Simulation-Based Methods for Blind Maximum-Likelihood Filter Identification

Olivier Cappé *, Arnaud Doucet †, Marc Lavielle ‡ and Eric Moulines *

* ENST Département TSI / CNRS URA 820
46 rue Barrault, 75634 Paris cedex 13, France
cappe, moulines @tsi.enst.fr

† CEN Saclay LETI/DEIN/SPE and ETIS - ENSEA Groupe Signal
95014 Cergy Pontoise cedex, France
Now with Cambridge University Department of Engineering, Signal Processing Group
ad2@eng.cam.ac.uk

‡ UFR de Mathématiques et Informatique, Université Paris V
45 rue des Saints-Pères, 75006 Paris, France
lavielle@math-info.univ-paris5.fr

Abstract

Blind linear system identification consists in estimating the parameters of a linear time-invariant system given its (possibly noisy) response to an unobserved input signal. Blind system identification is a crucial problem in many applications which range from geophysics to telecommunications, either for its own sake or as a preliminary step towards blind deconvolution (ie. recovery of the unknown input signal). This paper presents a survey of recent stochastic algorithms, related to the Expectation-Maximixation (EM) principle, that make it possible to estimate the parameters of the unknown linear system in the maximum likelihood sense. Emphasis is on the computational aspects rather than on the theoretical questions. A large section of the paper is devoted to numerical simulations techniques, adapted from the Markov Chain Monte Carlo (MCMC) methodology, and their efficient application to the noisy convolution model under consideration.

Keywords Blind system identification, Maximum likelihood estimation, Expectation Maximization (EM), Stochastic algorithms, Markov Chain Monte Carlo (MCMC)

1 Introduction

Blind linear system identification consists in estimating the parameters of a linear time-invariant system given its (possibly noisy) response to an unobserved input signal. In many applications, the ultimate goal is indeed blind deconvolution which aims at recovering the unobserved input signal itself. Most blind deconvolution approaches, however rely on the blind identification of the filter coefficients. These two problems have been the topic of a large number of contributions in the recent years (see Donoho [21] and [47] for early references; Nikias and Mendel [51], Haykin [30], Cadzow [6] and the references therein for an updated account). Many of these contributions deal with higher-order cumulants and polyspectral techniques, based on the pioneering contribution by Lii and Rosenblatt [33]. This approach was later extended in a series of papers by Giannakis and Mendel [27], Tugnait [65] (among many others).

In this paper, the focus is on the blind identification issue and *maximum likelihood* estimation of the model parameters (which include the filter coefficients, and possibly some characteristics of the noise and/or of the input signal) is considered. Contrary to cumulant or polyspectral techniques, maximum likelihood exploits all the available information on the probability distributions of the input and noise, which improves the accuracy of the parameter estimates. Maximum likelihood for noisy blind deconvolution problems has been only scarcely addressed in the signal processing literature, because the likelihood function cannot, in most cases, be expressed in a numerically tractable analytic form. In this context, simulation-based numerical optimization approaches provide a powerful alternative to their (more well-known) deterministic counterparts, such as the Expectation Maximization (EM) algorithm. This paper is mainly concerned with algorithmic issues, and intends to provide some answers on how to implement maximum likelihood in the blind deconvolution / system identification context. Related theoretical aspects such as consistency, asymptotic normality, asymptotic information bound, will thus be left aside.

The paper is organized as follows: In section 2, the general blind identification problem is described along with needed definitions and assumptions. Iterative stochastic algorithms for obtaining maximum-likelihood parameter estimates are introduced in section 3. These techniques usually relies on a data-augmentation strategy which requires conditional simulations of the missing inputs. Simulation strategies that may be used for carrying out such a task are far from being trivial and are discussed in detail in section 4. Finally, some simulation results are presented in section 5.

Remark

The question of knowing whether it is necessary to first identify the parameters of the model before attempting the deconvolution is an important and yet controversial methodological issue. The two-steps approach (identification of the filter parameters followed by deconvolution) is the most popular approach and (presumably) the most successful to date [40], [13]. In this approach, the recovery of the input sequence is usually carried out in a Bayesian framework by maximizing the posterior distribution of the input sequence given the filter parameters and the observations, or some computationally tractable approximation of this criterion. The use of a priori information in this context is of a prime importance since the deconvolution is in general an ill-behaved problem (with more "degrees of freedom" than the available number of observations) [47], [18]. The deterministic approaches devised for joint model identification and input recovery ("generalized likelihood", "deterministic maximum-likelihood"), although quite successful in the case of Single Input Multiple Output (SIMO) systems [42], [31], do not appear to be reliable for Single Input Single Output (SISO) systems [24]. A recent alternative approach for tackling this joint estimation problem consist in performing a so-called "fully Bayesian" simulation-based analysis (see, for instance, [23], [14], [15] and references therein). This way of proceeding is very different in spirit since it makes it possible to perform (at least conceptually)

various type of inferences, such as marginal estimation of the input sequence (where the filter coefficients are considered as a nuisance parameter and are marginalized out). In the rest of the paper, we will not discuss any further the deconvolution issue and we assume that the goal is indeed maximum likelihood filter identification.

2 Blind identification model

The estimation of non minimum phase system has received a considerable attention in the past decade. Most of the methods proposed to date are based on the higher-order statistics (the third-order or the fourth-order cumulants or the corresponding frequency-domain quantities, e.g. the bispectrum or the trispectrum) of the output (see for example [52],[65] and the references therein). These methods are most often straightforward to implement but are far from being optimal from a statistical point of view when an a priori information is available on the distribution of the input signal. Examples of this situation may be found in *geophysics* or in *digital communications* applications: In these cases, the distribution of the input signal is either known or at least can be modeled accurately. In these situations, important improvements in the performance of the estimates can be expected (and achieved in practice) by taking into account this information in the estimation procedure. A natural way to exploit this information consist in solving the blind identification problem in the maximum likelihood sense.

As outlined above, maximum likelihood has only scarcely be used for parameter estimation in noisy deconvolution problems, except when the input is discrete and belongs to a finite alphabet, a situation of interest in digital communication (see [34],[2],[60] and the references therein). Extensions to more general input models have only marginally been addressed.

2.1 Blind identification as an incomplete data problem

From a statistical point of view, blind identification is a typical example of a problem which involve unobserved data. Unobserved (also known as incomplete, or missing) data models forms a large and important class which has received a considerable interest in the statistical literature during recent years. Before going further, some notations and definitions are presented.

Let $\mathbf{y} \triangleq (y_1, \cdots, y_T)'$ denote the vector of observed data samples. It is assumed that

$$y_t = \sum_{l=0}^{p} h_l z_{t-l} + \sigma n_t \,, \tag{1}$$

where $\{z_t\}$ is the (*unobserved*) input sequence, $\mathbf{h} = (h_0, \cdots, h_p)'$ is the vector of MA coefficients, and $\{n_t\}$ is an (*unobserved*) additive noise. In this model, the input sequence $\mathbf{z} = (z_{1-p}, \cdots, z_T)'$ plays the role of the *missing data* and $(\mathbf{y}, \mathbf{z})$ is referred to as the *complete* data. It is further assumed that

- **(M1)** $\{z_t\}$ is an *iid.* sequence of random variables with known probability distribution function (pdf) $p(z_t)$ (with respect to some dominating measure μ_z).
- **(M2)** $\{n_t\}$ is an *iid.* sequence of zero-mean Gaussian variables with unit variance.
- **(M3)** The processes $\{z_t\}$ and $\{n_t\}$ are independent.

As is clear from above, we restrict ourself to the case of Single-Input Single-Output (SISO) moving-average (MA) models. The extension to Multiple-Input Multiple-Output (MIMO) MA models would be straightforward (except for added notational complexity). A more challenging question concerns the extension to, possibly non-causal, IIR filters models. Another possible extension is the case of non-Gaussian measurement noise $\{n_t\}$: assumption (M2) could be relaxed so as to allow for mixture of Gaussian, Laplacian, or other exponential family of pdf. These two

last points are discussed in some more details in the conclusion of the paper. Finally, assumption (M1) could be relaxed by assuming that the pdf $p(z_t)$ belongs to some known parametric family, and depends upon an unknown finite-dimensional parameter. The adaptations needed to handle this case are, at least conceptually, straightforward but it raises some important questions (identifiability, asymptotic efficiency) that are not considered here.

The parameters $\boldsymbol{\theta}$ of the model to be estimated includes both the filter coefficients $\mathbf{h}$ and the noise variance σ^2. Under these assumptions, the log-likelihood corresponding to the complete data is, up to constant terms,

$$\log p(\mathbf{z}, \mathbf{y}; \boldsymbol{\theta}) = -\frac{T}{2} \log \sigma^2 - \frac{1}{2\sigma^2} \sum_{t=1}^{T} (y_t - \mathbf{h}' \underline{\boldsymbol{z}}_t)^2 \,, \tag{2}$$

which can be rewritten as

$$\log p(\mathbf{z}, \mathbf{y}; \boldsymbol{\theta}) = L(\mathbf{S}_1(\mathbf{z}), \mathbf{S}_2(\mathbf{z}); \boldsymbol{\theta}) \,, \tag{3}$$

with

$$\begin{aligned} L(\mathbf{S}_1, \mathbf{S}_2; \boldsymbol{\theta}) &= -\frac{T}{2} \log \sigma^2 - \frac{1}{2\sigma^2} \left(\sum_{t=1}^{T} y_t^2 - 2\mathbf{h}' \mathbf{S}_1 + \mathbf{h}' \mathbf{S}_2 \mathbf{h} \right) \,, \\ \mathbf{S}_1(\mathbf{z}) &= \sum_{t=1}^{T} y_t \underline{\boldsymbol{z}}_t \,, \\ \mathbf{S}_2(\mathbf{z}) &= \sum_{t=1}^{T} \underline{\boldsymbol{z}}_t \underline{\boldsymbol{z}}_t' \,, \end{aligned} \tag{4}$$

where $\underline{\boldsymbol{z}}_t \triangleq (z_t, \cdots, z_{t-p})'$. $\mathbf{S}_1(\mathbf{z})$ and $\mathbf{S}_2(\mathbf{z})$ are the sufficient statistics for the complete-data model (dependence of $\mathbf{S}_i(\mathbf{z})$ on $\mathbf{y}$ is implicit). Maximum Likelihood Estimates (MLE) of the unknown parameters in the complete data model are given by

$$\begin{aligned} \hat{\mathbf{h}} &= \mathbf{S}_2(\mathbf{z})^{-1} \mathbf{S}_1(\mathbf{z}) \,, \\ \hat{\sigma}^2 &= \frac{1}{T} \left(\sum_{t=1}^{T} y_t^2 - \hat{\mathbf{h}}' \mathbf{S}_1(\mathbf{z}) \right) \,. \end{aligned} \tag{5}$$

Unfortunately, when the input data $\mathbf{z}$ is not observed, the actual likelihood corresponding to the observed data only is obtained by marginalization of (2), that is by integrating over the values of the unobserved input data sequence:

$$p(\mathbf{y}; \boldsymbol{\theta}) = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{+\infty} p(\mathbf{y}, \mathbf{z}; \boldsymbol{\theta}) \prod_{t=1-p-1}^{T} p(z_t) \mu_z(dz_t) \,. \tag{6}$$

Because of the multiple integration, (6) cannot in general be evaluated in a tractable analytic form.

2.2 The EM paradigm

The EM algorithm can be seen as an iterative method for finding the modes of the likelihood function, which is extremely useful for models where it is hard to maximize the likelihood directly but easy to work with the 'complete' data model. The EM algorithm formalizes a relatively old idea for handling missing data: Starting with a guess of the parameters, (1) replace the missing values by their expectations given the guessed parameters, (2) estimate parameters assuming the missing data are given by their estimated values, (3) reestimate the missing values assuming the

new parameter estimates are correct, (4) reestimate parameters, and so forth, until convergence. In fact, the EM algorithm is more efficient than these four steps would suggest since each missing data value is not estimated separately; instead those functions of the missing data that are needed to estimate the model parameters are estimated jointly.

The name "EM" comes from the two alternating steps: Computation of the *expectation* of the needed functions (or in other words, sufficient statistics) of the missing values, and estimation of the parameters by *maximization* using the expected values of the sufficient as if they had been computed from observed values of the missing data (see [19], [62], the historical review of [32], as well as recent developments in [48]). More precisely, denote $\theta^{(n-1)}$ the current fit of the parameter before the nth iteration of the algorithm. At iteration n, the E-step amounts to computing

$$Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(n-1)}) = E\left(\log p(\mathbf{y},\mathbf{z};\boldsymbol{\theta})|\,\mathbf{y};\theta^{(n-1)}\right)\,. \tag{7}$$

The M-step consists in finding the parameter $\boldsymbol{\theta}^{(n)}$ that maximizes $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(n-1)})$ in the feasible set Θ. For the convolution model, it is easily seen from (3) and (4) that $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(n-1)})$ may be written as

$$Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(n-1)}) = L(\bar{\boldsymbol{S}}_1(\boldsymbol{\theta}^{(n-1)}), \bar{\boldsymbol{S}}_2(\boldsymbol{\theta}^{(n-1)});\boldsymbol{\theta})\,, \tag{8}$$

where

$$\begin{aligned}\bar{\boldsymbol{S}}_1(\boldsymbol{\theta}) &= \sum_{t=1}^{T} y_t E(\underline{\boldsymbol{z}}_t|\mathbf{y};\boldsymbol{\theta})'\,,\\ \bar{\boldsymbol{S}}_2(\boldsymbol{\theta}) &= \sum_{t=1}^{T} E(\underline{\boldsymbol{z}}_t\underline{\boldsymbol{z}}_t'|\mathbf{y};\boldsymbol{\theta})\,.\end{aligned} \tag{9}$$

The maximization step is thus carried out as in (5), replacing the complete data sufficient statistics by their expected values. The main difficulty with this scheme is that direct computation of $E(\underline{\boldsymbol{z}}_t|\mathbf{y};\boldsymbol{\theta})$ and $E(\underline{\boldsymbol{z}}_t\underline{\boldsymbol{z}}_t'|\mathbf{y};\boldsymbol{\theta})$ is, for many source signal models, intractable. The only exceptions to that rule are when *(i)* the source is Gaussian - which is of course only of marginal interest in a blind identification context because of the inherent limited identifiability of the filter - and when *(ii)* source is discrete. The later case is of particular interest in digital communications applications. It has been addressed by many authors, after the pioneering contribution of Kaleh and Vallet [34] (see also [60] and [2]). The special feature of the discrete (finite) case is that the expectation of any function of the unobserved input signal z_t can be evaluated from the probabilities $P(z_t) = v_k$, where $v_1, \cdots, v_K$ are the possible values of the input signal. More precisely, the state vector $\underline{\boldsymbol{z}}_t$ defined in the previous section can take at most $M = K^{(p+1)}$ different values which we denote by $\underline{\boldsymbol{v}}_m$. Eq. (9) thus reduces to

$$\bar{\boldsymbol{S}}_i^{(n)} = \sum_{t=1}^{T} y_t \sum_{m=1}^{M} \mathbf{S}_i(\underline{\boldsymbol{v}}_m) P(\underline{\boldsymbol{z}}_t = \underline{\boldsymbol{v}}_m|\mathbf{y};\boldsymbol{\theta}^{(n-1)}) \quad \text{for} \quad i = 1,2\,. \tag{10}$$

Moreover, the posterior probabilities $P(\underline{\boldsymbol{z}}_t = \underline{\boldsymbol{v}}_m|\mathbf{y};\boldsymbol{\theta}^{(n-1)})$ that appear in (10), can be computed efficiently using a two-pass algorithm introduced by Baum and his colleagues in the early 1970s, which is known as the *Forward-backward* procedure [55], [45]. In all other situations, the EM paradigm is not directly exploitable, and need some adaptations.

3 EM-related stochastic algorithms

In this section, several possible variations around the basic EM paradigm are presented. The principle of all these methods consists in replacing the explicit computation of the expectations

by some kind of stochastic integration procedure. These methods thus all requires stochastic simulations of the missing data, or of some other auxiliary data. Applicable simulation procedures for the noisy convolution model will be considered in section 4.

A word of caution is needed here: For the sake of simplicity, all the algorithms presented in this section are described as if it was possible to perform exact independent simulations of the required stochastic quantities. We shall however see in section 4 that the difficulty of the simulation task itself should not be overlooked. For the model under consideration, we will consider Markov chain simulation techniques and show that the choice of a particular sampling strategy can substantially affect the convergence behavior of the algorithms.

3.1 MCEM: Monte Carlo EM algorithm

Monte Carlo EM, as proposed by Wei and Tanner [66], [62], consists in computing approximately the EM intermediate quantity defined by (9) by use of Monte Carlo integration. Basically, the nth E-step is replaced by the following procedure:

1. *Multiple simulations:* Draw $M(n)$ values $\mathbf{z}^{(n,i)}$ $(i = 1, \cdots, M(n))$ of the missing data vector under $p(\mathbf{z}|\mathbf{y}; \boldsymbol{\theta}^{(n-1)})$, the a posteriori distribution of the missing data given the observations and the current estimate of the parameters.

2. *Monte Carlo integration:* Approximate $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(n-1)})$ with

$$\hat{Q}(\boldsymbol{\theta}|\boldsymbol{\theta}^{(n-1)}) = \frac{1}{M(n)} \sum_{i=1}^{M(n)} \log p_{\boldsymbol{\theta}}(\mathbf{y}, \mathbf{z}^{(n,i)}) \,. \tag{11}$$

(3) and (4) imply that $\hat{Q}(\boldsymbol{\theta}|\boldsymbol{\theta}^{(n-1)}) = L(\hat{\boldsymbol{S}}_1^{(n)}, \hat{\boldsymbol{S}}_2^{(n)}; \boldsymbol{\theta})$ where

$$\hat{\boldsymbol{S}}_j^{(n)} = \frac{1}{M(n)} \sum_{i=1}^{M(n)} \mathbf{S}_1(\mathbf{z}^{(n,i)}) \quad \text{for} \quad j = 1, 2 \,, \tag{12}$$

where $\mathbf{S}_1(\mathbf{z})$ and $\mathbf{S}_2(\mathbf{z})$ are defined in (4).

In certain models, although it is not possible to compute directly $E(\mathbf{S}_i(\mathbf{z})|\mathbf{y}; \boldsymbol{\theta})$, it is feasible to compute $E(\mathbf{S}_i(\mathbf{z})|\mathbf{y}, \mathbf{q}; \boldsymbol{\theta})$, where $\mathbf{q}$ are some auxiliary missing variables. If sampling from $p(\mathbf{q}|\mathbf{y}; \boldsymbol{\theta})$ is simpler or more efficient than sampling directly from $p(\mathbf{z}|\mathbf{y}; \boldsymbol{\theta})$, it may be advantageous to adapt the MCEM scheme presented above. Bayes formula implies that

$$E(\mathbf{S}_i(\mathbf{z})|\mathbf{y}; \boldsymbol{\theta}) = \int E(\mathbf{S}_i(\mathbf{z})|\mathbf{y}, \mathbf{q}; \boldsymbol{\theta}) p(\mathbf{q}|\mathbf{y}, \boldsymbol{\theta}) \mu_{\mathbf{q}}(d\mathbf{q}) \,, \tag{13}$$

and the modified scheme goes as follows

1. *Multiple simulations:* Draw $M(n)$ values $\mathbf{q}^{(n,i)}$ $(i = 1, \cdots, M(n))$ of the auxiliary missing data vector under $p(\mathbf{q}|\mathbf{y}; \boldsymbol{\theta}^{(n-1)})$.

2. *Monte Carlo integration:* Approximate $E(\mathbf{S}_i(\mathbf{z})|\mathbf{y}; \boldsymbol{\theta})$ with

$$\hat{\boldsymbol{S}}_i^{(n)} = \frac{1}{M(n)} \sum_{i=1}^{M(n)} E(\mathbf{S}_i(\mathbf{z})|\mathbf{y}, q^{(n,i)}; \boldsymbol{\theta}^{(n-1)}) \,. \tag{14}$$

Eq. (14) relies on the exact computation of $E(\mathbf{S}_i(\mathbf{z})|\mathbf{y}, q; \boldsymbol{\theta})$ (which has to be feasible). Such schemes which mix simulation and analytic integration (an operation which is described as "parametric Rao-Blackwellization" in [11]) are often preferable because they make the estimates of the sufficient statistics more reliable.

There are very few available results concerning the convergence of Monte Carlo EM. It is important to note that unlike EM, Monte Carlo EM does not deterministically increases the actual likelihood of the parameters at each iteration. This situation which is characteristic of stochastic optimization algorithms makes the convergence analysis more complex to study. Under suitable technical conditions (see, e.g. [5, 44]), MCEM may be shown to converge with probability 1 to a stationary point of the likelihood when $M(n) \to \infty$ is increasing with the iteration index n at a appropriate rate (typically, $M(n) = O(n^\alpha)$, with $\alpha > 0$). Increasing the number of simulations at each stage, decreases the simulation variance of the Monte-Carlo approximation of the conditional expectation and thus the simulation variance of the parameter estimate. This is of course at the expense of the computational efficiency, and some practical trade-off must be found.

3.2 Stochastic EM

The Stochastic EM (SEM) algorithm of Celeux and Diebolt [12], [20] tries to circumvent the problems of MCEM by using only one single simulation of the unobserved data at each iteration (using $M(n) = 1$). This is really an illustration of the "filling-in" or imputation principle since at each step, a pseudo vector of the complete data is simulated using the information brought by the observations $\mathbf{y}$ and the currently available estimate of the parameters $\boldsymbol{\theta}^{(n-1)}$. With $M(n) = 1$, there is no stabilizing mechanism which would ensure that the sequence of parameter estimates $\{\boldsymbol{\theta}_n\}$ does converge (in some proper sense) to a deterministic value. Averaged estimates of the form

$$\hat{\boldsymbol{\theta}}^{(p)} = \frac{1}{n - m_0} \sum_{p=n_0}^{n} \boldsymbol{\theta}^{(p)} \,, \tag{15}$$

where m_0 is the length of the burn-in period during which the output estimates are discarded (so as to reduce the influence of the initial condition). Very few is known about the convergence of the SEM algorithm (see [32]). It has been shown, for some specific models [20] (e.g. mixture of Gaussian pdfs), that $\hat{\boldsymbol{\theta}}^{(n)}$ is a consistent and asymptotically normal estimate of the parameter (but $\hat{\boldsymbol{\theta}}^{(n)}$ does not necessarily converge to a maximum likelihood estimate or a significant mode of the likelihood function). Note that these results do not readily apply to the convolution model considered here, and the convergence of $\hat{\boldsymbol{\theta}}^{(n)}$ to a meaningful value still is an open question.

3.3 SAEM: Stochastic Approximation EM algorithm

The SAEM (for Stochastic Approximation EM) algorithm proposed by Lavielle *et al.* [41] uses a stochastic approximation procedure in order to estimate the conditional expectation of the complete data log-likelihood. The basic idea is that rather than using a large number of simulations at each step, the expectation needed to perform the E-step of EM can be approximated by accumulation of the statistics computed for all previous iterations, with some suitable forgetting mechanism. The nth E-step of SAEM consists in:

- *Simulation :* Sample one realization $\mathbf{z}^{(n)}$ of the missing data vector under $p(\mathbf{z}|\mathbf{y}; \boldsymbol{\theta}^{(n-1)})$.
- *Stochastic approximation :* Update the current approximation of the EM intermediate quantity according to

$$\hat{Q}^{(n)}(\boldsymbol{\theta}) = \hat{Q}^{(n-1)}(\boldsymbol{\theta}) + \gamma_n \left(\log p(\mathbf{y}, \mathbf{z}^{(n)}; \boldsymbol{\theta}) - \hat{Q}^{(n-1)}(\boldsymbol{\theta}) \right) , \tag{16}$$

  where (γ_j) is a sequence of positive step-sizes.

One of the interest of SAEM is that all the results obtained for stochastic approximation in a general framework can be used. In particular, an appropriate choice of the sequence of step-sizes guarantees almost-sure pointwise convergence of the sequence of parameter estimates to a local maxima of the likelihood for a wide class of probability models (see Lavielle [41] for technical details).

The step-size controls the amount of stochastic excitation that is fed into the algorithm at each iteration. The step-sizes should not decrease to rapidly in order to avoid the convergence towards spurious stationary points (e.g. saddle points or local minima). On the other hand, convergence of the estimates will only occur when the step-size becomes close to zero. Typical choices are $\gamma_n = (1/n)^\alpha$, with $1/2 < \alpha \leq 1$ [38]. All the techniques developed to speed up convergence of stochastic algorithms and to reduce the variance of the estimates can be used, such as Polyak's [54] averaging scheme or Kesten's [35] procedure for computing an optimal sequence of step-sizes.

For blind deconvolution models, (3) shows that updating $\hat{Q}^{(n-1)}(\boldsymbol{\theta})$ is equivalent to updating the approximations of the conditional expectation of the sufficient statistics defined in (4), so that (16) reduces to:

$$\hat{\boldsymbol{S}}_i^{(n)} = \hat{\boldsymbol{S}}_i^{(n-1)} + \gamma_n \left(\mathbf{S}_i(\mathbf{z}^{(n)}) - \hat{\boldsymbol{S}}_i^{(n-1)} \right) \quad \text{for} \quad i = 1, 2 \, . \tag{17}$$

Computation of $\boldsymbol{\theta}^{(n+1)}$ can then be carried out directly using (5).

Note that as was the case for the MCEM algorithm, the SAEM algorithm can also be adapted to cases where it is more appropriate to sample from auxiliary missing variables. Then, instead of (17) the updating of $\hat{\boldsymbol{S}}_i^{(n)}$ is be carried out as follows

$$\hat{\boldsymbol{S}}_i^{(n)} = \hat{\boldsymbol{S}}_i^{(n-1)} + \gamma_n \left[E(\mathbf{S}_i(\mathbf{z})|\mathbf{y}, \mathbf{q}^{(n)}; \boldsymbol{\theta}^{(n-1)}) - \hat{\boldsymbol{S}}_i^{(n-1)} \right] \, . \tag{18}$$

4 Simulation techniques

The stochastic versions of the EM algorithm presented in the previous section require simulation of the missing input samples under $p\,(\mathbf{z}\,|\mathbf{y};\boldsymbol{\theta}\,)$. This simulation step is not straightforward to implement as the input samples are not conditionally independent given the observed output samples. In this section, several possible methods for simulating the missing inputs are presented. In Subsection 4.1, a general Markov Chain Monte Carlo (MCMC) sampler is presented to simulate under the posterior $p(\mathbf{z}|\mathbf{y};\boldsymbol{\theta})$. More efficient sampling strategies, fitted to the case where the input data pdf is a mixture of Gaussian, are presented in section 4.3.

For notational convenience, the dependence on the current value of the parameter $\boldsymbol{\theta}$ of all the probability distribution functions is omitted in this section.

4.1 Basic Principles

MCMC (Markov Chain Monte Carlo) is a class of stochastic simulation methods designed for sampling from multivariate distributions (generally of high-dimensionality). These methods appeared in the statistical literature in the early 80's and are very useful in the fields image processing and computational statistics. MCMC techniques are well-documented in the literature (see [4], [25], [28], [57], [61] and references therein) and only a brief account of these methods is given here.

The idea is very simple. Suppose that we need to sample from a distribution $f\,(\underline{x})$ where $\underline{x} \triangleq (x_1, \cdots, x_n) \in \mathcal{X} \subseteq \mathbb{R}^n$ which is known (perhaps up to multiplicative constant). f will be referred to as the *target* distribution. If f is very complex so that it is no direct sampling method available, an indirect method for obtaining samples from f consists in constructing a Markov chain (aperiodic and irreducible), whose stationary (or invariant) distribution is $f(\underline{x})$. Then, if the chain is run for long enough, simulated values from the chain can be treated as

a dependent samples from the target distribution, and used as shown in the previous section. There are many important implementation issues associated with MCMC methods, including, amongst others, the choice of the chain's transition mechanism, and techniques to control the convergence to the limit distribution.

Gibbs Sampler

The Gibbs sampler was first introduced for image restoration by Geman and Geman [26] and Besag [3]. An extensive account of the Gibbs sampler may be found in the tutorials by Smith and Roberts [61], Gelfand and Smith [25] and Besag et al [4]. The Gibbs sampler proceeds by splitting the state vector into a number of components and updating each in turn by a series of Gibbs transitions. Suppose that the state vector is split into $q \leq n$ components $(\underline{x}_1, \cdots, \underline{x}_q)$. Having selected component $\underline{x}_i$ to be updated, the Gibbs transition kernel produces a new state vector $\underline{x}' = (\underline{x}_1, \cdots, \underline{x}_{i-1}, \underline{y}, \underline{x}_{i+1}, \underline{x}_q)$ where $\underline{y}$ is sampled from $f(\underline{x}_i|\underline{x}_{-i})$, the conditional distribution of $\underline{x}_i$, given the values of the other components $\underline{x}_{-i} = [\underline{x}_1, \cdots, \underline{x}_{i-1}, \underline{x}_{i+1}, \cdots, \underline{x}_q]$, $1 \leq i \leq q$. Ideally, the conditional distribution $f(\underline{x}_i|\underline{x}_{-i})$ should be easy to sample form (ie. is a "standard" distribution). However, in the cases where the conditional distribution is non-standard, there are ways to sample from the appropriate conditionals (see next section). The basic Gibbs sampler uses a fixed sequence of Gibbs transition kernels, each of which updates a different component of the state vector, as follows:

Algorithm 1 (Gibbs sampler)

1. Set an arbitrary starting value $\underline{x}^{(0)} = \left(\underline{x}_1^{(0)}, \cdots, \underline{x}_q^{(0)}\right)$ for the first iteration $(k = 1)$.
2. At iteration index k,
   - Sample $\underline{x}_1^{(k)}$ from $f\left(\underline{x}_1 | \underline{x}_{-1}^{(k)}\right)$,
   - Sample $\underline{x}_2^{(k)}$ from $f\left(\underline{x}_2 | \underline{x}_{-2}^{(k)}\right)$,
   - $\cdots$
   - Sample $\underline{x}_q^{(i)}$ from $f\left(\underline{x}_q^{(k)} \middle| \underline{x}_{-q}^{(k)}\right)$,

   where $\underline{x}_{-i}^{(k)} \triangleq \left(\underline{x}_1^{(k)}, \cdots, \underline{x}_{i-1}^{(k)}, \underline{x}_{i+1}^{(k-1)}, \cdots, \underline{x}_q^{(k-1)}\right)$.

Iteration of the *full* cycle of random variate generations from each of the full conditionals, produces a sequence which is a realization of a Markov chain with stationary distribution $f(\underline{x})$ (under conditions that are discussed in [63], [57]). This sampling algorithm, where each component is updated in turn, is sometimes referred to as the *systematic sweep Gibbs sampler*. However, the Gibbs transition kernel need not be used in this systematic manner, and many other implementations are possible, such as the *random sweep* Gibbs sampler, which randomly selects a component to be updated at each iteration, and thus uses a mixture (rather than a cycle) of Gibbs updates.

Metropolis-Hasting algorithm

The Metropolis-Hasting algorithm is an alternative and more general updating scheme, where values are drawn from an arbitrary (yet sensibly chosen) distributions and are *accepted or rejected* in such a way that, asymptotically, they behave as dependent random observations from the target distribution. This method is a form of generalized rejection sampling approach and is widely applicable [28], [57].

The Metropolis Hasting update proceeds as follows. Suppose we wish to update $\underline{x}$, first a candidate observation $\underline{y}$ is sampled form an arbitrary pdf $q(\underline{x}, \underline{y})$ that depends on the current state of the chain $\underline{x}$. The choice of q is essentially arbitrary (subject to the condition that the resulting Markov chain is aperiodic is irreducible): It is generally selected so that sampling from this (proposal) distribution is easy. The candidate $\underline{y}$ is accepted with probability

$$\alpha(\underline{x}, \underline{y}) = \min\left(1, \frac{f(\underline{y})\ q(\underline{y}, \underline{x})}{f(\underline{x})\ q(\underline{x}, \underline{y})}\right) . \tag{19}$$

In the case where the candidate is rejected, the chain remains in its current state $\underline{x}$. Note that f only enters through α and the ratio $f(\underline{y})/f(\underline{x})$, so that the knowledge of the distribution only up to a multiplicative constant is sufficient for implementation. There are an infinite range of choices for q, see Tierney [63] and Chib and Greenberg [16]. The most often used proposal are

Random Walk Metropolis If $q(\underline{x}, \underline{y}) = \phi(\underline{y} - \underline{x})$ for some arbitrary density ϕ, then the kernel driving the chain is a random walk. There are many common choices for ϕ including the uniform distribution on an hypersphere, a multivariate normal, or an over-dispersed multivariate student t-distribution.

The independent sampler If $q(\underline{x}, \underline{y}) = \phi(\underline{y})$, then the candidate observation is drawn independently of the current state of the chain. In this case, the acceptance probability can be written as,

$$\alpha(\underline{x}, \underline{y}) = \min(1, w(\underline{y})/w(\underline{x})) ,$$

where the ratio $w(\underline{x}) = f(\underline{x})/\phi(\underline{x})$ is known as the importance weight function [62].

Data augmentation sampling

In certain cases, it is more appropriate to sample not directly from $f(\underline{x})$ but from an *augmented* pdf $g(\underline{x}, \underline{v})$ such that $f(\underline{x})$ is the marginal distribution of $g(\underline{x}, \underline{v})$ with respect to $\underline{v}$. This is typically the case when sampling from the f alone is not so easy. This approach, known as *data augmentation* was introduced in the statistical literature by Tanner and Wong [62].

An example of a 'Gibbs-style' data augmentation sampler is given below, where the augmented state-vector is split in two blocks $\underline{x}$ and $\underline{v}$.

Algorithm 2 (Data augmentation sampler)

1. Set an arbitrary starting value $\underline{v}^{(0)}$
2. At iteration index k, sample
   - $\underline{x}^{(k)}$ from $p(\underline{x}|\underline{v}^{(k-1)})$,
   - and $\underline{v}^{(k)}$ from $p(\underline{v}|\underline{x}^{(k)})$,

   where $p(\underline{x}|\underline{v})$ (resp. $p(\underline{v}|\underline{x})$) denotes the conditional distribution of $\underline{x}$ given $\underline{v}$ (resp. $\underline{v}$ given $\underline{x}$), derived from g.

Sampling the two sub-components $\underline{x}$ and $\underline{v}$ in block, rather than element by element as in the basic Gibbs paradigm, is usually preferable (if it is feasible) because it reduces the correlation between subsequent outputs of the Markov simulation chain [8], [43], [58].

4.2 A general-purpose sampler for blind deconvolution

In the application under consideration, it is required to sample from $p(\mathbf{z}|\mathbf{y})$. A first, and perhaps not optimal, procedure proceeds by dividing the data vector $\mathbf{z}$ into its scalar components $z_{1-p}, \cdots, z_T$.

Gibbs Sampler

To implement a systematic sweep Gibbs sampler, we need to evaluate full conditional distribution. Under the assumptions stated above, the full conditional may be expressed as

$$p(z_t|y_{1:T}, z_{1-p:t-1}, z_{t+1:T}) \propto p(y_{1:T}|\underline{\boldsymbol{z}}_{1:T})p(z_t) \tag{20}$$

$$\propto \prod_{i=\max(t,1)}^{\min(t+p,T)} p(y_i|\underline{\boldsymbol{z}}_i)p(z_t). \tag{21}$$

where $\propto$ means "proportional to" and $y_{1:T} \triangleq (y_1, \cdots, y_T)'$, $z_{1-p:t} \triangleq (z_{1-p}, \cdots, z_t$, $\underline{\boldsymbol{z}}_{1:T} \triangleq (\underline{\boldsymbol{z}}_1, \cdots, \underline{\boldsymbol{z}}_T)'$, etc. Most often, the full-conditional distribution does not belong to a standard distribution family for which efficient sampling algorithms are readily available. One important exception occurs when the pdf of z_t is a mixture of Gaussian because the full conditional still is a mixture of Gaussian in this case. We shall see however in section 4.3 that there are more efficient sampling scheme for handling this particular case. In other situations, one need to resort to an hybrid strategy, mixing the Gibbs sampler and a Metropolis-Hasting procedure.

Single-component independent sampler

The *Metropolis-Hasting within Gibbs algorithm* (also known as a one-at-a-time Metropolis-Hastings scheme) consists in updating each individual component in turn, via a single Metropolis-Hasting update until all components have been visited. This solution is equivalent to the so-called hybrid Gibbs sampler, suggested by Muller [50]. The most straightforward solution consists in running an *independent sampler* using the prior distribution of z_t as the proposal distribution. The procedure for updating the t-th component z_t, $1 - P \leq t \leq T$, goes as follow

Algorithm 3 (Single-component independent sampler)

- Sample $\tilde{z}_i$ from the prior distribution $p(z)$.
- Accept $\tilde{z}_i$ with probability

$$\alpha(z_i, \tilde{z}_i) = \min\left(1, \exp\left[-(2\sigma^2)^{-1} \sum_{i=i_{min}}^{i_{max}} (y_i - h'\underline{\tilde{\boldsymbol{z}}}_t)^2 - (y_i - h'\underline{\boldsymbol{z}}_t)^2\right]\right),$$

  where $i_{min} = \max(t, 1)$, $i_{max} = \min(t + p, T)$ and $\underline{\tilde{\boldsymbol{z}}}_i = [z_i, \cdots, z_{t+1}, \tilde{z}_t, z_{t-1}, \cdots, z_{i-p}]'$, for $i_{min} \leq i \leq i_{max}$.

Compared to a random walk Metropolis-Hasting procedure, the independence sampler described above has the advantage that it doesn't necessitate any tuning of the proposal distribution. On the other hand, the independence sampler can only be used when simulation from $p(z)$ is feasible, and it may lead to high rejection ratios for certain distributions.

In practice, only a few iterations (or more precisely a few complete cycles) of the sampling procedure are performed to obtain the simulation needed at each SEM, MCEM or SAEM step. This is definitely not enough to guarantee convergence of the sampler to its limiting target distribution, but it proves in practice to be enough to ensure proper convergence of the estimates.

4.3 Sampling schemes for Gaussian mixtures

The mixture of Gaussian model deserves special attention; this model has been used extensively in geophysics for seismic trace inversion (see Mendel [47]). It is also frequently used to model sources with impulsive behavior, like neutronic sources (see Doucet *et al* [23] for applications). One particular case of interest, is the Bernoulli-Gaussian distribution which is a two components mixture of Gaussian with zero means and largely different variances [13], [40], [15]. A slightly different perspective consists in using mixture models in a semi-parametric context, where the distribution of the input data is not precisely known. The motivation for using mixtures here is that any 'smooth' probability distribution function $p(z_t)$ may be approximated by a mixture of Gaussian, provided that the number of components is large enough. Thus when K is sufficiently large, it can be expected that the estimate of the filter coefficients are "close to optimal" for a large class of input distributions $p(z_t)$ [49]. There are several theoretical as well as practical issues in that direction, that still need be answered.

A Gaussian mixture model has the form [64], [56]

$$p(z) = \sum_{k=1}^{K} \lambda_k \phi(z; \mu_k, \rho_k^2), \tag{22}$$

where λ_k are the statistical weights of the components of the mixture, and $\phi(\cdot; \mu_k, \rho_k)$ is the Gaussian probability density with mean μ_k and variance ρ_k^2. It is often enlightening to consider that the observations in a mixture models are incomplete since (22) corresponds to the following data-generation mechanism

$$z_t | q_t \sim \phi(z_t; \mu_{q_t}, \rho_{q_t}^2), \tag{23}$$

where q_t is an *unobservable* random variable taking its value in the set $\{1, \cdots, K\}$, with probability distribution $P(q_t = k) = \lambda_k$ $(1 \leq k \leq K)$. The variables q_t are often referred to as the labels or the categories, or more formally as the *mixture component indicators* (see Titterington *et al.* [64] for a complete account of mixture models).

It is worthwhile to note that, conditionally to $q_{1-p:T} = [q_{1-p}, \cdots, q_T]'$ and $y_{1:T} = [y_1, \cdots, y_T]'$, the random vector $z_{1-p:T} = [z_{1-p}, \cdots, z_T]'$ is Gaussian. It is shown below that it is possible to sample directly in block from $p(z_{1-p:T} | q_{1-p:T}, y_{1:T})$, using a recursive algorithm derived from the Kalman filter and smoother. Next, it is easily seen that

$$p(q_{1-p:T} | z_{1-p:T}, y_{1:T}) = \prod_{t=1-p}^{T} p(q_t | z_t)\,, \tag{24}$$

where $p(q_t | z_t)$ is a (discrete) multinomial distribution and

$$p(q_t = k | z_t) = \frac{\lambda_k \phi(z_t; \mu_k, \rho_k^2)}{\sum_{j=1}^{K} \lambda_j \phi(z_t; \mu_j, \rho_j^2)}\,. \tag{25}$$

It is thus straightforward to sample in block from the conditional distribution of $q_{1-p:T}$ given $z_{1-p:T}, y_{1:T}$. This suggests to use the *data augmentation* sampler introduced in the previous section. Two alternative methods to sample from $p(z_{1:T} | y_{1:T}, q_{1-p:T})$ are given below. The first, proposed by Carter and Kohn [8], is straightforward to implement. A somewhat more involved scheme with better numerical efficiency is described next.

Method I: State sampler

The sampling procedure developed below is based on the following observation: Conditional to $y_{1:T}$ and $q_{1-p:T}$, $\underline{\boldsymbol{z}}_{1:T}$ is an inhomogeneous Markov chain, in the sense that :

$$p(\underline{\boldsymbol{z}}_{1:T} | y_{1:T}, q_{1-p:T}) = p(\underline{\boldsymbol{z}}_T | y_{1:T}, q_{1-p:T}) \prod_{t=1}^{T-1} p(\underline{\boldsymbol{z}}_t | y_{1:t}, \underline{\boldsymbol{z}}_{t+1}, q_{1-p:T})\,. \tag{26}$$

Eq. (26) suggests the following strategy to sample from $p(\underline{\boldsymbol{z}}_{1:T}|y_{1:T}, q_{1-p:T})$: *(1)* First, sample from the conditional distribution of the last state vector $p(\underline{\boldsymbol{z}}_T|y_{1:T}, q_{1-p:T})$, *(2)* sample *backward-in-time* (for $t = T-1, ..., 1$), from $p(\underline{\boldsymbol{z}}_t|y_{1:t}, \underline{\boldsymbol{z}}_{t+1}, q_{1-p:T})$. This strategy requires sampling from $p(\underline{\boldsymbol{z}}_t|y_{1:t}, \underline{\boldsymbol{z}}_{t+1}, q_{1-p:T})$, which is feasible because $[\underline{\boldsymbol{z}}_t, y_{1:t}, \underline{\boldsymbol{z}}_{t+1}]$ is a Gaussian vector conditionally to $q_{1-p:T}$. All we need to compute is thus the conditional mean and variance of $\underline{\boldsymbol{z}}_t$ given $y_{1:t}$, $\underline{\boldsymbol{z}}_{t+1}$ and $q_{1-p:T}$. For that purpose, a Kalman filter is used.

A few additional notations are in order. First, it is convenient to consider the observation model in state-space form

$$y_t = \mathbf{h}'\underline{\boldsymbol{z}}_t + \sigma n_t \,, \tag{27}$$

$$\underline{\boldsymbol{z}}_{t+1} = \mathbf{S}\underline{\boldsymbol{z}}_t + (m_t + r_t u_t)\mathbf{e} \,, \tag{28}$$

where $\mathbf{S}$ is the down-shift matrix and $\mathbf{e} \triangleq (1, 0, \cdots, 0)'$. $\{u_t\}_{t\geq -p}$ and $\{n_t\}_{t\geq 1}$ are independent sequences of i.i.d. Gaussian standardized random variables. The attention of the reader is drawn on the fact that the convention used above requires that $m_t \triangleq \mu_{q_{t+1}}$ and $r_t \triangleq \rho_{q_{t+1}}$. Despite the minor disagreement of this index shift, the convention used in (28) is prefered since it corresponds to standard state-space form of a dynamic linear systems [7].

The simulation procedure proceeds in two pass: A *forward* pass, where the quantities of interest are computed using the Kalman filter recursions; A *backward* pass, where sampling is performed from a normal distribution with parameters determined from the quantities computed in the forward pass. Denote:

$\epsilon_t = y_t - y_{t\|t-1}$	innovation process
$d_t = E(\epsilon_t^2)$	variance of the innovation process
$\underline{\boldsymbol{z}}_{t\|t-1}$	one-step ahead state predictor
$\underline{\boldsymbol{z}}_{t\|t}$	filtered state estimate

(29)

Here the notation $y_{t|v}$ and (resp. $\underline{\boldsymbol{z}}_{t|v}$) denote the orthogonal projection (in the Hilbert space of square integrable random variables) of y_t (resp. $\underline{\boldsymbol{z}}_t$) onto the closed linear span of $\{1, y_1, \cdots, y_v\}$. Let:

$$\mathbf{\Gamma}_{t|t-1} = E\left[(\underline{\boldsymbol{z}}_t - \underline{\boldsymbol{z}}_{t|t-1})(\underline{\boldsymbol{z}}_t - \underline{\boldsymbol{z}}_{t|t-1})'\right] \quad \text{and} \quad \mathbf{\Gamma}_{t|t} = E\left[(\underline{\boldsymbol{z}}_t - \underline{\boldsymbol{z}}_{t|t})(\underline{\boldsymbol{z}}_t - \underline{\boldsymbol{z}}_{t|t})'\right] \,,$$

denote the one step-ahead state prediction and the state filter covariance matrices, respectively.

Algorithm 4 (Kalman filter) Initialize the recursion with $\underline{\boldsymbol{z}}_{1|0} = [m_0, m_{-1}, \cdots, m_{-p}]'$ and $\Gamma_{1|0} = \mathrm{diag}[r_0^2, r_{-1}^2, \cdots, r_{-p}^2]$, and compute, for $1 \leq t \leq T$,

$\epsilon_t = y_t - \mathbf{h}'\underline{\boldsymbol{z}}_{t\|t-1}$	innovation update
$d_t = \mathbf{h}'\mathbf{\Gamma}_{t\|t-1}\mathbf{h} + \sigma^2$	variance of the innovation
$\mathbf{k}_t = d_t^{-1}\mathbf{\Gamma}_{t\|t-1}\mathbf{h}$	Kalman gain update
$\underline{\boldsymbol{z}}_{t\|t} = \underline{\boldsymbol{z}}_{t\|t-1} + \mathbf{k}_t\epsilon_t$	state filtering equation
$\mathbf{\Gamma}_{t\|t} = \mathbf{\Gamma}_{t\|t-1} - d_t\mathbf{k}_t\mathbf{k}_t'$	covariance of the filtering error
$\underline{\boldsymbol{z}}_{t+1\|t} = \mathbf{S}\underline{\boldsymbol{z}}_{t\|t} + m_t\mathbf{e}$	state predictor update
$\mathbf{\Gamma}_{t+1\|t} = \mathbf{S}\mathbf{\Gamma}_{t\|t}\mathbf{S}' + r_t^2\mathbf{e}\mathbf{e}'$	covariance of the prediction error

(30)

When running the Kalman filter, the quantities $\underline{\boldsymbol{z}}_{t|t}$, $\mathbf{\Gamma}_{t|t}$, $\underline{\boldsymbol{z}}_{t+1|t}$ and $\mathbf{\Gamma}_{t+1|t}$ should be stored (note that the computation of $\underline{\boldsymbol{z}}_{t|t}$ and $\mathbf{\Gamma}_{t|t}$ during the forward pass can be skipped: In practise, it may be more convenient to store only $\underline{\boldsymbol{z}}_{t+1|t}$, $\mathbf{\Gamma}_{t+1|t}$, $\mathbf{k}_t$ and d_t and to perform the needed computations during the backward pass). The backward simulation proceeds as follow:

1. Simulate $\underline{\boldsymbol{z}}_T$ under a multivariate normal distribution with mean $\underline{\boldsymbol{z}}_{T|T}$ and covariance $\mathbf{\Gamma}_{T|T}$.

2. For $t = T-1, \cdots, 1$, sample z_{t-p} from a scalar Gaussian distribution with mean $\tilde{m}_{t-p}$ and variance $\tilde{r}_{t-p}$ given by

$$\begin{aligned} \tilde{m}_{t-p} &= \mathbf{j}' \left(\underline{\boldsymbol{z}}_{t|t} + \boldsymbol{\Gamma}_{t|t} \mathbf{S}' \boldsymbol{\Gamma}_{t+1|t}^{-1} \left(\underline{\boldsymbol{z}}_{t+1} - \underline{\boldsymbol{z}}_{t+1|t} \right) \right) , \\ \tilde{r}_{t-p} &= \mathbf{j}' \left(\boldsymbol{\Gamma}_{t|t} - \boldsymbol{\Gamma}_{t|t} \mathbf{S}' \boldsymbol{\Gamma}_{t+1|t}^{-1} S \boldsymbol{\Gamma}_{t|t} \right) \mathbf{j} , \end{aligned} \tag{31}$$

where the vector $\mathbf{j} \triangleq (0, 0, \cdots, 1)'$ selects the last component.

This simulation technique which correspond to the straightforward application of (26) requires the inversion of a $(p+1)\times(p+1)$ matrix at each iteration step, which can become computationally involved when the filter order p is large.

Method II: Disturbance sampler

For the deconvolution model, the fact that the dimension of the disturbance noise u_t is much less than that of the state vector $\mathbf{z}_t$ makes it possible to derive an equivalent simulation algorithm with a reduced complexity. The disturbance sampler introduced by De Jong and Shephard [17] samples directly a realization of the disturbance noise sequence $u_{1:T-1}$, which together with a simulation of the initial state $\underline{\boldsymbol{z}}_1$ can be used to obtain the complete sequence $z_{1:T}$. This algorithm is based on a backward Gram-Schmidt orthogonalization and on previous results on the so-called *disturbance smoother* established independently by Mendel [46] and Koopman [37].

Algorithm 5 (Disturbance sampler)

1. Kalman filter (Forward filtering): Run the Kalman filter as indicated by (30), and store ϵ_t, d_t and $\mathbf{L}_t \triangleq \mathbf{S}(\mathbf{I} - \mathbf{k}_t \mathbf{h}')$, where $\mathbf{I}$ denotes the $(p+1) \times (p+1)$ identity matrix.

2. Backward sampling: For $t = T-1, \cdots, 1$ compute

$$\mathbf{U}_t = \begin{cases} d_T^{-1} \mathbf{h}\mathbf{h}' & \text{for} \quad t = T-1 \\ \left[d_{t+1}^{-1} \mathbf{h}\mathbf{h}' + c_{t+1}^{-1} \mathbf{v}_{t+1} \mathbf{v}_{t+1}' + \mathbf{L}_{t+1}' \mathbf{U}_{t+1} \mathbf{L}_{t+1} \right] & \text{if} \quad t \leq T-2 \end{cases} \tag{32}$$

$$\mathbf{p}_t = \begin{cases} d_T^{-1} \epsilon_T \mathbf{h} & \text{for} \quad t = T-1 \\ \left[d_{t+1}^{-1} \epsilon_{t+1} \mathbf{h} - c_{t+1}^{-1} \tilde{\eta}_{t+1} \mathbf{v}_{t+1} + \mathbf{L}_{t+1}' \mathbf{p}_{t+1} \right] & \text{if} \quad t \leq T-2 \end{cases} \tag{33}$$

$$\mathbf{v}_t = r_t \mathbf{L}_t' \mathbf{U}_t \mathbf{e} , \tag{34}$$

and

$$c_t = 1 - r_t^2 \mathbf{e}' \mathbf{U}_t \mathbf{e} . \tag{35}$$

Simulate $\tilde{\eta}_t$ from $\phi(\cdot; r_t \mathbf{e}' \mathbf{p}_t, c_t)$ and compute

$$\tilde{z}_{t+1} = m_t + r_t \tilde{\eta}_t .$$

3. Initial state: Compute

$$\mathbf{U}_0 = \left[d_1^{-1} \mathbf{h}\mathbf{h}' + c_1^{-1} \mathbf{v}_1 \mathbf{v}_1' + \mathbf{L}_1' \mathbf{U}_1 \mathbf{L}_1 \right] , \tag{36}$$

$$\mathbf{p}_0 = \left[d_1^{-1} \epsilon_1 \mathbf{h} - c_1^{-1} \tilde{\eta}_1 \mathbf{v}_1 + \mathbf{L}_1' \mathbf{p}_1 \right] , \tag{37}$$

and set

$$\mathbf{m}_0 = (m_0, m_{-1}, \cdots, m_{-p})' \qquad \mathbf{R}_0 = \mathrm{diag}(r_0^2, r_{-1}^2, \cdots, r_{-p}^2)\,. \tag{38}$$

Simulate $\tilde{\underline{\boldsymbol{z}}}_1$ (in one block) from

$$\phi\left(\cdot,\ \mathbf{m}_0 + \mathbf{R}_0\mathbf{p}_0,\ \mathbf{R}_0 - \mathbf{R}_0\mathbf{U}_0\mathbf{R}_0\right)\,.$$

The basic principle of the above disturbance sampler consists in performing a recursive backward Gram-Schmidt orthogonalization of the disturbance noise sequence $u_t, 1 \leq t \leq T-1$ [17]. Thus, rather than trying to compute directly the mean and variance of u_t conditional to $u_{t+1:T-1}, y_{1:T}$, one considers the orthonormal sequence defined as $\eta_t = u_t - u_{t|\mathcal{H}_{t+1}}$, where $\mathcal{H}_{t+1}$ is the closed linear span of $u_{t+1:T-1}, y_{1:T}$ (which coincides, by construction, with that of $\epsilon_{t+1:T}, \eta_{t+1:T-1}$).

4.4 Sampling only the mixture indicators

As indicated at the end of section 3.1, it may be advantageous to sample from an auxiliary set of dummy variables rather than from the missing data vector $z_{1-p:T}$ itself. In the case of Gaussian mixture, there are evidence that MCMC algorithms that samples the indicator variables $q_{1-p:T}$ without sampling the missing data $z_{1-p:T}$ are more efficient (faster converging and mixing of the chain) than the data augmentation schemes described above [9].

For the moving average convolution model considered in this paper, it is extremely simple to derive a single site Gibbs sampling for simulating the mixture indicator variables q_t, $1 \leq t \leq T$. Eq. (20) implies that the conditional distribution $p(q_t|y_{1:T}, q_{1-p:t-1}, q_{t+1:T})$ may be expressed as

$$\begin{aligned} p(q_t|y_{1:T}, q_{1-p:t-1}, q_{t+1:T}) &\propto p\left(y_{1:T}|\, q_{1-p:T}\right) p\left(q_{1-p:T}\right) \\ &\propto p\left(y_{\max(t,1):\min(t+p,T)}\middle|\, q_{\max(t-p,1-p):\min(t+p,T)}\right) p\left(q_t\right)\,. \end{aligned} \tag{39}$$

In the following, we shall assume that $1 \leq t \leq T-p$ for the sake of brevity (the modifications needed to handle the case of boundaries are straightforward and omitted here). First note that, given $q_{t-p:t+p}$, $z_{t-p:t+p}$ is conditionally Gaussian with mean vector and covariance matrix

$$\boldsymbol{\mu}_t = (\mu_{q_{t-p}}, \cdots \mu_{q_{t+p}})' \qquad \boldsymbol{\Sigma}_t = \mathrm{diag}(\rho^2_{q_{t-p}}, \cdots \rho^2_{q_{t+p}})\,. \tag{40}$$

Then using (27), it is easy to see that

$$y_{t:t+p} = \mathcal{T}\left(\mathbf{h}\right) z_{t-p:t+p} + \sigma n_{t:t+p}\,,$$

where $\mathcal{T}\left(\mathbf{h}\right)$ is the $(p+1) \times (2p+1)$ Sylvester matrix defined as

$$\mathcal{T}\left(\mathbf{h}\right) \triangleq \begin{pmatrix} h_0 & h_1 & \cdots & h_p & 0 & \cdots & \cdots & 0 \\ 0 & h_0 & h_1 & \cdots & h_p & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & \cdots & 0 & h_0 & h_1 & \cdots & h_p \end{pmatrix}\,, \tag{41}$$

and $n_{t:t+p} = [n_t, \cdots, n_{t+p}]'$. Under the stated assumptions, $n_{t:t+p}$ is a $(p+1) \times 1$ standard normal vector, which implies that

$$p(q_t|y_{1:T}, q_{1-p:t-1}, q_{t+1:T}) \propto \lambda_{q_t}\, \phi\left(y_{t:t+p}\,;\, \mathcal{T}\left(\mathbf{h}\right)\boldsymbol{\mu}_t, \mathcal{T}\left(\mathbf{h}\right)\boldsymbol{\Sigma}_t\mathcal{T}\left(\mathbf{h}\right)' + \sigma^2\mathbf{I}\right)\,, \tag{42}$$

where $\mathbf{I}$ is the $(p+1) \times (p+1)$ identity matrix. Eq. (42) has to be evaluated for the K possible values of q_t, $q_t = 1, \cdots, K$ yielding the actual conditional probabilities after re-normalization. The above procedure is a much simplified version of the algorithm described in [9] for arbitrary conditionally Gaussian state-space models.

To complete the expectation step (see eqs. (14) and (18)), it is necessary to evaluate the following quantities

$$E(\underline{\boldsymbol{z}}_t|y_{1:T}, q_{1-p:T}) \text{ and } E(\underline{\boldsymbol{z}}_t\underline{\boldsymbol{z}}_t'|y_{1:T}, q_{1-p:T}) \text{ for } 1 \leq t \leq T \,.$$

Since the conditional distribution of $\underline{\boldsymbol{z}}_{1:T}$ given $y_{1:T}$ and $q_{1-p:T}$ is Gaussian, these quantities may be efficiently computed using a disturbance smoother which many similarities with the disturbance sampler described in section 4.3.

Algorithm 6 (Disturbance smoother)

1. Kalman filter (Forward filtering): Run the Kalman filter as indicated by (30), and store ϵ_t, d_t, $\boldsymbol{\Gamma}_{t+1|t}$ and $\mathbf{L}_t = \mathbf{S}(\mathbf{I} - \mathbf{k}_t\mathbf{h}')$.

2. Backward smoothing: For $t = T-1, \cdots, 1$ compute

$$\mathbf{U}_t = \begin{cases} d_T^{-1}\mathbf{h}\mathbf{h}' & \text{for} \quad t = T-1 \\ \left[d_{t+1}^{-1}\mathbf{h}\mathbf{h}' + \mathbf{L}_{t+1}'\mathbf{U}_{t+1}\mathbf{L}_{t+1}\right] & \text{if} \quad t \leq T-2 \end{cases} \tag{43}$$

$$\mathbf{p}_t = \begin{cases} d_T^{-1}\epsilon_T\mathbf{h} & \text{for} \quad t = T-1 \\ \left[d_{t+1}^{-1}\epsilon_{t+1}\mathbf{h} + \mathbf{L}_{t+1}'\mathbf{p}_{t+1}\right] & \text{if} \quad t \leq T-2 \end{cases} \tag{44}$$

   Store

$$E\left[z_{t+1}|y_{1:T}, q_{1-p:T}\right] = m_t + r_t^2\mathbf{e}'\mathbf{p}_t \,, \tag{45}$$

   and

$$E\left[\left(\underline{\boldsymbol{z}}_{t+1} - E[\underline{\boldsymbol{z}}_{t+1}|y_{1:T}, q_{1-p:T}]\right)\left(z_{t+1} - E[z_{t+1}|y_{1:T}, q_{1-p:T}]\right)\right] = r_t^2\left[(1 - r_t^2\mathbf{e}'\mathbf{U}_t\mathbf{e})\mathbf{e} - \mathbf{S}\boldsymbol{\Gamma}_{t|t-1}\mathbf{L}_t'\mathbf{U}_t\mathbf{e}\right] \,. \tag{46}$$

3. Initial state: Compute

$$\mathbf{U}_0 = \left[d_1^{-1}\mathbf{h}\mathbf{h}' + \mathbf{L}_1'\mathbf{U}_1\mathbf{L}_1\right] \,, \tag{47}$$

$$\mathbf{p}_0 = \left[d_1^{-1}\epsilon_1\mathbf{h} + \mathbf{L}_1'\mathbf{p}_1\right] \,, \tag{48}$$

   and compute

$$E\left[\underline{\boldsymbol{z}}_1|y_{1:T}, q_{1-p:T}\right] = \mathbf{m}_0 + \mathbf{R}_0\mathbf{p}_0 \,, \tag{49}$$

$$E\left[\left(\underline{\boldsymbol{z}}_1 - E[\underline{\boldsymbol{z}}_1|y_{1:T}, q_{1-p:T}]\right)\left(\underline{\boldsymbol{z}}_1 - E[\underline{\boldsymbol{z}}_1|y_{1:T}, q_{1-p:T}]\right)'\right] = \mathbf{R}_0 - \mathbf{R}_0\mathbf{U}_0\mathbf{R}_0 \,, \tag{50}$$

   where $\mathbf{m}_0$ and $\mathbf{U}_0$ are defined as in (38).

Equations (45) and (49) directly yield the a posteriori mean of the unobserved input signal, whereas (46) and (50) can be used to obtain recursively the a posteriori covariance of each state vector $Cov\{\underline{\boldsymbol{z}}_t - E[\underline{\boldsymbol{z}}_t|y_{1:T}, q_{1-p:T}]\}$ noting that for a MA system it is indeed only necessary to compute the first line or first column when $Cov\{\underline{\boldsymbol{z}}_{t-1} - E[\underline{\boldsymbol{z}}_{t-1}|y_{1:T}, q_{1-p:T}]\}$ is already known.

5 Simulation results

In this section, some numerical results and comparisons are provided to illustrate the behavior of the estimation methods as well as the influence of the choice of a particular simulation strategy. Only the SAEM method (see section 3.3) will be considered here since most conclusive convergence results obtained to date pertain to this method.

5.1 Finite-valued input

We begin with a very simple model, for which exact independent sampling is feasible, in order to illustrate the role of the step-size control strategy in the SAEM technique. Let's assume that the input signal z_t is *iid.* and takes values plus or minus one with probability 1/2, the order p of the filter is set to one and the unknown parameters include both the two coefficients of the filter and the Gaussian noise variance σ^2. This finite-valued input model is of much interest in the domain of digital communication and, as discussed in section 2.2, it is the only blind convolution model for which the EM algorithm can be directly applied using the forward-backward recursions. Conditional simulation of the input symbol sequence given the observations and the model parameters can also be carried out from the quantities computed during the forward-backward recursions [8], [22].

Fig. 1 displays the log-likelihood surface (optimized with respect to the noise variance σ^2 for each combination of the two filter coefficients). It is striking to see that even for such a simplistic case, the log-likelihood surface is already quite complex with a local maximum (point labeled LOC) distinct from the actual maximum of the likelihood (indicated by the MLE label). The sequence of dots visible on the surface represent the position of the first 50 iterates of a SAEM sequence with what seems to be the most appropriate tuning of the step-size decay for this particular model (see Fig. 2). Of course, since the algorithm is stochastic, the sequence of estimates depends not only on the initial guess of the parameters (there are positions from which the algorithm will eventually converge to the LOC point) but also on the particular run of the algorithm.

Fig. 2 illustrates the importance of the step-size decay scheme for a proper convergence of the SAEM algorithm. Note that for each setting of the step-size decay, five different runs of the algorithm are displayed in order to give an idea of the randomness of the sequences. Fig. 2 (a) and (b) corresponds to the two limit cases for which convergence of SAEM to a local maxima of the likelihood is guaranteed [41]: Fig. 2-(a) corresponds to a fast decay with very smooth sample paths, while Fig. 2-(b) display results obtained with a slow decay for which the sample paths are rougher and more variable from run to run. With a slow step-size decay, the algorithm converges much more quickly to the region of interest around the mode (in 10 to 20 iterations), but the stabilization of the estimates will take much longer than when using a fast decay. It is possible to improve over this behavior simply by starting to average the estimates at some point (see [54], [38] for a full account of the merits of averaging) as in Fig. 2-(c), where the estimates obtained after the tenth iteration are averaged. It is important to note that for optimal convergence behavior, the averaging should be performed afterwards and that in no way should the averaged estimates be used while running the SAEM. Finally, Fig. 2-(d) shows that in most cases it is very useful to allow for a burn-in period during which the step-size is set to 1 (as if running the SEM algorithm) so as to locate more rapidly the region of interest.

5.2 Impulsive input

We now consider a case which genuinely requires sophisticated methods such as the ones considered in this paper. This model, which is commonly used for the deconvolution of seismic traces measurements [47], [13], [39], assumes that the input signal is distributed as a mixture of two

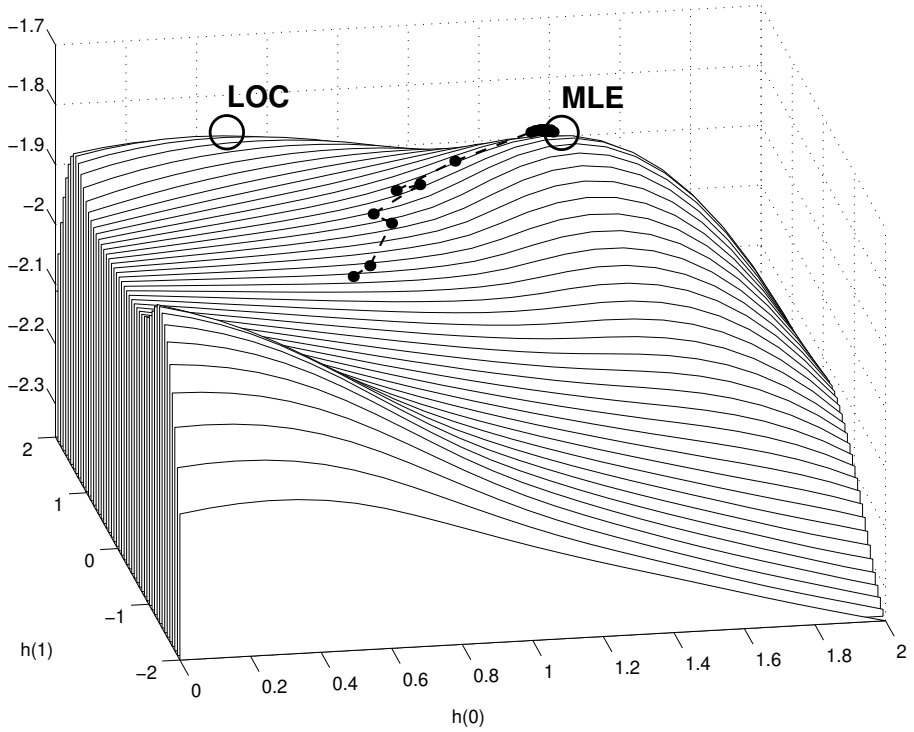

Figure 1: Profile log-likelihood $\max_{\sigma^2} \log p(\mathbf{y}; \mathbf{h}, \sigma^2)$ for a sequence of $T = 150$ observations (6 dB SNR). The dots corresponds to a sequence of SAEM estimates using the weight decay scheme of Fig. 2-(d). The point labeled LOC is a local maxima of the likelihood.

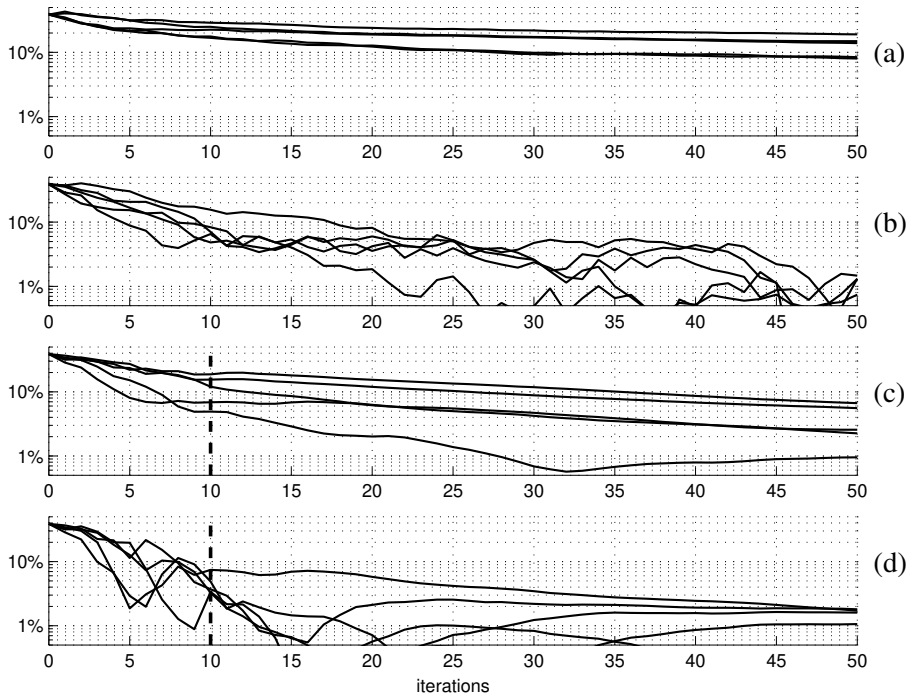

Figure 2: Relative deviation to the MLE (in L^2 norm) on a log-scale for five SAEM sample paths for different step-size decay schemes: **(a)** $\gamma_n = n^{-1}$; **(b)** $\gamma_n = n^{-1/2+\epsilon}$; **(c)** $\gamma_n = n^{-1/2+\epsilon}$, with averaging after the tenth iteration; **(d)** $\gamma_n = n^{-1/2+\epsilon}$ with averaging for $n > 10$, and a burn-in period of 10 iterations (with $\gamma_n = 1$ for $n \leq 10$).

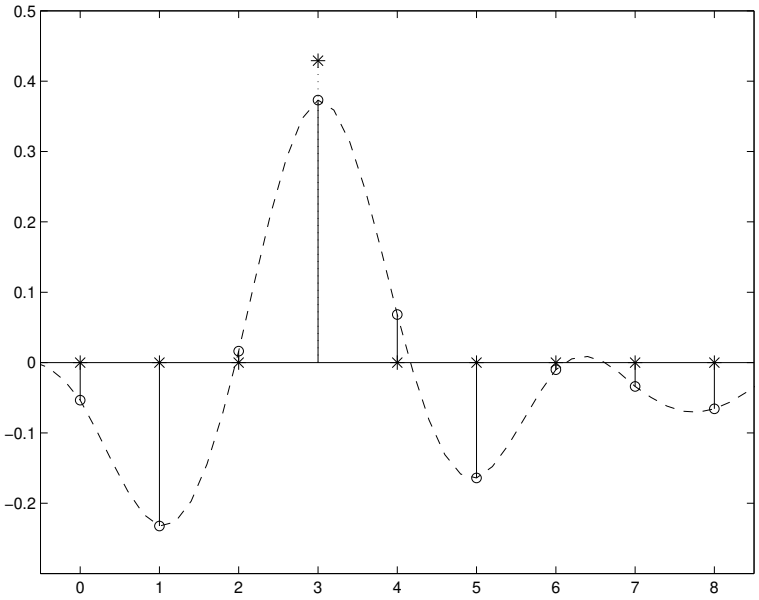

Figure 3: Actual convolution filter (circles) with initial guess (stars). The dotted line features the continuous-time wavelet.

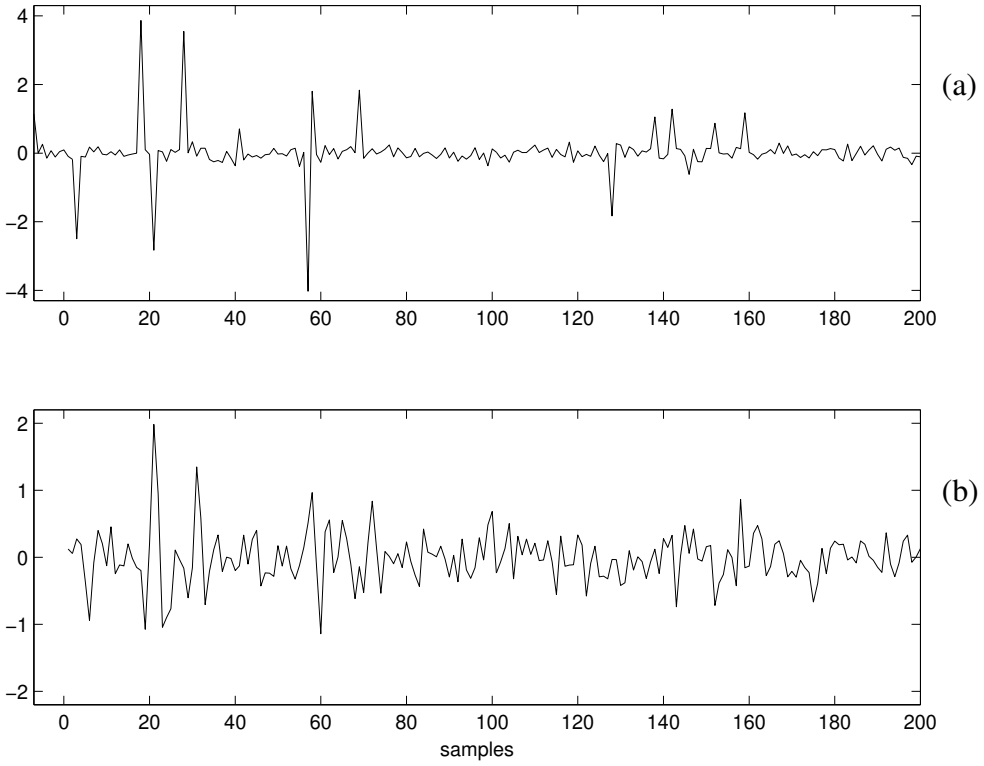

Figure 4: Input sequence and noise corrupted filter output (5 dB SNR).

Gaussian distributions

$$z_t \sim \lambda\phi(0, \rho_1^2) + (1 - \lambda)\phi(0, \rho_2^2),$$

where $\rho_2 \gg \rho_1$, and λ is close to 1. Although simple, this Bernouilli-Gaussian model provides a realistic characterization of impulsive input sequences. We use a synthetic data set whose parameters are taken from the analysis of actual seismic data: $\lambda = 0.9$, $\rho_1^2 = 0.02$, $\rho_2^2 = 5$, $T = 200$ (data size), $p = 8$ (filter order), Signal-to-Noise Ratio (SNR) 5 dB (white Gaussian noise). The filter used for the simulation is displayed on figure 3 (points indicated by circles), while the dotted line indicates the continuous time "wavelet" (see [47] for details concerning the geophysical application) from which it is sampled. Fig. 4 shows the input signal together with the observed signal (their time origins slightly differ since the input signal is extended to include the initial state $\mathbf{z}_1$). The parameters to be estimated include both the vector of filter coefficients $\mathbf{h}$ and the observation noise variance σ^2. The estimation problem is quite complicated both because of the low SNR and of the form of the convolution filter itself (non minimum-phase system, with only two zeros close to the unit circle).

Considering the multimodal aspect of the likelihood observed for the much simpler case of Fig. 1, it can be expected that the estimation will be sensitive to the choice of the initial guess of the parameters. It turns out that the key point is to start from a reliable estimate of the overall delay introduced by the filter, otherwise the procedure converge to time-shifted versions of the correct filter[1]. To allow for more meaningful comparisons, we will thus only discuss the results obtained with the initialization displayed on figure 3 (coefficients indicated by stars) which correspond to a simple 3 taps delay.

Compared to the example considered in the previous section (finite-valued input), there is a very significant conceptual difference: For the example under consideration, it is no longer possible to obtain exact independent simulations distributed under the conditional distribution $p(\mathbf{z}_{1:T}|y_{1:T}; \mathbf{h}, \sigma^2)$. It is thus necessary to use the Markov chain sampling techniques described in section 4. It would of course be very inefficient to re-initialize theses simulations chains and run them for a large number of iterations (so as to obtain independent and approximately distributed samples) in between each SAEM iteration. In practise, only a few cycles of the Markov chain simulation sampler are run at each SAEM iteration and the chains are not reinitialized but are rather re-started from their current value. Although very efficient in practise, this combination of SAEM and Markov chain simulation still needs to be evaluated theoretically since currently available results concern the case of exact independent sampling [41]. For the model under consideration, the use of 3 complete simulation cycles (ie. 3 consecutive simulations of the complete missing data vector) per SAEM iteration seems to be sufficient to guarantee a proper convergence behavior. The requirement to perform several complete simulation cycles was found to be specially stringent for the initial SAEM iterations where the parameters undergo large changes (which means that it would probably be possible to use only one simulation cycle when the estimates start to stabilize).

Figure 5 and 6 display a 3D representation the trajectories of the estimated filter coefficients (for a particular run of the algorithm), with the earliest iterations at the back of the figure and the filter taps from left to right. All the simulation presented in this section use an initial burn-in period (with no step-size decrease) of 100 iterations, and the 400 subsequent iterations are run with a slow step-size decay ($\gamma_n = n^{-0.6}$) and averaging (which corresponds to the settings of Fig. 2-(d) with a number of iterations multiplied by a factor 10). The need for a larger number of iterations than for the simple example of section 5.1 is due both to the intrinsic difficulty of the task considered here and to the fact that we are using Monte Carlo (ie. correlated and non-exactly distributed) simulations. Figure 5 corresponds to the use of the single component

[1] It can be shown that the blind convolution model is indeed identifiable up to a scale factor, as long as the filter order is correctly specified. In practise however, for a moderate observation length, shifted versions of the filter lead to high likelihood values, particularly in cases such as the one considered here where the filter has very small coefficients near the first and last taps.

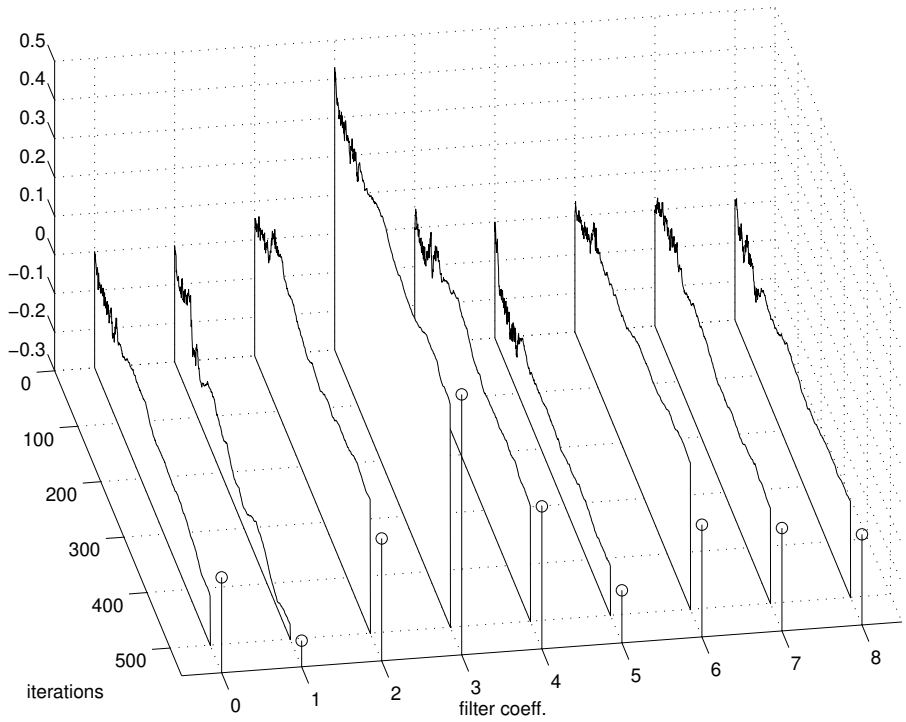

Figure 5: Estimates of the filter coefficients using single component independent sampler (Sec. 4.2) with 3 sampler cycles per SAEM iteration.

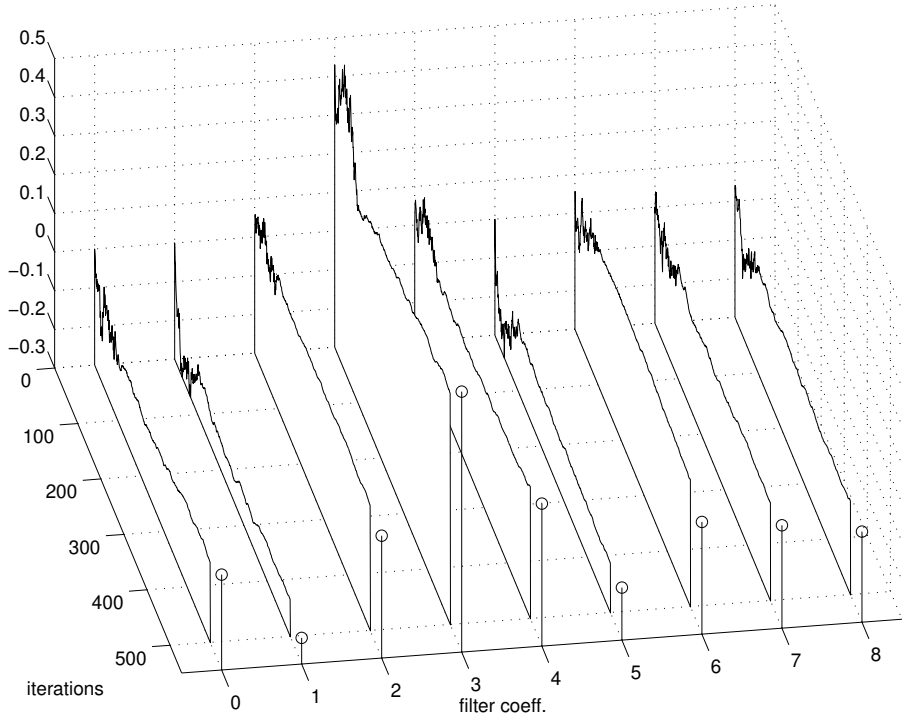

Figure 6: Estimates of the filter coefficients using block Gibbs updates (Sec. 4.3) with 3 cycles per SAEM iteration.

sampling scheme described in section 4.2, while figure 6 corresponds to the use of the block-Gibbs sampling procedure (which is based on data-augmentation using the mixture indicators $\mathbf{q}_{1:T}$ as auxiliary variables – see section 4.3). For comparison purpose, the coefficients of the filter used for simulating the data are displayed at the front of each diagram (points indicated by circles).

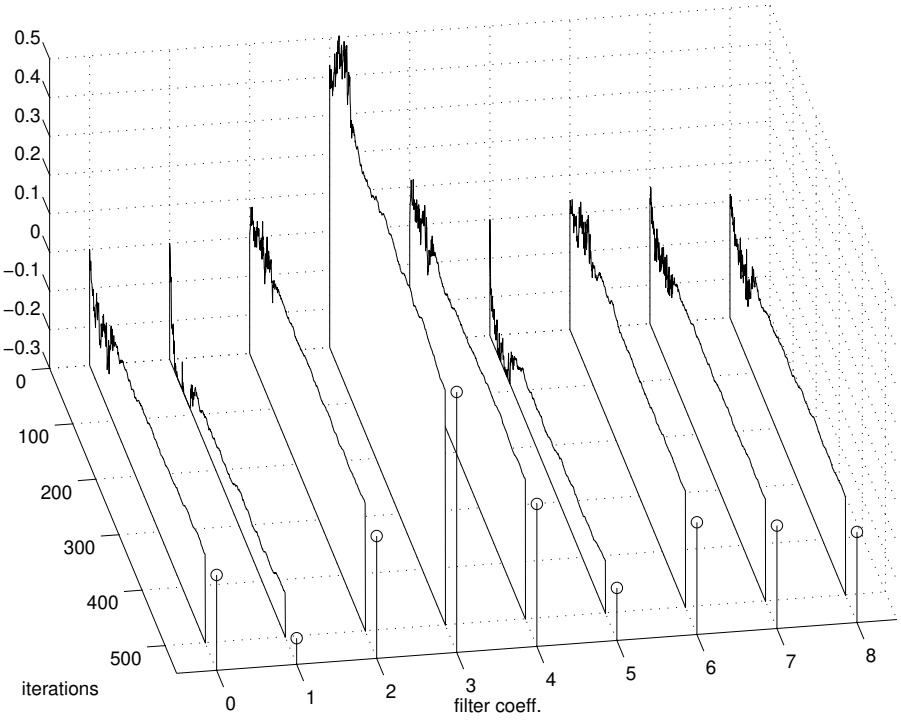

Figure 7: Estimates of the filter coefficients using single component independent sampler (Sec. 4.2) with 30 sampler cycles per SAEM iteration.

With the single site sampling technique (Fig. 5), the estimates converge more slowly and there are clear indications that the SAEM algorithm is indeed still far from having converged after 500 iterations (this is particularly obvious for taps 1 and 3). On the other hand, on Fig. 6, the estimates obtained with the block Gibbs sampling technique stabilize very quickly: After the first 100 iterations (which correspond to the initial burn in period) only slight adjustments are visible which is in contrast with the slow drifts observed for some taps on Fig. 5. This significant difference in the efficiency of the simulation samplers based on single site sampling and block sampling has been observed for many applications involving similar state space models [17], [8], [53]. Available results suggest that the block sampling approach significantly reduces the correlation between subsequent simulations compared to the single site update strategy and thus improve the mixing properties of the simulation chain. This point is illustrated on figure 7 where the single site update strategy is used with 30 complete simulation cycles per SAEM iteration: When allowed to run for larger numbers of simulation steps, the simpler approach does yield results which are quite comparable to that of the block sampling scheme of Fig. 6.

Without getting into very detailed argument about the respective computation load and memory requirement associated with the two sampling options, it is clear that the computation time needed for running a single cycle of the block sampling approach is slightly more than for a cycle of the single site update but certainly less than what is needed for 10 complete cycles of the single site update. In practise, the choice of the simulation component should thus take into account the possibility or not to obtain a Gaussian mixture representation for the source (and noise) signal, which is a requirement for using the more sophisticated sampling approaches of sections 4.3-4.4 (see [10], [36] for examples of the use of "approximate" mixture representations).

6 Further topics

As a conclusion, we would like to point out several directions in which the techniques presented in this paper could be extended.

The assumption that the noise is Gaussian may be unrealistic in some applications (in an impulsive noise environment, for instance). This assumption can be relaxed, the only strict requirement being that the complete data-likelihood still belongs to the curved exponential family. The most direct extension consists in using a Gaussian mixture model for the noise pdf [29], [23]. In this case, all the simulation techniques described in section 4 can be used since the model still is Gaussian conditionally to the source and noise mixture indicators (which corresponds to the most general setting used in [8], [17] or [9]). Mixture of more general exponential distributions can also be considered. Note however that in this case, one has to resort to the systematic sweep Metropolis-Hasting strategy (section 4.2).

As another possible extension, one may consider a general ARMA model instead of the MA model. Such an extension is straightforward when the underlying ARMA model has a Markovian representation similar to that given by eqs. (27)-(28). Extension to non-causal AR and ARMA models is still an open problem (see Rosenblatt [59] and the references therein).

Finally, as mentioned in the introduction, the techniques presented therein are closely related to the so-called *fully* Bayesian estimation. In this approach, a prior (possibly non-informative) pdf is selected for the model parameter and posterior mean of the parameter pdf is used as a point estimator. Several fully Bayesian blind deconvolution methods are presented in [29], [23], [1], [15], [14]. The potential advantage of this approach lies in the fact that it is, at least theoretically, insensitive to the initialization. These methods are however computationally more demanding because they need to explore the full posterior distribution of the parameters and not just only the neighborhood of a maximum of the likelihood function. Comparison of the approach presented in this paper with fully Bayesian estimation, especially for moderate sample sizes, is thus an interesting and important question.

References

[1] C. Andrieu, A. Doucet, and P. Duvaut. Bayesian estimation of filtered point processes using markov chain monte carlo methods. In *Proc. Conf. IEEE Asilomar*, 1997.

[2] C. Antón-Haro, J. A. R. Fonollosa, and J. R. Fonollosa. Blind channel estimation and data detection using hidden Markov models. *IEEE Trans. Signal Processing*, 45(1), January 1997.

[3] J. Besag. The statistical analysis of dirty picture. *J. Royal Statist. Soc. Ser. B*, 48(1):259–302, 1986.

[4] J. Besag, P. J. Green, D. Higdon, and K. Mengersen. Bayesian computation and stochastic systems. *Statistical Science*, 10(1):3–66, 1995.

[5] J. C. Biscarat. Almost sure convergence of class of stochastic algorithms. *Stoch. Proc. Appli.*, 50:83–99, 1994.

[6] J. A. Cadzow. Blind deconvolution via cumulant extrema. *IEEE Signal Proc. Magazine*, 1996.

[7] P. E. Caines. *Linear stochastic systems*. Wiley, 1988.

[8] C. K. Carter and R. Kohn. On Gibbs sampling for state space models. *Biometrika*, 81(3):541–553, 1994.

[9] C. K. Carter and R. Kohn. Markov chain Monte Carlo in conditionnaly Gaussian state space models. *Biometrika*, 83(3):589–601, 1996.

[10] C. K. Carter and R. Kohn. Semiparametric bayesian inference for time series with mixed spectra. *J. Royal Statist. Soc. Ser. B*, 59(1):255–268, 1997.

[11] G. Casella and C. P. Robert. Rao-Blackwellisation of sampling schemes. *Biometrika*, 83(1):81–94, 1996.

[12] G. Celeux and J. Diebolt. The SEM algorithm: a probabilistic teacher algorithm derived from the EM algorithm for the mixture problem. *Computational statistics*, 2:73–82, 1985.

[13] F. Champagnat, Y. Goussard, and J. Idier. Unsupervised deconvolution of sparse spike trains using stochastic approximation. *IEEE Trans. Signal Processing*, 44(12):2988–2998, 1996.

[14] R. Chen and T. Li. Blind restoration of linearly degraded discrete signals by gibbs sampling. *IEEE Trans. Signal Processing*, 43(10):2410–2413, 1995.

[15] Q. Cheng, R. Chen, and T. Li. Simultaneous wavelet estimation and deconvolution of reflection seismic signals via Gibbs sampler. *IEEE Trans. Geoscience and Remote Sensing*, 34:377–384, 1996.

[16] S. Chib and E. Greenberg. Understanding the metropolis-hastings algorithm. *The American Statistician*, 49:327–345, 1995.

[17] P. De Jong and N. Sheppard. The simulation smoother for time series models. *Biometrika*, 82(2):339–350, 1995.

[18] G. Demoment. Image reconstruction and restoration: Overview of common estimation structures and problems. *IEEE Trans. Acoust., Speech, Signal Processing*, 37(12):2024–2036, December 1989.

[19] A. Dempster, N. Laird, and D. Rubin. Maximum-likelihood from incomplete data via the EM algorithm. *J. Royal Statist. Soc. Ser. B*, 39:1–38, 1977.

[20] J. Diebolt and E. H. S. Ip. Stochastic EM: method and application. In W. R. Gilks, S. Richardson, and D. J. Spiegelhalter, editors, *Markov Chain Monte Carlo in Practice*, pages 259–273. Chapman & Hall, 1996.

[21] D. L. Donoho. On minimum entropy deconvolution. In D. F. Findley, editor, *Applied Time-Series Analysis*. Academic Press, 1981.

[22] A. Doucet and P. Duvaut. Fully baesian analysis of hidden Markov models. In *9th EUSIPCO conference*, Trieste, 1996.

[23] A. Doucet and P. Duvaut. Bayesian estimation of state-space models applied to deconvolution of Bernoulli-Gaussian processes. *Signal Processing*, 57(2):147–161, March 1997.

[24] E. Gassiat, F. Montfront, and Y. Goussard. On simultaneous signal estimation and parameter identification using a generalized likelihood approach. *IEEE Trans. Inform. Theory*, 38(1):157–162, 1992.

[25] A. E. Gelfand and A. F. M. Smith. Sampling based approach for calculating marginal densities. *J. Amer. Stat. Assoc.*, 85:398–409, 1990.

[26] S. Geman and D. Geman. Stochastic relaxation, Gibbs distribution, and the Bayesian restoration of images. *IEEE Trans. on Pattern Anal. Machine Intell.*, 6:721–741, 1984.

[27] G. Giannakis and J.M. Mendel. Identification of non-minimum phase systems using higher-order statistics. *IEEE Trans. Acoust., Speech, Signal Processing*, 37:360–377, 1989.

[28] W. R. Gilks, S. Richardson, and D. J. Spiegelhalter, editors. *Markov Chain Monte Carlo in Practice*. Interdisciplinary Statistics Series. Chapman & Hall, 1996.

[29] S. J. Godsill. 'bayesian enhancement of speech and audio signals which can be modelled as arma processes. *Int. Stat. Rev.*, 65(1):1–21, 1997.

[30] S. Haykin, editor. *Blind Deconvolution*. Englewood Cliffs,NJ,Pretice Hall, 1994.

[31] Y. Hua. Fast maximum likelihood for blind identification of multiple FIR channels. *IEEE Trans. Signal Processing*, 44(3):661–672, 1996.

[32] E. H. S. Ip. A stochastic EM estimator in the presence of missing data - Theory and applications. Technical Report 304, Department of Statistics, Stanford University, 1994.

[33] K. S. Lii and M. Rosenblatt. Deconvolution and estimation of transfer function phase and coefficients for non-gaussian linear processes. *Annals of Statistics*, 10:1195–1208, 1982.

[34] G. K. Kaleh and R. Vallet. Joint parameter estimation and symbol detection for linear or non-linear unknown channels. *IEEE Trans. Communications*, 42(7), 1994.

[35] H. Kesten. Accelerated stochastic approximation. *Annals of Mathematics and Statistics*, 29:41–59, 1958.

[36] S. Kim, Neil Shephard, and Siddhartha Chib. Stochastic volatility, likelihood inference and comparison with arch models. *Review of Economic Studies*, To appear, 1998.

[37] S. J. Koopman. Disturbance smoother for state space models. *Biometrika*, 80(1):117–126, 1993.

[38] H. J. Kushner and G. G. Yin. *Stochastic approximation algorithms and applications*. Springer-Verlag, New York, 1997.

[39] M. Lavielle. 2-D Bayesian deconvolution. *Geophysics*, 56(12):2008–2018, 1991.

[40] M. Lavielle. Bayesian deconvolution of Bernoulli-Gaussian processes. *Signal Processing*, 33:67–79, 1993.

[41] M. Lavielle, E. Moulines, and B. Delyon. Convergence of a stochastic approximation version of the EM algorithm. *Annals of Stat. (to appear)*, 1998.

[42] H. Liu and G. Xu. A deterministic approach to blind symbol estimation. *IEEE Signal Processing Letters*, 1(12):205–207, 1994.

[43] J. S. Liu, W. H. Wong, and A. Kong. Covariance structure of the gibbs sampler with applications to the comparisons of estimators and augmentation schemes. *Biometrika*, 81(1):27–40, 1994.

[44] D. P. Loti-Viaud. Random perturbations of recursive sequences with an application to epidemic models. *J. Applied Probability*, Jan 1995.

[45] I. L. MacDonald and W. Zucchini. *Hidden Markov models and other models for discrete-valued time series*. Chapman & Hall, 1997.

[46] J. Mendel. White-noise estimators for seismic data processing in oil exploration. *IEEE Trans. Automatic Control*, 22(5):694–706, 1977.

[47] J. Mendel. *Optimal seismic deconvolution.* Academic Press, 1983.

[48] X-L. Meng and D. van Dyk. The em algorithm – an old folk-song sung to a fast new tune (with discussion). *J. Royal Statist. Soc. Ser. B*, 59(3):511–567, 1997.

[49] E. Moulines, J.-F. Cardoso, and E. Gassiat. Maximum likelihood for blind separation and deconvolution of noisy signals using mixture models. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, pages 3617–3620, 1997.

[50] P. Muller. A generic approach to posterior integration and gibbs sampling. Technical report, Purdue. University. West Lafayette, 1991.

[51] C.L. Nikias and J.M. Mendel. Signal processing with higher-order spectra. *IEEE Signal Processing Magazine*, (3):10–37, July 1993.

[52] A. P. Petropulu and C. L. Nikias. Blind deconvolution using signal reconstruction from partial higher order cepstral information. *IEEE Trans. Signal Processing*, 41(6):2088–2100, 1993.

[53] M. K. Pitt and N. Shephard. Analytic convergence rates and parameterisation issues for the gibbs sampler applied to state space models. *J. Time Ser. Anal*, To appear, 1989.

[54] B.T. Polyak. New stochastic approximation type procedures. *Automatica i Telemekh.*, pages 98–107, 1990. translated in Autom. & Remote Contr. vol. 51(7).

[55] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE*, 77(2):257–285, February 1989.

[56] R. A. Redner and H. F. Walker. Mixture densities, maximum likelihood and the EM algorithm. *SIAM Review*, 26(2):195–239, April 1984.

[57] C. P. Robert. *Méthodes de Monte Carlo par chaînes de Markov.* Economica, 1996.

[58] G. O. Roberts and S. K. Sahu. Updating schemes, correlation structure, blocking and parameterization for the gibbs sampler. *J. Royal Statist. Soc. Ser. B*, 59(2), 1997.

[59] M. Rosenblatt. The likelihood of an autoregressive scheme. In P. M. Robinson and M. Rosenblatt, editors, *Athens conference on applied probability and time series (volume II)*, Lecture notes in statistics. Springer-Verlag, 1996.

[60] N. Seshadri. Joint data and channel estimation using blind trellis search techniques. *IEEE Trans. on Communications*, 42(2):1606–1616, 1994.

[61] A. F. M. Smith and G. O. Roberts. Bayesian computation via the gibbs sampler and related Markov chain Monte Carlo methods. *J. Royal Statist. Soc. Ser. B*, 55(1):3–23, 1993.

[62] M. Tanner. *Tools for statistical inference: methods for exploration of posterior distributions and likelihood functions.* Springer-Verlag, 1993.

[63] L. Tierney. Markov chains for exploring posterior distributions. *Annals of Statistics*, 22:1701–1762, 1994.

[64] D. M. Titterington, A. F. M. Smith, and U. E. Makov. *Statistical analysis of finite mixture distributions.* Wiley, 1985.

[65] J. K. Tugnait. Blind equalization and estimation of digital communication FIRq channels using cumulant matching. *IEEE Trans. on Communications.*, 43(2):1240–1260, 1995.

[66] G. Wei and M. Tanner. A Monte-Carlo implementation of the EM algorithm and the Poor's Man's data augmentation algorithm. *J. Amer. Stat. Assoc.*, 85:699–704, 1990.