A sequential framework for audio signal restoration

Rubén M. Clavería, and Simon J. Godsill, *Senior Member, IEEE*

***Abstract*—In this paper we present a sequential framework for audio signal restoration. Building upon Gabor regression models intended for batch inference, we represent the evolution of synthesis coefficients over time in terms of a state-space model amenable to Bayesian filtering strategies. Rather than utilising off-the-shelf particle filters, we exploit the probabilistic structure of the modified model to derive exact Gibbs updates for the parameters within each state. Thus, sequential inference is carried out by means of sequential Markov Chain Monte Carlo (also known as MCMC-Particle filter) plus a rejuvenating move to ensure particle diversity. As an exploratory test for the framework's potential, we employ the proposed sequential Markov Chain Monte Carlo algorithm to analyse a series of degraded audio signals. The resulting filtering distributions are used to generate a enhanced versions of the input. Results suggest that our sequential approach is competitive with batch strategies in terms of perceptual quality and waveform signal-to-noise ratio, and can readily address scenarios where existing batch schemes may struggle, such as non-stationary noise or variations of volume. Moreover, the sequential model shows potential for real-time applications—provided that enhanced inference strategies are developed.**

***Index Terms*—Audio denoising, Gabor frames, Bayesian filtering, Sequential MCMC**

I. Introduction

Sparse dictionary representations [1], [2] are at the core of many of the advances in the field of signal processing in the last decades, such as signal compression [3] and denoising [4]. Bayesian extensions of dictionary representation, in which additional assumptions on regression coefficients are incorporated through a prior distribution, have been successfully applied to the analysis of images, e.g. [5], and audio time series [6]. Crucially, structural and distributional information of the transform domain (e.g., wavelet hierarchies in image analysis), can be readily included in this class of models, yielding significant advantages over simpler unstructured sparsity approaches.

The Gabor regression model, proposed in [6], can be regarded as a dictionary representation endowed with a prior structure designed to favour both smoothness of the estimated signal and sparseness of the coefficient representation. Applications of this model are not limited to denoising (see [7], [8], [9] for other signal restoration tasks). Inference in Gabor regression models is typically carried out with Markov chain Monte Carlo methods (MCMC hereinafter, see [10] for a general review). Although the coupling of Gabor regression models and Monte Carlo inference schemes can lead to high quality reconstructions in practice, it has the typical shortcomings associated to MCMC: high computational expense, long run times and noticeable approximation errors due to the stochastic nature of the method.

With real time applications as the ultimate motivation, in this paper we investigate sequential alternatives to the batch restoration algorithms of [6] and subsequent works. Taking the assumptions of the batch Gabor regression model as a starting point–namely, atom representation of the signal, structured sparsity and heavy-tailed priors on synthesis coefficients–and replacing global parameters such as background noise σ^2 by slowly varying components (i.e., $\sigma^2_{1:t}$), we formulate a state-space model for the parameters of interest. Exploiting the tractability of the resulting full conditionals, we propose a sequential Markov chain Monte Carlo scheme [11], [12], [13] to approximate the filtering distributions of the target parameters. We test the proposed model on two denoising examples, showing that the sequential scheme can generate solutions comparable with those obtained using batch methods at a significantly lower computational expense.

II. Background: Gabor regression

In the Gabor regression model, the input signal is modelled as a weighted sum of Gabor atoms—a single Gabor window $w(t)$ shifted in time and frequency. Equivalently, it can be interpreted as a weighted sum of oscillating functions with limited time-domain support (i.e., windowed sinusoids). To account for both background noise in the observed signal $y(t)$ and possible limitations of the model[1], an additive noise ϵ_t is considered on each sample of the signal. Denoting m, n the terms indexing frequency and time, respectively, the dictionary representation is summarised by:

$$y(t) = \sum_{m=0}^{M-1} \sum_{n=1}^{N} c_{m,n} g_{m,n}(t) + \epsilon_t, \quad g_{m,n}(t) = w_n(t) \mathrm{e}^{-2\pi \mathrm{i} \frac{m}{M} t}, \tag{1}$$

where $w_n(t)$ are time-shifted replicas of $w(t)$, a smooth function with compact support (e.g., a Hann window). Sparsity assumptions are incorporated by means of a spike-and-slab prior [14] on coefficients $c_{m,n}$, with $c_{m,n}$ taking the exact value of 0 if the corresponding indicator variable $\gamma_{m,n}$ is 0, and being Gaussian-distributed if $\gamma_{m,n} = 1$. An appealing trait of this class of priors is their suitability to encode structured relations between indicator variables (e.g., such as Markov chain or Markov random fields [6], [15], which can be used to model domain-specific patterns). Another widely utilised device to encourage sparsity in linear regression models is the adoption of heavy-tailed priors on coefficients [16], [17].

[1]Fourier-inspired dictionary representations are a powerful tool for signal modelling and can reproduce most signals of interest. However, there are certain components that the model may fail to capture, e.g., in [8] a second layer of basis functions is included in order to model the transients that a standard one-layer representation fails to characterise.

Rubén M. Clavería and Simon J. Godsill are with the Department of Engineering, University of Cambridge, CB2 1PZ, UK (e-mail:rmc83@cam.ac.uk; sjg@eng.cam.ac.uk).

In audio applications, that assumption is instead adopted on the grounds of reflecting the wide range of coefficient values expected in non-stationary processes. Computationally, it is convenient to use the Scaled mixture of normals representation [18] (SMiN), under which the Student's t-distribution adopted here and in previous works can be expressed as the conditionally Gaussian distribution with an Inverse Gamma-distributed variance: $p(c) = \int \mathcal{N}(c; 0, v)\mathcal{IG}(v; \alpha, \beta)\mathrm{d}v$.

III. A STATE-SPACE MODEL FOR GABOR REGRESSION

Due to their potential for real-time applications, the development of sequential extensions of the Gabor regression apparatus has been in demand since the early days of this approach. Crafted for different, more specific tasks, relevant work in the field of sequential time-frequency inference can be found in [19], where amplitude demodulation is posed as a probabilistic inference problem and a real-time Kalman-like scheme is presented; [20], where a bank of switching oscillators at specific frequencies is used to model the occurrence of musical notes; [21], where Particle Filter is applied to music transcription. Importantly, [22] presents a general framework that, under specific assumptions, can yield probabilistic analogues of classic notions like filter banks and Short term Fourier transform or recover previous models like the Probabilistic Phase Vocoder [23]. As in Gabor regression, the idea underpinning many of this models is the superposition of band-limited elements slowly modulated in time. A recent review of time-frequency state-space models can be found in [24], where equivalences between the different formulations are explored.

As indicated in Eq. 1, Gabor differs from the approaches of [22], [23] in that amplitude modulations of frequency components are modelled by fixed smooth window functions (e.g., Hann) rather than of auto-regressive (AR) or Gaussian processes (GP). Although in principle more limited than certain GP or AR counterparts, the Gabor model (with overlapping frames) has proved adequate for the representation of most signals of interest, and reduces the dimensionality of the problem significantly—it suffices to estimate the coefficient associated to each frame rather of the instant values of the amplitude envelope. Furthermore, this approach sets a natural partition of the input signal: signal $y(t)$ can be processed sequentially in chunks of length W (the size of the support of window $w(t)$), each of which can be treated as a unit—the observation associated to an underlying *state*. Conversely, in schemes described in [22], full or partial smoothing of the the state vector is needed to obtain reliable estimates of the amplitude envelopes, due to the large number of unknowns associated to each sample $y(t)$.

A. Observation model

Assuming a real-valued signal $y(t)$ and arranging its values in vector $\mathbf{y}$, Eq. 1 admits the real-domain matrix representation $\mathbf{y} = \mathbf{D}\bar{\mathbf{c}} + \boldsymbol{\epsilon}$, where $\mathbf{D}$ is a dictionary matrix with basis functions $g_{m,n}(t)$ in its columns and $\bar{\mathbf{c}}$ is the vector containing all coefficients $c_{m,n}$ (real and imaginary parts are stacked in real-valued matrices, following [25]). We assume 50% overlap. Matrix $\mathbf{D}$ has a redundant block-sparse structure: due to the limited support of basis functions $g_{m,n}$, columns are zero everywhere outside the specific time window covered by w_n, and $\mathbf{D}$ consists of the concatenation of shifted versions of the same block $\mathbf{G}_c$ containing the windowed sinusoidal waveforms[2] from frequencies 1 to m. Thus, by partitioning $\mathbf{y}$ into segments $\mathbf{y}_1, \dots, \mathbf{y}_N$ of length W, expression $\mathbf{y} = \mathbf{D}\bar{\mathbf{c}} + \boldsymbol{\epsilon}$ can be written as:

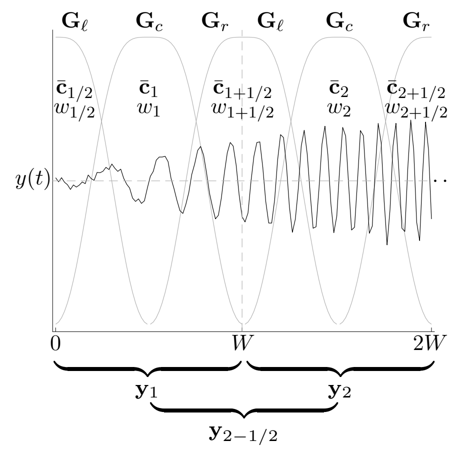

Fig. 1. Signal $y(t)$ is partitioned into segments $\mathbf{y}_1, \mathbf{y}_2, \dots$. Coefficients are grouped into subvectors $\mathbf{c}_n$ associated to the frame with support in window w_n. Due to frame overlap, half of the frames lie in the "interface" of consecutive chunks $\mathbf{y}_n$, $\mathbf{y}_{n+1}$. Coefficients associated to those frames are denoted $\mathbf{c}_{n+1/2}$, with $\mathbf{c}_n$ being the ones associated to the central frame. Terms $\mathbf{G}_\ell, \mathbf{G}_c, \mathbf{G}_r$ are the frames or basis functions used to reconstruct each chunk $\mathbf{y}_n$ (Eq. 2).

$$\mathbf{y}_n = \underbrace{[\mathbf{G}_\ell\ \mathbf{G}_c\ \mathbf{G}_r]}_{\mathbf{G}}\mathbf{c}_n + \boldsymbol{\epsilon}_n, \qquad \boldsymbol{\epsilon}_n \sim \mathcal{N}(\mathbf{0}; \sigma^2\mathbf{I}_W), \tag{2}$$

where terms $\mathbf{G}_\ell\ \mathbf{G}_c\ \mathbf{G}_r$ are the blocks of basic functions contributing to each segment of the signal (Figure 1): $\mathbf{G}_\ell$, left frame, upper half of the basis functions; $\mathbf{G}_c$, central frame; $\mathbf{G}_r$, right frame, lower half of the basis functions. Following the definitions of Figure 1, vector $\bar{\mathbf{c}}_n$ contains the coefficients relevant to frame y_n: $\mathbf{c}_n = [\bar{\mathbf{c}}_{n-1/2};\ \bar{\mathbf{c}}_n;\ \bar{\mathbf{c}}_{n+1/2}]$ (vertical vector).

B. State model

1) Sparsity: We would like to keep most of properties of the batch Gabor regression in our sequential framework: variable selection encoded in spike-and-slab prior on synthesis coefficients, structured sparsity and heavy tailed priors on coefficients (conditional to variable selection). Conversely, hyperparameters regulating the structure of indicator variables (denoted ϕ in [6], e.g., transition probabilities in a Markov chain prior) are taken as fixed for the sake of simplicity. Formally, the assumption of a spike-and-slab prior density on regression coefficients can be expressed as:

$$\begin{aligned} p(c_{m,n}|\gamma_{m,n}, s_{m,n}) =& (1-\gamma_{m,n})\delta(c_{m,n}) + \dots \\ & \gamma_{m,n}\, \mathcal{N}(c_{m,n}; \mathbf{0}_2, s_{m,n}\mathbf{I}_2)\,, \end{aligned} \tag{3}$$

[2]Except for a difference in phase that can be easily corrected by synthesis coefficients at the estimation stage.

where $\delta(\cdot)$ is a Dirac delta, γ_k is an indicator variable and s_k is the *a priori* variance of the coefficient. The normal distribution is bivariate because c_k contains the real and imaginary part of the original Gabor complex coefficients. As to the prior structure of indicators $\boldsymbol{\gamma}$, we set a Markov chain over time (indexed by n) with parameters ϕ_{00}, ϕ_{11} on each frequency m. Chains are independent, although ultimately connected through the likelihood function.

2) Variances: Heavy-tailed priors on c_k are imposed through a conditionally Gaussian SMiN structure. In particular, we choose a Student's t-distribution by setting an Inverse Gamma prior on coefficient variances s_k:

$$p(s_{m,n}) = \mathcal{IG}(s_{m,n}; \kappa, f(m)\nu_n), \tag{4}$$

where term $f(m)$ is a decreasing frequency-dependent modulating function mirroring the decaying profile of typical spectra (a Butterworth gain function works well in practice). Shape parameter κ is fixed and rate parameters ν_n are seen as variable (they are assigned their own prior) and time-dependent. In principle, the choice of ν_n is of practical nature: by making it time-dependent, it can be simply treated as an element of the state-space state vector—instead, a unique ν is considered across all frames in batch settings may complicate inference when trying to devise a sequential approach. It has been found, though, that this "relaxation" does not convey any noticeable disadvantage in practice. Moreover, it adapts smoothly to temporal variations of intensity in the signal. Either global or time-varying, tuning of parameter ν is an important component of this framework, since it allows for blind estimation of synthesis coefficients. Conversely, in scenarios such as standard LASSO-type approaches, the roughly equivalent equivalent regularisation parameter λ has to be hand-tuned according to characteristics of the signal. Prior density is set to be independent: $p(\nu_n) = \mathcal{G}(\nu_n|\alpha_\nu, \beta_\nu)$, although a Markov chain could be used to encourage smoothness.

As with ν, observation noise σ^2 is assumed unique across all frames in batch settings. Under the same argument presented for ν, we replace that assumption with a time-dependent noise $\sigma^2_{1:N}$ so that each σ^2_n can be incorporated in the state vector:

$$\begin{aligned} p(\sigma_1^2) &= \mathcal{IG}(\sigma_1^2|\alpha_{\sigma^2}, \beta_{\sigma^2}), \\ p(\sigma_n^2|\sigma_{n-1}^2) &= \mathcal{IG}(\sigma_n^2|\alpha_{\sigma^2}, (\alpha_{\sigma^2}-1)\sigma_{n-1}^2). \end{aligned} \tag{5}$$

To encourage smoothness across frames, the inverse-Gamma Markov chain prior is chosen so that the expected of σ^2_n given the previous value σ^2_{n-1} is σ^2_{n-1}. This assumption may be helpful when dealing with non-stationary but slowly varying noise (it can be easily incorporated into batch settings too).

3) Coefficients: Due to overlap, each chunk $\mathbf{y}_n$ shares frames $\bar{\mathbf{c}}_{n-1/2}$ and $\bar{\mathbf{c}}_{n+1/2}$ with the adjacent segments (Figure 1), making the association between an observation $\mathbf{y}_n$ and an underlying state less straightforward than in other cases (e.g., in tracking models each measurement y_t is typically associated to one position x_t). Moreover, due to the hierarchical structure of the Gabor regression model with variable selection, the influence of $\mathbf{c}_n$ on the coefficients of the adjacent frames is indirect. As a consequence, posing a recurrence of the form $p(\mathbf{c}_n|\mathbf{c}_{n-1})$ is not immediate. In order to facilitate a state-space formulation, we redefine $\mathbf{c}_n$ as $\mathbf{c}_n^\top = [\mathbf{c}_n^{(\ell)}; \mathbf{c}_n^{(c)}; \mathbf{c}_n^{(r)}]$, where ℓ, c, r indicate the position of the frame within segment $\mathbf{y}_n$, i.e., $\mathbf{c}_n^{(\ell)} = \bar{\mathbf{c}}_{n-1/2}$ corresponds to the frame on the left, etc. In this formulation $\mathbf{c}_n^{(r)}$ and $\mathbf{c}_{n+1}^{(\ell)}$ are seen as different variables, each one associated to a different state and observation, but the restriction $\mathbf{c}_n^{(r)} = \mathbf{c}_{n+1}^{(\ell)}$ must be incorporated in the model. In space-state parlance, the *state evolution equation* of the model can thus be represented as:

$$\mathbf{c}_n = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{I} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{c}_{n-1} + \boldsymbol{\gamma}_n \odot \begin{bmatrix} \mathbf{0} \\ \boldsymbol{\xi}_n^{(c)} \\ \boldsymbol{\xi}_n^{(r)} \end{bmatrix}, \tag{6}$$

where $\boldsymbol{\gamma}$ is the indicator vector corresponding to $\mathbf{c}_n$, $\odot$ denotes the element-wise product, and $\boldsymbol{\xi}_n^{(c)}$, $\boldsymbol{\xi}_n^{(r)}$ are realisations of Gaussians with the specific variances s associated to with coefficients $\mathbf{c}_n$ (Eq. 3). Re-stated from a probabilistic standpoint, Eq. 6 can be expressed as:

$$p(c_{m,n}^{(p)}|\mathbf{c}_{n-1}, \boldsymbol{\gamma}_{n-1}, \boldsymbol{\gamma}_n, \boldsymbol{s}_n) = \begin{cases} \text{Eq. 3} & \text{if } p \in \{c, r\}, \\ \delta(\mathbf{c}_n^{(\ell)} - \mathbf{c}_{n-1}^{(r)}), & \text{if } p = \ell. \end{cases} \tag{7}$$

This model may appear excessively stiff for filtering tasks, since there is no margin of accommodate components $\mathbf{c}_n^{(\ell)}$ in the light of new observation $\mathbf{y}_n$; however, as will be shown in Section IV, components $\mathbf{c}_n^{(\ell)}$ and $\mathbf{c}_{n-1}^{(r)}$ can be sampled jointly conditional to the neighboring states, in a *backwards* or *rejuvenating* move in the fashion of [26]. Putting together all components, the joint density of the proposed model can be written as:

$$\begin{aligned} &p(\mathbf{c}_{1:N}, \mathbf{s}_{1:N}, \boldsymbol{\gamma}_{1:N}, \sigma^2_{1:N}, \nu_{1:N}, \mathbf{y}) = \\ &\quad \left[(\mathbf{y}_1|\mathbf{c}_1, \sigma_1^2)p(\mathbf{c}_1|\boldsymbol{\gamma}_1, \mathbf{s}_1)p(\mathbf{s}_1)p(\sigma_1^2)p(\gamma_1)p(\nu_1)\right] \cdot \ldots \\ &\quad \prod_{n=2} \left[p(\mathbf{y}_n|\mathbf{c}_n, \sigma_n^2)p(\mathbf{c}_n|\mathbf{c}_{n-1}, \boldsymbol{\gamma}_n, \mathbf{s}_n)p(\mathbf{s}_n)p(\sigma_n^2|\sigma_{n-1}^2)\right. \\ &\quad \left. p(\gamma_n|\gamma_{n-1})p(\nu_n)\right], \end{aligned} \tag{8}$$

which has a hidden Markov model structure. Like $\mathbf{c}_n$, terms $\mathbf{s}_n$, $\boldsymbol{\gamma}_n$ in bold font groups variables sharing the same index n (it contains all frequencies m and positions $p \in \{\ell, c, l\}$). Density $p(\gamma_n|\gamma_{n-1})$ imposes $\gamma_n^{(\ell)} = \gamma_{n-1}^{(r)}$ with the Dirac delta trick (as in Eq. 7), and describes Markov chain dependencies between $\gamma_n^{(\ell)}$, $\gamma_n^{(c)}$, and $\gamma_n^{(r)}$ (in parallel for each frequency m).

IV. Computing the model

The aim is to devise a scheme to estimate the filtering distribution $p(x_k|y_{1:k})$. In this case, each observation corresponds to a chunk of length W of the original signal, denoted $\mathbf{y}_n$. Synthesis coefficients $\mathbf{c}_{1:N}$ are the quantities of interest, but nuisance and latent parameters such as indicator variables $\boldsymbol{\gamma}_{1:N}$, $\sigma^2_{1:N}$ and $\mathbf{s}_{1:N}$ must be estimated as well. Unlike other models (e.g., target tracking), the state evolution equation Eq. 6 is of little predictive value—it merely contains the restriction $\mathbf{c}_n^{(r)} = \mathbf{c}_{n+1}^{(\ell)}$ and the "process noise" $\boldsymbol{\xi}_n^{(c)}$, $\boldsymbol{\xi}_n^{(r)}$ (which indirectly encodes the heavy-tailed prior assumption). Thus, using traditional schemes like the bootstrap filter would hardly place the proposed samples in a zone of high probability.

Due to the high dimensionality of the problem, the importance sampling-based Particle filter may struggle to provide a reliable estimation of the target distribution. The so-called MCMC-based Particle Filter [12] (outlined earlier in [11]) has proved useful in challenging high-dimensional scenarios such as the multitarget tracking problem under heavy clutter studied in [13]. For efficiency reasons, we aim to develop an algorithm based on the non-marginalised version of the MCMC-based Particle Filter [13].

Instead of using importance sampling as traditional Particle filter methods, Sequential MCMC employs a Markov chain to generate samples from the filtering distribution, provided that a collection of samples from the filtering distribution of the previous time step is available. To avoid the high expense of computing marginalisation at each Metropolis-Hastings (MH) step of the Markov chain [12], the non-marginalised version [13] generates samples from the joint distribution $\hat{p}(x_k, x_{k-1}|y_{1:k})$ and subsequently extracts samples from the current state, $x_k^{(j)}$. This can be regarded as an alternative method to sample from marginal $\hat{p}(x_k|y_{1:k})$.

$$\hat{p}(x_k, x_{k-1}|y_{1:k}) \propto \hat{p}(x_{k-1}|y_{1:k-1})p(x_k|x_{k-1})p(y_k|x_k) \quad (9)$$

A. Sequential MCMC

Due to the conjugacy of the distributions used in the model (Eq. 8), the joint distribution of Eq. 9 can be sampled through an exact Gibbs sampling procedure, i.e., exact sampling from the full conditionals, without any costly MH steps. For each frame[3] n, the filtering distribution is approximated using the sampling procedure detailed below. Superscript $j = 1, \dots, N_{iter}$ indicates an iteration of the algorithm, whereas j' is the index of the particles in the the approximate filtering distribution of the previous frame, $\hat{p}(\mathbf{c}_{n-1}, \boldsymbol{\gamma}_{n-1}, \mathbf{s}_{n-1}, \sigma^2_{n-1}, \nu_{n-1}|\mathbf{y}_{1:n-1})$ (i.e., the samples resulting from the $n-1$-th cycle of the algorithm after thinning the chain and discarding the burn-in period).

1) **Initialisation**: extract a random ancestor particle (j') from the approximate filtering distribution of previous frame. Propose an initial value for σ_n^2. Since a degree of similarity is expected along frames, any value from the collection $[\sigma_{n-1}^2]^{(j')}$ works as an initial guess (alternatively, $[\sigma_n^2]^{(0)}$ can be sampled from the prior). Similarly, an initial value $\nu_n^{(0)}$ must be proposed to feed the MCMC chain.

2) **Gibbs sweep**:

- For $m \in \{0, \dots, M-1\}, p \in \{c, r\}$:

$$s_{m,n,p}^{(j)} \sim \mathcal{IG}\left(\kappa + \gamma_{m,n,p}^{(j-1)}, \frac{1}{2}\|c_{m,n,p}^{(j-1)}\|^2 + f(m)\nu_n^{(j-1)}\right) \quad (10)$$

In-frame position (p) was moved to the subscript to allow iteration index (j) to occupy the superscript field.

- For $m \in \{0, \dots, M-1\}, p \in \{c, r\}$, obtain pairs $(\gamma_{m,n,p}^{(j)}, c_{m,n,p}^{(j)})$ from the full conditionals using:

$$\begin{aligned} &\gamma_{m,n,p}^{(j)} \sim \text{Bernoulli}\left(\frac{\tau_{m,n,p}}{1+\tau_{m,n,p}}\right) \\ &c_{m,n,p}^{(j)} \begin{cases} = 0 & \text{if } \gamma_{m,n,p}^{(j)} = 0 \\ \sim \mathcal{N}(\mu_{m,n,p}, [\sigma_n^2]^{(j)}\Sigma_{m,n,p}) & \text{if } \gamma_{m,n,p}^{(j)} = 1 \end{cases} \end{aligned} \quad (11)$$

where $\mu_{m,n,p}, \Sigma_{m,n,p}$ and $\tau_{m,n,p}$ are the mean, covariance matrix and odds ratio, conditional to the observation and the current values of the rest of the parameters. The detailed formulae is found in Appendix A. A similar scheme is used in [6].

- Update scale parameter as in:

$$\nu_n^{(j)} \sim \mathcal{G}\left(\alpha_\nu + \kappa \cdot |\boldsymbol{\gamma}_{:,n,\{c,r\}}|, \beta_\nu + \sum_{\boldsymbol{\gamma}_{:,n,\{c,r\}}=1} \frac{f(m)}{s_{m,n,p}}\right), \quad (12)$$

where $|\boldsymbol{\gamma}_{:,n,\{c,r\}}|$ is the count of of indicator variables equal to one within frame n, positions $\{c, r\}$. Due to the dependency of v_ν on all indicator variables, a draw from the full conditional can lead to slow convergence. Instead, we use a draw ν_n from the reduced conditional (i.e., $\nu_n|\boldsymbol{\gamma}_{:,n,\{c,r\}}$), which in turn can be considered a joint draw of $\nu_n, \{s_{:,n,\{c,r\}}\} = 0|\{\boldsymbol{\gamma}_{:,n,\{c,r\}} = 1\}, \cdot$.

- Update observation noise as in:

$$\begin{aligned} [\sigma_n^2]^{(j)} \sim \mathcal{IG}\big(&\alpha_{\sigma^2} + W/2, (\alpha_{\sigma^2} - 1)[\sigma_{n-1}^2]^{(j')} + \\ &\frac{1}{2}\|\mathbf{y}_n - \mathbf{G}\mathbf{c}_n^{(j)}\|^2\big) \end{aligned} \quad (13)$$

3) **Rejuvenating move**: on the grounds of kernel invariance[4], a Gibbs move is applied on the current values of the ancestor particle j': $\mathbf{c}_{n-1,r}^{(j')}$, $\boldsymbol{\gamma}_{n-1,r}^{(j')}$ $\nu_{n-1}^{(j')}$, $[\sigma_{n-1}^2]^{(j')}$ (note that $\mathbf{c}_{n,\ell}^{(j)}$, $\boldsymbol{\gamma}_{n,\ell}^{(j)}$ are also updated, since the density imposes $\mathbf{c}_{n,\ell}^{(j)} = \mathbf{c}_{n-1,r}^{(j')}$, Eq. 7). The Gibbs updates are slightly different from those described in step 2, and a detailed description can be found in Appendix B.

4) **Sampling a new ancestor**: execute joint move of variables $\{\mathbf{c}_{n-1}, \boldsymbol{\gamma}_{n-1}, \sigma_{n-1}^2\}$ (plus $\mathbf{c}_{n,\ell}, \boldsymbol{\gamma}_{n,\ell}$) conditional to the current values of state n: $\mathbf{c}_{n,\{c,r\}}^{(j)}, \boldsymbol{\gamma}_{n,\{c,r\}}^{(j)}, [\sigma_n^2]^{(j)}$ and observations $\mathbf{y}_{1:n}$. Using the notation of Eq. 9, this step amounts to: i. weighting particles in $\hat{p}(x_{n-1}|\mathbf{y}_{1:n-1})$ (each of them having probability $1/P$) with densities $p(x_n|x_{n-1})$ and $p(\mathbf{y}_n|x_n)$ evaluated at the current values $x_n = x_n^{(j)}$; ii. sampling a new $x_{n-1}^{(j')}$ proportional to the resulting weights. An explanation is provided in Appendix C.

5) **Stopping condition:** If $j = N_{iter}$, stop, discard the burn-in period, thin the chain, and proceed to frame $n+1$. If not, set $j = j+1$ and repeat from step 2.

Akin to σ_n^2, the successive variables ν_n, ν_{n+1} could be linked through a Gamma Markov chain prior $p(\nu_n|\nu_{n-1})$ to

[3]Following the formulation of Eq. 8, we adopt n, the frame index, as the temporal dimension.

[4]Informally speaking, a transition kernel is invariant with respect to a distribution π if, when applied to an element drawn from π, the resulting sample is still marginally distributed according to π. Thus, if an invariant kernel is applied on the elements of a particle approximation $\hat{\pi}(x) = \{x_1, \dots, x_P\}, x_i \sim \pi(x)\ \forall i \in \{1, \dots, P\}$, the resulting collection of samples would still be a valid approximation of π.

mitigate temporal variations of the parameter, keeping the full conditional tractable. In practice, even an independent prior for each ν_n leads to good results. We conjecture that the assumption that values σ^2, ν are common to all frames (adopted in the batch approach, e.g., [8]), may be overly restrictive, whereas assuming slowly varying values may be better suited to capture local changes in the time series, such as variations in the background noise or the volume of the signal overall—synthesis coefficients tend to be smaller in sections where the volume is lower, and assuming a common value of ν for all frames may fail to represent this trend.

As anticipated in Section III, the state-space model is stiff in the sense that the coefficients associated to the frames in the "interfaces" between two consecutive chunks of the signal (e.g., $\bar{\mathbf{c}}_{1+1/2}$ in Figure 1) depend directly on both observations $\mathbf{y}_n$, $\mathbf{y}_{n+1}$. However, in a standard filtering scheme, the values of $\mathbf{c}_{n,\ell}$ are estimated "greedily", based on the information available up to the current time step and are left fixed for the next iteration (Eq. 6). The rejuvenating move of step 3 is introduced with the aim of updating the current values of the coefficients $\mathbf{c}_{n-1,r} = \mathbf{c}_{n,\ell}$ with the information provided by $\mathbf{y}_{n+1}$. Although not a full smoothing scheme, this additional step allows for refined estimates of the state and increased particle diversity, two aspects that ultimately translate into a better signal reconstruction.

The "ancestor sampling" procedure of step 4, although theoretically principled and empirically backed by previous works (e.g., [13]), has some drawbacks in this specific context: since the weights assigned to particles $\hat{p}(\mathbf{c}_{n-1}, \boldsymbol{\gamma}_{n-1}, \sigma^2_{n-1}, \nu_{n-1}|\mathbf{y}_{1:n-1})$ are proportional to the likelihood function $p(\mathbf{y}_n|\mathbf{c}_n^{(j')}, \sigma^2_n)$ (Eq. 15 in Appendix C), these values are dominated by the exponential term $\exp \frac{1}{2\sigma^2_n}\|\mathbf{y}_n - \mathbf{G}\mathbf{c}_n^{(j')}\|^2$, where $\mathbf{c}_n^{(j')} = [\mathbf{c}_{n-1,r}^{(j')}; \mathbf{c}_{n,c}^{(j)}; \mathbf{c}_{n,r}^{(j)}]$ (i.e., the leftmost set of coefficients taken from the particles in the approximate filtering distribution, and the rest taken from the current values of the MCMC iteration). Thus, despite containing additional terms such as $p(\sigma^2_n|\sigma^2_{n-1})$ and the Markov Chain dependencies of indicator variables, the weights act as a soft-max operator in practice, which results in the repeated selection of a same single particle over the successive MCMC iterations, leading to a lack of particle diversity (which is partially alleviated by the rejuvenating scheme of step 3). This issue sheds light on the difficulties of implementing an Particle Filter strategy to compute this model, since a similar behaviour could be expected in the resampling step. Similarly, when discussing Sequential Monte Carlo Samplers, [27], Chapter 17, points out that those techniques are of little help when the observations are highly informative, as occurs with the audio signal model proposed here.

Algorithm 1: Sequential MCMC and signal reconstruction

- Intialise chain values as explained in step 1 (Section IV). **for** $n = 1, \dots, N$ **do**
    - **for** $j = 1, \dots, N_{burn-in} + N_{samples}$ **do**
        - • Draw a new value of $s^{(j)}_{m,n,p}$ as in Eq. 10 (for $m = 0, \dots, M-1, p = c, r$)
        - **for** $p = c, r$ **do**
            - **for** $m = 0, \dots, M-1$ **do**
                - • Draw $c^{(j)}_{m,n,p}$ and $\gamma^{(j)}_{m,n,p}$ from Eq. 11
            - **end**
        - **end**
        - • Draw $\nu_n^{(j)}$ from Eq. 12
        - • Draw $[\sigma^2_n]^{(j)}$ from Eq. 13
    - **end**
    - **if** $n = 1$ **then**
        - • Perform a regular Gibbs sweep on $\mathbf{c}^{(j)}_{n,\ell}$, $\boldsymbol{\gamma}^{(j)}_{n,\ell}$.
    - **else**
        - • Apply rejuvenating move on $c^{(j)}_{m,n-1,r} = c^{(j)}_{m,n,\ell}$, $\gamma^{(j)}_{m,n-1,r} = \gamma^{(j)}_{m,n,\ell}$ $[\sigma^2_{n-1}]^{(j)}$, $\nu^{(j)}_{n-1}$, as explained in step 3 (Section IV) and Appendix B.
        - • Pick an ancestor (j') as explained in step 4 (Section IV) and Appendix C. Assign $\mathbf{c}^{(j+1)}_{n-1} = \mathbf{c}^{(j')}_{n-1}$, $\boldsymbol{\gamma}^{(j+1)}_{n-1} = \boldsymbol{\gamma}^{(j')}_{n-1}$, $\nu^{(j+1)}_{n-1} = \nu^{(j')}_{n-1}$, $[\sigma^2_{n-1}]^{(j+1)} = [\sigma^2_{n-1}]^{(j')}$
    - **end**
    - • $\hat{\mathbf{c}}_{n,\ell} = \sum_j \mathbf{c}^{(j)}_{n,\ell}$, $\hat{\mathbf{c}}_{n,c} = \sum_j \mathbf{c}^{(j)}_{n,c}$, where j goes from $N_{burn-in} + 1$ to $N_{burn-in} + N_{chain}$ $\big($and $\hat{\mathbf{c}}_{n,r} = \sum_j \mathbf{c}^{(j)}_{n,r}$ if $n = N\big)$.
- **end**

B. Signal reconstruction

In batch settings, estimates of the underlying audio signal are typically based on the conditional distribution $p(\mathbf{c}_{1:N}|\mathbf{y}_{1:N})$. Similarly, in sequential settings, estimates of the trajectory $\mathbf{c}_{1:N}$ can be refined through a backwards smoothing scheme. The filtering scheme adopted here generates samples of trajectories up to time n ($\mathbf{c}_{1:n}$) that could arguably be used to calculate estimates of the form $\hat{\mathbf{c}}_{1:n} = \sum_j \mathbf{c}^{(j)}_{1:n}$. However, in Particle Filter and related schemes, current particle values at time n (x_n) can normally be traced back to a single or just a few ancestor particles, thus losing much of the *finesse* with which the filtering distribution is characterised at previous iterations ($\hat{p}(x_{n-1}|y_{1:n-1})$, $\hat{p}(x_{n-2}|y_{1:n-2})$, and so on). For that reason, and bearing in mind that the proposed framework has real-time applications as its ultimate objective, we construct the estimate in an online fashion, largely based on the approximate filtering distribution at time n, $\hat{p}(\mathbf{c}_n|\mathbf{y}_{1:n}) = \sum_j \delta_{\mathbf{c}_n^{(j)}}(\mathbf{c}_n)$, rather than on estimates of the entire preceding trajectories. Due to the incorporation of a rejuvenating scheme, values of $\hat{\mathbf{c}}_{n,r}$ resulting from the n-th iteration are not used in the state estimate $\hat{\mathbf{c}}_n$; instead, $\hat{\mathbf{c}}_{n+1,\ell} = \hat{\mathbf{c}}_{n,r}$ are updated after the $(n+1)$-th iteratin is concluded. Both the filtering strategy and the (**semi-heuristic? DISCUSS**) signal reconstruction procedure are summarised in Algorithm 1.

V. Experiments

The proposed algorithm is tested on three music excerpts: a short glockenspiel record sampled at 22,050 [Hz] (3 seconds long, input SNR 10 dB, window size 1024 samples); a longer gramophone recording of a jazz band with solo trumpet sampled at 22,050 [Hz] (10 seconds long, input SNR 20 dB, window size 1024 samples); a vibraphone extract sampled at 22,050 [Hz] (8.5 seconds long, SNR input = 20 dB, window size 2048). Audible results available in https://matclaveria.github.io/sMCMC.html.

Output SNR results missing in this version. Informal report: they are inferior to the output SNR values obtained with batch methods but display "nice" audible qualities: less artifacts than batch Gibbs sampling [6] and a more "organic" sound texture than the reconstructions obtained with the EMVS algorithm proposed in my other draft (EMVS produces extremely midi-like, robotic sounding output signals in low SNR regimes, although has great potential for real-time restoration in medium SNR scenarios).

VI. Final remarks

(one short paragraph explaining) 1 validation of the sequential model 2 lower computational expense than batch, but still not feasible for real-time 3 a sMCMC scheme was proposed and tested, but faster methods are in demand. potentially interesting: variational approx. of filtering distributions. Particle filter and SMC samplers may

Appendix

A. Conditional distribution formulae

The quantities used in the joint sampling of indicator (γ) and regression coefficients (c) are detailed below. All of them stem from the full conditionals $p(c, \gamma | \dots)$. Term $-m$ denotes $\{0, M-1\} \setminus \{m\}$.

$$
\begin{aligned}
\Sigma_{m,n,p} &= \left(\mathbf{G}_{m,p}^\top \mathbf{G}_{m,p} + \frac{[\sigma_n^2]^{(j-1)}}{s_{m,n,p}^{(j)}} \mathbf{I}_2 \right)^{-1} \\
\mu_{m,n,p} &= \Sigma_{m,n,p} \mathbf{G}_{m,p}^\top [\mathbf{y}_n - \mathbf{G}_{-m,p} \mathbf{c}_{-m,n,p}] \\
\tau_{m,n,p} &= \frac{p(\gamma_{m,n,p} = 1 | \boldsymbol{\gamma}_{1:n}^{(j)} \setminus \gamma_{m,n,p}, \phi)}{p(\gamma_{m,n,p} = 0 | \boldsymbol{\gamma}_{1:n}^{(j)} \setminus \gamma_{m,n,p}, \phi)} \frac{[\sigma_n^2]^{(j-1)}}{s_{m,n,p}^{(j)}} \cdot \\
&\qquad |\Sigma_{m,n,p}|^{1/2} \exp\left(\frac{\mu_{m,n,p}^\top \Sigma_{m,n,p}^{-1} \mu_{m,n,p}}{2[\sigma_n^2]^{(j-1)}} \right)
\end{aligned}
$$

B. Rejuvenating step formulae

The formulae in Appendix A is valid for $p \in \{c, r\}$. In the rejuvenating step, where a Gibbs kernel is applied on the current values of $\mathbf{c}_n^{(\ell)} = \mathbf{c}_{n-1}^{(r)}$ in order to add diversity to the samples, slightly different formulae stem from the respective conditional distribution.

$$
\begin{aligned}
\mathbf{I}_\ell &= \begin{bmatrix} [\sigma_{n-1}^2]^{(j')} \mathbf{I}_{W/2} & \mathbf{0} \\ \mathbf{0} & [\sigma_n^2]^{(j-1)} \mathbf{I}_{W/2} \end{bmatrix} \\
\Sigma_{m,n,\ell} &= \left(\mathbf{G}_{m,c}^\top \mathbf{I}_\ell^{-1} \mathbf{G}_{m,c} + \frac{1}{s_{m,n,\ell}^{(j)}} \mathbf{I}_2 \right)^{-1} \\
\mu_{m,n,p} &= \Sigma_{m,n,\ell} \mathbf{G}_{m,c}^\top \mathbf{I}_\ell^{-1} [\mathbf{y}_{n-1/2} - \mathbf{G}_{-m,c} \mathbf{c}_{-m,n,\ell}] \\
\tau_{m,n,\ell} &= \frac{p(\gamma_{m,n,\ell} = 1 | \boldsymbol{\gamma}^{(j)} \setminus \gamma_{m,n,\ell}, \phi)}{p(\gamma_{m,n,\ell} = 0 | \boldsymbol{\gamma}^{(j)} \setminus \gamma_{m,n,\ell}, \phi)} \frac{|\Sigma_{m,n,\ell}|^{1/2}}{s_{m,n,\ell}^{(j)}} \cdot \\
&\qquad \exp\left(\frac{\mu_{m,n,\ell}^\top \Sigma_{m,n,\ell}^{-1} \mu_{m,n,\ell}}{2} \right)
\end{aligned}
$$

$$
\begin{aligned}
\gamma_{m,n,\ell}^{(j)} &\sim \text{Bernoulli}\left(\frac{\tau_{m,n,\ell}}{1 + \tau_{m,n,\ell}} \right) \\
c_{m,n,\ell}^{(j)} &\begin{cases} = 0 & \text{if } \gamma_{m,n,\ell}^{(j)} = 0 \\ \sim \mathcal{N}(\mu_{m,n,\ell}, \Sigma_{m,n,\ell}) & \text{if } \gamma_{m,n,\ell}^{(j)} = 1 \end{cases}
\end{aligned}
$$

Whereas ν_{n-1} can be updated exactly as in step 2 (Section IV), the updates of σ_{n-1}^2 in the rejuvenation step must be executed in a slightly different way. Since we are conditioning on the remaining values of the "path" defined by particle (j), $[\sigma_{n-1}^2]^{(j)}$ now depends on the values of both the previous and the next states. The resulting conditional distribution is a Generalised inverse Gaussian distribution ($\mathcal{GIG}$) with the values described below (index j omitted for ease of notation):

$$
\begin{aligned}
&\sigma_{n-1}^2 \sim \mathcal{GIG}(\sigma_{n-1}^2; a, b, p), \qquad a = \frac{\alpha_{\sigma^2} - 1}{\sigma_n^2}, \\
&p = -\alpha_{\sigma^2} - \frac{W}{2}, \; b = \left[\frac{1}{2} (\mathbf{y}_{n-1} - \mathbf{G} \mathbf{c}_{n-1}) + (\alpha_{\sigma^2} - 1) \sigma_{n-2}^2 \right],
\end{aligned}
$$

where:

$$
\mathcal{GIG}(x; a, b, p) = \frac{(a/b)^{p/2}}{2 K_p(\sqrt{ab})} x^{(p-1)} e^{-(ax + b/x)/2}.
$$

Note that σ_{n-1}^2 depends on σ_{n-2}^2 (two time steps behind current state n). That does not pose a problem since, although equations in Section IV are written in terms of (x_k, x_{k+1}), the particle approximation actually saves the entire trajectory from time 1 to the previous time step k: $\{x_{1:k}^{(j)}\}_{j'} = 1, \dots, P$. Then, the updates described for $\mathbf{c}_{n-1,r}^{(j)}$, $\gamma_{n-1,r}^{(j)}$ and $[\sigma_{n-1}^2]^{(j')}$ correspond to MCMC "perturbations" applied to the respective trajectory, as explained in [26].

C. Sampling the previous state

For clarity, we show the derivaton of Eq. 9:

$$
\begin{aligned}
p(x_k, x_{k+1} | y_{1:k+1}) &= \frac{p(x_k, x_{k+1}, y_{k+1} | y_{1:k})}{p(y_{k+1} | y_{1:k})} \\
&= \frac{p(y_{k+1} | x_{k+1}) p(x_k, x_{k+1}, | y_{1:k})}{p(y_{k+1} | y_{1:k})} \\
&= \frac{p(y_{k+1} | x_{k+1}) p(x_k | y_{1:k}) p(x_{k+1} | x_k)}{p(y_{k+1} | y_{1:k})}
\end{aligned}
$$

The sequential MCMC algorithm described in Section IV relies on Gibbs sampling to generate samples from $p(x_k, x_{k+1}|y_{1:k+1})$, alternating draws from $p(x_{k+1}|x_k, y_{1:k+1})$ and $p(x_k|x_{k+1}, y_{1:k+1})$. Whereas sampling from $p(x_{k+1}|x_k, y_{1:k+1})$ is straightforward (provided that a suitable strategy exists to sample the resulting conditional, e.g., tractable component-wise Gibbs steps), the procedure to sample $x_k|x_{k+1}, y_{1:k+1}$ may not be obvious when $p(x_k|y_{1:k})$ is a particle approximation. However, conditioning on x_{k+1} just introduces normalisation variables:

$$\begin{aligned} p(x_k|x_{k+1}, y_{1:k+1}) =& \frac{p(x_k, x_{k+1}|y_{1:k+1})}{p(x_{k+1}|y_{1:k+1})} \\ =& \frac{p(y_{k+1}|x_{k+1})p(x_k|y_{1:k})p(x_{k+1}|x_k)}{p(y_{k+1}|y_{1:k})p(x_{k+1}|y_{1:k+1})} \\ \propto& p(x_k|y_{1:k})p(x_{k+1}|x_k). \end{aligned} \tag{14}$$

Replacing $p(x_k|y_{1:k})$ for $\hat{p}(x_k|y_{1:k})$, it is easy to see that sampling from the approximate distribution amounts to weight the particles by terms depending on both x_k and x_{k+1}:

$$\hat{p}(x_k|x_{k+1}, y_{1:k+1}) \propto \sum_{j'} \delta_{x_k^{(j')}}(x_k)p(x_{k+1}|x_k^{(j')})$$

The restriction $\mathbf{c}_{n,\ell} = \mathbf{c}_{n-1,r}$ (analogous for $\boldsymbol{\gamma}$) renders impossible to sample a particle different from the current value $\mathbf{c}_{n-1}^{(j')}$ using the scheme described above, since none of the remaining particles satisfies the condition $\mathbf{c}_{n-1,r} = \mathbf{c}_{n,\ell}^{(j)}$. However, a joint move of $\mathbf{c}_{n_1}, \boldsymbol{\gamma}_{n-1}, \sigma_{n-1}^2, \nu_{n-1}$ (denoted $\{\mathbf{c}, \boldsymbol{\gamma}, \sigma^2, \nu\}_{n-1}$ for brevity) and $\mathbf{c}_{n,\ell}, \boldsymbol{\gamma}_{n,\ell}$ is possible following a similar reasoning to that of Eq. 14:

$$\begin{aligned} & p(\{\mathbf{c}, \boldsymbol{\gamma}, \sigma^2, \nu\}_{n-1}, \mathbf{c}_{n,\ell}, \boldsymbol{\gamma}_{n,\ell}|\sigma_n^2, \nu_n, \mathbf{c}_{n,\{c,r\}}, \boldsymbol{\gamma}_{n,\{c,r\}}, \mathbf{y}_{1:n}) \\ =& \frac{p(\{\mathbf{c}, \boldsymbol{\gamma}, \sigma^2, \nu\}_{n-1}, \mathbf{c}_{n,\ell}, \boldsymbol{\gamma}_{n,\ell}, \sigma_n^2, \nu_n, \mathbf{c}_{n,\{c,r\}}, \boldsymbol{\gamma}_{n,\{c,r\}}|\mathbf{y}_{1:n})}{p(\sigma_n^2, \nu_n, \mathbf{c}_{n,\{c,r\}}, \boldsymbol{\gamma}_{n,\{c,r\}}|\mathbf{y}_{1:n})} \\ =& \frac{p(\{\mathbf{c}, \boldsymbol{\gamma}, \sigma^2, \nu\}_{n-1}|\mathbf{y}_{1:n-1}) \cdot p(\mathbf{y}_n|\mathbf{c}_{n,\ell}, \mathbf{c}_{n,\{c,r\}}, \sigma_n^2)}{p(\mathbf{y}_n|\mathbf{y}_{1:n-1})} \cdot \\ & \frac{p(\mathbf{c}_n, \boldsymbol{\gamma}_n, \sigma_n^2, \nu_n|\mathbf{c}_{n-1}, \boldsymbol{\gamma}_{n-1}, \sigma_{n-1}^2, \nu_{n-1})}{p(\sigma_n^2, \nu_n, \mathbf{c}_{n,\{c,r\}}, \boldsymbol{\gamma}_{n,\{c,r\}}|\mathbf{y}_{1:n})}, \end{aligned}$$

thus, replacing $p(\{\mathbf{c}, \boldsymbol{\gamma}, \sigma^2, \nu\}_{n-1}|\mathbf{y}_{1:n-1})$ with the particle approximation and keeping the terms of $p(\mathbf{c}_n, \boldsymbol{\gamma}_n, \sigma_n^2, \nu_n|\mathbf{c}_{n-1}, \boldsymbol{\gamma}_{n-1}, \sigma_{n-1}^2, \nu_{n-1})$ that actually depend on previous state, the approximate conditional density of $\{\mathbf{c}, \boldsymbol{\gamma}, \sigma^2, \nu\}_{n-1}, \mathbf{c}_{n,\ell}, \boldsymbol{\gamma}_{n,\ell}$ has the form:

$$\begin{aligned} & \hat{p}(\{\mathbf{c}, \boldsymbol{\gamma}, \sigma^2, \nu\}_{n-1}, \mathbf{c}_{n,\ell}, \boldsymbol{\gamma}_{n,\ell}|\sigma_n^2, \nu_n, \mathbf{c}_{n,\{c,r\}}, \boldsymbol{\gamma}_{n,\{c,r\}}, \mathbf{y}_{1:n}) \\ & \propto \sum_{j'} \delta_{\mathbf{c}_{n-1}^{j'}, \boldsymbol{\gamma}_{n-1}^{j'}, [\sigma_{n-1}^2]^{j'}, \nu_{n-1}^{j'}}(\mathbf{c}_{n-1}, \boldsymbol{\gamma}_{n-1}, \sigma_{n-1}^2, \nu_{n-1}) \cdot \\ & \qquad \delta_{\mathbf{c}_{n-1,r}^{j'}, \boldsymbol{\gamma}_{n-1,r}^{j'}}(\mathbf{c}_{n,\ell}, \mathbf{c}_{n,\ell}) \cdot p(\mathbf{y}_n|\mathbf{c}_{n,\ell}, \mathbf{c}_{n,\{c,r\}}, \sigma_n^2) \cdot \\ & \qquad p(\boldsymbol{\gamma}_{n,c}|\boldsymbol{\gamma}_{n,\ell}) \cdot p(\sigma_n^2|[\sigma_{n-1}^2]^{j'}), \end{aligned} \tag{15}$$

which can be sampled–again–just by weighting the particles by the relevant densities evaluated at $\mathbf{c}_{n,\ell}, \boldsymbol{\gamma}_{n,\ell}, \sigma_n^2$. Note that, due to the constraints imposed by Eq. 7, $p(\mathbf{y}_n|\mathbf{c}_{n,\ell}, \mathbf{c}_{n,\{c,r\}}, \sigma_n^2)$ takes values $p(\mathbf{y}_n|\mathbf{c}_{n,\ell} = \mathbf{c}_{n-1,r}^{j'}, \mathbf{c}_{n,\{c,r\}}, \sigma_n^2)$.

References

[1] S. G. Mallat and Z. Zhang, “Matching pursuits with time-frequency dictionaries,” *IEEE Transactions on signal processing*, vol. 41, no. 12, pp. 3397–3415, 1993.

[2] S. Chen and D. Donoho, “Basis pursuit,” in *Proceedings of 1994 28th Asilomar Conference on Signals, Systems and Computers*, vol. 1. IEEE, 1994, pp. 41–44.

[3] R. Neff and A. Zakhor, “Very low bit-rate video coding based on matching pursuits,” *IEEE Transactions on circuits and systems for video technology*, vol. 7, no. 1, pp. 158–171, 1997.

[4] F. Abramovich, T. Sapatinas, and B. W. Silverman, “Wavelet thresholding via a bayesian approach,” *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 60, no. 4, pp. 725–749, 1998.

[5] M. S. Crouse, R. D. Nowak, and R. G. Baraniuk, “Wavelet-based statistical signal processing using hidden markov models,” *IEEE Transactions on signal processing*, vol. 46, no. 4, pp. 886–902, 1998.

[6] P. J. Wolfe, S. J. Godsill, and W.-J. Ng, “Bayesian variable selection and regularization for time–frequency surface estimation,” *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 66, no. 3, pp. 575–589, 2004.

[7] P. J. Wolfe and S. J. Godsill, “Interpolation of missing data values for audio signal restoration using a gabor regression model,” in *Proceedings.(ICASSP'05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.*, vol. 5. IEEE, 2005, pp. v–517.

[8] C. Févotte, B. Torrésani, L. Daudet, and S. J. Godsill, “Sparse linear regression with structured priors and application to denoising of musical audio,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 1, pp. 174–185, 2007.

[9] J. Murphy and S. Godsill, “Joint bayesian removal of impulse and background noise,” in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2011, pp. 261–264.

[10] C. Robert and G. Casella, “Monte carlo statistical methods,” *Springer, New York*, 2004.

[11] C. Berzuini, N. G. Best, W. R. Gilks, and C. Larizza, “Dynamic conditional independence models and markov chain monte carlo methods,” *Journal of the American Statistical Association*, vol. 92, no. 440, pp. 1403–1412, 1997.

[12] Z. Khan, T. Balch, and F. Dellaert, “Mcmc-based particle filtering for tracking a variable number of interacting targets,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 27, no. 11, pp. 1805–1819, 2005.

[13] F. Septier, S. K. Pang, A. Carmi, and S. Godsill, “On mcmc-based particle methods for bayesian filtering: Application to multitarget tracking,” in *2009 3rd IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*. IEEE, 2009, pp. 360–363.

[14] E. I. George and R. E. McCulloch, “Approaches for bayesian variable selection,” *Statistica sinica*, pp. 339–373, 1997.

[15] F. Li and N. R. Zhang, “Bayesian variable selection in structured high-dimensional covariate spaces with applications in genomics,” *Journal of the American statistical association*, vol. 105, no. 491, pp. 1202–1214, 2010.

[16] T. Park and G. Casella, “The bayesian lasso,” *Journal of the American Statistical Association*, vol. 103, no. 482, pp. 681–686, 2008.

[17] M. E. Tipping, “Sparse bayesian learning and the relevance vector machine,” *Journal of machine learning research*, vol. 1, no. Jun, pp. 211–244, 2001.

[18] D. F. Andrews and C. L. Mallows, “Scale mixtures of normal distributions,” *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 36, no. 1, pp. 99–102, 1974.

[19] R. E. Turner and M. Sahani, “Demodulation as probabilistic inference,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 8, pp. 2398–2411, 2011.

[20] A. T. Cemgil, H. J. Kappen, and D. Barber, “A generative model for music transcription,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 2, pp. 679–694, 2006.

[21] R. G. Everitt, C. Andrieu, and M. Davy, “Online bayesian inference in some time-frequency representations of non-stationary processes,” *IEEE transactions on signal processing*, vol. 61, no. 22, pp. 5755–5766, 2013.

[22] R. E. Turner and M. Sahani, “Time-frequency analysis as probabilistic inference,” *IEEE Transactions on Signal Processing*, vol. 62, no. 23, pp. 6171–6183, 2014.

[23] A. T. Cemgil and S. J. Godsill, “Probabilistic phase vocoder and its application to interpolation of missing values in audio signals,” in *2005 13th European Signal Processing Conference*. IEEE, 2005, pp. 1–4.

[24] W. J. Wilkinson, M. R. Andersen, J. D. Reiss, D. Stowell, and A. Solin, "Unifying probabilistic models for time-frequency analysis," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 3352–3356.

[25] T. Strohmer, "Numerical algorithms for discrete gabor expansions," in *Gabor analysis and algorithms*. Springer, 1998, pp. 267–294.

[26] W. R. Gilks and C. Berzuini, "Following a moving target—monte carlo inference for dynamic bayesian models," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 63, no. 1, pp. 127–146, 2001.

[27] N. Chopin and O. Papaspiliopoulos, "Smc samplers," in *An Introduction to Sequential Monte Carlo*. Springer, 2020, pp. 329–355.