High resolution NMF for modeling mixtures of non-stationary signals in the time-frequency domain

Roland Badeau

High resolution NMF for modeling mixtures of non-stationary signals in the time-frequency domain

NMF à haute résolution pour la modélisation de mélanges de signaux non-stationnaires dans le domaine temps-fréquence

Roland Badeau
Institut Mines-Télécom, Télécom ParisTech, CNRS LTCI
Email : roland.badeau@telecom-paristech.fr

Abstract

Nonnegative Matrix Factorization (NMF) is a powerful tool for decomposing mixtures of non-stationary signals in the Time-Frequency (TF) domain. However, unlike the High Resolution (HR) methods dedicated to mixtures of complex exponentials, its spectral resolution is limited by that of the underlying TF representation. In this paper, we present a unified probabilistic model called HR-NMF, that permits to overcome this limit by taking both phases and local correlations in each frequency band into account. This model is estimated with a recursive implementation of the Expectation-Maximization (EM) algorithm. Its capabilities are illustrated in the context of audio source separation and audio inpainting.

Index Terms

Nonnegative Matrix Factorization, High Resolution methods, Expectation-Maximization algorithm, Source separation, Audio inpainting.

Résumé

La NMF (*Nonnegative Matrix Factorization*) est un outil puissant pour décomposer des mélanges de signaux non-stationnaires dans le domaine Temps-Fréquence (TF). Cependant, contrairement aux méthodes à Haute Résolution (HR) dédiées aux mélanges d'exponentielles complexes, sa résolution spectrale est limitée par celle de la représentation TF sous-jacente. Dans cet article, nous présentons un modèle probabiliste unifié appelé HR-NMF, qui permet de s'affranchir de cette limite en tenant compte à la fois des phases et des corrélations locales dans chaque bande de fréquences. Ce modèle est estimé à l'aide d'une implémentation récursive de l'algorithme Espérance-Maximisation (EM). Son potentiel est illustré dans le contexte de la séparation de sources audio et de la restauration de signaux audio.

Mots clés

Nonnegative Matrix Factorization, Méthodes à Haute Résolution, Algorithme Espérance-Maximisation, Séparation de sources, Restauration de signaux audio.

This work is supported by the French National Research Agency (ANR) as a part of the DReaM project (ANR-09-CORD-006-03) and partly supported by the Quaero Program, funded by OSEO.

I. INTRODUCTION

NONNEGATIVE matrix factorization was originally introduced as a rank-reduction technique, which approximates a non-negative matrix $\boldsymbol{V} \in \mathbb{R}^{F\times T}$ as a product of two non-negative matrices $\boldsymbol{W} \in \mathbb{R}^{F\times K}$ and $\boldsymbol{H} \in \mathbb{R}^{K\times T}$ with $K < \min(F,T)$ [1]. In audio signal processing, it is often used for decomposing a magnitude or power TF representation, such as the spectrogram [2]. The columns of matrix $\boldsymbol{W}$ are then interpreted as a dictionary of spectral templates, whose temporal activations are represented in the rows of matrix $\boldsymbol{H}$. Several applications to audio have been addressed, such as multi-pitch estimation [3], [4], [5], automatic music transcription [6], [7], musical instrument recognition [8], or source separation [9], [10], [11].

In the literature, many variants of NMF have been proposed, in order to enforce some desired properties in the factorization, such as the harmonicity of the spectral templates [2], [5], [7], the smoothness of the spectral envelopes [12], [5], [7], the smoothness of the temporal activations [2], [13], [7], the sparsity in $\boldsymbol{W}$ or $\boldsymbol{H}$ [12], [4], or for modeling some spectral non-stationarities [14], [15]. The proposed approaches generally consist of parameterizing $\boldsymbol{W}$ or $\boldsymbol{H}$, or using a predefined dictionary $\boldsymbol{W}$ (either parametric or non-parametric, possibly learned beforehand), or they rely on Bayesian inference, which involves some prior distributions of the model parameters. Several probabilistic models, involving latent variables, have thus been designed for introducing some a priori knowledge in NMF. They also permit to exploit well-known statistical inference techniques in order to estimate the model parameters. Those models include NMF with additive Gaussian noise [12], Probabilistic Latent Component Analysis (PLCA) [16], mixtures of Poisson components [2], and mixtures of Gaussian components [13].

Since phases are generally discarded in these models[1], reconstructing the phase field requires employing ad-hoc methods [19]. To the best of our knowledge, apart the complex NMF which was designed in a deterministic framework [20], [21], the only probabilistic model that takes the phase field into account (but in a non-informative way) is the Itakura-Saito (IS)-NMF [13]. Separating the signal components is then proven equivalent to Wiener filtering. However, the phase field is still ignored when estimating IS-NMF, and the spectral resolution of IS-NMF is limited by that of the TF representation (sinusoids in the same frequency band cannot be properly separated). In other respects, IS-NMF assumes that all TF coefficients are independent, which is not the case of sinusoidal signals for instance. In the literature, Markov models have thus been proposed for taking the local dependencies between contiguous TF coefficients of a magnitude or power TF representation into account [22], [23], [24].

In [25], we introduced a unified model called HR-NMF, which natively takes both phases and local correlations in each frequency band into account. This approach avoids using a phase reconstruction algorithm, and we showed that it overcomes the spectral resolution of the TF representation. It can be used with both complex-valued and real-valued TF representations (like the short-time Fourier transform or the modified discrete cosine transform). In this paper, we go further into detail in the study of HR-NMF. The Expectation-Maximization (EM) algorithm designed for estimating this model is improved:

- numerical stability issues encountered in the previous implementation [25] have been solved;
- Kalman filtering/smoothing is now implemented in square root form, in order to guarantee the positive-definiteness of covariance matrices;
- multiplicative update rules are proposed for initializing the EM algorithm.

Besides, all mathematical derivations are now provided in the Appendix. This paper is organized as follows: HR-NMF is introduced in section II, and our recursive implementation of the EM algorithm for estimating this model is presented in section III. Section IV is devoted to experimental results, and conclusions are drawn in section V.

Notation

The following notation will be used throughout the paper:

- x: scalar (normal letter),
- $\boldsymbol{v}$: column vector (bold lower case letter),
- v_i: i-th entry of $\boldsymbol{v}$ (indexed lower case letter),
- $\boldsymbol{M}$: matrix (bold upper case letter),
- $M_{(i,j)}$: (i,j)-th entry of $\boldsymbol{M}$ (indexed upper case letter),

[1]More precisely, phases are discarded in the source signal models, but in the case of multichannel mixtures, phase relationships in the mixture model can be taken into account via the mixing matrix [17], [18].

- $[\boldsymbol{M}, \boldsymbol{N}]$: horizontal concatenation of $\boldsymbol{M}$ and $\boldsymbol{N}$,
- $[\boldsymbol{M}; \boldsymbol{N}]$: vertical concatenation of $\boldsymbol{M}$ and $\boldsymbol{N}$,
- $\mathbf{0}$: vector or matrix whose entries are all equal to 0,
- $\mathbf{1}$: vector whose entries are all equal to 1,
- $\mathrm{diag}(.)$: (block)-diagonal matrix,
- $\boldsymbol{I}$: identity matrix ($\boldsymbol{I} = \mathrm{diag}(\mathbf{1})$),
- $\boldsymbol{M}^*$: conjugate of matrix (or vector) $\boldsymbol{M}$,
- $\boldsymbol{M}^H$: conjugate transpose of matrix (or vector) $\boldsymbol{M}$,
- $\boldsymbol{M}^\dagger$: Moore-Penrose pseudo-inverse of matrix $\boldsymbol{M}$,
- $\mathbb{E}[.]_{/x}$: conditional expectation of (.) given x,
- $\mathcal{N}_{\mathbb{F}}(\boldsymbol{\mu}, \boldsymbol{R})$: real (if $\mathbb{F} = \mathbb{R}$) or circular complex (if $\mathbb{F} = \mathbb{C}$) multivariate normal distribution of mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{R}$.
- for a given vector $\overline{\boldsymbol{v}}$ of dimension $\overline{D}$, and any subvector $\boldsymbol{v}$ of dimension $D \leq \overline{D}$ (whose entries are a subset of those of $\overline{\boldsymbol{v}}$), $\boldsymbol{J}_v^{\overline{v}}$ denotes the $\overline{D} \times D$ selection matrix such that $\boldsymbol{v} = \boldsymbol{J}_v^{\overline{v}H}\, \overline{\boldsymbol{v}}$.

II. Time-Frequency Mixture Model

The HR-NMF mixture model of TF data $x(f,t) \in \mathbb{F}$ (where $\mathbb{F} = \mathbb{R}$ or $\mathbb{C}$) is defined for all discrete frequencies $1 \leq f \leq F$ and times $1 \leq t \leq T$ as the sum of K latent components $c_k(f,t) \in \mathbb{F}$ plus a white noise $n(f,t) \sim \mathcal{N}_{\mathbb{F}}(0, \sigma^2)$:

$$x(f,t) = n(f,t) + \sum_{k=1}^{K} c_k(f,t) \tag{1}$$

where

- $c_k(f,t) = \sum_{p=1}^{P(k,f)} a(p,k,f)\, c_k(f, t-p) + b_k(f,t)$ is obtained by autoregressive filtering of a non-stationary signal $b_k(f,t) \in \mathbb{F}$ (where $a(p,k,f) \in \mathbb{F}$ and $P(k,f) \in \mathbb{N}$ is such that $a(P(k,f),k,f) \neq 0$),
- $b_k(f,t) \sim \mathcal{N}_{\mathbb{F}}(0, v_k(f,t))$ where $v_k(f,t)$ is defined as

$$v_k(f,t) = w(k,f)\, h(k,t), \tag{2}$$

  with $w(k,f) \geq 0$ and $h(k,t) \geq 0$,
- processes n and $b_1 \dots b_K$ are mutually independent.

Moreover, $\forall (k,f) \in \{1 \dots K\} \times \{1 \dots F\}$, the random vectors $\boldsymbol{c}_k(f,0) = [c_k(f,0); \dots; c_k(f, -P(k,f)+1)]$ are assumed to be independent and distributed according to the prior distribution $\boldsymbol{c}_k(f,0) \sim \mathcal{N}_{\mathbb{F}}(\boldsymbol{\mu}_k(f), \boldsymbol{Q}_k(f)^{-1})$, where the mean $\boldsymbol{\mu}_k(f)$ and the precision matrix $\boldsymbol{Q}_k(f)$ are fixed parameters[2]. Lastly, we assume that $\forall f \in \{1 \dots F\}$, $\forall t \leq 0$, $x(f,t)$ is unobserved. The parameters to be estimated are σ^2, $a(p,k,f)$, $w(k,f)$, and $h(k,t)$. This time-frequency model generalizes some very popular models, widely used in various signal processing communities:

- If $\sigma^2 = 0$ and $\forall k, f, P(k,f) = 0$, (1) becomes $x(f,t) = \sum_{k=1}^{K} b_k(f,t)$, thus $x(f,t) \sim \mathcal{N}_{\mathbb{F}}(0, \widehat{V}_{ft})$, where $\widehat{\boldsymbol{V}}$ is defined by the NMF $\widehat{\boldsymbol{V}} = \boldsymbol{W}\,\boldsymbol{H}$ with $W_{fk} = w(k,f)$ and $H_{kt} = h(k,t)$. The maximum likelihood estimation of $\boldsymbol{W}$ and $\boldsymbol{H}$ is then equivalent to the minimization of the IS-divergence between the matrix model $\widehat{\boldsymbol{V}}$ and the spectrogram $\boldsymbol{V}$ (where $V_{ft} = |x(f,t)|^2$), that is why this model is referred to as IS-NMF [13].
- For given values of k and f, if $\forall t$, $h(k,t) = 1$, then $c_k(f,t)$ is an autoregressive process of order $P(k,f)$.
- For given values of k and f, if $P(k,f) \geq 1$ and $\forall t \geq P(k,f)+1$, $h(k,t) = 0$, then $c_k(f,t)$ can be written in the form $c_k(f,t) = \sum_{p=1}^{P(k,f)} \alpha_p\, z_p^t$, where $z_1 \dots z_{P(k,f)}$ are the roots of the polynomial $z^{P(k,f)} - \sum_{p=1}^{P(k,f)} a(p,k,f) z^{P(k,f)-p}$. This corresponds to the Exponential Sinusoidal Model (ESM)[3] commonly used in HR spectral analysis of time series [26].

For these reasons, model (1) is referred to as HR-NMF.

[2]In practice we choose $\boldsymbol{\mu}_k(f) = [0; \dots; 0]^\top$ and $\boldsymbol{Q}_k(f)^{-1} = \xi \boldsymbol{I}$, where $\boldsymbol{I}$ is the identity matrix and ξ is small relative to 1, in order to both enforce the causality of the latent components and avoid singular matrices.

[3]Actually HR-NMF also encompasses the more general Polynomial Amplitude Complex Exponentials (PACE) model introduced in [26].

III. Expectation-Maximization Algorithm

The EM algorithm [27] is an iterative method which aims to estimate the parameters of a probabilistic model involving both observed and latent random variables, by recursively increasing the log-likelihood of the observed variables at each iteration. It consists of two steps called Expectation (E-step) and Maximization (M-step).

In order to estimate the HR-NMF model parameters, the EM algorithm is applied to the observed data x and the latent components $c_1 \dots c_K$ (here the complete data is $\{x, c_1 \dots c_K\}$). In order to handle the case of missing data, we define $\delta(f,t) = 1$ if $x(f,t)$ is observed, and $\delta(f,t) = 0$ else.

A. Square root implementation

As some numerical stability issues have been encountered with the previous implementation of the EM algorithm presented in [25], a new implementation is introduced here, where covariance matrices are represented in their square root form. We call "square root" of a positive definite matrix $\boldsymbol{R}$ any square matrix[4] $\sqrt{\boldsymbol{R}}$ such that $\sqrt{\boldsymbol{R}}\sqrt{\boldsymbol{R}}^H = \boldsymbol{R}$. This factorization guarantees that $\boldsymbol{R}$ remains positive definite, even in the presence of rounding errors due to the finite machine precision. In the following mathematical derivations, $\boldsymbol{R}$ will often be obtained in the form $\boldsymbol{R} = \boldsymbol{M}\,\boldsymbol{M}^H$, where matrix $\boldsymbol{M}$ has more columns than rows. The square matrix $\sqrt{\boldsymbol{R}}$ can then be computed by means of a function $\mathcal{LT}$ (to be understood as the acronym of "lower triangular"), such that $\sqrt{\boldsymbol{R}} = \mathcal{LT}(\boldsymbol{M})$. For instance, function $\mathcal{LT}$ can compute the thin QR factorization $\boldsymbol{M}^H = \boldsymbol{Q}\sqrt{\boldsymbol{R}}^H$, where $\boldsymbol{Q}^H\boldsymbol{Q} = \boldsymbol{I}$ and $\sqrt{\boldsymbol{R}}^H$ is upper triangular, which implies that $\boldsymbol{R} = \boldsymbol{M}\,\boldsymbol{M}^H = \sqrt{\boldsymbol{R}}\sqrt{\boldsymbol{R}}^H$.

B. Maximization Step (M-step)

In the EM algorithm, the M-step aims to maximize the conditional expectation (given the observations) of the log-likelihood of the complete data w.r.t. the model parameters. We first note that:

$$\begin{aligned}
p(c_1 \dots c_K, x) &= p(x/c_1 \dots c_K) \prod_{k=1}^{K} p(c_k) \\
p(x/c_1 \dots c_K) &\propto \prod_{f=1}^{F}\prod_{t=1}^{T} \exp\left(\delta(f,t)\left(-\ln(\sigma^2) + \tfrac{|x(f,t)-\sum_{k=1}^{K} c_k(f,t)|^2}{\sigma^2}\right)\right) \\
\forall k,\ p(c_k) &\propto \prod_{f=1}^{F}\prod_{t=1}^{T} \exp\left(-\ln(w(k,f)h(k,t)) + \tfrac{|c_k(f,t)-\sum_{p=1}^{P(k,f)} a(p,k,f)\,c_k(f,t-p)|^2}{w(k,f)h(k,t)}\right),
\end{aligned}$$

where symbol $\propto$ denotes equality up to a multiplicative factor. Thus the conditional expectation of the log-likelihood of the complete data is

$$\begin{aligned}
Q &= \mathbb{E}_{/x}\left[\ln(p(c_1 \dots c_K, x))\right] \\
&= \mathbb{E}_{/x}\left[\ln(p(x/c_1 \dots c_K))\right] + \textstyle\sum_{k=1}^{K} \mathbb{E}_{/x}\left[\ln(p(c_k))\right].
\end{aligned}$$

It can be written in the form[5] $Q \overset{\mathsf{C}}{=} Q_0 + \sum_{k=1}^{K} Q_k$ where

$$Q_0 = -\sum_{f=1}^{F}\sum_{t=1}^{T} \delta(f,t)\ln(\sigma^2) + e(f,t)/\sigma^2, \tag{3}$$

$$Q_k = -\sum_{f=1}^{F}\sum_{t=1}^{T} \ln(w(k,f)h(k,t)) + \tfrac{\boldsymbol{a}(k,f)^H \boldsymbol{S}(k,f,t)\,\boldsymbol{a}(k,f)}{w(k,f)h(k,t)}, \tag{4}$$

$$e(f,t) = \delta(f,t)\,\mathbb{E}_{/x}\left[\left|x(f,t) - \sum_{k=1}^{K} c_k(f,t)\right|^2\right], \tag{5}$$

$$\boldsymbol{\alpha}(k,f) = -[a(1,k,f); \dots; a(P(k,f),k,f)] \tag{6}$$

$$\boldsymbol{a}(k,f) = [1; \boldsymbol{\alpha}(k,f)], \tag{7}$$

[4]Note that there is an infinity of such square roots.

[5]$\overset{\mathsf{C}}{=}$ denotes equality up to additive and multiplicative constants which do not depend on the model parameters to be estimated.

and $\forall k, f$, for all $0 \leq p_1, p_2 \leq P(k,f)$,

$$S_{(p_1,p_2)}(k,f,t) = \mathbb{E}_{/x}[c_k(f,t-p_1)^* \, c_k(f,t-p_2)]. \tag{8}$$

Maximizing Q is thus equivalent to independently maximizing Q_0 with respect to (w.r.t.) σ^2 and each Q_k w.r.t. $h(k,t)$, $w(k,f)$ and $\boldsymbol{a}(k,f)$. Since the maximization of Q_k does not admit a closed form solution, Q_k is recursively maximized w.r.t. $(w(k,f), \boldsymbol{a}(k,f))$ and w.r.t. $h(k,t)$. Either this recursive maximization is repeated until convergence inside the M-step, which results in an exact EM algorithm, or only a few iterations are performed, which leads to a generalized EM (GEM) algorithm [27] [6].

The full mathematical derivation of the M-step is provided in Appendix A. The pseudo-code is summarized in Table I [7]. Its complexity is $O(FTK(1+P)^3)$, where $P = \max_{k,f} P(k,f)$. Note that parallel computing permits to process the K components simultaneously, reducing the computational time to $O(FT(1+P)^3)$.

Inputs: $\delta(f,t)$, $e(f,t)$, $\sqrt{\boldsymbol{S}}(k,f,t)$, $h(k,t)$	**Eq.**
$\sigma^2 = \sum_{f=1}^{F}\sum_{t=1}^{T} e(f,t) \Big/ \sum_{f=1}^{F}\sum_{t=1}^{T} \delta(f,t)$	(16)
For $k = 1$ to K, [6]	
Repeat (as many times as wanted) [7]:	
For $f = 1$ to F, [6]	
$\sqrt{\boldsymbol{\Sigma}}(k,f) = \mathcal{LT}\left(\left[\frac{\sqrt{\boldsymbol{S}}(k,f,1)}{\sqrt{Th(k,1)}}, \ldots, \frac{\sqrt{\boldsymbol{S}}(k,f,T)}{\sqrt{Th(k,T)}}\right]\right)$	(19)
$\boldsymbol{a}(k,f) = \left[1, -\left(\boldsymbol{J}_1^{\boldsymbol{a}\,H}\sqrt{\boldsymbol{\Sigma}}(k,f)\right)\left(\boldsymbol{J}_{\boldsymbol{\alpha}}^{\boldsymbol{a}\,H}\sqrt{\boldsymbol{\Sigma}}(k,f)\right)^{\dagger}\right]^H$	(20)
$w(k,f) = \|\sqrt{\boldsymbol{\Sigma}}(k,f)^H \boldsymbol{a}(k,f)\|^2$	(18)
End for f;	
For $t = 1$ to T, [6]	
$h(k,t) = \sum_{f=1}^{F}\left(\frac{\|\sqrt{\boldsymbol{S}}(k,f,t)^H \boldsymbol{a}(k,f)\|}{\sqrt{F w(k,f)}}\right)^2$	(17)
End for t;	
Normalization of the NMF: $H_k = \max_t(h(k,t))$,	
$w(k,f) = H_k \, w(k,f)$, $h(k,t) = h(k,t)/H_k$	
End repeat;	
End for k;	
Outputs: σ^2, $\boldsymbol{a}(k,f), w(k,f), h(k,t)$.	

TABLE I
PSEUDO-CODE OF THE M-STEP

C. Expectation Step (E-step)

The purpose of the E-step is to determine the a posteriori distribution[10] of the latent components $c_k(f,t)$ given the observations $x(f,t)$. Since these random variables are mutually independent for different values of f, the E-step can process each f separately. Nevertheless, the computational complexity of a direct implementation of the E-step would be $O(FT^3K^2)$, which is prohibitively expensive when T becomes high. We thus propose a faster recursive implementation of the E-step based on a linear state space representation and Kalman filtering theory [28].

Note that similar state space representations have already been proposed for estimating mixtures of autoregressive processes in the literature of time series analysis [29], and for addressing the particular problem of blind source separation (see *e.g.* [30], [31], [32], [33]). However, those state space representations were designed for modeling mixtures of (frame-wise) stationary signals in the temporal domain, whereas the proposed HR-NMF model deals with mixtures of non-stationary signals in the time-frequency domain. We thus present in this section a specific

[6] The GEM algorithm still guarantees that the log-likelihood of the observed data is non-decreasing.
[7] According to the notation introduced in section I, $\boldsymbol{J}_1^{\boldsymbol{a}} = [1; 0; \ldots; 0]$.
[6] This loop can be processed in parallel.
[7] This loop has to be processed sequentially.
[10] and more precisely, $e(f,t)$ and $\boldsymbol{S}(k,f,t)$ defined in equations (5) and (8).

linear state space representation and the corresponding Kalman filter/smoother, which have been especially designed for the HR-NMF model.

Let us first introduce the linear state space representation of the HR-NMF model defined in equation (1):

$$\boldsymbol{\gamma}(f,t) = \boldsymbol{A}(f)\,\boldsymbol{\gamma}(f,t-1) + \boldsymbol{b}'(f,t), \tag{9}$$

$$x(f,t) = \boldsymbol{u}(f)^H \boldsymbol{\gamma}(f,t) + n'(f,t), \tag{10}$$

where

- $\mathcal{K}(f)$ denotes the set $\{k \in \{1 \dots K\} / P(k,f) \geq 1\}$;
- $\forall f,t$, $\forall k \in \mathcal{K}(f)$,
$$\boldsymbol{c}(k,f,t) = \left[c_k(f,t); \dots ; c_k(f,t-P(k,f)+1)\right];$$
- the state vector $\boldsymbol{\gamma}(f,t)$ contains $\boldsymbol{c}(k,f,t)$ $\forall k \in \mathcal{K}(f)$;
- the state transition matrix is
$$\boldsymbol{A}(f) = \mathrm{diag}(\{\boldsymbol{A}(k,f)\}_{k \in \mathcal{K}(f)}),$$
where $\boldsymbol{A}(k,f) = a(1,k,f)$ if $P(k,f) = 1$, otherwise $\boldsymbol{A}(k,f) =$
$$\begin{bmatrix} a(1,k,f) \dots a(P(k,f)-1,k,f) & a(P(k,f),k,f) \\ \boldsymbol{I} & \boldsymbol{0} \end{bmatrix}$$
- $\forall f,t$, vector $\boldsymbol{c}(f,t)$ contains $c_k(f,t)$ for all $k \in \mathcal{K}(f)$,
- the process noise is
$$\boldsymbol{b}'(f,t) = \boldsymbol{J}_{c(f,t)}^{\gamma(f,t)}\,\boldsymbol{b}(f,t), \tag{11}$$
where notation $\boldsymbol{J}_{c(f,t)}^{\gamma(f,t)}$ was defined in section I, and $\boldsymbol{b}(f,t)$ contains $b_k(f,t)$ $\forall k \in \mathcal{K}(f)$; thus $\boldsymbol{b}(f,t) \sim \mathcal{N}_{\mathbb{F}}(\boldsymbol{0}, \boldsymbol{R}_{v(f,t)})$, where $\boldsymbol{R}_{v(f,t)} = \mathrm{diag}(\boldsymbol{v}(f,t))$ and $\boldsymbol{v}(f,t)$ contains $v_k(f,t)$ $\forall k \in \mathcal{K}(f)$;
- the observation is $x(f,t)$;
- the observation matrix is $\boldsymbol{u}(f)^H$, where
$$\boldsymbol{u}(f) = \boldsymbol{J}_{c(f,t)}^{\gamma(f,t)}\,\boldsymbol{1}; \tag{12}$$
- the white observation noise is $n'(f,t) = n(f,t) + \sum_{k/P(k,f)=0} b_k(f,t) \sim \mathcal{N}_{\mathbb{F}}(0, \sigma^2(f,t))$, where
$$\sigma^2(f,t) = \sigma^2 + \sum_{k/P(k,f)=0} v_k(f,t). \tag{13}$$

The full mathematical derivation of the E-step is provided in Appendix B. The pseudo-code is summarized in Table II. Its overall computational complexity is $O(FTK^3(1+P)^3)$. Note that parallel computing permits to process the F frequencies simultaneously, reducing the computational time to $O(TK^3(1+P)^3)$.

In Table II, the following notation has been used:

- $\forall f,t$, $\boldsymbol{d}(f,t)$ contains $c_k(f,t-P(k,f))$ $\forall k \in \mathcal{K}(f)$,
- $\forall f,t$, vector $\boldsymbol{c}'(f,t)$ contains $c_k(f,t)$ for all $k \notin \mathcal{K}(f)$,
- $\forall f,t$, vector $\boldsymbol{v}'(f,t)$ contains $v_k(f,t)$ for all $k \notin \mathcal{K}(f)$,
- $\forall f,t$, $\forall k \in \mathcal{K}(f)$,
$$\overline{\boldsymbol{c}}(k,f,t) = \left[c_k(f,t); \dots ; c_k(f,t-P(k,f))\right],$$
- $\forall f,t$, $\overline{\boldsymbol{\gamma}}(f,t)$ contains $\overline{\boldsymbol{c}}(k,f,t)$ for all $k \in \mathcal{K}(f)$,
- $\forall f,t$, and for any random vector $\boldsymbol{v}$, $\boldsymbol{v}^{f,t}$ is the conditional expectation of $\boldsymbol{v}$ given $\{x(f,1) \dots x(f,t)\}$. Besides, $\boldsymbol{R}_{\boldsymbol{v}}^{f,t}$ is the conditional expectation of $(\widetilde{\boldsymbol{v}}^{f,t})\,(\widetilde{\boldsymbol{v}}^{f,t})^H$ given $\{x(f,1) \dots x(f,t)\}$, where $\widetilde{\boldsymbol{v}}^{f,t} = \boldsymbol{v} - \boldsymbol{v}^{f,t}$. Similarly, for any vectors $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$, $\boldsymbol{R}_{\boldsymbol{v}_1,\boldsymbol{v}_2}^{f,t}$ is the conditional expectation of $(\widetilde{\boldsymbol{v}}_1^{f,t})\,(\widetilde{\boldsymbol{v}}_2^{f,t})^H$ given $\{x(f,1) \dots x(f,t)\}$.

Other letters in Table II denote temporary variables used in the computations.

D. Initialization of the EM algorithm

Because the proposed implementation of the EM algorithm remains computationally demanding, we present in this section a fast initialization method that permits to reduce the number of subsequent EM iterations to be performed.

Inputs after the M-step: $x(f,t)$, $\delta(f,t)$, σ^2, $\boldsymbol{A}(f)$, $v_k(f,t)$	**Eq.**
Initialization: $\forall f,\ L(f,0)=0,\ \boldsymbol{\gamma}^{f,0}(f,0)=\mathbf{0},\ \sqrt{\boldsymbol{R}}^{f,0}_{\gamma(f,0)}=\sqrt{\xi}\boldsymbol{I}$	
$\forall f,\ \overline{\boldsymbol{A}}(f)=\boldsymbol{J}^{\overline{\gamma}(f,t)}_{\gamma(f,t)}\boldsymbol{A}(f)+\boldsymbol{J}^{\overline{\gamma}(f,t)}_{d(f,t)}\boldsymbol{J}^{\gamma(f,t-1)\,H}_{d(f,t)}$	(28)
$\forall f,\ \boldsymbol{u}(f)=\boldsymbol{J}^{\gamma(f,t)}_{c(f,t)}\mathbf{1}$	(12)
$\forall f,t,\ \sigma^2(f,t)=\sigma^2+\sum_{k/P(k,f)=0} v_k(f,t)$	(13)
For $f=1$ to F, [6]	
For $t=1$ to T (**forward pass**): [7]	
Predict phase:	
$\sqrt{\boldsymbol{R}}^{f,t-1}_{\overline{\gamma}(f,t)}=\mathcal{LT}\left(\left[\overline{\boldsymbol{A}}(f)\sqrt{\boldsymbol{R}}^{f,t-1}_{\gamma(f,t-1)},\boldsymbol{J}^{\overline{\gamma}(f,t)}_{c(f,t)}\sqrt{\boldsymbol{R}}_{v(f,t)}\right]\right)$	(29)
$\boldsymbol{\Phi}^{f,t-1}=\left(\sqrt{\boldsymbol{R}}^{f,t-1\,H}_{\overline{\gamma}(f,t)}\boldsymbol{J}^{\overline{\gamma}(f,t)}_{\gamma(f,t)}\right)^{\dagger}\sqrt{\boldsymbol{R}}^{f,t-1\,H}_{\overline{\gamma}(f,t)}\boldsymbol{J}^{\overline{\gamma}(f,t)}_{d(f,t)}$	(30)
$\sqrt{\boldsymbol{\Psi}}^{f,t-1}=\mathcal{LT}\left((\boldsymbol{J}^{\overline{\gamma}(f,t)}_{d(f,t)}-\boldsymbol{J}^{\overline{\gamma}(f,t)}_{\gamma(f,t)}\boldsymbol{\Phi}^{f,t-1})^H\sqrt{\boldsymbol{R}}^{f,t-1}_{\overline{\gamma}(f,t)}\right)$	(32)
$\sqrt{\boldsymbol{R}}^{f,t-1}_{\gamma(f,t)}=\mathcal{LT}\left(\boldsymbol{J}^{\overline{\gamma}(f,t)\,H}_{\gamma(f,t)}\sqrt{\boldsymbol{R}}^{f,t-1}_{\overline{\gamma}(f,t)}\right)$	(31)
$\boldsymbol{d}^{f,t-1}(f,t)=\boldsymbol{J}^{\gamma(f,t-1)\,H}_{d(f,t)}\boldsymbol{\gamma}^{f,t-1}(f,t-1)$	(26)
$\boldsymbol{\gamma}^{f,t-1}(f,t)=\boldsymbol{A}(f)\,\boldsymbol{\gamma}^{f,t-1}(f,t-1)$	(24)
$\boldsymbol{\phi}^{f,t-1}=\boldsymbol{d}^{f,t-1}(f,t)-\boldsymbol{\Phi}^{f,t-1\,H}\boldsymbol{\gamma}^{f,t-1}(f,t)$	(23)
Update phase:	
$\boldsymbol{\mu}(f,t)=(\sqrt{\boldsymbol{R}}^{f,t-1}_{\gamma(f,t)})^H\boldsymbol{u}(f)$	(35)
$\varepsilon(f,t)=\sigma^2(f,t)+\|\boldsymbol{\mu}(f,t)\|^2$	(36)
$\boldsymbol{\lambda}(f,t)=\frac{\delta(f,t)}{\varepsilon(f,t)}\sqrt{\boldsymbol{R}}^{f,t-1}_{\gamma(f,t)}\,\boldsymbol{\mu}(f,t)$	(37)
$\sqrt{\boldsymbol{R}}^{f,t}_{\gamma(f,t)}=\sqrt{\boldsymbol{R}}^{f,t-1}_{\gamma(f,t)}-\boldsymbol{\lambda}(f,t)\frac{\boldsymbol{\mu}(f,t)^H}{1+\sqrt{\frac{\sigma^2(f,t)}{\varepsilon(f,t)}}}$	(34)
$\epsilon^{f,t}(f,t)=x(f,t)-\boldsymbol{u}(f)^H\boldsymbol{\gamma}^{f,t-1}(f,t)$	(39)
$\boldsymbol{\gamma}^{f,t}(f,t)=\boldsymbol{\gamma}^{f,t-1}(f,t)+\boldsymbol{\lambda}(f,t)\,\epsilon^{f,t}(f,t)$	(38)
$L(f,t)\overset{\zeta}{=}L(f,t-1)-\delta(f,t)\left(\ln(\varepsilon(f,t))+\frac{\|\epsilon^{f,t}(f,t)\|^2}{\varepsilon(f,t)}\right)$	(40)
End for t;	
For $t=T$ downto 1 (**backward pass**): [7]	
Wiener filtering phase:	
$\epsilon^{f,T}(f,t)=x(f,t)-\boldsymbol{u}(f)^H\boldsymbol{\gamma}^{f,T}(f,t)$	(43)
$\boldsymbol{c}'^{f,T}(f,t)=\frac{\delta(f,t)}{\sigma^2(f,t)}\,\boldsymbol{v}'(f,t)\,\epsilon^{f,T}(f,t)$	(42)
$\boldsymbol{\mu}'(f,t)=(\sqrt{\boldsymbol{R}}^{f,T}_{\gamma(f,t)})^H\boldsymbol{u}(f)$	(47)
$e'(f,t)=\delta(f,t)\left(\|\epsilon^{f,T}(f,t)\|^2+\|\boldsymbol{\mu}'(f,t)\|^2\right)$	(46)
$e(f,t)=\delta(f,t)\frac{\sigma^2}{\sigma^2(f,t)}\left((\sigma^2(f,t)-\sigma^2)+\frac{\sigma^2}{\sigma^2(f,t)}e'(f,t)\right)$	(45)
$\forall k\notin\mathcal{K}(f),\sqrt{\boldsymbol{S}}(k,f,t)=\sqrt{\frac{v_k(f,t)}{\sigma^2(f,t)}}$ $\sqrt{(\sigma^2(f,t)-\delta(f,t)v_k(f,t))+\frac{v_k(f,t)}{\sigma^2(f,t)}e'(f,t)}$	(48)
Smoothing phase:	
$\sqrt{\boldsymbol{R}}^{f,T}_{\overline{\gamma}(f,t)}=\boldsymbol{J}^{\overline{\gamma}(f,t)}_{d(f,t)}\boldsymbol{\Phi}^{f,t-1\,H}\sqrt{\boldsymbol{R}}^{f,T}_{\gamma(f,t)}\boldsymbol{J}^{\overline{\gamma}(f,t)\,H}_{\gamma(f,t)}$ $+\boldsymbol{J}^{\overline{\gamma}(f,t)}_{\gamma(f,t)}\sqrt{\boldsymbol{R}}^{f,T}_{\gamma(f,t)}\boldsymbol{J}^{\overline{\gamma}(f,t)\,H}_{\gamma(f,t)}+\boldsymbol{J}^{\overline{\gamma}(f,t)}_{d(f,t)}\sqrt{\boldsymbol{\Psi}}^{f,t-1}\boldsymbol{J}^{\overline{\gamma}(f,t)\,H}_{d(f,t)}$	(53)
$\sqrt{\boldsymbol{R}}^{f,T}_{\gamma(f,t-1)}=\mathcal{LT}\left(\boldsymbol{J}^{\overline{\gamma}(f,t)\quad H}_{\gamma(f,t-1)}\sqrt{\boldsymbol{R}}^{f,T}_{\overline{\gamma}(f,t)}\right)$	(54)
$\boldsymbol{d}^{f,T}(f,t)=\boldsymbol{\phi}^{f,t-1}+\boldsymbol{\Phi}^{f,t-1\,H}\boldsymbol{\gamma}^{f,T}(f,t)$	(56)
$\overline{\boldsymbol{\gamma}}^{f,T}(f,t)=\boldsymbol{J}^{\overline{\gamma}(f,t)}_{\gamma(f,t)}\boldsymbol{\gamma}^{f,T}(f,t)+\boldsymbol{J}^{\overline{\gamma}(f,t)}_{d(f,t)}\boldsymbol{d}^{f,T}(f,t)$	(55)
$\boldsymbol{\gamma}^{f,T}(f,t-1)=\boldsymbol{J}^{\overline{\gamma}(f,t)\quad H}_{\gamma(f,t-1)}\overline{\boldsymbol{\gamma}}^{f,T}(f,t)$	(57)
$\forall k\in\mathcal{K}(f),\sqrt{\boldsymbol{S}}(k,f,t)=$ $\mathcal{LT}\left(\boldsymbol{J}^{\overline{\gamma}(f,t)\quad H}_{\overline{c}(k,f,t)}\left[\sqrt{\boldsymbol{R}}^{f,T}_{\overline{\gamma}(f,t)},\overline{\boldsymbol{\gamma}}^{f,T}(f,t)\right]^*\right)$	(58)
End for t;	
End for f;	
Outputs: $e(f,t)$, $\sqrt{\boldsymbol{S}}(k,f,t)$, and $L(f,T)$ and $c^{f,T}_k(f,t)$ if wanted	

TABLE II
PSEUDO-CODE OF THE E-STEP

1) Multiplicative update rules: When initializing the EM algorithm, all $P(k,f)$ are first assumed to be zero. In this case, the log-likelihood of the observed mixture $x(f,t)$ defined in equation (1) can be simply written as follows:

$$L \stackrel{\text{c}}{=} -\sum_{f=1}^{F}\sum_{t=1}^{T}\delta(f,t)\left(\ln\left(\sigma^2(f,t)\right)+\frac{|x(f,t)|^2}{\sigma^2(f,t)}\right) \tag{14}$$

where

$$\sigma^2(f,t)=\sigma^2+\sum_{k=1}^{K}w(k,f)h(k,t). \tag{15}$$

Differentiating this log-likelihood w.r.t any parameter $\theta\in\{\sigma^2,w(k,f),h(k,t)\}$ yields

$$\frac{\partial L}{\partial\theta}=-\sum_{f=1}^{F}\sum_{t=1}^{T}\delta(f,t)\left(\frac{1}{\sigma^2(f,t)}-\frac{|x(f,t)|^2}{\sigma^4(f,t)}\right)\frac{\partial\sigma^2(f,t)}{\partial\theta}$$

The standard multiplicative update rule [34] for maximizing L w.r.t. the nonnegative parameter θ can then be written as:

$$\theta\leftarrow\theta\,\frac{\sum_{f=1}^{F}\sum_{t=1}^{T}\delta(f,t)\frac{|x(f,t)|^2}{\sigma^4(f,t)}\frac{\partial\sigma^2(f,t)}{\partial\theta}}{\sum_{f=1}^{F}\sum_{t=1}^{T}\delta(f,t)\frac{1}{\sigma^2(f,t)}\frac{\partial\sigma^2(f,t)}{\partial\theta}}.$$

Since $\frac{\partial\sigma^2(f,t)}{\partial\theta}\geq 0$, this rule guarantees that θ remains nonnegative over the iterations. As proven in Appendix C1, the multiplicative update rules summarized in Table III make the log-likelihood L non-decreasing. Their overall computational complexity is $O(FTK)$.

2) First E-step: After the multiplicative updates, the E-step should be run with the desired values of $P(k,f)$, which implies that $\boldsymbol{\alpha}(k,f)$ are set to zero. However, the E-step as presented in Table II assumes that $\forall k\in\mathcal{K}(f)$, $a(P(k,f),k,f)\neq 0$, which guarantees that matrix $\boldsymbol{A}(f)$ is invertible, so that all matrix inverses and divisions in table II are well-defined. Since this property does not stand at initialization, a specific implementation of the first E-step must be used, which is derived in Appendix C2 and summarized in Table III. Its computational complexity is $O(FTK(1+P)^3)$.

3) Summary of the proposed estimation method: The proposed method for estimating HR-NMF consists of three steps:

1) Initialize all filters to identity, and all nonnegative parameters to random values;
2) Make all non-negative parameters converge and perform the first E-step as in table III;
3) Make the log-likelihood further increase by estimating the filters along with the other parameters, by means of the EM algorithm summarized in tables I and II.

IV. Applications

This section aims to provide a basic proof of principle of HR-NMF. We consider two examples of straightforward applications[11]: audio source separation (section IV-B) and audio inpainting (section IV-C). Indeed, since the E-step determines the a posteriori distribution of the latent components $c_k(f,t)$ given $x(f,t)$, even at time-frequency bins where the observation is missing, this distribution can be used to reconstruct the latent components[12][13]. The test signal is a real piano sound, composed of a C4 tone played alone at $t=0$ ms, and a C3 tone played at $t=680$ ms while the C4 tone is maintained. The sampling frequency is 8600 Hz, and $x(f,t)$ is obtained by computing the STFT of the input signal with $F=400$ and $T=60$, using 90 ms-long Hann windows with 75% overlap (the corresponding spectrogram is plotted in Figure 1).

[11]The Matlab code, as well as the sound files of the various signals computed in these experiments, are available online at http://perso.telecom-paristech.fr/rbadeau/unrestricted/HR-NMF-simulations.zip.

[12]As could be expected, our experiments showed that the a posteriori distribution permits to reconstruct the latent components much more accurately than the a priori distribution.

[13]In the following experiments, $c_k(f,t)$ will be estimated as the posterior mean $c_k^{f,T}(f,t)$, but in future work the second moments of the a posteriori distribution should also be taken into account in order to achieve a more realistic synthesis.

Inputs: $x(f,t)$, $\delta(f,t)$	**Eq.**
Initialize σ^2, $w(k,f)$, $h(k,t)$ to nonnegative random values	
Multiplicative update rules:	
$\forall f,t, \quad \sigma^2(f,t) = \sigma^2 + \sum_{k=1}^{K} w(k,f)h(k,t)$	(15)
Repeat (as many times as wanted)[7]:	
$\sigma^2 = \sigma^2 \dfrac{\sum_{f=1}^{F}\sum_{t=1}^{T} \frac{\delta(f,t)\lvert x(f,t)\rvert^2}{\sigma^4(f,t)}}{\sum_{f=1}^{F}\sum_{t=1}^{T} \frac{\delta(f,t)}{\sigma^2(f,t)}}$	(59)
$\forall k,t, \quad h(k,t) = h(k,t) \dfrac{\sum_{f=1}^{F} w(k,f) \frac{\delta(f,t)\lvert x(f,t)\rvert^2}{\sigma^4(f,t)}}{\sum_{f=1}^{F} w(k,f) \frac{\delta(f,t)}{\sigma^2(f,t)}}$	(60)
$\forall f,t, \quad \sigma^2(f,t) = \sigma^2 + \sum_{k=1}^{K} w(k,f)h(k,t)$	(15)
$\sigma^2 = \sigma^2 \dfrac{\sum_{f=1}^{F}\sum_{t=1}^{T} \frac{\delta(f,t)\lvert x(f,t)\rvert^2}{\sigma^4(f,t)}}{\sum_{f=1}^{F}\sum_{t=1}^{T} \frac{\delta(f,t)}{\sigma^2(f,t)}}$	(59)
$\forall k,f, \quad w(k,f) = w(k,f) \dfrac{\sum_{t=1}^{T} h(k,t) \frac{\delta(f,t)\lvert x(f,t)\rvert^2}{\sigma^4(f,t)}}{\sum_{t=1}^{T} h(k,t) \frac{\delta(f,t)}{\sigma^2(f,t)}}$	(61)
$\forall f,t, \quad \sigma^2(f,t) = \sigma^2 + \sum_{k=1}^{K} w(k,f)h(k,t)$	(15)
$L \overset{\mathrm{C}}{=} -\sum_{f=1}^{F}\sum_{t=1}^{T} \delta(f,t) \left(\ln\left(\sigma^2(f,t)\right) + \frac{\lvert x(f,t)\rvert^2}{\sigma^2(f,t)} \right)$	(14)
Normalization of the NMF: $\forall k$, $H_k = \max_t(h(k,t))$, $w(k,f) = H_k\, w(k,f)$, $h(k,t) = h(k,t)/H_k$	
End repeat;	
$\boldsymbol{a}(k,f) = \boldsymbol{J}_1^{\boldsymbol{a}}$	
First E-step:	
$\forall f,t, \quad e(f,t) = \delta(f,t) \frac{\sigma^2}{\sigma^2(f,t)} \left((\sigma^2(f,t) - \sigma^2) + \frac{\sigma^2 \lvert x(f,t)\rvert^2}{\sigma^2(f,t)} \right)$	(62)
$\forall k,f,t, \quad c_k^{f,T}(f,t) = \delta(f,t) \frac{v_k(f,t)}{\sigma^2(f,t)} x(f,t)$	(63)
$\forall k,f,t, \quad \sqrt{R}_{c_k(f,t)}^{f,T} = \sqrt{\frac{v_k(f,t)}{\sigma^2(f,t)}} \sqrt{\sigma^2(f,t) - \delta(f,t) v_k(f,t)}$	(64)
$\forall k,f,t, \quad \sqrt{\boldsymbol{R}}_{\overline{c}(k,f,t)}^{f,T} = \mathrm{diag}\left(\sqrt{R}_{c_k(f,t)}^{f,T} \dots \sqrt{R}_{c_k(f,t-P(k,f))}^{f,T} \right)$	(65)
$\forall k,f,t, \quad \sqrt{\boldsymbol{S}}(k,f,t) = \mathcal{LT}\left(\left[\sqrt{\boldsymbol{R}}_{\overline{c}(k,f,t)}^{f,T}, \overline{\boldsymbol{c}}^{f,T}(k,f,t) \right]^* \right)$	(66)
Outputs: σ^2, $\boldsymbol{a}(k,f)$, $w(k,f)$, $h(k,t)$, $e(f,t)$, $\sqrt{\boldsymbol{S}}(k,f,t)$.	

TABLE III
PSEUDO-CODE OF THE INITIALIZATION METHOD

A. Monotonicity of the log-likelihood and computation time

Before processing this mixture piano sound, our first experiment consists of learning the spectral parameters $w(k,f)$ and $\boldsymbol{a}(k,f)$ from the fully observed STFT ($\delta(f,t) = 1$) of the first 680 ms of the two isolated tones (in the case of C4, this corresponds to the first half of the STFT represented in Figure 1). Each piano tone is represented by a HR-NMF model of order $K = 1$, involving autoregressive filters of order $P(k,f) = 2$, which permit to model the beating in the partials of piano strings. The two HR-NMF models are thus estimated by running 30 iterations of the multiplicative update rules in Table III, and 10 iterations of the EM algorithm in Tables I and II. In order to make a comparison, the IS-NMF models of the two tones (with $K = 1$, $\sigma^2 = 0$ and $P(k,f) = 0$) are estimated by running 30 iterations of the standard multiplicative update rules [8] (defined by equations (15), (60) and (61) in Table III).

Figure 2-(a) represents the log-likelihood of the estimated C4 models as a function of the iteration number. The black dash-dotted line corresponds to the IS-NMF model: as expected, the multiplicative update rules make the log-likelihood (defined in equations (14) and (15) with $\sigma^2 = 0$) non-decreasing. The red solid line corresponds to the initialization of the HR-NMF model in Table III: as proven in Appendix C1, the multiplicative update rules make the log-likelihood non-decreasing again. Finally, the blue dashed line corresponds to the EM estimation of the HR-NMF model: as explained in section III-D3, the EM algorithm makes the log-likelihood (computed in equation (40) in Table II) further increase.

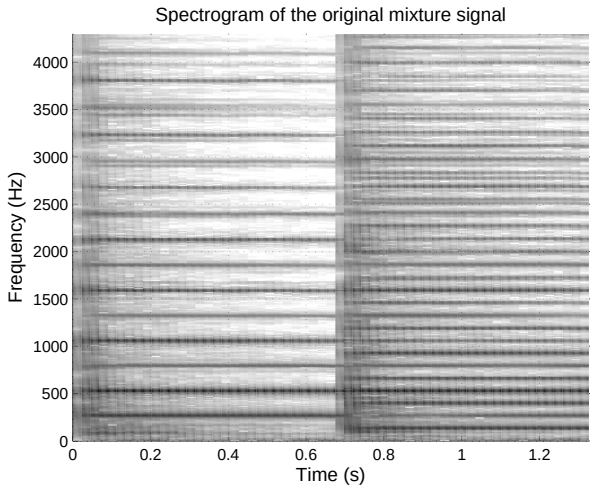

Fig. 1. Spectrogram of the input piano sound

Figure 2-(b) represents the same log-likelihoods as functions of the elapsed time[14]. It can be observed that one iteration of the EM algorithm is much more time consuming than one iteration of the multiplicative update rules. Thus the initialization with multiplicative updates permits to reduce the overall computational time.

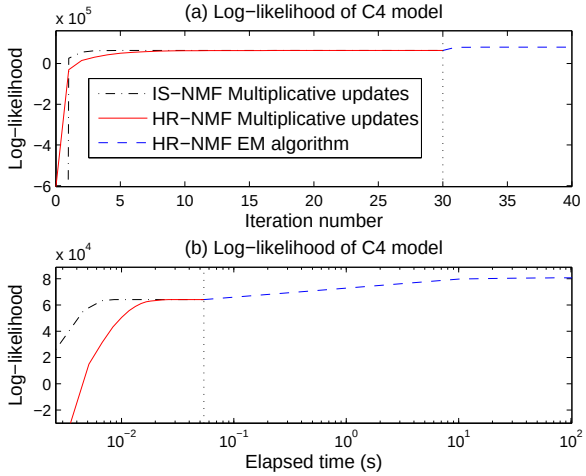

Fig. 2. Monotonicity of the log-likelihood

B. Source separation

Here the observation is the whole STFT $x(f,t)$ represented in Figure 1 ($\delta(f,t)=1$), and the objective is to separate $K=2$ components $c_k(f,t)$. The spectral parameters $w(k,f)$ and $\boldsymbol{a}(k,f)$ are learned as explained in section IV-A, and parameters $h(k,t)$ and σ^2 have to be estimated from the mixture. Again, an IS-NMF model (with $P(k,f)=0$ and $\sigma^2=0$) is estimated by running 30 iterations of the multiplicative update rules defined by equations (15) and (60) in Table III. The HR-NMF model (with $P(k,f)=2$) is estimated by running 60 iterations of the EM algorithm, initialized with the temporal activations $h(k,t)$ of the IS-NMF model.

Figure 3 focuses on the results obtained in the frequency band f which corresponds to the second harmonic of C4 and to the fourth harmonic of C3 (around 540 Hz). These two sinusoidal components (whose real parts

[14]This experiment was performed with Matlab(R) 7.10 64-bit, run in a Windows 7 system with 2.66 GHz Intel(R) Xeon(R) CPU and 6 Go RAM.

are represented as red solid lines) have very close frequencies, which makes them hardly separable. As expected, IS-NMF does not properly separate the components when they overlap, from $t = 680$ ms to 1.36 s: the observed mixture signal is wrongly fully assigned to the second component (the estimated components are represented as black dash-dotted lines). As a comparison, the components estimated by HR-NMF (blue dashed lines) better fit the ground truth.

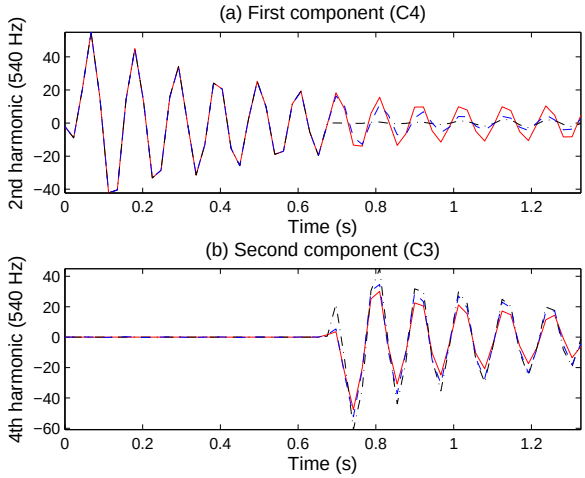

Fig. 3. Separation of two sinusoidal components. The real parts of the two components are plotted as red solid lines, their IS-NMF estimates are plotted as black dash-dotted lines, and their HR-NMF estimates are plotted as blue dashed lines.

C. Audio inpainting

In this last experiment, the second part of the STFT (from $t = 680$ ms to 1.36 s) is unobserved ($\delta(f,t) = 0$), and in the first part (from $t = 0$ ms to $t = 680$ ms), only 50% of the TF coefficients $x(f,t)$ are randomly observed. Since the second tone is completely unobserved, our purpose is to recover $K = 1$ component. Again, the spectral parameters $w(k,f)$ and $\boldsymbol{a}(k,f)$ are learned as explained in section IV-A, and parameters $h(k,t)$ and σ^2 have to be estimated from the observations. The IS-NMF model (with $P(k,f) = 0$ and $\sigma^2 = 0$) is estimated by running 10 iterations of the multiplicative update rules[15] defined by equations (15) and (60) in Table III. The HR-NMF model (with $P(k,f) = 2$) is estimated by running 10 iterations of the EM algorithm, initialized with the temporal activations $h(k,t)$ of the IS-NMF model. Figure 4 shows that the C4 tone is correctly reconstructed with the HR-NMF model[16]. Moreover, the noise in the unobserved part has been removed. As a comparison, IS-NMF is not appropriate for audio inpainting, because it does not take the correlations between contiguous TF coefficients into account: the missing coefficients are estimated as their posterior mean, which is zero.

V. Conclusions

In this paper, we presented a new method for modeling mixtures of non-stationary signals in the time-frequency domain. The HR-NMF model was introduced and an expectation-maximization algorithm was designed for estimating its parameters. This technique was successfully applied to source separation and audio inpainting. Compared to standard IS-NMF, the proposed approach natively takes both phases and local correlations in each frequency band into account. It was shown that it achieves high resolution, which means that two sinusoids of different frequencies can be properly separated within the same frequency band[17]. Besides, HR-NMF is also suitable for modeling stationary and non-stationary noise.

[15]In the second part of the sound, parameters $h(k,t)$ cannot be estimated since there is no observation; there are thus set to zero.

[16]A listening test did not permit to perceive any artifact in the signal reconstructed from the estimated TF component by a standard overlap-add technique.

[17]Note that contrary to standard high resolution methods, the proposed approach is able to handle mixtures of amplitude-modulated sinusoids starting at different times; it also performs the clustering of these sinusoids into several components according to their temporal dynamics.

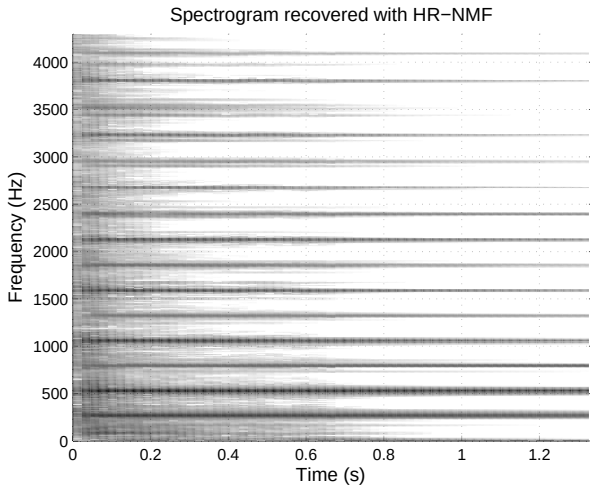

Fig. 4. Recovery of the full C4 piano tone

In future work, this approach could be transposed into a Bayesian framework, by applying some prior distributions to the model parameters, in order to enforce some desirable properties such as harmonicity, sparsity, or smoothness. Besides, the basic NMF that has been used for modeling the non-stationarities in the distribution of $b_k(f,t)$ could be replaced by any non-stationary parametric model, such as one of the many variants of NMF. The model could also be extended in several ways, for instance by taking the correlations across frequencies and/or across components into account, or by representing multichannel signals. Since the proposed EM algorithm is time consuming when processing realistic data without using parallel computing, alternative methods with better computational complexity and convergence properties should be developed for estimating the model, for instance by using conjugate gradient or modified Newton-Raphson algorithms, or by introducing approximations such as those used in variational Bayesian inference.

The possible applications of this work are numerous: beyond source separation and audio inpainting, all usual applications of NMF and PLCA can be considered, such as multi-pitch estimation and automatic music transcription. Besides, we plan to address new applications, such as the physical analysis of impact sounds involving a mixture of damped sinusoids and non-stationary noise, or the development of a new hybrid audio coder, in-between transform coding and parametric coding.

APPENDIX

A. Mathematical derivations of the M-step

1) Maximization of Q_0 w.r.t. σ^2: The maximization of Q_0 defined in equation (3) w.r.t. σ^2 leads to

$$\sigma^2 = \sum_{f=1}^{F}\sum_{t=1}^{T} e(f,t) \Big/ \sum_{f=1}^{F}\sum_{t=1}^{T} \delta(f,t). \tag{16}$$

2) Maximization of Q_k w.r.t. $h(k,t)$, $w(k,f)$, $\boldsymbol{a}(k,f,t)$: The global maximization of Q_k defined in equation (4) does not admit a closed form solution. Thus an iterative algorithm is proposed below, which recursively maximizes Q_k w.r.t. $h(k,t)$ and w.r.t. $(w(k,f), \boldsymbol{a}(k,f,t))$.

First, the maximization of Q_k w.r.t. $h(k,f)$ leads to

$$h(k,t) = \frac{1}{F}\sum_{f=1}^{F} \frac{\boldsymbol{a}(k,f)^H \boldsymbol{S}(k,f,t)\, \boldsymbol{a}(k,f)}{w(k,f)},$$

which can be rewritten in the following square root form:

$$h(k,t) = \sum_{f=1}^{F} \left(\frac{\|\sqrt{\boldsymbol{S}}(k,f,t)^H \boldsymbol{a}(k,f)\|}{\sqrt{F w(k,f)}} \right)^2. \tag{17}$$

Then the maximization of Q_k w.r.t. $w(k,f)$ leads to

$$w(k,f) = \boldsymbol{a}(k,f)^H \boldsymbol{\Sigma}(k,f)\, \boldsymbol{a}(k,f),$$

where $\boldsymbol{\Sigma}(k,f) = \frac{1}{T}\sum_{t=1}^{T} \frac{\boldsymbol{S}(k,f,t)}{h(k,t)}$, which can be rewritten in the following square root form:

$$w(k,f) = \|\sqrt{\boldsymbol{\Sigma}}(k,f)^H \boldsymbol{a}(k,f)\|^2, \tag{18}$$

where

$$\sqrt{\boldsymbol{\Sigma}}(k,f) = \mathcal{LT}\left(\left[\frac{\sqrt{\boldsymbol{S}}(k,f,1)}{\sqrt{Th(k,1)}}, \ldots, \frac{\sqrt{\boldsymbol{S}}(k,f,T)}{\sqrt{Th(k,T)}}\right]\right). \tag{19}$$

Finally, the maximization of Q_k w.r.t. $\boldsymbol{a}(k,f)$ is equivalent to the minimization of $\|\sqrt{\boldsymbol{\Sigma}}(k,f)^H \boldsymbol{a}(k,f)\|^2$ under the constraint that the first coefficient of $\boldsymbol{a}(k,f)$ be equal to 1. The solution to this quadratic programming problem is

$$\boldsymbol{a}(k,f) = \left[1, -\left(\boldsymbol{J}_1^{\boldsymbol{a}\,H}\sqrt{\boldsymbol{\Sigma}}(k,f)\right)\left(\boldsymbol{J}_{\boldsymbol{\alpha}}^{\boldsymbol{a}\,H}\sqrt{\boldsymbol{\Sigma}}(k,f)\right)^{\dagger}\right]^H. \tag{20}$$

Like in most NMF algorithms, the NMF factors $w(k,f)$ and $h(k,t)$ are normalized at the end of the M-step in table I, in order to prevent any possible numerical instability (this normalization does not affect the resulting variances $v_k(f,t)$).

B. Mathematical derivations of the E-step

Note that the E-step in Table II consists of two passes:

- the *forward pass* computes $\boldsymbol{\gamma}^{f,t}(f,t)$ and $\sqrt{\boldsymbol{R}}^{f,t}_{\boldsymbol{\gamma}(f,t)}$ for $t = 1$ to T;
- the *backward pass* computes $\boldsymbol{\gamma}^{f,T}(f,t)$ and $\sqrt{\boldsymbol{R}}^{f,T}_{\boldsymbol{\gamma}(f,t)}$ for $t = T$ downto 1.

1) Forward pass: The forward pass (for $t = 1$ to T) consists of two phases:

- the *predict phase* computes $\boldsymbol{\gamma}^{f,t-1}(f,t)$ and $\sqrt{\boldsymbol{R}}^{f,t-1}_{\boldsymbol{\gamma}(f,t)}$;
- the *update phase* computes $\boldsymbol{\gamma}^{f,t}(f,t)$ and $\sqrt{\boldsymbol{R}}^{f,t}_{\boldsymbol{\gamma}(f,t)}$.

a) Predict phase: As will be explained in section B2b, in order to later perform the backward pass, it is necessary to compute the following matrices and vector in the forward pass:

$$\boldsymbol{\Phi}^{f,t-1} = \left(\boldsymbol{R}^{f,t-1}_{\boldsymbol{\gamma}(f,t)}\right)^{-1} \boldsymbol{R}^{f,t-1}_{\boldsymbol{\gamma}(f,t),d(f,t)}, \tag{21}$$

$$\boldsymbol{\Psi}^{f,t-1} = \boldsymbol{R}^{f,t-1}_{d(f,t)} - \boldsymbol{\Phi}^{f,t-1\,H} \boldsymbol{R}^{f,t-1}_{\boldsymbol{\gamma}(f,t),d(f,t)}, \tag{22}$$

$$\boldsymbol{\phi}^{f,t-1} = \boldsymbol{d}^{f,t-1}(f,t) - \boldsymbol{\Phi}^{f,t-1\,H} \boldsymbol{\gamma}^{f,t-1}(f,t). \tag{23}$$

Taking the expectation of (9) given $x(f,1)\ldots x(f,t-1)$ yields

$$\boldsymbol{\gamma}^{f,t-1}(f,t) = \boldsymbol{A}(f)\, \boldsymbol{\gamma}^{f,t-1}(f,t-1). \tag{24}$$

Then subtracting equation (24) to equation (9) yields

$$\widetilde{\boldsymbol{\gamma}}^{f,t-1}(f,t) = \boldsymbol{A}(f)\, \widetilde{\boldsymbol{\gamma}}^{f,t-1}(f,t-1) + \boldsymbol{b}'(f,t). \tag{25}$$

Note that $\boldsymbol{d}(f,t) = \boldsymbol{J}^{\boldsymbol{\gamma}(f,t-1)\,H}_{d(f,t)} \boldsymbol{\gamma}(f,t-1)$, thus

$$\boldsymbol{d}^{f,t-1}(f,t) = \boldsymbol{J}^{\boldsymbol{\gamma}(f,t-1)\,H}_{d(f,t)} \boldsymbol{\gamma}^{f,t-1}(f,t-1), \tag{26}$$

$$\widetilde{\boldsymbol{d}}^{f,t-1}(f,t) = \boldsymbol{J}^{\boldsymbol{\gamma}(f,t-1)\,H}_{d(f,t)} \widetilde{\boldsymbol{\gamma}}^{f,t-1}(f,t-1). \tag{27}$$

Equations (11), (25) and (27) yield

$$\widetilde{\overline{\boldsymbol{\gamma}}}^{f,t-1}(f,t) = \overline{\boldsymbol{A}}(f) \widetilde{\overline{\boldsymbol{\gamma}}}^{f,t-1}(f,t-1) + \boldsymbol{J}^{\overline{\boldsymbol{\gamma}}(f,t)}_{c(f,t)} \boldsymbol{b}(f,t)$$

where

$$\overline{\boldsymbol{A}}(f) = \boldsymbol{J}^{\overline{\gamma}(f,t)}_{\gamma(f,t)} \boldsymbol{A}(f) + \boldsymbol{J}^{\overline{\gamma}(f,t)}_{d(f,t)} \boldsymbol{J}^{\gamma(f,t-1)^H}_{d(f,t)}. \tag{28}$$

Taking the expectation of $\widetilde{\overline{\boldsymbol{\gamma}}}^{f,t-1}(f,t)\,\widetilde{\overline{\boldsymbol{\gamma}}}^{f,t-1}(f,t)^H$ given $x(f,1)\dots x(f,t-1)$ finally yields

$$\boldsymbol{R}^{f,t-1}_{\overline{\gamma}(f,t)} = \overline{\boldsymbol{A}}(f) \boldsymbol{R}^{f,t-1}_{\gamma(f,t-1)} \overline{\boldsymbol{A}}(f)^H + \boldsymbol{J}^{\overline{\gamma}(f,t)}_{c(f,t)} \boldsymbol{R}_{v(f,t)} \boldsymbol{J}^{\overline{\gamma}(f,t)^H}_{c(f,t)},$$

which can be rewritten in the following square root form:

$$\sqrt{\boldsymbol{R}}^{f,t-1}_{\overline{\gamma}(f,t)} = \mathcal{LT}\left(\left[\overline{\boldsymbol{A}}(f)\sqrt{\boldsymbol{R}}^{f,t-1}_{\gamma(f,t-1)}, \boldsymbol{J}^{\overline{\gamma}(f,t)}_{c(f,t)}\sqrt{\boldsymbol{R}}_{v(f,t)}\right]\right). \tag{29}$$

Besides, note that $\boldsymbol{\Phi}^{f,t-1}$ as defined in equation (21) is the solution to the following quadratic programming problem:

$$\boldsymbol{\Phi}^{f,t-1} = \underset{\boldsymbol{\Phi}}{\operatorname{argmin}} \left\| \sqrt{\boldsymbol{R}}^{f,t-1\,H}_{\overline{\gamma}(f,t)} \left(\boldsymbol{J}^{\overline{\gamma}(f,t)}_{d(f,t)} - \boldsymbol{J}^{\overline{\gamma}(f,t)}_{\gamma(f,t)} \boldsymbol{\Phi} \right) \right\|^2.$$

Therefore $\boldsymbol{\Phi}^{f,t-1}$ can be rewritten in square root form:

$$\boldsymbol{\Phi}^{f,t-1} = \left(\sqrt{\boldsymbol{R}}^{f,t-1\,H}_{\overline{\gamma}(f,t)} \boldsymbol{J}^{\overline{\gamma}(f,t)}_{\gamma(f,t)} \right)^{\dagger} \left(\sqrt{\boldsymbol{R}}^{f,t-1\,H}_{\overline{\gamma}(f,t)} \boldsymbol{J}^{\overline{\gamma}(f,t)}_{d(f,t)} \right). \tag{30}$$

Finally, note that $\sqrt{\boldsymbol{R}}^{f,t-1}_{\gamma(f,t)}$ and $\boldsymbol{\Psi}^{f,t-1}$ as defined in equation (22) can also be rewritten in square root form:

$$\sqrt{\boldsymbol{R}}^{f,t-1}_{\gamma(f,t)} = \mathcal{LT}\left(\boldsymbol{J}^{\overline{\gamma}(f,t)^H}_{\gamma(f,t)} \sqrt{\boldsymbol{R}}^{f,t-1}_{\overline{\gamma}(f,t)} \right), \tag{31}$$

$$\sqrt{\boldsymbol{\Psi}}^{f,t-1} = \mathcal{LT}\left(\left(\boldsymbol{J}^{\overline{\gamma}(f,t)}_{d(f,t)} - \boldsymbol{J}^{\overline{\gamma}(f,t)}_{\gamma(f,t)} \boldsymbol{\Phi}^{f,t-1} \right)^H \sqrt{\boldsymbol{R}}^{f,t-1}_{\overline{\gamma}(f,t)} \right). \tag{32}$$

b) Update phase: Whether $x(f,t)$ be observed or not,

$$\begin{array}{rl} & \ln(p(\boldsymbol{\gamma}(f,t)/x(f,1)\dots x(f,t))) \\ = & \ln(p(\boldsymbol{\gamma}(f,t)/x(f,1)\dots x(f,t-1))) \\ & +\delta(f,t)\ln(p(x(f,t)/\boldsymbol{\gamma}(f,t))) \\ & -\delta(f,t)\ln(p(x(f,t)/x(f,1)\dots x(f,t-1))). \end{array} \tag{33}$$

Identifying the quadratic terms in $\boldsymbol{\gamma}(f,t)$ in equation (33) and substituting equation (10) leads to the precision matrix $\boldsymbol{Q}^{f,t}_{\gamma(f,t)}$ (defined as the inverse of the covariance matrix $\boldsymbol{R}^{f,t}_{\gamma(f,t)}$):

$$\boldsymbol{Q}^{f,t}_{\gamma(f,t)} = \boldsymbol{Q}^{f,t-1}_{\gamma(f,t)} + \delta(f,t)\frac{\boldsymbol{u}(f)\,\boldsymbol{u}(f)^H}{\sigma^2(f,t)}.$$

Then the matrix inversion lemma [35, pp. 18-19] yields

$$\boldsymbol{R}^{f,t}_{\gamma(f,t)} = \boldsymbol{R}^{f,t-1}_{\gamma(f,t)} - \delta(f,t)\frac{\boldsymbol{R}^{f,t-1}_{\gamma(f,t)}\,\boldsymbol{u}(f)\boldsymbol{u}(f)^H\boldsymbol{R}^{f,t-1}_{\gamma(f,t)}}{\sigma^2(f,t) + \boldsymbol{u}(f)^H\boldsymbol{R}^{f,t-1}_{\gamma(f,t)}\,\boldsymbol{u}(f)}$$

which can be rewritten in the following square root form:

$$\sqrt{\boldsymbol{R}}^{f,t}_{\gamma(f,t)} = \sqrt{\boldsymbol{R}}^{f,t-1}_{\gamma(f,t)} - \boldsymbol{\lambda}(f,t)\frac{\boldsymbol{\mu}(f,t)^H}{1+\sqrt{\frac{\sigma^2(f,t)}{\varepsilon(f,t)}}} \tag{34}$$

where

$$\boldsymbol{\mu}(f,t) = (\sqrt{\boldsymbol{R}}^{f,t-1}_{\gamma(f,t)})^H\boldsymbol{u}(f) \tag{35}$$

$$\varepsilon(f,t) = \sigma^2(f,t) + \|\boldsymbol{\mu}(f,t)\|^2 \tag{36}$$

$$\boldsymbol{\lambda}(f,t) = \frac{\delta(f,t)}{\varepsilon(f,t)}\sqrt{\boldsymbol{R}}^{f,t-1}_{\gamma(f,t)}\boldsymbol{\mu}(f,t). \tag{37}$$

Then identifying the linear terms in $\boldsymbol{\gamma}(f,t)$ in (33) yields

$$\boldsymbol{\gamma}^{f,t}(f,t) = \boldsymbol{\gamma}^{f,t-1}(f,t) + \boldsymbol{\lambda}(f,t)\epsilon^{f,t}(f,t) \tag{38}$$

where

$$\epsilon^{f,t}(f,t) = x(f,t) - \boldsymbol{u}(f)^H \boldsymbol{\gamma}^{f,t-1}(f,t). \tag{39}$$

Finally, identifying the terms which do not depend on $\boldsymbol{\gamma}(f,t)$ in equation (33) shows that if $x(f,t)$ is observed,

$$x(f,t)/x(f,1)\dots x(f,t-1) \sim \mathcal{N}_{\mathbb{F}}\left(\boldsymbol{u}(f)^H \boldsymbol{\gamma}^{f,t-1}(f,t), \varepsilon(f,t)\right).$$

This permits to recursively evaluate the log-likelihood $L(f,t) = \ln(p(x(f,1)\dots x(f,t)))$ of the observed data:

$$L(f,t) \overset{\text{c}}{=} L(f,t-1) - \delta(f,t)\left(\ln(\varepsilon(f,t)) + \frac{|\epsilon^{f,t}(f,t)|^2}{\varepsilon(f,t)}\right). \tag{40}$$

2) Backward pass: The backward pass (for $t = T$ downto 1) consists of two phases:

- the *Wiener filtering phase* calculates $\overline{\boldsymbol{c}}^{f,T}(f,t)$ and $\sqrt{\boldsymbol{R}}^{f,T}_{\overline{c}(f,t)}$ (where $\overline{\boldsymbol{c}}(f,t) = [\boldsymbol{c}(f,t); \boldsymbol{c}'(f,t)]$);
- the *smoothing phase* computes $\boldsymbol{\gamma}^{f,T}(f,t-1)$ and $\sqrt{\boldsymbol{R}}^{f,T}_{\gamma(f,t-1)}$.

a) Wiener filtering phase: Note that, whether $x(f,t)$ be observed or not,

$$\begin{array}{rl} & \ln(p(\boldsymbol{c}(f,t), \boldsymbol{c}'(f,t)/x(f,1)\dots x(f,T))) \\ = & \ln(p(\boldsymbol{c}(f,t)/x(f,1)\dots x(f,T))) \\ & +\ln(p(\boldsymbol{c}'(f,t)/\boldsymbol{c}(f,t), x(f,t))) \\ = & \ln(p(\boldsymbol{c}(f,t)/x(f,1)\dots x(f,T))) + \ln(p(\boldsymbol{c}'(f,t))) \\ & +\delta(f,t)\ln(p(x(f,t)/\boldsymbol{c}'(f,t), \boldsymbol{c}(f,t))) \\ & -\delta(f,t)\ln(p(x(f,t)/\boldsymbol{c}(f,t))). \end{array} \tag{41}$$

Identifying the quadratic terms in $\overline{\boldsymbol{c}}(f,t)$ in equation (41) leads to the precision matrix $\boldsymbol{Q}^{f,T}_{\overline{c}(f,t)}$:

$$\begin{array}{l} \boldsymbol{Q}^{f,T}_{\overline{c}(f,t)} = \operatorname{diag}\left(\boldsymbol{Q}^{f,T}_{c(f,t)}, \operatorname{diag}(\boldsymbol{v}'(f,t))^{-1}\right) \\ +\delta(f,t)\begin{bmatrix} \mathbf{1} & \mathbf{1} \\ \mathbf{1} & \mathbf{0} \end{bmatrix} \operatorname{diag}\left(\frac{1}{\sigma^2}, \frac{1}{\sigma^2(f,t)}\right) \begin{bmatrix} \mathbf{1} & \mathbf{1} \\ \mathbf{1} & \mathbf{0} \end{bmatrix}^H. \end{array}$$

Applying the matrix inversion lemma [35, pp. 18-19] yields

$$\begin{array}{l} \boldsymbol{R}^{f,T}_{\overline{c}(f,t)} = \operatorname{diag}\left(\boldsymbol{R}^{f,T}_{c(f,t)}, \operatorname{diag}(\boldsymbol{v}'(f,t))\right) - \frac{\delta(f,t)}{\sigma^2(f,t)} \times \\ \begin{bmatrix} \mathbf{0} & (\boldsymbol{R}^{f,T}_{c(f,t)}\mathbf{1})\boldsymbol{v}'(f,t)^H \\ \boldsymbol{v}'(f,t)(\boldsymbol{R}^{f,T}_{c(f,t)}\mathbf{1})^H & \left(1 - \frac{\mathbf{1}^H \boldsymbol{R}^{f,T}_{c(f,t)}\mathbf{1}}{\sigma^2(f,t)}\right)\boldsymbol{v}'(f,t)\boldsymbol{v}'(f,t)^H \end{bmatrix} \end{array}$$

Identifying the linear terms in $\overline{\boldsymbol{c}}(f,t)$ in equation (41) yields

$$\boldsymbol{c}'^{f,T}(f,t) = \frac{\delta(f,t)}{\sigma^2(f,t)}\boldsymbol{v}'(f,t)\,\epsilon^{f,T}(f,t), \tag{42}$$

where

$$\epsilon^{f,T}(f,t) = x(f,t) - \boldsymbol{u}(f)^H \boldsymbol{\gamma}^{f,T}(f,t). \tag{43}$$

Then $e(f,t)$ introduced in equation (5) is obtained as

$$e(f,t) = \delta(f,t)\,\mathbb{E}_{/x}\left[\left|x(f,t) - \mathbf{1}^H \boldsymbol{c}(f,t) - \mathbf{1}^H \boldsymbol{c}'(f,t)\right|^2\right] \tag{44}$$

$$= \delta(f,t)\frac{\sigma^2}{\sigma^2(f,t)}\left((\sigma^2(f,t) - \sigma^2) + \frac{\sigma^2}{\sigma^2(f,t)}e'(f,t)\right) \tag{45}$$

where

$$e'(f,t) = \delta(f,t)\left(|\epsilon^{f,T}(f,t)|^2 + \|\boldsymbol{\mu}'(f,t)\|^2\right), \tag{46}$$

$$\boldsymbol{\mu}'(f,t) = (\sqrt{\boldsymbol{R}}^{f,T}_{\gamma(f,t)})^H \boldsymbol{u}(f), \tag{47}$$

and $\forall k \notin \mathcal{K}$, $\sqrt{\boldsymbol{S}}(k,f,t)$ in (8) is obtained as

$$\begin{aligned}\sqrt{\boldsymbol{S}}(k,f,t) &= \sqrt{\mathbb{E}_{/x}\left[|c_k(f,t)|^2\right]} \\ &= \sqrt{\tfrac{v_k(f,t)}{\sigma^2(f,t)}}\sqrt{(\sigma^2(f,t) - \delta(f,t)v_k(f,t)) + \tfrac{v_k(f,t)}{\sigma^2(f,t)}e'(f,t)}.\end{aligned} \tag{48}$$

b) Smoothing phase: Note that

$$\begin{aligned} & \ln(p(\overline{\boldsymbol{\gamma}}(f,t)/x(f,1)\dots x(f,T))) \\ = \; & \ln\left(p(\boldsymbol{d}(f,t)/\boldsymbol{\gamma}(f,t), x(f,1)\dots x(f,t-1))\right) \\ & + \ln\left(p(\boldsymbol{\gamma}(f,t)/x(f,1)\dots x(f,T))\right) \\ = \; & \ln\left(p(\overline{\boldsymbol{\gamma}}(f,t)/x(f,1)\dots x(f,t-1))\right) \\ & - \ln\left(p(\boldsymbol{\gamma}(f,t)/x(f,1)\dots x(f,t-1))\right) \\ & + \ln\left(p(\boldsymbol{\gamma}(f,t)/x(f,1)\dots x(f,T))\right). \end{aligned} \tag{49}$$

Identifying the quadratic terms in $\overline{\boldsymbol{\gamma}}(f,t)$ in equation (49) leads to the precision matrix $\boldsymbol{Q}^{f,T}_{\overline{\gamma}(f,t)}$:

$$\boldsymbol{Q}^{f,T}_{\overline{\gamma}(f,t)} = \boldsymbol{Q}^{f,t-1}_{\overline{\gamma}(f,t)} + \boldsymbol{J}^{\overline{\gamma}(f,t)}_{\gamma(f,t)}\left(\boldsymbol{Q}^{f,T}_{\gamma(f,t)} - \boldsymbol{Q}^{f,t-1}_{\gamma(f,t)}\right)\boldsymbol{J}^{\overline{\gamma}(f,t)^H}_{\gamma(f,t)}.$$

Then $\boldsymbol{R}^{f,T}_{\overline{\gamma}(f,t)}$ can be updated from $\boldsymbol{R}^{f,T}_{\gamma(f,t)}$, according to

$$\begin{aligned}\boldsymbol{R}^{f,T}_{\overline{\gamma}(f,t)} &= \boldsymbol{J}^{\overline{\gamma}(f,t)}_{\gamma(f,t)}\boldsymbol{R}^{f,T}_{\gamma(f,t)}\boldsymbol{J}^{\overline{\gamma}(f,t)^H}_{\gamma(f,t)} + \boldsymbol{J}^{\overline{\gamma}(f,t)}_{d(f,t)}\boldsymbol{R}^{f,T}_{d(f,t)}\boldsymbol{J}^{\overline{\gamma}(f,t)^H}_{d(f,t)} \\ &+\boldsymbol{J}^{\overline{\gamma}(f,t)}_{\gamma(f,t)}\boldsymbol{R}^{f,T}_{\gamma(f,t),d(f,t)}\boldsymbol{J}^{\overline{\gamma}(f,t)^H}_{d(f,t)} + \boldsymbol{J}^{\overline{\gamma}(f,t)}_{d(f,t)}\boldsymbol{R}^{f,T}_{d(f,t),\gamma(f,t)}\boldsymbol{J}^{\overline{\gamma}(f,t)^H}_{\gamma(f,t)}\end{aligned} \tag{50}$$

where

$$\boldsymbol{R}^{f,T}_{\gamma(f,t),d(f,t)} = \boldsymbol{R}^{f,T}_{\gamma(f,t)}\,\boldsymbol{\Phi}^{f,t-1}, \tag{51}$$

$$\boldsymbol{R}^{f,T}_{d(f,t)} = \boldsymbol{\Psi}^{f,t-1} + \boldsymbol{R}^{f,T}_{d(f,t),\gamma(f,t)}\boldsymbol{\Phi}^{f,t-1}, \tag{52}$$

and matrices $\boldsymbol{\Phi}^{t-1}$ and $\boldsymbol{\Psi}^{t-1}$ were defined in equations (21) and (22). Equation (50) can be rewritten in the following square root form:

$$\begin{aligned}\sqrt{\boldsymbol{R}}^{f,T}_{\overline{\gamma}(f,t)} &= \boldsymbol{J}^{\overline{\gamma}(f,t)}_{d(f,t)}\boldsymbol{\Phi}^{f,t-1^H}\sqrt{\boldsymbol{R}}^{f,T}_{\gamma(f,t)}\boldsymbol{J}^{\overline{\gamma}(f,t)^H}_{\gamma(f,t)} \\ &+\boldsymbol{J}^{\overline{\gamma}(f,t)}_{\gamma(f,t)}\sqrt{\boldsymbol{R}}^{f,T}_{\gamma(f,t)}\boldsymbol{J}^{\overline{\gamma}(f,t)^H}_{\gamma(f,t)} + \boldsymbol{J}^{\overline{\gamma}(f,t)}_{d(f,t)}\sqrt{\boldsymbol{\Psi}}^{f,t-1}\boldsymbol{J}^{\overline{\gamma}(f,t)^H}_{d(f,t)}.\end{aligned} \tag{53}$$

Then $\boldsymbol{R}^{f,T}_{\gamma(f,t-1)}$ is extracted from $\boldsymbol{R}^{f,T}_{\overline{\gamma}(f,t)}$ as

$$\boldsymbol{R}^{f,T}_{\gamma(f,t-1)} = \boldsymbol{J}^{\overline{\gamma}(f,t)}_{\gamma(f,t-1)}{}^H\boldsymbol{R}^{f,T}_{\overline{\gamma}(f,t)}\boldsymbol{J}^{\overline{\gamma}(f,t)}_{\gamma(f,t-1)},$$

which can be rewritten in the following square root form:

$$\sqrt{\boldsymbol{R}}^{f,T}_{\gamma(f,t-1)} = \mathcal{LT}\left(\boldsymbol{J}^{\overline{\gamma}(f,t)}_{\gamma(f,t-1)}{}^H\sqrt{\boldsymbol{R}}^{f,T}_{\overline{\gamma}(f,t)}\right). \tag{54}$$

Similarly, identifying the linear terms in $\overline{\boldsymbol{\gamma}}(f,t)$ in (49), $\overline{\boldsymbol{\gamma}}^{f,T}(f,t)$ can be updated from $\boldsymbol{\gamma}^{f,T}(f,t)$ according to

$$\overline{\boldsymbol{\gamma}}^{f,T}(f,t) = \boldsymbol{J}^{\overline{\gamma}(f,t)}_{\gamma(f,t)}\boldsymbol{\gamma}^{f,T}(f,t) + \boldsymbol{J}^{\overline{\gamma}(f,t)}_{d(f,t)}\boldsymbol{d}^{f,T}(f,t) \tag{55}$$

where

$$\boldsymbol{d}^{f,T}(f,t) = \boldsymbol{\phi}^{f,t-1} + \boldsymbol{\Phi}^{f,t-1^H}\boldsymbol{\gamma}^{f,T}(f,t), \tag{56}$$

and vector $\boldsymbol{\phi}^{f,t-1}$ was defined in equation (23).

Then $\boldsymbol{\gamma}^{f,T}(f,t-1)$ is extracted from $\overline{\boldsymbol{\gamma}}^{f,T}(f,t)$ as

$$\boldsymbol{\gamma}^{f,T}(f,t-1) = \boldsymbol{J}^{\overline{\gamma}(f,t)}_{\gamma(f,t-1)}{}^H\overline{\boldsymbol{\gamma}}^{f,T}(f,t). \tag{57}$$

Finally, $\forall k \in \mathcal{K}$, $\boldsymbol{S}(k,f,t)$ in (8) is obtained as

$$\boldsymbol{S}(k,f,t) = \boldsymbol{J}_{\overline{c}(k,f,t)}^{\overline{\gamma}(f,t)\ H} \overline{\boldsymbol{S}}(f,t) \boldsymbol{J}_{\overline{c}(k,f,t)}^{\overline{\gamma}(f,t)},$$

where $\overline{\boldsymbol{S}}(f,t) = (\boldsymbol{R}_{\overline{\gamma}(f,t)}^{f,T} + \overline{\boldsymbol{\gamma}}^{f,T}(f,t)\,\overline{\boldsymbol{\gamma}}^{f,T}(f,t)^H)^*$, which can be rewritten in the following square root form:

$$\sqrt{\boldsymbol{S}}(k,f,t) = \mathcal{LT}\left(\boldsymbol{J}_{\overline{c}(k,f,t)}^{\overline{\gamma}(f,t)\ H} \left[\sqrt{\boldsymbol{R}}_{\overline{\gamma}(f,t)}^{f,T}, \overline{\boldsymbol{\gamma}}^{f,T}(f,t) \right]^* \right). \tag{58}$$

C. Mathematical derivations of the initialization method

1) Analysis of the multiplicative update rules:

Proposition 1. *The log-likelihood L in equation* (14) *is non-decreasing under the following updates:*

$$\hat{\sigma}^2 = \sigma^2 \frac{\sum_{f=1}^{F}\sum_{t=1}^{T} \frac{\delta(f,t)|x(f,t)|^2}{\sigma^4(f,t)}}{\sum_{f=1}^{F}\sum_{t=1}^{T} \frac{\delta(f,t)}{\sigma^2(f,t)}} \tag{59}$$

$$\forall k,t,\ \hat{h}(k,t) = h(k,t) \frac{\sum_{f=1}^{F} w(k,f) \frac{\delta(f,t)|x(f,t)|^2}{\sigma^4(f,t)}}{\sum_{f=1}^{F} w(k,f) \frac{\delta(f,t)}{\sigma^2(f,t)}} \tag{60}$$

Proof of Proposition 1:

$$\begin{aligned}
& L(\hat{\sigma}^2, \hat{h}(k,t), w(k,f)) \\
= & -\sum_{f=1}^{F}\sum_{t=1}^{T} \delta(f,t) \Big(\ln\big(\sigma^2(f,t) + (\hat{\sigma}^2 - \sigma^2) \\
& + \sum_{k=1}^{K} w(k,f)(\hat{h}(k,t) - h(k,t))\big) \\
& + \frac{|x(f,t)|^2}{\frac{\sigma^2}{\sigma^2(f,t)}\frac{\hat{\sigma}^2\sigma^2(f,t)}{\sigma^2} + \sum_{k=1}^{K} \frac{w(k,f)h(k,t)}{\sigma^2(f,t)} \frac{\hat{h}(k,t)\sigma^2(f,t)}{h(k,t)}} \Big) \\
\geq & -\sum_{f=1}^{F}\sum_{t=1}^{T} \delta(f,t) \Big(\ln\big(\sigma^2(f,t)\big) \\
& + \frac{1}{\sigma^2(f,t)} \big((\hat{\sigma}^2 - \sigma^2) + \sum_{k=1}^{K} w(k,f)(\hat{h}(k,t) - h(k,t)) \big) \\
& + \frac{\sigma^2}{\sigma^2(f,t)} \frac{|x(f,t)|^2}{\frac{\hat{\sigma}^2\sigma^2(f,t)}{\sigma^2}} + \sum_{k=1}^{K} \frac{w(k,f)h(k,t)}{\sigma^2(f,t)} \frac{|x(f,t)|^2}{\frac{\hat{h}(k,t)\sigma^2(f,t)}{h(k,t)}} \Big) \\
= & \ L(\sigma^2, h(k,t), w(k,f))
\end{aligned}$$

The first equality is a rewriting of $L(\hat{\sigma}^2, \hat{h}(k,t), w(k,f))$; the inequality is due to the concavity of function $\ln(.)$ (which is upper bounded by its first order Taylor expansion) and to the convexity of function $1/(.)$ (which is upper bounded by Jensen's inequality); the last equality is obtained by substituting the updates (59) and (60) and identifying the resulting expression with $L(\sigma^2, h(k,t), w(k,f))$. ■

Proposition 2. *The log-likelihood L in equation* (14) *is non-decreasing under the updates* (59) *and*

$$\forall k,f,\ \hat{w}(k,f) = w(k,f) \frac{\sum_{t=1}^{T} h(k,t) \frac{\delta(f,t)|x(f,t)|^2}{\sigma^4(f,t)}}{\sum_{t=1}^{T} h(k,t) \frac{\delta(f,t)}{\sigma^2(f,t)}} \tag{61}$$

The proof of Proposition 2 is the same as that of Proposition 1.

2) First E-step: Given x, all time-frequency samples $n(f,t)$ are independent and

$$n(f,t) \sim \mathcal{N}_{\mathbb{F}}\left(\delta(f,t)\frac{\sigma^2 x(f,t)}{\sigma^2(f,t)}, \frac{\sigma^2(\sigma^2(f,t)-\delta(f,t)\sigma^2)}{\sigma^2(f,t)}\right)$$

thus $e(f,t)$ defined in equation (5) is obtained as

$$e(f,t) = \delta(f,t)\frac{\sigma^2}{\sigma^2(f,t)}\left((\sigma^2(f,t)-\sigma^2) + \frac{\sigma^2\left|x(f,t)\right|^2}{\sigma^2(f,t)}\right). \tag{62}$$

In the same way, given x, $\forall k$ all time-frequency samples $c_k(f,t)$ are independent and

$$c_k(f,t) \sim \mathcal{N}_{\mathbb{F}}\left(c_k^{f,T}(f,t), R_{c_k(f,t)}^{f,T}\right)$$

where

$$c_k^{f,T}(f,t) = \delta(f,t)\frac{v_k(f,t)}{\sigma^2(f,t)}x(f,t), \tag{63}$$

$$\sqrt{R}_{c_k(f,t)}^{f,T} = \sqrt{\frac{v_k(f,t)}{\sigma^2(f,t)}}\sqrt{\sigma^2(f,t)-\delta(f,t)v_k(f,t)}. \tag{64}$$

Thus $\forall k,f,t$,

$$\sqrt{\boldsymbol{R}}_{\overline{c}(k,f,t)}^{f,T} = \operatorname{diag}\left(\sqrt{R}_{c_k(f,t)}^{f,T}\ldots\sqrt{R}_{c_k(f,t-P(k,f))}^{f,T}\right). \tag{65}$$

Finally, $\forall k \in \mathcal{K}$, $\boldsymbol{S}(k,f,t)$ in (8) is obtained as

$$\boldsymbol{S}(k,f,t) = \left(\boldsymbol{R}_{\overline{c}(k,f,t)}^{f,T} + \overline{\boldsymbol{c}}^{f,T}(k,f,t)\,\overline{\boldsymbol{c}}^{f,T}(k,f,t)^H\right)^*,$$

which can be rewritten in the following square root form:

$$\sqrt{\boldsymbol{S}}(k,f,t) = \mathcal{LT}\left(\left[\sqrt{\boldsymbol{R}}_{\overline{c}(k,f,t)}^{f,T}, \overline{\boldsymbol{c}}^{f,T}(k,f,t)\right]^*\right). \tag{66}$$

REFERENCES

[1] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, pp. 788–791, Oct. 1999.

[2] T. Virtanen, A. Cemgil, and S. Godsill, "Bayesian extensions to non-negative matrix factorisation for audio signal modelling," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Las Vegas, Nevada, USA, Apr. 2008, pp. 1825–1828.

[3] S. A. Raczyński, N. Ono, and S. Sagayama, "Multipitch analysis with harmonic nonnegative matrix approximation," in *Proc. of the 8th International Society for Music Information Retrieval Conference (ISMIR)*, Vienne, Autriche, Sep. 2007.

[4] P. Smaragdis, "Relative pitch tracking of multiple arbitrary sounds," *Journal of the Acoustical Society of America (JASA)*, vol. 125, no. 5, pp. 3406–3413, May 2009.

[5] E. Vincent, N. Bertin, and R. Badeau, "Adaptive harmonic spectral decomposition for multiple pitch estimation," vol. 18, no. 3, pp. 528–537, Mar. 2010.

[6] P. Smaragdis and J. C. Brown, "Non-negative matrix factorization for polyphonic music transcription," in *Proc. of IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, New Paltz, New York, USA, Oct. 2003, pp. 177–180.

[7] N. Bertin, R. Badeau, and E. Vincent, "Enforcing harmonicity and smoothness in Bayesian non-negative matrix factorization applied to polyphonic music transcription," vol. 18, no. 3, pp. 538–549, Mar. 2010.

[8] A. Cichocki, R. Zdunek, A. H. Phan, and S.-i. Amari, *Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-way Data Analysis and Blind Source Separation*. Wiley, Nov. 2009.

[9] T. Virtanen, "Monaural sound source separation by nonnegative matrix factorization with temporal continuity and sparseness criteria," vol. 15, no. 3, pp. 1066–1074, Mar. 2007.

[10] D. FitzGerald, M. Cranitch, and E. Coyle, "Extended nonnegative tensor factorisation models for musical sound source separation," *Computational Intelligence and Neuroscience*, vol. 2008, pp. 1–15, May 2008, article ID 872425.

[11] A. Liutkus, R. Badeau, and G. Richard, "Informed source separation using latent components," in *Proc. of 9th International Conference on Latent Variable Analysis and Signal Separation (LVA/ICA)*, ser. Lecture Notes in Computer Science, V. Vigneron, V. Zarzoso, E. Moreau, R. Gribonval, and E. Vincent, Eds., vol. 6365. Saint Malo, France: Springer, Sep. 2010, pp. 498–505.

[12] M. N. Schmidt and H. Laurberg, "Non-negative matrix factorization with Gaussian process priors," *Computational Intelligence and Neuroscience*, vol. 2008, pp. 1–10, 2008, article ID 361705.

[13] C. Févotte, N. Bertin, and J.-L. Durrieu, "Nonnegative matrix factorization with the Itakura-Saito divergence. With application to music analysis," *Neural Computation*, vol. 21, no. 3, pp. 793–830, Mar. 2009.

[14] R. Hennequin, R. Badeau, and B. David, "Time-dependent parametric and harmonic templates in non-negative matrix factorization," in *Proc. of the 13th International Conference on Digital Audio Effects (DAFx)*, Graz, Autriche, Sep. 2010.
[15] ——, "NMF with time-frequency activations to model non-stationary audio events," vol. 19, no. 4, pp. 744–753, May 2011.
[16] P. Smaragdis, *Blind Speech Separation*. Springer, 2007, ch. Probabilistic decompositions of spectra for sound separation, pp. 365–386.
[17] A. Ozerov and C. Févotte, "Multichannel nonnegative matrix factorization in convolutive mixtures for audio source separation," vol. 18, no. 3, pp. 550–563, Mar. 2010, special issue on Signal Models and Representations of Musical and Environmental Sounds.
[18] A. Ozerov, E. Vincent, and F. Bimbot, "A general flexible framework for the handling of prior information in audio source separation," vol. 20, no. 4, pp. 1118–1133, May 2012.
[19] D. Griffin and J. Lim, "Signal reconstruction from short-time Fourier transform magnitude," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 32, no. 2, pp. 236–243, 1984.
[20] H. Kameoka, N. Ono, K. Kashino, and S. Sagayama, "Complex NMF: A new sparse representation for acoustic signals," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Apr. 2009, pp. 3437–3440.
[21] J. Le Roux, H. Kameoka, E. Vincent, N. Ono, K. Kashino, and S. Sagayama, "Complex NMF under spectrogram consistency constraints," in *Proc. of the Acoustical Society of Japan Autumn Meeting*, no. 2-4-5, Sep. 2009.
[22] A. Ozerov, C. Févotte, and M. Charbit, "Factorial scaled hidden markov model for polyphonic audio representation and source separation," in *Proc. of IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, New Paltz, New York, USA, Oct. 2009, pp. 121–124.
[23] O. Dikmen and A. T. Cemgil, "Gamma Markov random fields for audio source modeling," vol. 18, no. 3, pp. 589–601, Mar. 2010.
[24] G. Mysore, P. Smaragdis, and B. Raj, "Non-negative hidden Markov modeling of audio with application to source separation," in *Proc. of 9th international Conference on Latent Variable Analysis and Signal Separation (LCA/ICA)*, St. Malo, France, Sep. 2010.
[25] R. Badeau, "Gaussian modeling of mixtures of non-stationary signals in the time-frequency domain (HR-NMF)," in *Proc. of IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, New Paltz, New York, USA, Oct. 2011, pp. 253–256.
[26] R. Badeau, B. David, and G. Richard, "High resolution spectral analysis of mixtures of complex exponentials modulated by polynomials," *IEEE Trans. Signal Process.*, vol. 54, no. 4, pp. 1341–1350, Apr. 2006.
[27] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society*, vol. 39, no. 1, pp. 1–38, 1977.
[28] G. Bierman, *Factorization methods for discrete sequential estimation*. Academic Press, 1977.
[29] J. J. Commandeur and S. J. Koopman, *An Introduction to State Space Time Series Analysis*, ser. Practical Econometrics. Oxford University Press, Jul. 2007.
[30] L. Zhang and A. Cichocki, "Blind separation of filtered sources using state-space approach," in *Advances in Neural Information Processing Systems (NIPS)*, vol. 11, Denver, Colorado, USA, Nov. 1998.
[31] R. K. Olsson and L. K. Hansen, "Linear state-space models for blind source separation," *Journal of Machine Learning Research*, vol. 7, pp. 2585–2602, 2006.
[32] A. T. Cemgil, H. J. Kappen, and D. Barber, "A generative model for music transcription," vol. 14, no. 2, pp. 679–694, Mar. 2006.
[33] S. Bensaid, A. Schutz, and D. T. M. Slock, "Single microphone blind audio source separation using EM-Kalman filter and short+long term AR modeling," in *Proc. of 9th International Conference on Latent Variable Analysis and Signal Separation (LVA/ICA)*, ser. Lecture Notes in Computer Science, vol. 6365/2010, Saint Malo, France, Sep. 2010, pp. 106–113.
[34] R. Badeau, N. Bertin, and E. Vincent, "Stability analysis of multiplicative update algorithms and application to non-negative matrix factorization," *IEEE Trans. Neural Netw.*, vol. 21, no. 12, pp. 1869–1881, Dec. 2010.
[35] R. A. Horn and C. R. Johnson, *Matrix analysis*. Cambridge: Cambridge University Press, 1985.