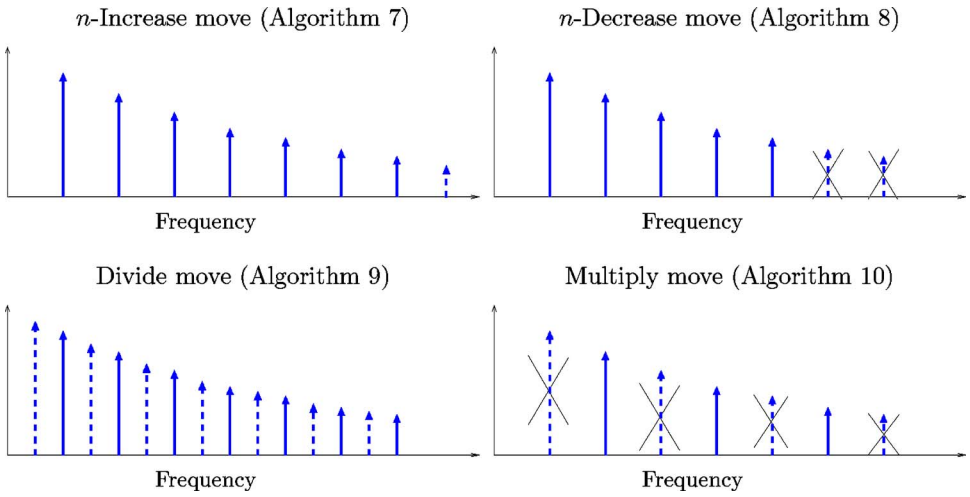

FIG. 3. (Color online) Schematic description of the four moves used to sample the number of partials M_k. From the initial situations, partials are either added (in dashed lines) or removed (in dashed lines, crossed). The partials plotted in solid lines are left unchanged.

In Algorithm 5, the probabilities b_{M_k} and d_{M_k} are computed in the same way as μ_K and ν_K (see Appendix B). The number of partials can be changed by two possible reversible moves as depicted in Fig. 3: either n partials are added (resp. removed) in the n-increase move (resp. n-decrease move) [see Algorithm 7 (resp. see Algorithm 8)] or the number of partials is multiplied by 2 (resp. divided by 2) in the divide move (resp. multiply move) [see Algorithm 10 (resp. see Algorithm 9)]. The multiply and divide moves are synchronized with fundamental frequency division/multiplication by two in order to avoid octave ambiguities: assume a note with frequency 440 Hz is played by an instrument. The notes with frequencies 220 and 880 Hz have partials that are almost superimposed with the partials of the actual note. Without these moves, the Markov chain may be stuck on incorrect fundamental frequencies, namely 220 or 880 Hz, which is not satisfactory. Note that the random parameter e is only aimed at ensuring reversibility of the divide and multiply moves. The parameter λ, selected by users in $[0;1]$, tunes the rate multiply-divide moves/increase-divide-update moves. A small λ is selected to avoid octave errors, because it favors multiply/divide moves. Algorithm 6, aimed at updating partial frequencies, is presented below.

Algorithm 6: Frequency update Metropolis-Hastings algorithm

- With probability λ_{note}, try the *global update*: sample $\boldsymbol{\omega}_k^{\star}$ according to $q_{\text{note}}(\boldsymbol{\omega}_k|\mathbf{r}_k,M_k)$ and perform an accept/reject test with respect to $p(\boldsymbol{\omega}_k,M_k,K=1|\mathbf{r}_k)$ given in Eq. (20) so as to obtain $\tilde{\boldsymbol{\omega}}_k^{(l)}$. Sample the amplitudes according to $p(\boldsymbol{\beta}_k|\tilde{\sigma}_v^{2(l-1)},\tilde{\boldsymbol{\omega}}_k^{(l)},\tilde{M}_k^{(l)},\mathbf{r}_k)$
- Otherwise, try the *local update* as follows: for $m=1,\ldots,M_k$,
  - Compute the residual $\mathbf{r}_{k,m}=\mathbf{r}_k-\mathbf{D}_{k,-m}\boldsymbol{\beta}_{k,-m}$ where $\boldsymbol{\beta}_{k,-m}$ (resp. $\mathbf{D}_{k,-m}$) is the vector of amplitudes $\boldsymbol{\beta}_{-k}$ (resp. the matrix of Gabor atoms $\mathbf{D}_{k,-m}$) where the entries related to partial (k,m) have been removed.
  - Sample $\boldsymbol{\omega}_{k,m}^{\star}$ according to $q_{\text{local}}(\boldsymbol{\omega}|\tilde{\boldsymbol{\omega}}_{k,m}^{(l-1)},\{m\})$
  - Perform an accept/reject test with respect to $p(\boldsymbol{\omega}_{k,m},\mathbf{M}=[1],K=1|\mathbf{r}_{k,m})$ given in Eq. (20) so as to obtain $\tilde{\omega}_{k,m}^{(l)}$
  - Resample the amplitudes $\tilde{\boldsymbol{\beta}}_{k,m}^{(l-1)}$ of partial k,m according to $p(\boldsymbol{\beta}_{k,m}|\tilde{\sigma}_v^{2(l-1)},\tilde{\boldsymbol{\omega}}_{k,m}^{(l)},\tilde{M}_k^{(l)},\mathbf{r}_{k,m})$ given in Eq. (22)

In Algorithm 6, two possible frequency updates are mixed: the global move consists of sampling the candidate $\boldsymbol{\omega}_{k,m}^{\star}$ in all the frequency range, whereas in the local move, the candidate $\boldsymbol{\omega}_{k,m}^{\star}$ is kept close to the previous value $\tilde{\boldsymbol{\omega}}_{k,m}^{(l-1)}$, and the partial frequencies are sampled one at a time. The global move probability λ_{note} is generally set to 0.25, and it tunes the rate of global/local moves so as to explore both the full space and the neighborhood. The proposal distributions q_{note} and q_{local} are presented in Sec. III E below.

D. Fast computations

In Algorithms 2–6, as well as in Algorithms 7–10 presented in Appendix C, we extensively use the following trick: each accept/reject test involves the computation of MH ratios of the form $p(\boldsymbol{\omega}^{\star},\mathbf{M}^{\star},K^{\star}|\mathbf{y})/p(\tilde{\boldsymbol{\omega}}^{(l-1)},\tilde{\mathbf{M}}^{(l-1)},\tilde{K}^{(l-1)}|\mathbf{y})$, which requires the costly inversion of two $2R(I+1)$-dimensional matrices of the form $\mathbf{D}^{\mathrm{t}}\mathbf{D}+(1/\xi^2)\mathbf{I}$ [see Eq. (20)]. However, we notice that only a fraction of the components in $\tilde{\boldsymbol{\omega}}^{(l-1)}$ are changed to form $\boldsymbol{\omega}^{\star}$. Similarly, only the component #k in $\tilde{\mathbf{M}}^{(l-1)}$ is modified to form $\mathbf{M}^{\star}$. More generally, assume we know the parameters that define completely some of the partials. In other words, we know $\omega_{k,m},M_k,a_{k,m,i},b_{k,m,i}$ for all $i=0,\ldots,I$ and a limited set of partial indexes denoted $\{k,m\}_a$. Let $\boldsymbol{\beta}_a$, $\boldsymbol{\omega}_a$, and $\mathbf{M}_a$ be the parameters corresponding to $\{k,m\}_a$ with related matrix $\mathbf{D}_a$. Similarly, we denote $\boldsymbol{\beta}_b$, $\boldsymbol{\omega}_b$, and $\mathbf{M}_b$ the remaining parameters with related matrix $\mathbf{D}_b$. By linearity, Eq. (6) can be written $\mathbf{y}=\mathbf{D}_a\boldsymbol{\beta}_a+\mathbf{D}_b\boldsymbol{\beta}_b+\mathbf{v}$. Introducing the residual $\mathbf{r}_a=\mathbf{y}-\mathbf{D}_a\boldsymbol{\beta}_a$, this is equivalent to $\mathbf{r}_a=\mathbf{D}_b\boldsymbol{\beta}_b+\mathbf{v}$ where the dimension of $\mathbf{D}_b$ and $\boldsymbol{\beta}_b$ is smaller than the original dimension of $\mathbf{D}$ and $\boldsymbol{\beta}$. Clearly, the posterior distribution $p(\boldsymbol{\omega}_b,\mathbf{M}_b|\mathbf{y},\boldsymbol{\beta}_a,\boldsymbol{\omega}_a,\mathbf{M}_a)$ is $p(\boldsymbol{\omega}_b,\mathbf{M}_b|\mathbf{r}_a)$ with exactly the same structure as $p(\boldsymbol{\omega},\mathbf{M}|\mathbf{y})$ in Eq. (20), with the major difference that the matrix $\mathbf{D}^{\mathrm{t}}\mathbf{D}$ to be inverted is now $\mathbf{D}_a^{\mathrm{t}}\mathbf{D}_a$, with smaller size. In practice, the indexes $\{k,m\}_a$ correspond to either one note #k (i.e., M_k partials) as in Algorithm 5, or

to a single partial as in Algorithm 6, thus the inversion of $\mathbf{D}_a^t\mathbf{D}_a$ is dramatically quicker than that of $\mathbf{D}^t\mathbf{D}$ (matrix inversion algorithms complexity scales as N_{mat}^3, where N_{mat} is the matrix dimension). More details about this trick can be found in Ref. 40, where several fast methods are also proposed to compute $\mathbf{r}_a^t\mathbf{P}_a\mathbf{r}_a$, $\det(\mathbf{S}_a)$, and $\boldsymbol{\mu}_a$ and to sample from $\mathcal{N}(\boldsymbol{\mu}_a, \sigma_v^2\mathbf{S}_a)$ based on the Choleski decomposition of $\mathbf{S}_a^{-1}$ (where the notation with subscript a is the obvious extension of the above notations).

In practice, the MH ratios $p(\boldsymbol{\omega}^\star, \mathbf{M}^\star, K^\star|\mathbf{y})/p(\tilde{\boldsymbol{\omega}}^{(l-1)}, \tilde{\mathbf{M}}^{(l-1)}, \tilde{K}^{(l-1)}|\mathbf{y})$ are replaced by similar ratios involving only the set of parameters to be changed, as written in Algorithms 2–10.

E. Proposal distributions

In Algorithms 2–6, the new frequencies are sampled randomly according to several proposal distributions. These distributions are crucial: a bad proposal distribution samples candidates distant from likely solutions which are rejected almost always, yielding an inefficient algorithm.

The local proposal distribution q_{local} consists of sampling the new frequency close to the last accepted one. For the model proposed in Eq. (6), q_{note} may have the following structure: the fundamental frequency $\omega_{k,1}$ is sampled according to $q_{\text{local}}(\omega_{k,1}|\tilde{\omega}_{k,m}^{(l-1)})$ chosen as the Gaussian distribution of mean $\tilde{\omega}_{k,1}^{(l-1)}$ and user defined variance $\sigma_{\text{RW},1}^2$. Then, for $m=1,\ldots,M_k$, the δ detuning parameters are updated one at a time and $q_{\text{local}}(\delta|\tilde{\delta}_{k,m}^{(l-1)}, m)$ is a Gaussian distribution of mean $\tilde{\delta}_{k,m}^{(l-1)}$ and user-defined variance $\sigma_{\text{RW},\delta}^2$. In practice, there is a different q_{local} for each partial frequency model. The distribution $q_{\text{partials}}(\omega|\mathbf{r}_k, \omega_{k,1}, \{\text{list of partials}\})$ samples the partial frequencies listed according to $q_{\text{local}}(\delta|\tilde{\delta}_{k,m}^{(l-1)}, m)$.

F. Convergence issues

The Metropolis-Hastings-type algorithm presented above is designed so as to generate a Markov chain with target distribution $p(\boldsymbol{\beta}, \sigma_v^2, \boldsymbol{\omega}, \mathbf{M}, K|\mathbf{y})$. It can be shown that this algorithm is built so as to produce an ergodic Markov chain, which ensures convergence whatever the initial sample. Moreover, it can be shown that it converges uniformly geometrically to the desired probability distribution:[49]

$$\|p^{(l)}(\boldsymbol{\beta}, \sigma_v^2, \boldsymbol{\omega}, \mathbf{M}, K) - p(\boldsymbol{\beta}, \sigma_v^2, \boldsymbol{\omega}, \mathbf{M}, K|\mathbf{y})\|_{TV} \leqslant C\rho^l \quad (27)$$

where $p^{(l)}(\boldsymbol{\beta}, \sigma_v^2, \boldsymbol{\omega}, \mathbf{M}, K)$ is the actual probability distribution of $\tilde{\boldsymbol{\beta}}^{(l)}$, $\tilde{\sigma}_v^{2(l)}$, $\tilde{\boldsymbol{\omega}}^{(l)}$, $\tilde{\mathbf{M}}^{(l)}$, $\tilde{K}^{(l)}$ and $\|\cdot\|_{TV}$ is the total variation norm; C and $\rho<1$ are constants. In practice, it would be necessary to run millions of iterations to have samples distributed according to the posterior over the full space (including label switching); however, simulations show that a few hundred iterations are enough to explore one of the global maxima. Empirical study of the simulation of similar algorithms can be found in Ref. 40.

All the elements have now been described to enable implementation of the proposed method. Results are presented in Sec. IV.

IV. RESULTS

In this section, we report results obtained by processing real music signals from the McGill Database. In order to illustrate the performance of the approach in monophonic and polyphonic contexts, we investigate the four cases $K=1$, $K=2$, $K=3$, and $K=4$. Signals in cases $K\geqslant 2$ are generated by randomly mixing monophonic signals (i.e., signals with $K=1$ note) from the database. In all experiments, we have extracted signals with duration of 544 ms from the original signals. Signals are downsampled from 44100 to 11 025 Hz, thus the processed music extracts have $N=6000$ samples. For the sake of clarity, signals are numbered as $K.xx$ in the following, where K is the number of notes and xx is the extract number. In all the experiments, one Markov chain with 800 iterations is generated using the algorithm presented above, and the model parameters are estimated as explained above. We would like to stress that **only one** simulation is run. We did not run several simulations and keep the best ones, as this would not provide performance results of practical interest. In these estimations, the 100 final samples in the chain are kept in order not to use the 700 first samples where the chain may not have converged and which may not be distributed according to $p(\boldsymbol{\beta}, \sigma_v^2, \boldsymbol{\omega}, \mathbf{M}|\mathbf{y})$. All results concerning pitch estimation are obtained by converting a frequency to a pitch number τ in semitones, relative to A 440 Hz, as follows,

$$\tau = 12\frac{\log(\omega/2\pi) - \log(440)}{\log(2)}. \quad (28)$$

In the main simulations, we used the partials frequencies model in Eq. (7) and the frequencies prior is as given in Eq. (14). In further comparative simulations, different simpler versions of the model are tested for comparative purposes with earlier work. The model parameters are $I=14$, $M_{\min}=2$, $M_{\max}=30$, and $\sigma_\delta^2=3\times10^{-8}$. The hyperparameter ξ^2, Λ_k and Λ' are incorporated into the set of unknown parameters to be estimated, and are sampled as in Refs. 39 and 50. The algorithm proposal parameters used were $\lambda=0.3$, $\lambda_{\text{note}}=0.25$, $\sigma_{\text{RW},1}^2=0.01/N$, and $\sigma_{\text{RW},\delta}^2=10^{-3}$. Moreover, the notes number K is initialized to $K=1$—thus choosing $q_{\text{init}}(K)=\delta_1(K)$—so as to let the number of notes increase until it reaches the actual number of notes.

On a dual processor Pentium IV (2.6 GHz) computer, and Matlab code, the average computation time is 1.35 s per iteration per note, rather a high load, but note that the segments of music being analyzed are fairly substantial in length (roughly 0.5 s computed in 1 h in total).

In the next subsection, we consider the monophonic case ($K=1$). In Sec. IV B, we address the polyphonic cases ($K=2$ to $K=4$). In both Secs. IV B and IV A, the number of notes is assumed known and is used in the algorithm by setting $K_{\max}$ to the true number of notes. This issue is further discussed in Sec. IV E.

A. Monophonic case: $K=1$

We first present simulations aimed at assessing the Markov chain convergence, then we focus on three typical simulations. Finally, we propose some statistics obtained by processing 20 different music signals.

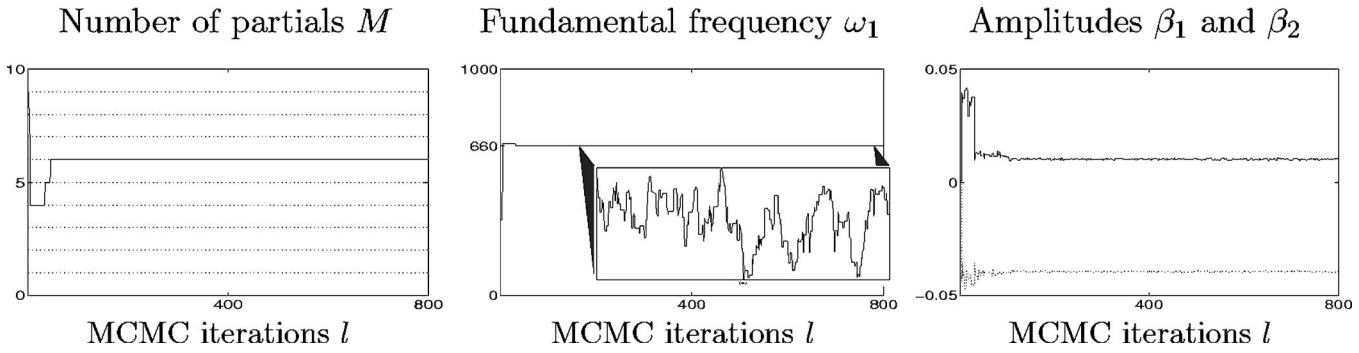

FIG. 4. Monophonic case, test signal 1.1. Convergence of the Markov chain generated by the algorithm in the monophonic case ($K=1$). Some components of the sampled random vector $\tilde{\boldsymbol{\beta}}^{(l)}, \tilde{\sigma}_v^{2(l)}, \tilde{\boldsymbol{\omega}}^{(l)}, \tilde{\mathbf{M}}^{(l)}$ are represented for $l=1,\ldots,1000$. As can be seen, convergence is reached within the 100 first iterations. Samples $l=701$ to $l=800$ are kept for estimation purposes. As illustrated in the central plot, though convergence is reached, the chain concerning ω_1 keeps fluctuating. The true fundamental frequency is 660 Hz ($\tau=7$).

1. Convergence of the sampling algorithm

In order to illustrate the convergence of the above MCMC algorithm, we have plotted the evolution of the random samples generated at each iteration $l=1,\ldots,800$ (see Fig. 4). As can be seen, the chain converges to a steady state after only 100 iterations, and the very final samples can be assumed to be distributed according to $p(\boldsymbol{\beta}, \sigma_v^2, \boldsymbol{\omega}, \mathbf{M}, K=1|\mathbf{y})$. In practice, we keep the final 100 samples (i.e., iterations $l=701$ to $l=800$) to estimate the parameter values (see Sec. III B). In Fig. 4, the magnified portion of $\tilde{\omega}_1^{(l)}$ shows that the parameter keeps fluctuating after convergence is reached: this is a normal behavior which is explained as follows. The Markov chain has converged *in terms of probability density function*; this means that the samples generated are random (thus, the chain fluctuates), but their distribution is fixed. All 20 tests we ran showed very similar behavior to this simulation. In other words, the Markov Chain plotted in Fig. 4 in representative of the 20 Markov chains generated for the 20 tests, whatever the instrument playing, and whatever the played note pitch.

2. Estimation results

The estimation performance is now analyzed. We focus on three test signals, namely test signals #1.1, #1.4, and #1.8. For each test signal, the reconstructed signal $\hat{\mathbf{y}}$ is computed using the model in Eq. (3) with $K=1$ and the estimated parameters $\hat{\boldsymbol{\beta}}$, $\hat{\boldsymbol{\omega}}$, $\hat{\mathbf{M}}$ computed using the last 100 samples of the Markov chain. In Fig. 5, we have plotted the original signals $\mathbf{y}$ as well as $\hat{\mathbf{y}}$ and the residuals $\mathbf{y}-\hat{\mathbf{y}}$, in the time domain. The difference between $\mathbf{y}$ and $\hat{\mathbf{y}}$ is quite small in all three cases (i.e., the residual energy is small compared to the energy of $\mathbf{y}$). Figure 6 displays the spectra of $\mathbf{y}$, $\hat{\mathbf{y}}$ and residuals $\mathbf{y}-\hat{\mathbf{y}}$. The model in Eq. (3) has been able to capture all harmonic components except for some high-frequency partials. The amplitudes of these partials are, however, about 20 dB smaller than partials with smaller frequency. Moreover, the flute extract #1.4 displayed in Figs. 5 and 6 shows that our Gabor-based approach models accurately time-varying amplitudes with significant variations.

Table I reports pitch estimation results for the 20 test signals, in terms of which semitone in the scale is closest to the estimated pitch number. As can be seen, there is no pitch estimation error. Estimated pitch values may differ slightly from the exact integer values, because they are computed relative to A 440 Hz. Due to recording artifacts, instrument tuning, etc., the true A frequency may differ from 440 Hz.

B. Polyphonic cases: $K=2$, $K=3$, and $K=4$

In this subsection, we present results obtained in polyphonic contexts. As for monophonic tests, generated Markov chains converge quite quickly to the parameter posterior distribution. (For the sake of brevity, we do not present convergence examples again here.) Table II reports pitch estimation results for the 20 test signals. Similar to the monophonic case, most notes are accurately estimated. There are, however, some octave error cases. Octave errors happen when the estimated pitch $\hat{\tau}$ is one or more octaves (12 note numbers) away from the true pitch. Example errors can be found with extracts numbers #2.10, #2.12–#2.15, and #2.18. For extract number #2.16, the two notes have the same pitch ($\tau=-12$), so it is very difficult for the algorithm to separate them into two superimposed notes. The algorithm actually assigned the correct pitch $\hat{\tau}=-12$ to one note, and set $\hat{\tau}=7$ to the other note. The partials of the note with pitch $\hat{\tau}=7$ are superimposed with partials of the note with $\hat{\tau}=-12$. Beside these problems, no errors occurred.

Figure 7 displays estimation results in the frequency domain for the extracts #2.1 (both notes are correctly estimated), #2.14 (one note is correctly estimated, and the other shows an octave error), and #2.16 (the two notes have the same frequency). The time domain plots are very similar to those presented in Fig. 5 (monophonic case), thus they are not presented here. In these more difficult contexts, the majority of notes are still accurately estimated. There are more octave errors than with $K=2$, but octave errors do not increase much when related to the actual number of notes to be estimated (three or four notes, instead of two). In Fig. 8, we have plotted the spectra corresponding to extracts #3.1 and #4.2. Again, the model has captured most harmonic information, apart from some high-frequency partials. Aside from octave ambiguities, some new error cases occur: they are examined in details in Sec. IV C.

Cases with $K=3$ and $K=4$ notes exhibit the same kind of behavior. In summary, pitch is correctly estimated for 84.6%

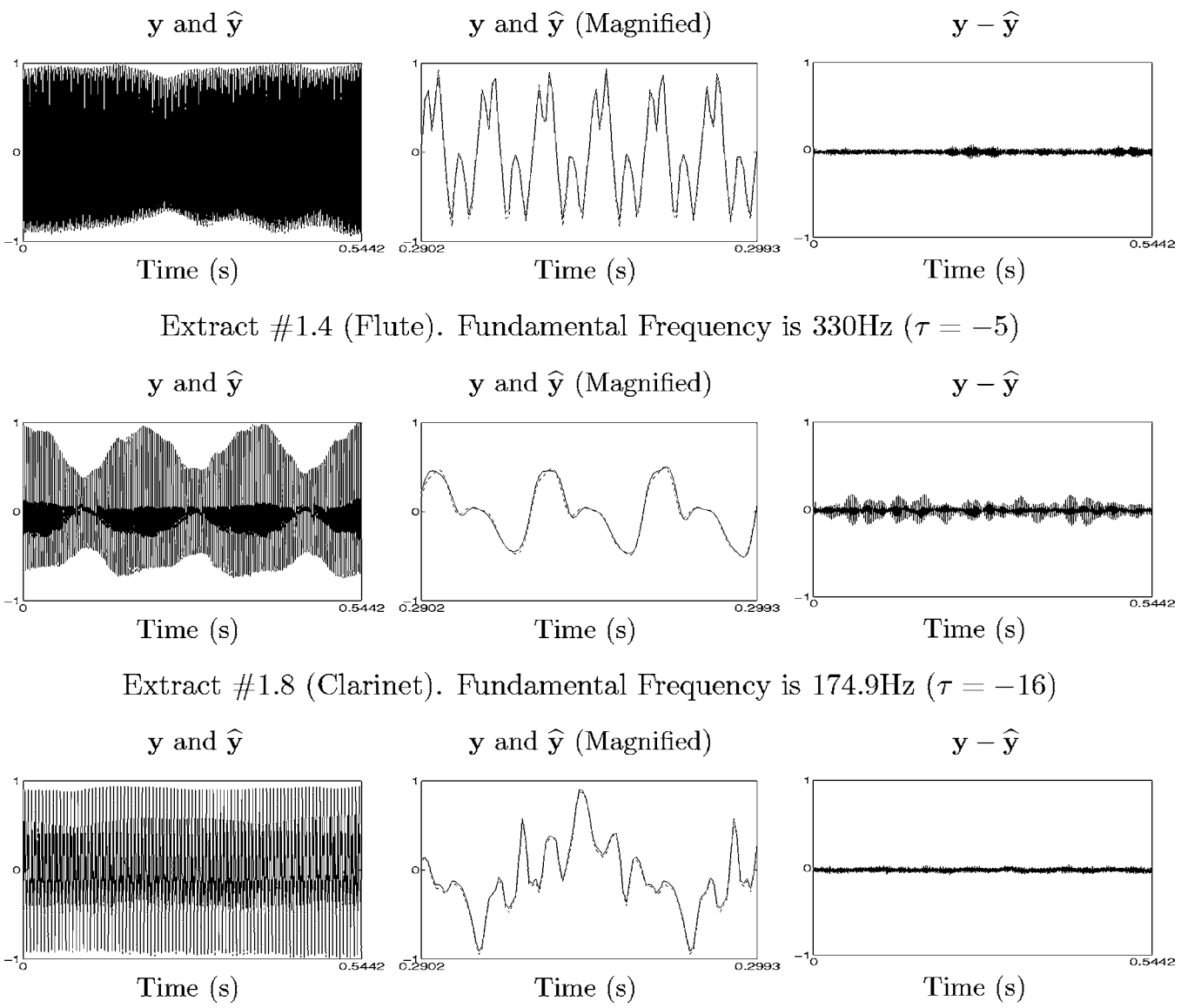

FIG. 5. Monophonic case, estimation results for three test signals. Time domain plots of original signals $\mathbf{y}$ (dashed lines), reconstructed signals $\hat{\mathbf{y}}$ (solid lines), and the residuals $\mathbf{y}-\hat{\mathbf{y}}$.

of notes when $K=2$ (94.9% if we omit octave errors), for 75% of notes when $K=3$ (92.8% if we omit octave errors), and for 71.0% of notes when $K=4$ (80.3% if we omit octave errors).

C. Study of some error cases

As outlined above, some error cases occur when the estimated pitch of a note is not related by octave errors to the true pitch. Among these error cases, we will focus on extracts #3.15 and #4.8. Figure 9 displays the spectra of these extracts, together with the spectra of the estimated notes. As can be seen, the error notes are not located at random frequencies: for extract #3.15, the error note is located at pitch $\hat{\tau}\approx -4$, which is one octave below an already existing note ($\tau=8$). The note at $\tau=-13$ has simply been missed, certainly because its amplitude is lower than that of other existing notes. In order to test the stability of this solution, we have run again the MCMC algorithm three times. The pitch estimation results are $(-4.19,-6.99,8.02)$, $(-3.97,-7.01,1.19)$, $(-13,02,-6.98,8.01)$, and $(-3.99,-7.00,-19.75)$, whereas the true pitches are $(-13,-7,8)$. The corresponding residuals energy, expressed as a fraction of the original signal energy, are 2.85%, 4.00%, 0.09%, and 2.43%. These four solutions are all stable local solutions, and the overall one (found at MCMC run #3) is not reached at each time. It seems in these cases that the algorithm may be trapped in local probability maxima, thus not exploring all of the high-probability regions. Longer MCMC chains could be run to overcome this, but convergence might still not be achieved in a reasonable computation time. Multiple chain approaches are perhaps more promising here, either using informal ideas such as picking the chain with highest posterior probability or lowest

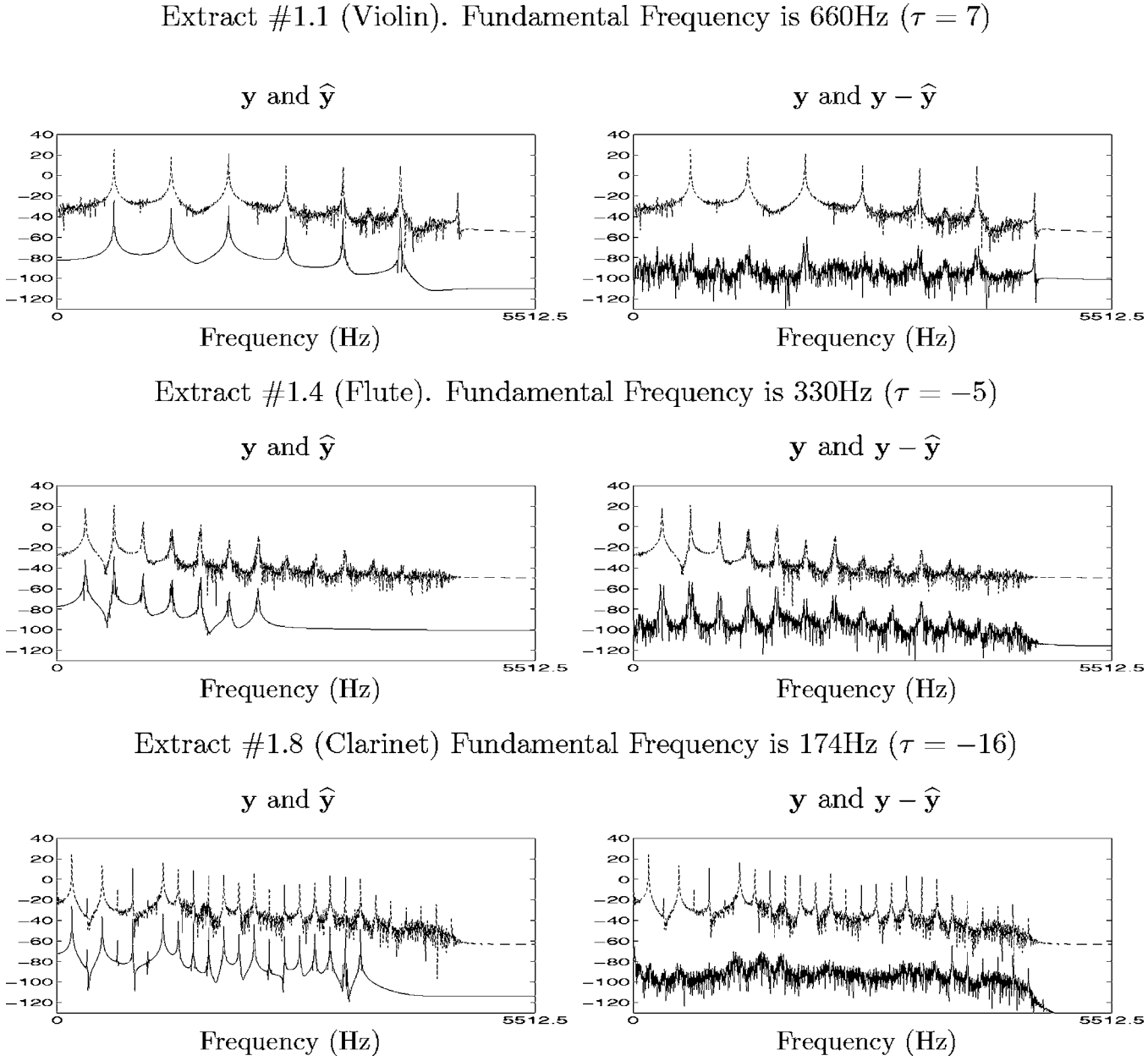

FIG. 6. Monophonic case, estimation results for three test signals. Frequency domain spectra (in dB) of original signals $\mathbf{y}$ (dashed lines), reconstructed signals $\hat{\mathbf{y}}$ (solid lines), and the residuals $\mathbf{y}-\hat{\mathbf{y}}$ (solid lines). The spectra of $\hat{\mathbf{y}}$ and of $\mathbf{y}-\hat{\mathbf{y}}$ are plotted with a −50-dB offset for improved visibility.

TABLE I. Estimation situations and results for 20 monophonic tests.

Extract	Instrument	Pitch τ	Estimated pitch $\hat{\tau}$	Extract	Instrument	Pitch τ	Estimated pitch $\hat{\tau}$
1.1	Violin	7	7.01	1.11	Trumpet	10	9.98
1.2	Fr. Horn	−27	−26.89	1.12	Oboe	13	12.99
1.3	Oboe	−7	−6.98	1.13	Flute	−7	−8.68
1.4	Flute	−5	−4.94	1.14	Flute	−4	−3.89
1.5	Trumpet	−5	−5.03	1.15	Fr. Horn	−27	−26.96
1.6	Trumpet	−9	−8.98	1.16	Violin	−4	−4.09
1.7	Fr. Horn	−17	−17.01	1.17	Oboe	13	12.99
1.8	Clarinet	−16	−15.97	1.18	Oboe	−7	−6.98
1.9	Violin	−9	−9.01	1.19	Flute	−2	−1.91
1.10	Viola	3	2.99	1.20	Violin	−3	−2.94

TABLE II. Estimation situations and results for 20 polyphonic tests, with $K=2$.

Extract	Pitch τ		Estimated pitch $\hat{\tau}$		Extract	Pitch τ		Estimated pitch $\hat{\tau}$	
2.1	1	5	1.00	5.00	2.11	6	−7	5.99	−7.03
2.2	−1	7	−1.06	7.06	2.12	−7	−1	−7.03	−13.05
2.3	−9	−2	−1.96	−2.06	2.13	7	−3	7.07	19.07
2.4	1	11	1.048	11.03	2.14	12	−13	12.00	−25.00
2.5	−7	13	−6.87	13.00	2.15	10	−2	−1.94	−1.90
2.6	−11	10	−11.03	10.06	2.16	−12	−12	−11.99	7.02
2.7	−15	−19	−14.98	−18.96	2.17	−13	−4	−13.26	−3.79
2.8	−13	−4	−12.95	−4.17	2.18	2	8	20.02	8.02
2.9	10	4	10.07	4.07	2.19	−3	3	−2.87	3.03
2.10	−10	5	−10.15	−6.8	2.20	−1	14	−0.91	14.22

residual error, or more formally using multiple-chain MCMC methods and annealing (see Refs. 51 and 52). These promising possibilities will be investigated in our next paper.

Concerning extract #4.8, the error can be explained as follows: though this is not an octave error, we see that the third partial (i.e., the partial such that $m=3$) of the note with $\tau=-8$ is located at $\tau=11$ (this is the pitch of the note that has been missed). Its fourth partial ($m=4$) is located at $\tau=16$, which also corresponds to the third partial ($m=3$) of the note located at $\tau=-3$. This explains the stability of this solution: positioning a note at $\tau=-8$ captures some of the energy of the two notes located at $\tau=\{-3,11\}$. Similar error cases can be found for extracts #4.8, #4.10, #4.11, etc.: they can be easily understood by noticing that the third partial of any note with pitch τ is located at pitch $\tau+19$. No doubt these occasional problems are a combination of misconverged

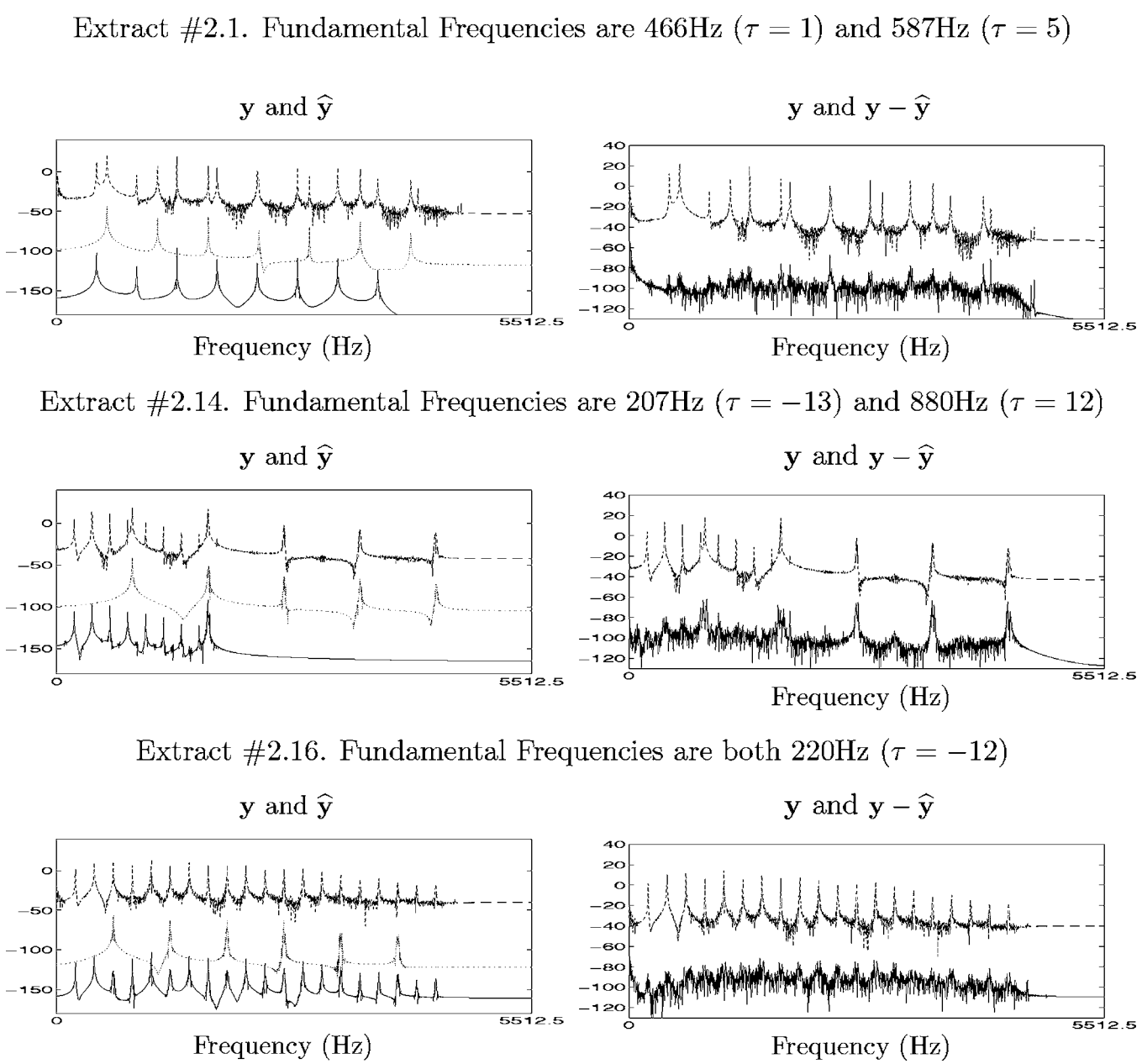

FIG. 7. Polyphonic case ($K=2$), estimation results for three test signals. Frequency domain spectra (in dB) of original signals $\mathbf{y}$ (dashed lines), reconstructed signals $\hat{\mathbf{y}}$ (note #1 in solid lines with −50-dB offset, note #2 in dotted lines with −100-dB offset) and the residuals $\mathbf{y}-\hat{\mathbf{y}}$ (solid lines with −50-dB offset).

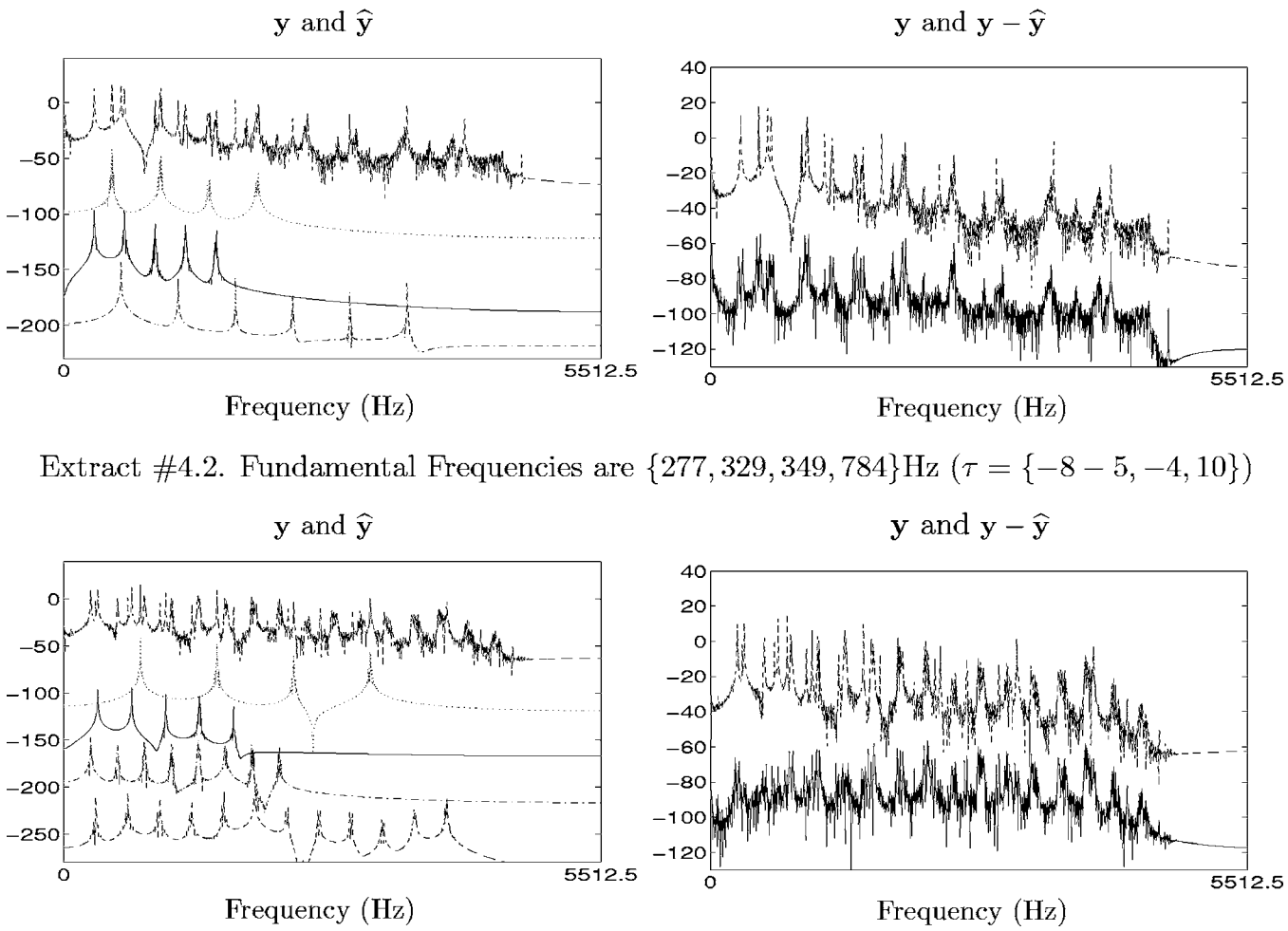

FIG. 8. Polyphonic case ($K=3$ and $K=4$), estimation results for two test signals. Frequency domain spectra (in dB) of original signals $\mathbf{y}$ (dashed lines), reconstructed signals $\hat{\mathbf{y}}$ (note #1 in solid lines with −50-dB offset, note #2 in dotted lines with −100-dB offset, note #3 in dash-dotted lines with −150-dB offset, and note #4 in dashed lines with −200-dB offset) and the residuals $\mathbf{y}-\hat{\mathbf{y}}$ (solid lines with −50-dB offset).

MCMC runs and/or ambiguous data for our models. We have found no straightforward solutions to this problem, although one possibility is to look at the estimated partial amplitudes: for the error note located at pitch $\tau=-8$, partials $m=1$, $m=2$, and $m=5$ have low amplitude compared to partials $m=3$, $m=4$, $m=6$, and $m=7$. In notes produced by natural instruments, such an amplitude profile would be quite unlikely and so priors based on more regular amplitude laws could be incorporated, thus penalizing such erroneous solutions.

D. Performance from the auditory viewpoint

Aside from pitch estimation, a very promising feature of this Bayesian method is the possibility to separate several notes from a mixture, even if only one microphone is used to record the extract (monophonic extracts as opposed to stereophonic extracts). A very promising result is that there is almost no audible difference between $\mathbf{y}$ and $\hat{\mathbf{y}}$ in most cases. Moreover, in the case of octave errors, the corresponding reconstructed note typically sounds very close to the original note. Thus, from the auditory viewpoint, the performance of the method is higher than from the pitch estimation viewpoint. We note that Virtanen and Kalpuri[53] have used their frequency domain models to perform separation of notes in a chord.

E. Estimation of *K*

In the results presented above, the number of notes is assumed known. In principle, though, the model and the algorithm are designed to estimate K. In practice, however, letting K as a free parameter systematically leads to too many notes with, typically, two notes located at the same position. This may be overcome by imposing small K via the prior over Λ', but this approach fails in examples with many true notes.

The birth/death moves are nevertheless quite important insofar as K is initialized to $K=1$ and slowly increases until reaching $K_{\max}$ (the true K assume known). A good solution to actually estimate K is to run MCMC simulations with increasing $K_{\max}$ from one to, e.g., ten and monitoring the minimum residual energy over several MCMC runs for each $K_{\max}$. When $K_{\max}$ reaches the true number K, the residual

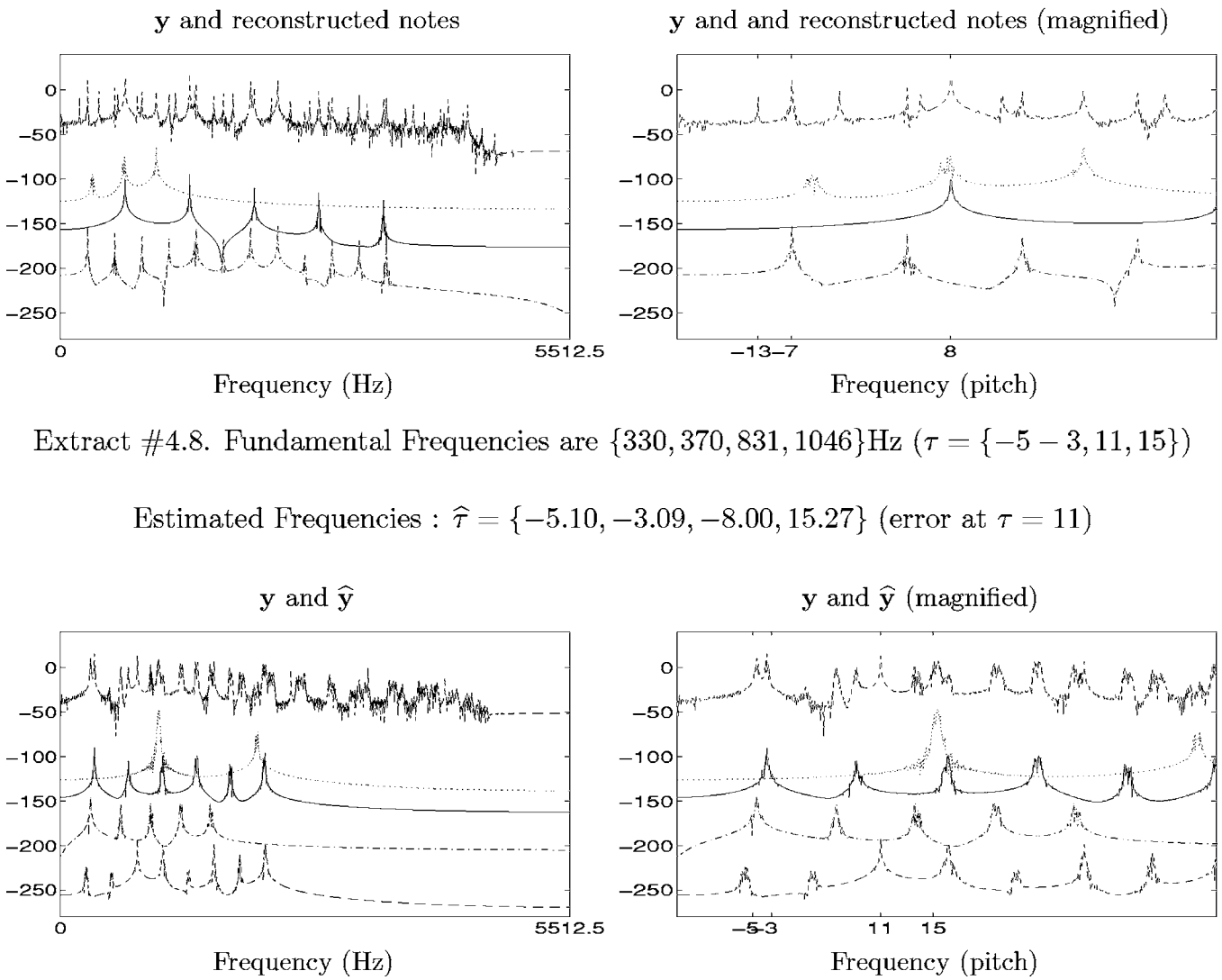

FIG. 9. Polyphonic case ($K=3$ and $K=4$), estimation results for two test signals. Frequency domain spectra (in dB) of original signals $\mathbf{y}$ (dashed lines), and reconstructed notes (note #1 in solid lines with −50-dB offset, note #2 in dotted lines with −100-dB offset, note #3 in dash-dotted lines with −150-dB offset, and note #4 in dashed lines with −200-dB offset).

energy suddenly stops decreasing, yielding the actual K. It should be noted that estimating K is a complicated problem, even for trained human listeners.

F. Comparison with previous algorithms

The model presented here is an improved version of previous Bayesian models using MCMC.[27–30] In order to illustrate the improvement in practice, we have processed the 20 three-note extracts with a model with constant amplitudes (that is, our model with a single rectangular window ϕ_0) and no inharmonicity (that is, with $\omega_{k,m}=m\omega_{k,1}$). The remaining algorithm parameters are kept the same. Then we are essentially back to the model by Walmsley *et al.*[27,28] The pitch is correctly estimated for 62.5% of notes (here $K=3$) and 78.6% if we omit octave errors. This is substantially lower than with the new model (75% and 92.8%). Moreover, the notes are poorly reconstructed with this model: the average residual energy calculated over the 20 extracts is 27.5% of the original signal energy. This is much higher than with the full model (only 5%). This is mainly due to two reasons: (1) the amplitudes are nonconstant with real instruments over time and (2) without the inharmonicity parameters, the model cannot accurately fit partials with order m larger than about 4, because their inharmonicity can be too great [the $\delta_{k,m}$'s in model (7) were found to be nonzero for most of the sounds processed]. A similar improvement in performance with the new model is obtained when comparing with a previous inharmonicity model having additive $\delta_{k,m}$ parameters rather than multiplicative, as we initially proposed in our

previous works:[29,30] for large m, the additive $\delta_{k,m}$ parameter should be large, which is not allowed by the prior distribution of $\delta_{k,m}$.

V. CONCLUSIONS AND FUTURE WORK

In this paper, we have presented a novel technique for the estimation and analysis of western tonal (pitched) music. By using a Gabor atomic model together with a specially designed MCMC algorithm, we have shown that it is possible to estimate a high number of parameters from many simultaneously playing notes. Our model includes time-varying amplitudes, inharmonicity (detuned partials), and unknown number of partials. We obtain good pitch estimation results on a database of polyphonic real music extracts containing randomly generated chords with up to four notes. These results are found to be a significant improvement over previous versions of related models. More notes than four are possible with these models, but clearly the performance degrades as the number increases. Nevertheless, in informal trials the methods have been successfully able to identify and separate out the notes in chords of up to eight pitches. In addition to pitch estimation, our technique enables note separation from chords. The separated notes sound perceptually quite similar to the original, unmixed notes, thus giving some confidence in the method's ability to model the individual note components accurately, even when many partials are overlapping between different notes.

We have identified some error cases, and discussed their causes, suggesting improvements to help eliminate them, using multiple chain approaches or more informative amplitude priors, for example. In addition, considering the chords to be in the context of real musical sequences, we can usefully model musical dependencies from frame to frame in order to aid the estimation process. For example, fundamental frequency priors that depend on the previous notes played would build in a useful extra layer of prior information for the models.

APPENDIX A: PIANO MODEL FREQUENCIES DISTRIBUTION

Fletcher and Rossing[2] propose the following piano model, where partial frequencies follow the law:

$$\omega_{k,m} = m\omega_{k,1}\sqrt{\frac{1+m^2B}{1+B}}. \tag{A1}$$

In our harmonic model, one can simply implement this formula with unknown B (a typical value is $B \approx 4\times 10^{-4}$), which yields a piano music model. The frequency prior distribution can be as follows:

$$p(\boldsymbol{\omega}|K) = p(\boldsymbol{\omega}_1|K)p(B_1)\cdots p(B_K), \tag{A2}$$

where $\boldsymbol{\omega}_1=[\omega_{1,1},\ldots,\omega_{K,1}]$ is the vector of fundamental frequencies and B_k $(k=1,\ldots,K)$ are the parameters B in Eq. (A1) for each note. The piano fundamental frequency prior may encode the strong knowledge we have about possible notes, which is determined by strings tuning (equally tempered scale). For simplicity, we assume note fundamental frequencies are independent, which means

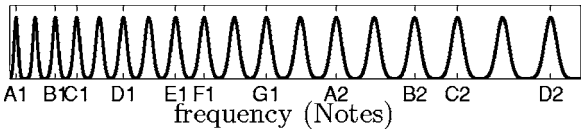

FIG. 10. Prior distribution of the fundamental frequencies $p(\omega_{k,1})$ in the piano model. The spread of the Gaussian function φ_j increases with frequency, i.e., with the strings note labels $j=\{\text{A1},\text{A1\#},\text{B1},\ldots\}$. This is aimed at limiting areas where the prior is zero.

$$p(\boldsymbol{\omega}_1|K) = \prod_{k=1}^{K} p_{\text{plano}}(\boldsymbol{\omega}_{k,1}), \tag{A3}$$

and for each note $k=1,\ldots,K$,

$$p_{\text{piano}}(\omega_{k,1}) \propto \sum_{j=1}^{N_{\text{strings}}} \varphi_j(\omega_{k,1}), \tag{A4}$$

where φ_j is for example a Gaussian function centered on the frequency of string $\#j$, and N_{strings} is the number of piano strings. The shape of the resulting prior is plotted in Fig. 10.

APPENDIX B: PROBABILITIES OF PERFORMING BIRTH/DEATH MOVES

The probabilities $\mu(K)$ and $\nu(K)$ governing the birth and death moves are such that

$$\mu(K) = c\min\left\{1, \frac{p(K+1|\Lambda')}{p(K|\Lambda')}\right\} \quad \text{for } K < K_{\max} \quad \text{with } \nu(1)=0, \tag{B1}$$

$$\nu(K+1) = c\min\left\{1, \frac{p(K|\Lambda')}{p(K+1|\Lambda')}\right\} \quad \text{for } K > 1 \quad \text{with } \mu(K_{\max})=0, \tag{B2}$$

where $c=0.15$ and $p(K|\Lambda')$ is the prior probability of K.

APPENDIX C: *n*-INCREASE/DECREASE AND MULTIPLY/DIVIDE MOVES

In Algorithm 5, the number of partials can be changed in two ways: either n partials are added (resp. removed) in the n-increase move (resp. n-decrease move) [see Algorithm 7 (resp. see Algorithm 8)] or the number of partials is multiplied by 2 (resp. divided by 2) in the divide move (resp. multiply move) [see Algorithm 10 (resp. see Algorithm 9)].

Algorithm 7: n-Increase move

- Compute the residual $\mathbf{r}_0=\mathbf{y}-\tilde{\mathbf{D}}^{(l-1)}\tilde{\boldsymbol{\beta}}^{(l-1)}$
- Set $M_k^{\star} \leftarrow \tilde{M}_k^{(l-1)}+n$,
- Sample the frequencies $[\omega_{k,M_k+1}^{\star},\ldots,\omega_{k,M_k+n}^{\star}]$ from the proposal distribution $q_{\text{partials}}(\boldsymbol{\omega}|\mathbf{r}_0,\omega_{k,1},\{M_k+1,\ldots,M_k+n\})$
- Set $\boldsymbol{\omega}_k^{\star} \leftarrow [\boldsymbol{\omega}_k^{(l-1)},\omega_{k,M_k+1}^{\star},\ldots,\omega_{k,M_k+n}^{\star}]$.
- Let $\mathbf{P}^{\star}$ and $\mathbf{S}^{\star}$ be the $\mathbf{P}$ and $\mathbf{S}$ matrices related to candidate set of partials from $m=\tilde{M}_k^{(l-1)}+1$ to $m=M_k^{\star}$ with frequencies $[\omega_{k,M_k+1}^{\star},\ldots,\omega_{k,M_k+n}^{\star}]$. Compute

$$g_{\text{inc}} = \left(\frac{\gamma_0 + \|\mathbf{r}_0\|^2}{\gamma_0 + \mathbf{r}_0^{\text{t}} \mathbf{P}^{\bigstar} \mathbf{r}_0} \right)^{(N+\nu_0)/2} \times \frac{p(\boldsymbol{\omega}_k^{\bigstar}, M_k^{\bigstar})}{p(\boldsymbol{\omega}_k^{(l-1)}, \tilde{M}_k^{(l-1)})} \frac{d_{M_k+n}}{b_{M_k}} \det(\mathbf{S}^{\bigstar})^{1/2} \times [q_{\text{partials}}(\omega_{k,M_k+1}^{\bigstar}, \ldots, \omega_{k,M_k+n}^{\bigstar} | \mathbf{r}_0, \omega_{k,1}, \{M_k + 1, \ldots, M_k + n\})_b ig]^{-1} \quad \text{(C1)}$$

- With probability $\min(1, g_{\text{inc}})$, accept the candidate: Set $\tilde{M}_k^{(l)} \leftarrow M_k^{\bigstar}$ and $\tilde{\boldsymbol{\omega}}_k^{(l)} \leftarrow \boldsymbol{\omega}_k^{\bigstar}$ and sample $\tilde{\boldsymbol{\beta}}_{M_k+1,\ldots,M_k+n}^{(l)}$ according to $p(\boldsymbol{\beta} | \tilde{\sigma}_v^{2(l-1)}, \boldsymbol{\omega}_{k,M+n}^{\bigstar}, M = n, K = 1, \mathbf{r}_0)$ given in Eq. (22).

To ensure the theoretical convergence of the algorithm, it is necessary to implement the reverse move, the n-decrease move, presented in Algorithm 8 below.

Algorithm 8: n-Decrease move

- Set $M_k^{\bigstar} \leftarrow \tilde{M}_k^{(l-1)} - n$,
- Remove the frequencies $[\tilde{\omega}_{k,M_k-n+1}^{(l-1)}, \ldots, \tilde{\omega}_{k,M_k}^{(l-1)}]$ in $\tilde{\boldsymbol{\omega}}_k^{(l-1)}$ so as to form $\boldsymbol{\omega}_k^{\bigstar}$
- Compute the residual $\mathbf{r}_{k,M-n} = \mathbf{r}_k - \mathbf{D}_{k,-\{M+1,\ldots,M+n\}} \boldsymbol{\beta}_{k,-\{M+1,\ldots,M+n\}}$ where $\boldsymbol{\beta}_{k,-\{M+1,\ldots,M+n\}}$ (resp. $\mathbf{D}_{k,-\{M+1,\ldots,M+n\}}$) is the vector of amplitudes $\tilde{\boldsymbol{\beta}}^{(l-1)}$ (resp. the matrix of Gabor atoms $\tilde{\mathbf{D}}^{(l-1)}$) computed with parameters $\tilde{\mathbf{M}}^{(l)}$ and $\tilde{\boldsymbol{\omega}}^{(l)}$ where the entries related to partials $\{M_k - n + 1, \ldots, M_k\}$ have been removed.
- Let $\mathbf{P}_n$ and $\mathbf{S}_n$ be the $\mathbf{P}$ and $\mathbf{S}$ matrices related to the partials $M_k - n + 1, \ldots, M_k$ to be removed. Compute

$$g_{\text{dec}} = \left(\frac{\gamma_0 + \mathbf{r}_{k,M-n}^{\text{t}} \mathbf{P}_n \mathbf{r}_{k,M-n}}{\gamma_0 + \|\mathbf{r}_{k,M-n}\|^2} \right)^{(N+\nu_0)/2} \frac{p(\boldsymbol{\omega}_k^{\bigstar}, M_k^{\bigstar})}{p(\boldsymbol{\omega}_k^{(l-1)}, \tilde{M}_k^{(l-1)})} \times \frac{b_{M_k-n}}{d_{M_k}} \frac{q_{\text{partials}}(\tilde{\omega}_{k,M_k-n+1}^{(l-1)}, \ldots, \tilde{\omega}_{k,M_k}^{(l-1)} | \mathbf{r}_{k,M-n}, \omega_{k,1}, \{M_k - n + 1, \ldots, M_k\})}{\det(\mathbf{S}_n)^{1/2}} \quad \text{(C2)}$$

- With probability $\min(1, g_{\text{dec}})$, accept the candidate: Set $\tilde{M}_k^{(l)} \leftarrow M_k^{\bigstar}$, $\tilde{\boldsymbol{\omega}}_k^{(l)} \leftarrow \boldsymbol{\omega}_k^{\bigstar}$ and remove the components corresponding to partials $M_k - n + 1, \ldots, M_k$ from $\tilde{\boldsymbol{\beta}}^{(l-1)}$ so as to obtain $\tilde{\boldsymbol{\beta}}^{(l)}$.

The multiply and divide moves are synchronized with fundamental frequency division/multiplication by 2 in order to detect octave ambiguities: assume a note with frequency 440 Hz is played by an instrument. The notes with frequencies 220 and 880 Hz have partials that are almost superimposed with the partials of the actual note. Without these moves, the Markov chain may be stuck on incorrect fundamental frequencies, namely 220 or 880 Hz, which is not satisfactory.

Algorithm 9: Divide move

- Compute the residual $\mathbf{r}_0 = \mathbf{y} - \tilde{\mathbf{D}}^{(l-1)} \tilde{\boldsymbol{\beta}}^{(l-1)}$
- Set $M_k^{\bigstar} \leftarrow 2\tilde{M}_k^{(l-1)} + e$
- Form $\boldsymbol{\omega}_k^{\bigstar}$ by setting $\omega_{k,2m}^{\bigstar} \leftarrow \tilde{\omega}_{k,m}^{(l-1)}$ for $m = 1, \ldots, \lfloor M_k/2 \rfloor$.
- Sample frequencies $\tilde{\omega}_{k,2m-1}^{\bigstar}$ for $m = 1, \ldots, \lfloor M_k/2 \rfloor$ using the proposal distribution $q_{\text{partials}}(\omega | \mathbf{r}_0, \omega_{k,1}, 2m - 1)$
- Set $\boldsymbol{\omega}_k^{\bigstar} \leftarrow \boldsymbol{\omega}_k^{(l-1)} \cup \{\omega_{k,2m}^{\bigstar} \leftarrow \tilde{\omega}_{k,m}^{(l-1)}\}_{m=1,\ldots,\lfloor M_k/2 \rfloor}$.
- Let $\mathbf{P}^{\bigstar}$ and $\mathbf{S}^{\bigstar}$ be the $\mathbf{P}$ and $\mathbf{S}$ matrices related to candidate set of partials $(k, 2m-1)$ with frequencies $\tilde{\omega}_{k,2m-1}^{\bigstar}$, $m = 1, \ldots, \lfloor M_k/2 \rfloor$. Compute

$$g_{\text{div}} = \left[\frac{\gamma_0 + \|\mathbf{r}_0\|^2}{\gamma_0 + \mathbf{r}_0^{\text{t}} \mathbf{P}^{\bigstar} \mathbf{r}_0} \right]^{(N+\nu_0)/2} \det(\mathbf{S}^{\bigstar})^{1/2} \frac{p(\boldsymbol{\omega}_k^{\bigstar}, M_k^{\bigstar})}{p(\boldsymbol{\omega}_k^{(l-1)}, \tilde{M}_k^{(l-1)})} \times \prod_{m=1,\ldots,\lfloor M_k/2 \rfloor} [q_{\text{partials}}(\omega | \mathbf{r}_0, \omega_{k,1}, 2n - 1)]^{-1} \quad \text{(C3)}$$

- With probability $\min(1, g_{\text{div}})$, accept the candidate: Set $\tilde{M}_k^{(l)} \leftarrow M_k^{\bigstar}$ and $\tilde{\boldsymbol{\omega}}_k^{(l)} \leftarrow \boldsymbol{\omega}_k^{\bigstar}$ and sample $\tilde{\beta}_m$, $m = 1, \ldots, \lfloor M_k/2 \rfloor$ according to $p(\boldsymbol{\beta} | \tilde{\sigma}_v^{2(l-1)}, \tilde{\omega}_{k,2m-1}^{\bigstar}, M = 1, \mathbf{r}_0)$ given in Eq. (22).

As for any reversible jump, the divide move comes together with the reverse, the *multiply move* presented in Algorithm 10 below.

Algorithm 10: Multiply move

- Set $M_k^{\bigstar} \leftarrow (\tilde{M}_k^{(l-1)} - e)/2$,
- Form $\boldsymbol{\omega}_k^{\bigstar}$ by setting $\omega_{k,m}^{\bigstar} \leftarrow \tilde{\omega}_{k,2m}^{(l-1)}$ for $m = 1, \ldots, M_k - e$.
- Compute the residual $\mathbf{r}_{k,\text{odd}} = \mathbf{r}_k - \mathbf{D}_{k,-\{1,3,\ldots\}} \boldsymbol{\beta}_{k,-\{1,3,\ldots\}}$ where $\boldsymbol{\beta}_{k,-\{1,3,\ldots\}}$ (resp. $\mathbf{D}_{k,-\{1,3,\ldots\}}$) is the vector of amplitudes $\tilde{\boldsymbol{\beta}}^{(l-1)}$ (resp. the matrix of Gabor atoms $\tilde{\mathbf{D}}^{(l-1)}$) computed with parameters $\tilde{\mathbf{M}}^{(l)}$ and $\tilde{\boldsymbol{\omega}}^{(l)}$ where the entires related to partials {1, 3,…} have been removed.
- Let $\mathbf{P}_{\text{odd}}$ and $\mathbf{S}_{\text{odd}}$ be the $\mathbf{P}$ and $\mathbf{S}$ matrices related to the partials 1, 3,… to be removed. Compute

$$g_{\text{mul}} = \left[\frac{\gamma_0 + \mathbf{r}_{k,\text{odd}}^{\text{t}} \mathbf{P}_{\text{odd}} \mathbf{r}_{k,\text{odd}}}{\gamma_0 + \|\mathbf{r}_{k,\text{odd}}\|^2} \right]^{(N+\nu_0)/2} \times \det(\mathbf{S}_{\text{odd}})^{-1/2} \frac{p(\boldsymbol{\omega}_k^{\star}, M_k^{\star})}{p(\boldsymbol{\omega}_k^{(l-1)}, \widetilde{M}_k^{(l-1)})} \times \prod_{m=1,3,\ldots,} q_{\text{partials}}(\omega | \mathbf{r}_{k,\text{odd}}, \omega_{k,1}, 2m-1) \tag{C4}$$

- With probability $\min(1, g_{\text{mul}})$, accept the candidate: Set $\widetilde{M}_k^{(l)} \leftarrow M_k^{\star}$, $\widetilde{\boldsymbol{\omega}}_k^{(l)} \leftarrow \boldsymbol{\omega}_k^{\star}$ and remove the components corresponding to partials 1, 3,… from $\widetilde{\boldsymbol{\beta}}^{(l-1)}$ so as to obtain $\widetilde{\boldsymbol{\beta}}^{(l)}$.

Note: In all the ratios g presented above, the Jacobian of the transforms for birth/death, increase/decrease, divide/multiply should appear (see Ref. 43). Here, it equals 1, thus it does not appear.

[1] A. Bregman, *Auditory Scene Analysis* (MIT, Cambridge, USA, 1990).
[2] N. Fletcher and T. Rossing, *The Physics of Musical Instruments*, 2nd ed. (Springer, Berlin, 1998).
[3] M. McIntyre, R. Schumacher, and J. Woodhouse, "On the oscillations of musical instruments," J. Acoust. Soc. Am. **74**, 1325–1345 (1983).
[4] A. de Cheveigné, "Separation of concurrent harmonic sounds: Fundamental frequency estimation and a time-domain cancellation model for auditory processing," J. Acoust. Soc. Am. **93**, 3271–3290 (1993).
[5] A. de Cheveigné and H. Kawahara, "Multiple period estimation and pitch perception model," Speech Commun. **27**, 175–185 (1999).
[6] T. Virtanen and A. Klapuri, "Separation of harmonic sound sources using sinusoidal modeling," in *International Conference on Acoustics, Speech, and Signal Processing*, IEEE ICASSP, Istanbul, Turkey, 2000, Vol. **2**, pp. 765–768.
[7] T. Virtanen and A. Klapuri, "Separation of harmonic sounds using linear models for the overtone series," in *International Conference on Acoustics, Speech, and Signal Processing*, IEEE ICASSP, Orlando, FL, 2002.
[8] R. Irizarry, "Local harmonic estimation in musical sound signals," J. Am. Stat. Assoc. **96**(454), 357–367 (2001).
[9] R. Irizarry, "Weighted estimation of harmonic components in a musical sound signal," J. Time Ser. Anal. **23**(1), 29–48 (2002).
[10] H. Kameoka, T. Nishimoto, and S. Sagayama, "Separation of harmonic structures based on tied gaussian mixture model and information criterion for concurrent sounds," in *International Conference on Acoustics, Speech, and Signal Processing*, IEEE ICASSP, Montreal, Canada, 2004.
[11] K. Kashino, K. Nakadai, T. Kinoshita, and H. Tanaka, "Organisation of hierarchical perceptual sounds: Music scene analysis with autonomous processing modules and a quantitative information integration mechanism," in *International Joint Conference on Artificial Intelligence*, IJCAI, Montreal, Quebec, 1995, pp. 158–164.
[12] K. Kashino and H. Murase, "A sound source identification system for ensemble music based on template adaptation and music stream extraction," Speech Commun. **27**, 337–349 (1999).
[13] K. Kashino and S. J. Godsill, "Bayesian estimation of simultaneous musical notes based on frequency domain modelling," in *International Conference on Acoustics, Speech, and Signal Processing*, IEEE ICASSP, Montreal, Canada, 2004.
[14] C. Raphael, "Automatic transcription of piano music," in *International Conference on Music Information Retrieval*, ISMIR, Paris, France, 2002.
[15] C. Raphael and J. Stoddard, "Harmonic analysis with probabilistic graphical models," in *International Conference on Music Information Retrieval*, ISMIR, Baltimore, MD, 2003.
[16] J. Tabrikian, S. Dubnov, and Y. Dickalov, "Maximum a-posteriori probability pitch tracking in noisy environments using harmonic model," IEEE Trans. Speech Audio Process. **12**(1), 76–87 (2004).
[17] A. Sterian, M. Simoni, and G. Wakefield, "Model-based musical transcription," in *International Computer Music Conference*, ICMC, Beijing, China, 1999.
[18] L. Parra and U. Jain, "Approximate Kalman filtering for the harmonic plus noise model," in *Workshop on Applications of Signal Processing to Audio and Acoustics*, IEEE WASPAA, New Paltz, NY, 2001.
[19] A. Cemgil, H. Kappen, and D. Barber, "Generative model based polyphonic music transcription," in *Workshop on Applications of Signal Processing to Audio and Acoustics*, IEEE WASPAA, New Paltz, NY, 2003.
[20] A. Cemgil, B. Kappen, and D. Barber, "A generative model for music transcription," IEEE Trans. Speech Audio Process. (to be published).
[21] H. Laurent and C. Doncarli, "Stationarity index for abrupt changes detection in the time-frequency plane," IEEE Signal Process. Lett. **5**(2), 43–45 (1998).
[22] M. Davy and S. Godsill, "Detection of abrupt spectral changes using support vector machines an application to audio signal segmentation," in *International Conference on Acoustics, Speech, and Signal Processing*, IEEE ICASSP, Orlando, FL, 2002, Vol. **2**, pp. 1313–1316.
[23] F. Desobry, M. Davy, and C. Doncarli, "An online kernel change detection algorithm," IEEE Trans. Signal Process. **53**(5), 2961–2974 (2005).
[24] M. Basseville and I. Nikiforov, *Detection of Abrupt Changes-Theory and Application* (Prentice-Hall, Englewood Cliffs, NJ, 1993).
[25] J. Paulus and A. Klapuri, "Measuring the similarity of rhythmic patterns," in *International Conference on Music Information Retrieval*, ISMIR, Paris, France, 2002.
[26] C. Duxbury, M. Sandler, and M. Davies, "A hybrid approach to musical note onset detection," in *International Conference on Digital Audio Effects*, DAFX, Hamburg, Germany, 2002.
[27] P. Walmsley, S. J. Godsill, and P. J. W. Rayner, "Multidimensional optimisation of harmonic signals," in *European Signal Processing Conference*, EURASIP, Island of Rhodes, Greece, 1998.
[28] P. J. Walmsley, S. J. Godsill, and P. J. W. Rayner, "Polyphonic pitch tracking using joint Bayesian estimation of multiple frame parameters," in *Workshop on Applications of Signal Processing to Audio and Acoustics*, IEEE WASPAA, New Paltz, NY, 1999.
[29] M. Davy and S. Godsill, "Bayesian harmonic models for musical signal analysis," in *Seventh Valencia International Meeting, Bayesian Statistics 7* (Oxford U.P., Tenerife, Spain, 2002).
[30] S. Godsill and M. Davy, "Bayesian harmonic models for musical pitch estimation and analysis," in *International Conference on Acoustics, Speech, and Signal Processing*, IEEE ICASSP, Orlando, FL, 2002.
[31] H. Feichtinger and T. Strohmer, *Gabor Analysis and Algorithms: Theory and Applications, Applied and Numerical Harmonic Analysis* (Birkhauser, Boston, 1998).
[32] P. J. Wolfe, S. J. Godsill, and W. Ng, "Bayesian variable selection and regularisation for time-frequency surface estimation," J. R. Stat. Soc. Ser. B. Methodol. **66**(3), 575–589 (2004).
[33] R. Gribonval and E. Bacry, "Harmonic decomposition of audio signals with matching pursuit," IEEE Trans. Signal Process. **51**(1), 101–111 (2003).
[34] J. Rissanen, "A universal prior for integers and estimation by minimum description length," Ann. Stat. **11**(2), 416–431 (1983).
[35] H. Akaike, "A new look at statistical model identification," IEEE Trans. Autom. Control **19**, 716–723 (1974).
[36] G. Schwartz, "Estimating the dimension of a model," Ann. Stat. **6**(2), 461–464 (1985).
[37] M. Stephens, "Dealing with label-switching in mixture models," J. R. Stat. Soc. Ser. B. Methodol. **62**(4), 795 (2000).
[38] A. Jasra, C. Holmes, and D. Stephens, "MCMC and the label switching problem in Bayesian mixture models," Stat. Sci. **20**(1), 50–67 (2005).
[39] C. Andrieu and A. Doucet, "Joint Bayesian detection and estimation of noisy sinusoids via reversible jump MCMC," IEEE Trans. Signal Process. **47**(10), 2667–2676 (1999).
[40] M. Davy and J. Idier, "Fast MCMC computations for the estimation of sparse processes from noisy observations," in *International Conference on Acoustics, Speech, and Signal Processing*, IEEE ICASSP, Montreal, Canada, 2004.
[41] In practice, these parameters are chosen as $\nu_0 = \gamma_0 = 0$. The corresponding prior is no longer a proper probability density function (because its integral over σ_v^2 is infinite), but this is not a problem because the posterior distribution remains a proper probability density function.
[42] *Markov Chain Monte Carlo in Practice*, edited by W. R. Gilks, S. Richardson, and D. J. Spiegelhalter (Chapman & Hall, London, 1996).
[43] C. Robert and G. Casella, *Monte Carlo Statistical Methods* (Springer, New York, 2000).
[44] S. J. Godsill and P. J. W. Rayner, *Digital Audio Restoration: A Statistical Model-Based Approach* (Springer, Berlin, 1998).
[45] S. J. Godsill, "On the relationship between Markov chain Monte Carlo

methods for model uncertainty," J. Comput. Graph. Stat. **10**(2), 230–248 (2001).
[46]J. Vermaak, C. Andrieu, A. Doucet, and S. J. Godsill, "Reversible jump Markov chain Monte Carlo strategies for Bayesian model selection in autoregressive processes," J. Time Ser. Anal. **25**(6), 785–809 (2004).
[47]D. Battle, P. Gerstoft, W. Hodgkiss, W. A. Kuperman, and P. Nielsen, "Bayesian model selection applied to self-noise geoacoustic inversion," J. Acoust. Soc. Am. **116**, 2043–2056 (2004).
[48]P. J. Green, "Reversible jump Markov-chain Monte Carlo computation and Bayesian model determination," Biometrika **82**(4), 711–732 (1995).
[49]For a proof, see Ref. 39, where a related frequency estimation algorithm is analyzed.
[50]M. Davy and S. Godsill, "Bayesian harmonic models for musical pitch estimation and analysis," Technical Report No. CUED/F-INFENG/TR.431, Department of Engineering, University of Cambridge, Cambridge, UK (unpublished).
[51]W. R. Gilks and G. O. Roberts, in *Markov Chain Monte Carlo in Practice*, edited by W. R. Gilks, S. Richardson, and D. J. Spiegelhalter (Chapman & Hall, London, 1996); Chap. Strategies for Improving MCMC, pp. 89–110. See Ref. 42.
[52]P. Del Moral and A. Doucet, "On a class of genealogical and interacting metropolis models," in *Séminaire de Probabilités*, Vol. **XXXVII** of *Lecture Notes in Mathematics 1832*, edited by J. Azéma, M. Emery, M. Ledoux, and M. Yor (Springer, Berlin, 2003), pp. 415–446.
[53]T. Virtanen and A. Klapuri, "Separation of harmonic sounds using multipitch analysis and iterative parameter estimation," in *Workshop on Applications of Signal Processing to Audio and Acoustics*, IEEE WASPAA, New Paltz, NY, 2001, pp. 83–86.