BAYESIAN METHODS FOR DECONVOLUTION OF SPARSE PROCESSES

by

Sinan Yıldırım

B.S., Electrical and Electronics Engineering, Boğaziçi University, 2007

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Electrical and Electronics Engineering
Boğaziçi University

2009

BAYESIAN METHODS FOR DECONVOLUTION OF SPARSE PROCESSES

APPROVED BY:

Prof. Ayşın Baytan Ertüzün (Thesis Supervisor)	.....
Dr. Ali Taylan Cemgil (Thesis Co-supervisor)	.....
Prof. Mustafa Aktar	.....
Assist. Prof. F. Kerem Harmancı	.....
Prof. Bülent Sankur	.....

DATE OF APPROVAL: 12.06.2009

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my supervisor, Prof. Ayşın Baytan Ertüzün, for her invaluable guidance and help throughout this study. Her initial support motivated me to start my academic study, which would have been so much harder without her intimate attitude, patience and confidence in my ability to succeed.

I am grateful to my co-supervisor Dr. Ali Taylan Cemgil for his enlightening knowledge on technical issues as well as his invaluable friendship. Meeting him was the 'Renaissance' of my master. During my study in Cambridge, he and his family were the source of reliance, friendship, motivation and support.

My valuable thanks go to Prof. Mustafa Aktar and Yaman Özakın for their technical guidance and comments. I also would like to thank Prof. Bülent Sankur and Assist. Prof. F. Kerem Harmancı for their interests and guidance.

Many thanks to my past and present colleagues at BUSIM: Oya Çeliktutan, Erinç Dikici, Doğan Can, Onur Özyeşil, Ebru Arısoy, Sergül Aydöre, Ekin Olcan Şahin, Özgür Dalkılıç, Arman Savran, Hatice Çınar, and Temuçin Som, for providing the happy moments of this stressful period of work.

I would like to thank TÜBİTAK Project number 107E050 which funded my project and provided the financial and motivational means for this research.

I owe a special word of appreciation to Hatice Kılavuz, who shared all the happiness, sorrow, excitement with me. Her presence always made me feel better in my bad times.

Last but not least, I would like to express my deepest gratitude to my dear family, who have supported me with their endless love and understanding.

ABSTRACT

BAYESIAN METHODS FOR DECONVOLUTION OF SPARSE PROCESSES

In this work, various Bayesian methods for deconvolution and blind deconvolution of sparse processes are studied. By using the prior assumption of sparsity, deconvolution and blind deconvolution operations are mapped to inference and parameter estimation methods in a Bayesian framework.

For blind deconvolution of sparse processes, inverse-gamma model is proposed as a relaxation of the well known Bernoulli-Gaussian model. Methods based on expectation-maximization algorithm are investigated for both models, and several statistical inference and parameter estimation techniques are presented for expectation and maximization steps. The improvement in the performance is demonstrated by experiments on simulated data.

Receiver function analysis, a research topic in seismology, is studied as a real life application. Bayesian deconvolution is proposed as an alternative method to iterative deconvolution for estimating receiver functions. The superiority of Bayesian deconvolution is demonstrated both by experiments on both simulated and real data. Also, in this way, the assumption of sparsity for receiver functions is validated by the obtained results.

Finally, a preliminary theoretical solution to a challenging problem of blind estimation of receiver function analysis is developed. The performances of proposed methods for the solution are tested on simulated data.

ÖZET

SEYREK SÜREÇLERİN TERS EVRİŞİMİ İÇİN BAYESÇİ YÖNTEMLER

Bu çalışmada, seyrek süreçlerin ters evrişimi ve gözü kapalı ters evrişimi için çeşitli Bayesçi yöntemler üzerinde çalışıldı. Seyreklik önsel bilgisi kullanılarak, ters evrişim ve gözü kapalı ters evrişim işlemleri Bayesçi bir çerçevede istatistiksel çıkarım ve parametre kestirimi yöntemlerine eşlemlendi.

Seyrek süreçlerin gözü kapalı ters evrişimi için Bernoulli-Gauss modelinin bir yumuşatılması olarak ters-gamma modeli önerildi. Modeller için, beklenti-enbüyütme algoritmasını temel alan yöntemler incelendi, beklenti ve enbüyütme aşamaları için çeşitli istatistiksel çıkarım ve parametre kestirimi teknikleri sunuldu. Başarımlardaki iyileşme, benzetimli veriler üzerinde yapılan deneylerle gösterildi.

Gerçek hayat uygulaması için, bir yerbilimi araştırma konusu olan alıcı fonksiyon analizi üzerinde çalışıldı. Alıcı fonksiyonların kestirminde kullanılan döngülü ters evrişim yöntemine alternatif olarak Bayesçi ters evrişim yöntemi önerildi. Bayesçi ters evrişimin üstünlüğü hem benzetimli hem gerçek veriler üzerinde yapılan deneylerle gösterildi. Ayrıca, bu yolla, alıcı fonksiyonlar için olan seyreklik varsayımı alınan sonuçlar tarafından doğrulandı.

Son olarak, zor bir problem olan alıcı fonksiyonların gözü kapalı ters evrişimi için başlangıç aşamasında bir çözüm geliştirildi. Çözüm için önerilen yöntemlerin başarımı benzetimli veriler üzerinde yapılan deneylerle ölçüldü.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS/ABBREVIATIONS

a	Gaussian filter-width parameter
$\mathbf{A}$	State-transition matrix
$\mathbf{c}$	Indicator sequence
$\mathbf{C}$	Diagonal matrix of $\mathbf{c}$
c_k	Indicator value at time k
e	State noise
exp	Exponential
$\mathbf{g}$	Vector presentation of the gaussian filter
G	Gaussian filter in frequency domain
$\mathbf{G_s}$	Convolution matrix corresponding to $\mathbf{g}$ when convolved with $\mathbf{s}$
$\mathbf{G_h}$	Convolution matrix corresponding to $\mathbf{g}$ when convolved with $\mathbf{h}$
h	(Sparse) filter component
$\mathbf{h}$	Vector presentation of h
$\mathbf{H}$	Convolution matrix corresponding to h
$H(\cdot)$	Differential entropy
$\mathbf{I}$	Identity matrix
$\mathcal{IG}$	Inverse-gamma distribution
KL	Kullback-Leibler distance
log	Natural logarithm
$\mathbf{m}$	Gaussian mean
M	Order of autoregressive process
N	Number of samples
$\mathcal{N}$	Gaussian distribution
$p(\cdot)$	Probability density function
$p(\cdot\|\cdot)$	Conditional probability density function
$P(\cdot)$	Target distribution
$q(\cdot)$	Proposal distribution

$Q(\cdot)$	Variational distribution
r	Sparse process filtered with the Gaussian filter
$\mathbf{r}$	Vector presentation of r
$\mathbf{R}$	The matrix of autoregressive coefficients
s	Source component
$\mathbf{s}$	Vector presentation of s
$\mathbf{S}$	Vector presentation of s
$s * h$	Convolution of s and h
S_f	Sufficient statistics of f
T	Data length
T_s	Length of the source component
T_y	Length of the observation signal
T_h	Length of the (sparse) filter component
$tr(\cdot)$	Trace
$\mathbf{U}$	State noise matrix
v	Observation noise component
$\mathbf{v}$	Vector presentation of v
w	Noise component in the autoregressive process
$\mathbf{w}$	Vector presentation of w
x	Latent variable
$x \sim p(x)$	x comes from $p(x)$
$x^{(i)}$	$^{(th)}$ sample for x
x'	New candidate sample of x
$\hat{x}$	Estimated value for x
$\mathcal{VB}_Q$	Variational bound of Q
y	Noisy convolution result
$\mathbf{y}$	Vector presentation of y
z	Noiseless convolution
$\mathbf{z}$	Vector presentation of z
$\mathcal{Z}$	Normalizing constant
α	Shape parameter of inverse-gamma distribution

β	Scale parameter of inverse-gamma distribution
δ	Delta-dirac function
ϵ	Acceptance ratio
$\Gamma(\cdot)$	Gamma function
λ	Spike density
ν	Weight parameter in Monte-Carlo related expectation-maximization algorithms
ρ	Autoregressive coefficients
σ_0^2	Lower variance of the Bernoulli-Gaussian distribution
σ_1^2	Higher variance of the Bernoulli-Gaussian distribution
$\sigma_{h,k}^2$	Sample variance of the sparse process
σ_v^2	Observation noise variance
σ_w^2	AR driving noise variance
$\mathbf{\Sigma}$	Gaussian covariance
Θ	Parameter set
ω	Angular frequency
Ω	Intermediate quantity of expectation-maximization
$\lvert \cdot \rvert$	Determinant
$[\cdot]_{i,j}$	i,j^{th} element of the matrix
$[\cdot]_i$	i^{th} element of the vector
$\approx$	Approximately equal to
$\langle \cdot \rangle_{p(\cdot)}$	Expectation under p
$[\cdot]^\top$	Matrix transpose
$[\cdot]^{-1}$	Matrix inversion
$\propto$	Proportional to
$\equiv$	Definition

AR	Autoregressive
CGLSSM	Conditionally Gaussian linear state-space model
EM	Expectation-maximization
FIR	Finite impulse response
IIR	Infinite impulse response

IS	Importance sampling
i.i.d	Independently and identically distributed
KL	Kullback Leibler
LTI	Linear time invariant
MCEM	Monte-Carlo expectation-maximization
MCMC	Markov chain Monte-Carlo
MCMC1	First Markov chain Monte-Carlo method
MCMC2	Second Markov chain Monte-Carlo method
MCMC	Markov chain Monte-Carlo
MLE	Maximum-likelihood estimation
MSE	Mean squared error
nMSE	Normalized mean squared error
pdf	Probability density function
RS	Rejection Sampling
SAEM	Stochastic approximation expectation-maximization
SEM	Stochastic expectation-maximization
SNR	Signal-to-noise ratio

1. INTRODUCTION

The problem of the estimating signals which are distorted by *linear time invariant* (LTI) systems and possibly some additive noise arises in many fields such as seismic exploration, [1, 2, 3], acoustics [4], communication [5] and speech enhancement [4]. This problem is usually modeled with a discrete-time (noisy) *convolution* for the observations. The aim of *deconvolution* is to estimate the latent signals when the observation and the system are known. When the LTI system is unknown as well, we have the *blind deconvolution* problem.

There are several methods for deconvolution in literature. When there is no noise, or the noise is negligible, *inverse filtering* technique can be used [6]. Inverse filtering is performed simply by division in the frequency domain. However the frequency domain deconvolution is not often preferred because small or zero values of the denominator cause numerical problems in the calculation. There are several approaches to avoid this problem, the simplest is the ad hoc approach called *water-level deconvolution*, which was adopted by Langston in 1979 [7], works efficiently when the data quality is good. In water-level deconvolution, the way we avoid division by small numbers is to replace small values in the denominator with a fraction of the maximum value of the denominator. The fraction is called the *water-level parameter* and is often chosen by trial and error. The fraction multiplied by the maximum denominator amplitude is called the *water level*. It is also possible to avoid the numerical problem of the inverse filtering by using the *iterative deconvolution* method [8], which is a time domain deconvolution method. Iterative deconvolution approach is a least-squares minimization of the difference between the observed signal and its predicted version generated by the convolution of the estimated signal with the given component. It is not only numerically stable, but also enables *a priori* constructing the deconvolution as a sum of shifted versions of a certain waveform, such as a Gaussian pulse [9].

In many situations, there is considerable noise corrupting the observing signal. In that case, if we use the methods above, which assume a noiseless system, the error

in the results will get higher as the *signal-to-noise ratio* (SNR) decreases. However, if we have at least some knowledge of the type of noise in the data (for example, white noise), we may be able to improve our estimate through techniques such as Wiener deconvolution [10] and finite impulse response (FIR) Wiener filters [11].

Many algorithms have been proposed for blind deconvolution problem. The algorithms usually differ according to whether the unknown system is minimum phase or non-minimum phase. If the unknown systems are known to be minimum phase, causal filters are used as deconvolving filters. Many algorithms work well in learning the coefficients of causal filters, such as the second-order statistical approaches [12, 13, 14], higher-order statistical approaches [15, 16], and the Bussgang algorithms [17, 18]. For non-minimum phase systems, Amari et al. [19] used doubly sided infinite impulse response (IIR) filters as deconvolving models. Cascade structures for the deconvolving filters were also introduced by several authors, including Labat *et al* [20] and Xia *et al* [21].

A reasonable assumption that holds for many practical deconvolution or blind deconvolution problems is that the latent input signal is *sparse*. We call a signal sparse if it is zero -or very close to zero- almost everywhere but has a few non-zero values far from zero, often referred to as *spikes* [22]. In many applications, the goal is to estimate the locations and amplitudes of these spikes.

We encounter sparse signals in many deconvolution problems such as reflectivity sequence estimation using marine seismic data [3], room impulse response estimation [4], and *receiver function analysis* in seismology [9, 8]. We will now go in details of receiver function analysis, which constitutes the real life application part of our work.

Receiver functions are time series, computed from three-component seismograms, which show the relative response of earth structure near the receiver [9]. A three-component seismogram records P and S waves where the initial P wave is directly caused by a seismic event under the ground and it is refracted to an S wave due to velocity discontinuities in the earth structure. Due to the refractions from the P wave

to the S wave, it is assumed that the S wave is the convolution of the P wave with a function indicating the local velocity structure, and this function is called *the receiver function*. The P wave is largely recorded on the vertical component of the seismogram, whereas the radial component mostly consists of the S wave. Hence receiver function estimation methods are based on performing deconvolution of the radial component using the vertical component. From receiver functions, one can learn characteristics of the geological structure near the receiver, such as crustal thickness and shallow velocity structure [23].

The earliest receiver function studies such as Phinney [6] worked in the frequency domain. The method of Phinney was extended by Langston [7], who included phase information by using a complex frequency-domain ratio and inverse transforming back into the time domain. For deconvolution, he used the water-level decomposition method and a low-pass Gaussian filter to remove high-frequency noise in the results. Ammon [9] described and applied an iterative, time-domain deconvolution approach to receiver-function estimation and illustrate the reliability and advantages of the technique using synthetic simulations and experiments with real data. Özakın [23] applied iterative deconvolution for extracting the crustal structure of Southwestern Anatolia, and reported that iterative deconvolution outperformed the inverse filtering and water level deconvolution methods.

Since the convolution model in receiver function analysis arises from the refractions from the P wave to the S wave, the receiver function can be assumed as sparse, where its spikes are the indicators of these refractions. However, the sparsity assumption is not used in the convolution methods stated above. Only in iterative deconvolution, the number of iterations is limited to some value which is chosen heuristically, so that the estimate remains sparse. However, since this prevents the algorithm from converging to a solution by doing this, we believe that it is not a very reliable way to handle sparsity. Moreover, the signals recorded by the seismogram are noisy, which reduces the performance of the deconvolution algorithms above. Finally, iterative deconvolution needs the complete convolution data to produce reliable results. However, this is not the case in seismic recordings, where the vertical component is as long as

the radial component, hence we have a truncated observation.

In receiver function analysis, there are also some cases where the vertical component is too noisy to be trustable. This is the case for daily recorded data, when big earthquakes do not occur. We can not apply deconvolution methods on such data for the estimation of receiver functions. Hence, blind deconvolution is needed for such cases.

Bayesian theory allows us to formulate our prior knowledge about the data which is ignored by the classical frequentist approaches. Mainly, by means of Bayesian approach, uncertainties about the parameters of interest are taken into account. That is, parameters are considered to be random variables unlike the frequentist approaches where they are assumed to be deterministic. Basically, our *a priori* knowledge on these variables of interest is shaped by the *likelihood* of the observed data, giving the *a posteriori* knowledge. This intuitive idea is formulated by the well known *Bayes rule* [24], and used effectively in the solution of many problems. Since the sparsity is an important information about the characteristics of a signal, using Bayesian methods for deconvolution and blind deconvolution of sparse signals can significantly improve performance. Moreover, since the Bayesian methods are model-based, they can handle observation noise and truncation problems.

The most crucial point of applying Bayesian methods in sparse applications is the choice of the form of the prior distribution, that is, the statistical model used for sparsity. In literature, a popular approach is to model the coefficients of a sparse process by a *Bernoulli-Gaussian* distribution [25, 3, 26, 27]. For example, Rosec *et al* [3] used the Bernoulli-Gaussian model for reflectivity sequence estimation to improve the resolution of blind seismic images. Labat *et al* [25] used the same model for sparsity to perform sparse blind deconvolution accounting for time-shift ambiguity. The Bernoulli-Gaussian model is so popular in literature because it is very intuitive and its statistical properties allow easy implementation of Bayesian methods such as *Markov chain Monte-Carlo* (MCMC) and the *expectation-maximization* (EM) [26, 27]. However, this model suffers from multi-modality and needs very good initializations.

This severely effects the performance when we do not have prior knowledge about the unknown variables of interest.

Modeling samples of a sparse signal with a Gaussian random variable whose variance is an *inverse-gamma* random variable could be an alternative approach. This prior assumption is used in source separation and blind deconvolution problems [28, 29, 30]. It was also used by Févotte and Godsill [31] for blind separation of sparse sources. Yıldırım *et al* [32] also used the inverse-gamma assumption for the variance to provide better initialization for blind deconvolution of Bernoulli-Gaussian processes. In contrast to the Bernoulli-Gaussian case, the inference problems tend to be easier, since inverse-gamma assumption leads to more smooth distributions/objective functions. In fact, relaxation strategies to create smoother surfaces already exist in signal processing area. For example, *simulated annealing* is a powerful method in Bayesian signal processing for calculating quantities or maximization over an intractable function [33, 34]. Another example is the *gradient vector flow* method, which is used in the study of *active contours* [35]. In this line of thinking, we can say that the inverse-gamma model is a relaxation of the Bernoulli-Gaussian model. Moreover, like the Bernoulli-Gaussian model, the inverse-gamma model enables use of statistical inference and parameter estimation using classical methods such as *Gibbs sampling*, *variational Bayes*, and EM.

1.1. Motivations and Objectives of the Thesis

In this work, we use Bayesian methods for deconvolution and blind deconvolution of sparse signals. We address the problem of multi-modality and the necessity of good initialization when the Bernoulli-Gaussian model is used for sparsity. Our main contribution in this work is to propose the inverse-gamma model as a relaxation of the Bernoulli-Gaussian model, and to significantly increase the performance of inference and parameter estimation algorithms for blind deconvolution. The possible inference and parameter estimation methods for both models are investigated, and it is shown that the performance of the inverse-gamma model is better for every combination of these methods.

Another contribution of this work is to introduce a Bayesian deconvolution approach for receiver function analysis. We have already stated the disadvantages of the iterative deconvolution that it does not use the sparsity assumption, it is not reliable in noisy cases and it does not handle truncation. We show through experiments with both simulated and real data that Bayesian deconvolution is more robust to those problems, and produce better results. For real life experiments, we will use three sets of data recorded in different seismic stations in Turkey by Boğaziçi University Kandilli Observatory and Earthquake Research Institute.

We believe that a Bayesian approach can help blind estimation of receiver functions in cases where the source is not reliable. Motivated with this idea, we also investigated methods for blind deconvolution where we assign statistical models for both of the components in the convolution, that is, both the system and the source. Specifically, we deal with the case when the source signal is an AR process, which is a reasonable assumption for the vertical component, and the filter is sparse. This is a very challenging problem and in this thesis we present a preliminary solution for this problem.

1.2. Scope of the Thesis

The organization of the thesis is as follows: In chapter 2, the theoretical background is given. In chapter 3, various Bayesian methods for deconvolution and blind deconvolution for the Bernoulli-Gaussian and inverse-gamma models are investigated, and the simulation based comparison between two models are given. We also compare the performances of Bayesian deconvolution and iterative deconvolution using both simulated and real seismic data. In chapter 4, we deal with blind deconvolution of sparse filters when the source is an autoregressive process. Finally, we discuss the results with concluding remarks and give possible directions for future work in Chapter 5. The Appendix section consists of some mathematical derivations and further details about our work.

2. THEORETICAL BACKGROUND

In this chapter, we introduce the theoretical basis of our work. We start with the concepts of convolution, deconvolution and blind deconvolution. Then we relate the use of Bayesian signal processing to deconvolution and blind deconvolution problems. We will elaborate the Bayesian methods used in the thesis, such as Monte-Carlo methods, Gibbs sampling, variational Bayes, and EM algorithm.

2.1. Convolution, Deconvolution and Blind Deconvolution

In mathematics, convolution is a mathematical operation on two functions s and h, written as $s*h$, producing a third function that is typically viewed as a modified version of one of the original functions. Mathematically, the convolution of two functions s and h is defined by the integral transform [36]

$$z(t) = (s * h)(t) = \int_{-\infty}^{\infty} s(\tau)h(t-\tau)d\tau. \tag{2.1}$$

In many applications, t in (2.1) usually denotes for time, though it need not represent the time domain. As well as for continuous time functions, the *discrete convolution* can be defined for functions on the set of integers. For functions s and h defined on the set of integers, the discrete convolution of f and g is defined as [36]

$$z_n = (s * h)_n = \sum_{m=-\infty}^{\infty} s_m h_{n-m}. \tag{2.2}$$

This generalization of the convolution has applications in the design and implementation of LTI systems in discrete time in *signal processing*. Furthermore, when s and h are zero outside a finite interval, we can handle the convolution operation in a *matrix-vector representation*. Assume, for example, $s_n = 0$ outside $T_{s_1} < n \leq T_{s_2}$, and $h_n = 0$ outside $T_{h_1} < n \leq T_{h_2}$. If $T_s = T_{s_2} - T_{s_1}$ and $T_h = T_{h_2} - T_{h_1}$, then (2.2) can be written

as

$$\mathbf{z} = \mathbf{H}\mathbf{s}, \tag{2.3}$$

where the $(T_s + T_h - 1) \times 1$ and $T_s \times 1$ vectors $\mathbf{z}, \mathbf{s}$ and the $(T_s + T_h - 1) \times T_s$ matrix $\mathbf{H}$ are organized as $\mathbf{z} = \begin{bmatrix} z_{T_{s_1}+T_{h_1}+2} & \dots & z_{T_{s_2}+T_{h_2}} \end{bmatrix}^\top$, $\mathbf{s} = \begin{bmatrix} s_{T_{s_1}+1} & \dots & s_{T_{s_2}} \end{bmatrix}^\top$, and $\mathbf{H}_{i,j} = h_{i-j+1+T_{h_1}-T_{s_1}}$, $i = 1, \dots, T_s + T_h - 1$, $j = 1, \dots, T_s$, where $[\cdot]^\top$ denotes vector transposition.

Note that the convolution operation is commutative, hence $(s * h)_n = (h * s)_n$. Thus, we can write (2.3) the other way, that is,

$$\mathbf{z} = \mathbf{S}\mathbf{h}, \tag{2.4}$$

where $\mathbf{h} = \begin{bmatrix} h_{T_{h_1}+1} & \dots & h_{T_{h_2}} \end{bmatrix}^\top$, and $\mathbf{S}_{i,j} = s_{i-j+1+T_{s_1}-T_{h_1}}$, $\quad i = 1, \dots, T_s + T_h - 1$, $j = 1, \dots, T_h$.

In (2.3) and (2.4), the matrices $\mathbf{H}$ and $\mathbf{S}$ are called the *convolution matrices*, whose columns have the shifted versions of the vectors $\mathbf{h}$ and $\mathbf{s}$, respectively. The matrix-vector product expression for the convolution operation is useful for expressing LTI systems where the input passes through a *finite impulse response* (FIR) filter.

In many applications in signal processing, the observed signal is not the convolution itself, but the convolution corrupted by an additive *observation noise*. Then, we slightly modify (2.2) as

$$y_n = (s * h)_n + v_n = \sum_{m=-\infty}^{\infty} s_m h_{n-m} + v_n. \tag{2.5}$$

The matrix vector representation corresponding to (2.5) is

$$\mathbf{y} = \mathbf{z} + \mathbf{v} = \mathbf{H}\mathbf{s} + \mathbf{v} = \mathbf{S}\mathbf{h} + \mathbf{v},$$

with $\mathbf{y} = \begin{bmatrix} y_{T_{s_1}+T_{h_1}+2} & \dots & y_{T_{s_2}+T_{h_2}} \end{bmatrix}^\top$, and $\mathbf{v} = \begin{bmatrix} v_{T_{s_1}+T_{h_1}+2} & \dots & v_{T_{s_2}+T_{h_2}} \end{bmatrix}^\top$.

Deconvolution is an algorithm-based process used to reverse the effects of convolution on observed data. The objective of deconvolution in discrete time is to find a solution to the convolution equation in (2.2) or (2.5) for s (or h) when the observation (z or y) and h (or s) are given. Unlike the deconvoution problem, where we have one of the convolving components as well as observed signal, there are cases when we do not know neither s nor h in (2.2) or (2.5), but we have only the result of the convolution z, or its noisy version y. The problem of estimating the convolving components when only the convolution result is available is referred to as the *blind deconvolution* problem.

2.2. Bayesian Signal Processing

If we have a statistical *prior knowledge*, or belief, about the hidden variables in deconvolution or blind deconvolution problems, how could we make use of it? The answer is to use *Bayesian methods*, where we can include our *prior knowledge* into the solution of our problems.

In Bayesian theory, our knowledge on a variable of interest, say x, given some observed data y, is expressed by their *a posteriori* distribution, given by the Bayes rule below: [24]

$$p(x|y) = \frac{p(x)p(y|x)}{p(y)}, \tag{2.6}$$

where the densities $p(x|y)$, $p(x)$ and $p(y|x)$ are called *a posteriori* probability density function (pdf) (or posterior) , *a priori* pdf (or prior) and the *likelihood* function, respectively. The density $p(y)$ is known as the *evidence* and can be calculated by the integration

$$p(y) = \int_x p(x)p(y|x)dx.$$

In most applications in Bayesian signal processing, the problem is modeling the prior and the likelihood densities, and to calculate the posterior density. Another problem of concern is to find the best parameter set given the observed data, in case of latent variables. That is, we desire to find the parameter set $\hat{\Theta}$ such that

$$\hat{\Theta} = \arg\max_{\Theta} p(y|\Theta). \tag{2.7}$$

We can deal with (2.7) using our prior knowledge about the latent variable x for a given Θ, $p(x|\Theta)$. In that way, we connect Θ to y via x by the following integration:

$$p(y|\Theta) = \int_x p(x|\Theta)p(y|x,\Theta)dx.$$

During the interpretation of *a posteriori* pdf, evidence is usually discarded , since it is not dependent on the parameter of interest x. Therefore, (2.6) is usually written as follows in Bayesian literature [37, 24, 38]

$$p(x|y) \propto p(x)p(y|x). \tag{2.8}$$

As (2.8) implies, our posterior information is formed when our *a priori* belief is shaped by the observed data, i.e. the likelihood function. For some certain prior and posterior densities, the posterior has a closed form expression, hence it is straightforward to calculate (2.8), or the sufficient statistics of x under the posterior distribution. In order to deal with easy posterior densities, a rule of thumb is to choose a reasonable *conjugate prior* for the likelihood density. A class of prior probability distributions $p(x)$ is said to be *conjugate* to a class of likelihood functions $p(y|x)$ if the resulting posterior distributions $p(x|y)$ are in the same family as $p(x)$; the prior and posterior are then called *conjugate distributions*, and the prior is called a *conjugate prior for the likelihood* [24]. Below are two examples of conjugate priors for Gaussian likelihood, which we will use several times in the thesis.

Likelihood is Gaussian, prior is Gaussian: Assume that $\mathbf{x}$, $\mathbf{y}$ are vector valued random variables of sizes $K \times 1$ and $L \times 1$; and

$$p(\mathbf{x}) \sim \mathcal{N}(\mathbf{x}; \mathbf{m}_{\mathbf{x}}, \mathbf{\Sigma}_{\mathbf{x}}),\ p(\mathbf{y}|\mathbf{x}) \sim \mathcal{N}(\mathbf{y}; \mathbf{A}\mathbf{x}, \mathbf{\Sigma}_{\mathbf{y}|\mathbf{x}}).$$

where $\mathbf{m}_{\mathbf{x}}$ and $\mathbf{m}_{\mathbf{y}|\mathbf{x}} \equiv \mathbf{A}\mathbf{x}$ are vector valued means of $\mathbf{x}$ and $\mathbf{y}$ with sizes $K \times 1$ and $L \times 1$, $\mathbf{A}$ is a $L \times K$ matrix, and $\mathbf{\Sigma}_{\mathbf{x}}$ and $\mathbf{\Sigma}_{\mathbf{y}|\mathbf{x}}$ are the covariance matrices of $\mathbf{x}$ and $\mathbf{y}$ with sizes $K \times K$ and $L \times L$. The notation $\mathcal{N}(\mathbf{z}; \mathbf{m}, \mathbf{\Sigma})$ denotes the very well-known multivariate Gaussian distribution for the random variable $\mathbf{z}$ with mean $\mathbf{m}$ and covariance matrix $\mathbf{\Sigma}$ [39],

$$\mathcal{N}(\mathbf{z}; \mathbf{m}, \mathbf{\Sigma}) = \frac{1}{\sqrt{|2\pi\mathbf{\Sigma}|}} \exp\left[-0.5\,(\mathbf{z}-\mathbf{m})^{\top}\,\mathbf{\Sigma}^{-1}\,(\mathbf{z}-\mathbf{m})^{\top}\right].$$

Then, the posterior is another Gaussian

$$p(\mathbf{x}|\mathbf{y}) = \mathcal{N}(\mathbf{x}; \mathbf{m}_{\mathbf{x}|\mathbf{y}}, \mathbf{\Sigma}_{\mathbf{x}|\mathbf{y}})$$

with $\mathbf{\Sigma}_{\mathbf{x}|\mathbf{y}} = \left(\mathbf{\Sigma}_{\mathbf{x}}{}^{-1} + \mathbf{A}^{\top}\mathbf{\Sigma}_{\mathbf{y}|\mathbf{x}}{}^{-1}\mathbf{A}\right)^{-1}$ and $\mathbf{m}_{\mathbf{x}|\mathbf{y}} = \mathbf{\Sigma}_{\mathbf{x}|\mathbf{y}}\left(\mathbf{A}^{\top}\mathbf{\Sigma}_{\mathbf{y}|\mathbf{x}}{}^{-1}\mathbf{y} + \mathbf{\Sigma}_{\mathbf{x}}{}^{-1}\mathbf{m}\right)$.

Likelihood is Gaussian, prior is inverse-gamma: Another conjugate prior for the Gaussian likelihood is the inverse-gamma distribution which is given by [39]

$$\mathcal{IG}(x; \alpha, \beta) = \frac{\beta^{\alpha}}{\Gamma(\alpha)}\left(\frac{1}{x}\right)^{\alpha+1}\exp\left(-\frac{\beta}{x}\right),$$

where α and β are called respectively the *scale* and the *shape* parameters of the inverse-gamma distribution for x, and $\Gamma(\cdot)$ denotes the *gamma function* [39]. Assume the relation for scalar random variables x and y:

$$p(x) \sim \mathcal{IG}(x; \alpha, \beta),\ p(y|x) \sim \mathcal{N}(y; 0, x).$$

It can be shown that the posterior of x is another inverse-gamma distribution

$$p(x|y) \sim \mathcal{IG}(x; \alpha + 0.5, \beta + 0.5y^2).$$

For derivations of these two posteriors, see Appendix A.

In cases where posterior pdf and its integrations cannot be expressed analytically, numerical sampling and variational methods are of utmost importance to perform these tasks. In the following, we will review some of those methods, the ones we use in the thesis to perform deconvolution and blind deconvolution of sparse processes. These are MCMC methods (specifically *Gibbs sampling*), *variational Bayes*, EM algorithm, and *Monte-Carlo related EM*.

2.2.1. MCMC Methods - Gibbs Sampling

MCMC methods are widely used in applications where intractable densities are approximated by a finite number of samples, and Gibbs sampling is one of the most popular type of these methods. To comprehend the theory behind MCMC methods, it is useful to review some related concepts first, such as Monte-Carlo integration, proposal distribution and independently and identically distributed (i.i.d) sampling methods.

2.2.1.1. Monte-Carlo Integration. Numerical sampling methods are used in cases where a pdf and its integrations cannot be expressed analytically. The idea is that, if we can sample random variables from the desired (target) pdf $P(x)$, we can estimate any sufficient statistics of x by numerical optimization methods. If we sample N random variables, $x^{(i)}, i = 1, 2, ..., N$ from $P(x)$, the analytically inexpressible pdf can be represented in terms of the drawn samples as shown below:

$$\hat{P}(x) = \frac{1}{N} \sum_{i=1}^{N} \delta(x - x^{(i)}), \tag{2.9}$$

where $\delta(\cdot)$ is the *delta-dirac* function. Then, using (2.9), the expected value of any function of x under $P(x)$ can be approximated as

$$\langle f(x)\rangle_{P(x)} \approx \frac{1}{N}\sum_{i=1}^{N} f(x^{(i)}) \tag{2.10}$$

where $\langle f(x)\rangle_{P(x)}$ denotes the expectation of $f(x)$ under the distribution of $P(x)$. This method is called *Monte-Carlo integration* in literature [40].

To perform the Monte-Carlo integration in (2.10), we have to be able to sample from the target distribution, which is tedious or not possible in many cases. Also, it is well known that sampling from the target distribution may suffer from computational complexity (MacKay, 1993). In literature, sampling methods like *importance sampling* (IR) and *rejection sampling* (RS) are used to overcome these difficulties [41, 24, 42, 43]. In these methods, a *proposal distribution* $q(x)$ is chosen to sample from. This pdf is chosen so that drawing samples from this distribution is easier than that of the desired one. Such sampling methods are called *independently and identically distributed (i.i.d.) sampling* methods since they always sample from the same distribution, which is the proposal density $q(x)$.

Rejection Sampling: In RS, first a pdf $q(x)$ is proposed in such a way that its product by a constant c, i.e., $cq(x)$ takes higher values than the desired pdf $P(x)$ over the whole support region of random variables. The sampling scheme for all instants is as follows:

- Draw x' from $q(x)$.
- Draw uniformly distributed random variable u over the interval $[0, cq(x')]$.
- If $u < P(x')$, accept x'; otherwise, reject.

Note that RS can be used to sample from $P(x)$, as well as to estimate the expectations of functions of x by using the samples in (2.10).

Importance Sampling: In IS, we do not obtain samples for $P(x)$, we rather use it for estimating expectations of functions of x. The trick is, again, to use a simpler and more efficient distribution $q(x)$ for sampling. For any distribution $q(x)$, we can write

$$\begin{aligned}\langle f(x)\rangle_{P(x)} &= \int_x P(x)f(x)dx \\ &= \frac{\int_x \frac{P(x)}{q(x)} f(x)q(x)dx}{\int_x \frac{P(x)}{q(x)} q(x)dx} \\ &= \frac{\left\langle \frac{P(x)}{q(x)} f(x) \right\rangle_{q(x)}}{\left\langle \frac{P(x)}{q(x)} \right\rangle_{q(x)}}. \end{aligned} \tag{2.11}$$

Hence, if we apply Monte-Carlo integration for the nominator and the denominator of (2.11) by sampling from $q(x)$, we obtain the approximation

$$\langle f(x)\rangle_{P(x)} \approx \frac{1}{\sum_{i=1}^{N} w^{(i)}} \sum_{i=1}^{N} f(x^{(i)})w^{(i)}$$

where $w^{(i)} = P(x^{(i)})/q(x^{(i)})$'s are called the *importance weights*.

2.2.1.2. MCMC Methods. It is well known in literature that i.i.d. sampling methods, like RS and IS, suffer from the computational complexity and high estimation variance problems. Furthermore, in RS, the proposal pdf $q(x)$ should look like the desired pdf $P(x)$ throughout the whole support region of x, which is very difficult and becomes even harder when the dimension size increases [44, 24, 45, 43, 41]. To overcome these problems, instead of choosing such a whole distribution to fit, we can construct a Markov chain whose samples are asymptotically distributed according to the desired pdf $P(x)$. This is known as the MCMC technique [45, 41].

Let the probability distribution of the state is represented by $p^{(t)}(x)$ at iteration t. The objective is to find a Markov chain such that as $t \to \infty$, $p^{(t)}(x)$ approaches to the desired distribution $P(x)$. In order to define a Markov chain, an initial probability distribution $p^{(0)}(x)$ and a *transition probability* $T(x'; x)$ need to be specified. The

probability distribution of the state at the $(t+1)^{\text{th}}$ iteration of the Markov chain is given by the following equation:

$$p^{(t+1)}(x') = \int_x p^{(t)}(x) T(x'; x) dx.$$

When designing a MCMC method, the chain should satisfy certain conditions: The desired distribution must be an *invariant* distribution of the chain, and the chain must be *ergodic*. For *ergodicity*, the chain must be *irreducible* and *aperiodic* [41].

Many useful transition probabilities have the *detailed balance* property [41]:

$$T(x_a; x_b) P(x_b) = T(x_b; x_a) P(x_a) \text{ for all } x_a \text{ and } x_b.$$

Detailed balance is an important property because it implies invariance of $P(x)$ under the chain, which is one of the conditions to be satisfied.

Metropolis-Hasting Method: We have stated that it is very difficult to construct a proposal distribution $q(x)$ which should resemble the target density $P(x)$ in the whole space. This necessity can be relaxed by a Markov chain to propose new states based on the current ones. These new states, $x^{(t+1)}$ are drawn from simpler proposal distributions $q(x'; x^{(t)})$, which are located near $x^{(t)}$. The procedure for drawing new samples in *Metropolis-Hasting* method is in the following:

- Draw x' form $p(x'; x^{(t)})$.
- Calculate the acceptance ratio

$$\epsilon = \frac{P(x') q(x^{(t)}; x')}{P(x^{(t)}) q(x'; x^{(t)})}. \tag{2.12}$$

- If $\epsilon \geq 1$, $x^{(t+1)} = x'$; otherwise:
  - Draw uniformly distributed random variable u over the interval $[0, 1]$.
  - If $u < \epsilon$, $x^{(t+1)} = x'$, otherwise $x^{(t+1)} = x^{(t)}$.

This method, first proposed by Metropolis *et al* [44]. in 1953, is known as the *Metropolis-Hastings method.* If $q(x;x') = q(x';x)$, then the method is called as *Metropolis method* [41].

Gibbs Sampling: Gibbs sampling is an MCMC sampling method which can be used when the dimension of x, say n, is at least two. The idea of Gibbs sampling is that, even though it is difficult to draw samples from $P(x) = p(x_1, x_2, \ldots, x_n)$, it may be easier to do from the conditional densities $p(x_i|x_1, \ldots x_{i-1}, x_{i+1}, \ldots, x_n)$. The general sampling scheme of Gibbs sampling for $P(x)$ can be expressed by the following [45, 41]:

$$\begin{aligned} x_1^{(t+1)} &\sim p(x_1|x_2^{(t)}, x_3^{(t)}, \ldots, x_n^{(t)}) \\ x_2^{(t+1)} &\sim p(x_2|x_1^{(t+1)}, x_3^{t}, \ldots, x_n^{(t)}) \\ &\vdots \\ x_n^{(t+1)} &\sim p(x_n|x_1^{(t+1)}, x_2^{(t+1)}, \ldots, x_{n-1}^{(t+1)}). \end{aligned}$$

It has been shown that, under mild conditions, when $t \to \infty$, the samples are guaranteed to come from the exact distribution $P(x)$. The algorithm is named after the physicist J. W. Gibbs, in reference to an analogy between the sampling algorithm and statistical physics. The algorithm was devised by Stuart Geman and Donald Geman in 1984 [46], about eight decades after the passing of Gibbs.

Gibbs sampling is a special kind of Metropolis-Hasting method. At time t and iteration i, we have $x^{(t,i)} = (x_{1:i-1}^{(t+1)}, x_i^{(t)}, x_{i+1:n}^{(t)})$. So,

$$\begin{aligned} p(x^{(t,i)})q(x'|x^{(t,i)}) &= p(x^{(t,i)})\delta(x'_{1:i-1,i+1:n} - x^{(t,i)}_{1:i-1,i+1})p(x'_i|x^{(t,i)}_{1:i-1,i+1:n}) \\ &= p(x^{(t,i)}, x'_i)\delta(x'_{1:i-1,i+1:n} - x^{(t,i)}_{1:i-1,i+1:n}) \\ &= \delta(x^{(t,i)}_{1:i-1,i+1:n} - x'_{1:i-1,i+1:n})p(x^{(t,i)}, x'_i) \\ &= \delta(x^{(t,i)}_{1:i-1,i+1:n} - x'_{1:i-1,i+1:n})p(x_i^t, x') \\ &= p(x')q(x^{(t,i)}|x') \end{aligned} \tag{2.13}$$

Substituting (2.13) into (2.12), we find out that the acceptance ratio is always equal to 1. Therefore, we can say that Gibbs sampling is a Metropolis-Hasting method with the acceptance ratio always being equal to 1.

2.2.2. Variational Bayes

As an alternative for Monte-Carlo methods, *variational Bayesian methods*, also called ensemble learning, are another family of techniques for approximating intractable integrals arising in Bayesian statistics [47, 41]. These methods often provide an analytical approximation to the posterior probability which is useful for prediction. In variational inference, a posterior distribution of x given the data y, $p(x|y)$, is approximated by a *variational distribution*, $Q(x)$. The approximation seeks for minimization of dissimilarity function between $p(x|y)$, and $Q(x)$. One useful dissimilarity function where this minimization is tractable is the Kullback-Leibler (KL) divergence, which is given below [48]:

$$KL(Q||p) = \int_x Q(x) \log \frac{Q(x)}{p(x|y)} dx \tag{2.14}$$

We now observe that any $Q(x)$ gives rise to a lower bound to the log-evidence of the data:

$$\begin{aligned}
\log p(y) &= \log \int_x dx p(y,x) \\
&= \log \int_x dx Q(x) \frac{p(y,x)}{Q(x)} \\
&\geq \int_x dx Q(x) \log \frac{p(y,x)}{Q(x)} \\
&= \int_x dx Q(x) \log p(y,x) - \int_x dx Q(x) \log Q(x) \\
&= \langle \log p(y,x) \rangle_{Q(x)} + H(Q) \\
&\equiv \mathcal{VB}_Q
\end{aligned} \tag{2.15}$$

where $H(Q)$ is the differential entropy of $Q(x)$, and the inequality is a direct result of Jensen's inequality and concavity of the $\log(\cdot)$ function [48]. The lower bound $\mathcal{VB}_Q$ in

(2.15) is called the *variational bound*.

On the other hand, we have

$$\begin{aligned}
\log p(y) &= \int_x dx Q(x) \log p(y) \\
&= \int_x dx Q(x) \log \frac{p(y,x)}{p(x|y)} \\
&= \int_x dx \left(Q(x) \log Q(x) - Q(x) \log p(x|y)\right) \\
&- \int_x dx \left(Q(x) \log Q(x) - Q(x) \log p(x,y)\right) \\
&= \int_x dx Q(x) \log \frac{Q(x)}{p(x|y)} + \langle \log p(y,x) \rangle_{Q(x)} + H(Q) \\
&= KL(Q||p) + \mathcal{VB}_Q. \qquad (2.16)
\end{aligned}$$

Since $\log p(y)$ is fixed with respect to $Q(x)$, *maximizing the variational bound* $\mathcal{VB}_Q$ *minimizes the KL divergence* in (2.14), which is our objective.

It can be easily shown by substituting $Q(x) = p(x|y)$ in (2.15) and observing the equality that, the log-evidence $\log p(y)$ is maximized by the exact posterior, i.e., when $Q(x) = p(x|y)$. However, since $p(x|y)$ is not in a friendly form, we try to find the best fitting $Q(x)$ among a limited group of distributions, and the approximation is done by maximizing $\mathcal{VB}_Q$ over this limited group of distributions.

Mean Field Approximation of the Posterior: The *mean field* approximation is the case in which each $Q(x)$ is fully factorized over the hidden variables $x_1, x_2, \ldots, x_n$:

$$Q(x) = \prod_{i=1}^{n} q(x_i). \qquad (2.17)$$

Maximization of the variational bound $\mathcal{VB}_Q$ over the factorized distribution in (2.17) gives for each $q(x_i)$ the following equation [47]:

$$q(x_i) = \frac{1}{\mathcal{Z}_i} \exp \langle \log p(y,x) \rangle_{\prod_{j=1, j\neq i}^{n} q(x_j)}, \text{ for } i = 1, \ldots, n \qquad (2.18)$$

where $\mathcal{Z}_i$ is the normalizing constant. Notice that we do not need to know $\mathcal{Z}_i$ in most cases, since the proportionality already gives the form of the distribution. Using (2.18), a line maximization can be performed as follows:

$$\begin{aligned} q(x_1)^{(t+1)} &\propto \exp \langle \log p(y,x) \rangle_{\prod_{j=2}^{n} q(x_j)^{(t)}} \\ q(x_2)^{(t+1)} &\propto \exp \langle \log p(y,x) \rangle_{q(x_1)^{(t+1)} \prod_{j=3}^{n} q(x_j)^{(t)}} \\ &\vdots \\ q(x_n)^{(t+1)} &\propto \exp \langle \log p(y,x) \rangle_{\prod_{j=1}^{n-1} q(x_j)^{(t+1)}} \end{aligned}$$

2.2.3. EM Algorithm for Parameter Estimation

While dealing with deconvolution or blind deconvolution problems, there may be some deterministic parameters as well as statistical hidden variables. For statistical parameters, we have seen methods for calculating their posteriors. For estimating deterministic parameters, we have the EM algorithm and its variants. The EM algorithm was explained and given its name in a classic 1977 paper by Arthur Dempster, Nan Laird, and Donald Rubin [49].

We seek for maximum-likelihood estimation (MLE) for a parameter set Θ:

$$\hat{\Theta} = \arg\max_{\Theta} \log p(y;\Theta).$$

EM algorithm applies two iterative steps to perform the maximization in (2.2.3):

- Expectation (E) step: Calculate the *EM quantity* $\Omega^{(i)}(\Theta)$, which is the expectation of log-joint distribution of x and y under the posterior distribution of x given y and the previous parameter set $\Theta^{(i)}$:

$$\Omega^{(i)}(\Theta) = \langle \log p(x,y|\Theta) \rangle_{p(x|y,\Theta^{(i)})} \tag{2.19}$$

- Maximization (M) step: Find the parameter which maximizes this quantity:

$$\Theta^{(i+1)} = \arg\max_{\Theta} \Omega^{(i)}(\Theta) \tag{2.20}$$

It can be shown that $\log p(y|\Theta^{(i)})$ monotonically increases [50]:

$$\begin{aligned}
\log p(y|\Theta^{(i+1)}) &= \int_x dx \log p(y|\Theta^{(i+1)}) p(x|y,\Theta^{(i)}) \\
&= \int_x dx \log p(y,x|\Theta^{(i+1)}) p(x|y,\Theta^{(i)}) - \int_x dx \log p(x|y,\Theta^{(i+1)}) p(x|y,\Theta^{(i)}) \\
&= \Omega^{(i)}(\Theta^{(i+1)}) - \int_x dx \log p(x|y,\Theta^{(i+1)}) p(x|y,\Theta^{(i)})
\end{aligned} \tag{2.21}$$

Therefore, using (2.14), (2.19), and (2.21), we have

$$\begin{aligned}
&\log p(y|\Theta^{(i+1)}) - \log p(y|\Theta^{(i)}) \\
&\qquad = \Omega^{(i)}(\Theta^{(i+1)}) - \Omega^{(i)}(\Theta^{(i)}) + KL(p(x|y,\Theta^{(i)})||p(x|y,\Theta^{(i)})) \\
&\qquad \geq 0
\end{aligned} \tag{2.22}$$

since, by (2.20), $\Omega^{(i)}(\Theta^{(i+1)}) = \max_{\Theta} \Omega^{(i)}(\Theta) \geq \Omega^{(i)}(\Theta^{(i)})$, and the Kullback-Leibler distance is non-negative, which is obvious from (2.15) and (2.16). Also, notice that we have equality in (2.22) when $\Theta^{(i+1)} = \Theta^{(i)}$, hence, the algorithm has converged to a local maximum point.

2.2.3.1. Monte-Carlo related EM Methods. Recall that the E-step of the EM algorithm amounts to evaluating the function $\langle \log p(x,y|\Theta) \rangle_{p(x|y,\Theta^{(i)})}$. We now consider cases where direct numerical evaluation of this expectation is not available. The principle here is to use the Monte-Carlo sampling methods to approximate the intractable E-step with the empirical averages based on the simulated data [HMM book]. In this line of approach, we will state three different Monte-Carlo related EM methods. These methods are *Monte-Carlo EM*, (MCEM) [51], *Stochastic Approximation EM* (SAEM) [52], and *stochastic EM* (SEM), [53]. The E and M steps of all these methods can be generalized as the following [50]:

- Start with the initial $\hat{\Omega}^{(0)}$.
- for $i = 1, 2, \ldots$
  - *Simulation:* Draw N_i samples for x; $(x^{(i,1)}, ..., x^{(i,N_i)}$ from $p(x|y; \Theta^{(i-1)})$ using a Monte-Carlo method.
  - *Maximization:* Compute $\Theta^{(i)}$ that maximizes $\hat{\Omega}^{(i)}$, where

$$\hat{\Omega}^{(i)} = \nu_i \frac{1}{N_i} \sum_{j=1}^{N_i} \log p(y, x^{(i,j)}|\Theta^{(i-1)}) + (1 - \nu_i)\hat{\Omega}^{(i-1)}.$$

  where ν_i is a weight parameter. We have:

- SAEM: $\{\nu_i\}_{i \geq 1} > 0$, $\nu_1 \neq 1$, and $\nu_i \leq \nu_{i+1} \leq 1$.
- MCEM: $\nu_i = 1$ and N_i increases with time.
- SEM: $\nu_i = 1$ and N_i is constant.

Note that, the approximation for any sufficient statistics of x, $S_f^{(i)} = \langle f(x) \rangle_{p(x|\mathbf{y};\Theta^{(i-1)})}$ is

$$\hat{S}_f^{(i)} = (1 - \nu_i)\hat{S}_f^{(i-1)} + \nu_i \frac{1}{N_i} \sum_{j=1}^{N_i} f(x^{(i,j)}).$$

3. DECONVOLUTION AND BLIND DECONVOLUTION OF SPARSE PROCESSES

In this chapter, we will investigate Bayesian methods for deconvolution of sparse processes in case of unknown parameters. The common approach in these methods is to model sparse waveforms in a statistical way, and treat other unknowns of the convolution system as non-statistical parameters. This approach leads us to iterative maximum likelihood inference methods for parameter estimation, where statistical inference methods must be performed in every iteration.

While considering our problem, we assume that an input signal, which we call *source*, passes through an unknown FIR sparse filter, and the resulting convolution is observed in additive Gaussian noise. Here, the source, the sparse filter, and the noise correspond to the convolving components and the noise component in (2.5). The unknown parameter set may or may not include the source coefficients. In fact, when the source is unknown, our problem becomes a *blind deconvolution* problem. However, there is no methodological difference in treating the deconvolution and blind deconvolution problems, the only difference is that in blind deconvolution, the source coefficients are in the set of unknown parameters. Therefore, we will introduce methods that are applicable to the most general version of our problem, i.e., when both the source coefficients and the sparse filter are unknown.

The approaches to solve our problem differ in the way they model the unknown statistical variables, or the statistical inference/optimization methods they use. We will introduce two models and several methods in order to solve the blind deconvolution problem. The two models are the *Bernoulli-Gaussian* and the *inverse-gamma* models, which can produce quantitatively similar sparse processes. In the Bernoulli-Gaussian model, when both the source and the filter are jointly estimated, the true posterior is typically highly multimodal. Therefore, when not properly initialized, standard stochastic inference methods tend to get stuck and suffer from poor convergence.

In the inverse-gamma model, the Bernoulli-Gaussian prior is relaxed, and it will be shown with empirical evidence that inference in the relaxed model is not only efficient, but also provides better results compared to the Bernoulli-Gaussian model. We will also provide simulation studies that compare the results obtained with and without our initialization method for several combinations of state inference and parameter estimation methods used for the Bernoulli-Gaussian model. At the end of the chapter, we will use the inverse-gamma model for deconvolution of seismological signals and compare its performance with the iterative deconvolution method.

3.1. Noisy Convolution Model and Problem Statement

In a discrete time noisy convolution model, we consider the finite impulse response (FIR) filtering process where a latent input signal is passed through a LTI system with unknown sparse impulse response and the resulting signal is corrupted by additive Gaussian noise, giving the observed signal. This process can be formulated as follows:

$$y_t = \sum_{k=1}^{T_s} s_k h_{t-k+1} + v_t, \qquad t = 1, 2, ..., T \tag{3.1}$$

In (3.1), the input signal is represented by a $1 \times T_s$ vector of coefficients, $\mathbf{s} \equiv \begin{bmatrix} s_1 & s_2 & \dots & s_{T_s} \end{bmatrix}^\top$, and $\mathbf{h} \equiv \begin{bmatrix} h_1 & h_2 & \dots & h_T \end{bmatrix}^\top$ denotes the vector of filter coefficients of length T. $\mathbf{v} \equiv \begin{bmatrix} v_1 & v_2 & \dots & v_T \end{bmatrix}^\top$ is the observation noise vector with v_k being i.i.d Gaussian with zero mean and variance σ_v^2, that is, $v_k \sim \mathcal{N}(v_k; 0, \sigma_v^2)$. Finally, $\mathbf{y} \equiv \begin{bmatrix} y_1 & y_2 & \dots & y_T \end{bmatrix}^\top$ is the vector of noisy observation coefficients. Note that, we assume $T_h = T_y = T$. The reason for this assumption is its simplicity in expressing the developed methods in this chapter, which will be understood while we go into these methods.

The key point here is the choice of the prior distribution of $\mathbf{h}$, so that it is a sparse process. In the following, we introduce two simple choices, that are qualitatively similar but have significantly different behaviors for inference.

3.1.1. Bernoulli-Gaussian Model

It is a common approach in the literature to model the coefficients of a sparse waveform as i.i.d. Bernoulli-Gaussian random variables[3, 26]. According to this model, we assume that the samples h_k are i.i.d. Gaussian with their variances conditioned on an *indicator* (Bernoulli random variable) c_k along with success probability λ. That is,

$$h_k \sim \mathcal{N}(h_k; 0, \sigma^2_{c_k}), \quad c_k \in \{0, 1\}, \quad Pr(c_k = 1) = \lambda,$$

where $Pr(A)$ denotes the probability of the event A. Taking $\sigma_0^2 \ll \sigma_1^2$ and assigning a small value for λ, one can obtain a sparse and impulsive trace (see Figure 3.1). Here, the indicators c_k can be used to indicate the locations of spikes. In many applications, such as seismology [3], correct estimation of the spike locations is of primary concern, and this parametrization provides an intuitive semantic interpretation.

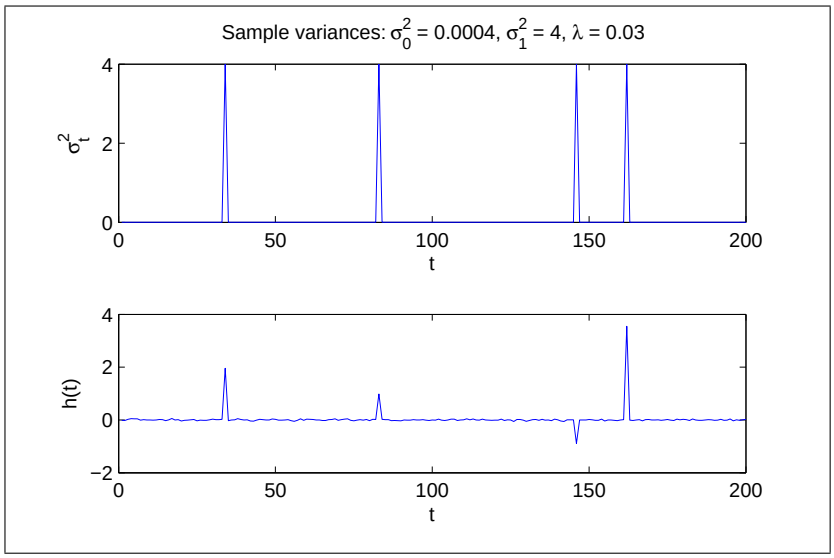


Figure 3.1. Sparse trace obtained using the Bernoulli-Gaussian model: $\sigma_0^2 = 0.0004, \sigma_1^2 = 4, \lambda = 0.03$

.

3.1.2. Inverse-Gamma Model

An alternative to the Bernoulli-Gaussian model is to model the samples of the sparse filter as a Gaussian with zero mean and the variance being an inverse-gamma random variable. That is,

$$h_k \sim \mathcal{N}(h_k; 0, \sigma_{h,k}^2) \quad \sigma_{h,k}^2 \sim \mathcal{IG}(\sigma_k^2; \alpha, \beta).$$

It is important to note that, in this model the samples h_k are *Student-t* random variables [31]. However, we call this model as the inverse-gamma model since we believe it is more expressive to emphasize the choice for the variances, rather than the resulting distribution for h_k's.

Choosing suitable values for α and β (especially, when $\beta/\alpha \ll 1$), one can obtain sparse and impulsive traces (see Figure 3.2). If we compare the two figures, we can see that the traces produced by these processes are quantitatively similar.

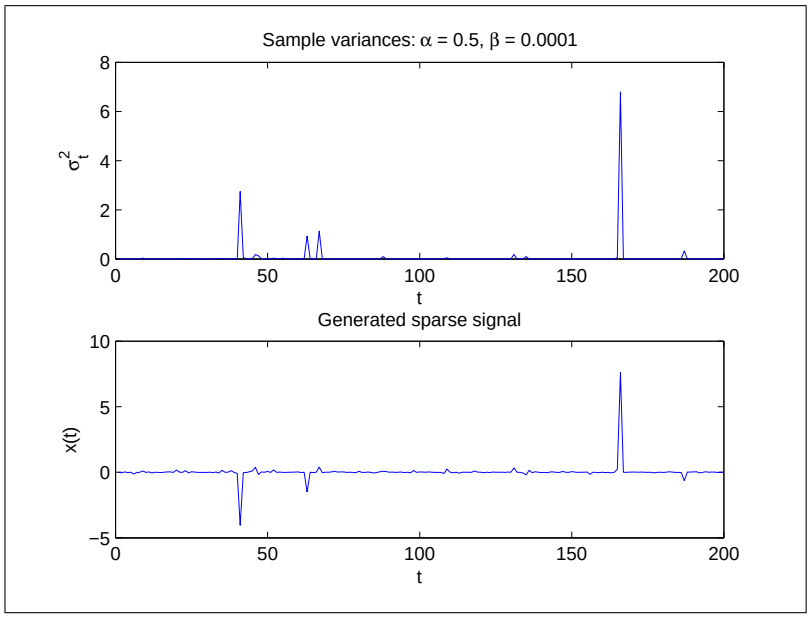


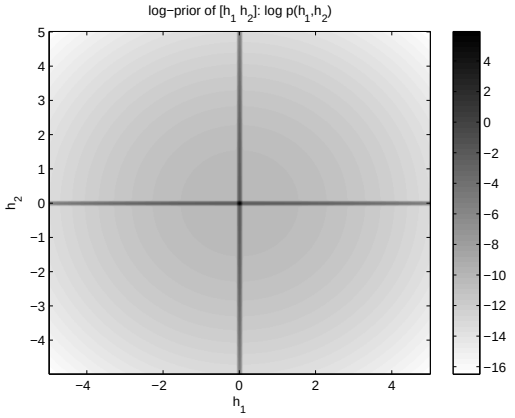
Figure 3.2. Sparse trace obtained using the inverse-gamma model: $\alpha = 0.5, \beta = 0.0001$

.

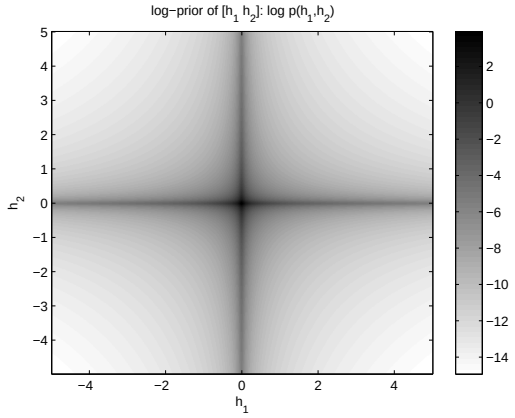
Inverse-gamma model is a relaxation of the Bernoulli-Gaussian model: The reason for choosing an inverse-gamma distribution for σ_k^2 is its analytical convenience leading to a relatively easier objective function than the one for the Bernoulli-Gaussian model [28, 31]. This lowers the effect of initialization, and thus helps preventing the inference and parameter estimation methods from converging to local optimum points.

This relaxing effect of the inverse-gamma model is illustrated in Figure 3.3. The top figure at the left hand side shows the log-prior distribution of $[h_1, h_2]$, if $p(h_i)$ is a Bernoulli-Gaussian random variable with parameters $\sigma_0^2 = 0.0004$, $\sigma_1^2 = 4$, and $\lambda = 0.03$, whereas the one on the right hand side is the log-prior of $[h_1, h_2]$ when they are normal random variables with their variances coming from the inverse-gamma distribution with parameters $\alpha = 0.5$, and $\beta = 0.0001$. The figures in the middle row show the log-likelihood of the data, when the system admits $y = 2h_1 + h_2 + \sigma_v v$, with $y = 3$ and $\sigma_v^2 = 0.05$. Finally, the bottom figures on the left and right hand sides show the log-posteriors of $[h_1, h_2]$ given y, for the Bernoulli-Gaussian and the inverse-gamma models, respectively. Note that the log-posteriors are illustrated as superimposed on the log-prior distributions, for the sake of understandability. It can be seen from the posteriors that, the posterior in the inverse-gamma model is much more smoother than the one in the Bernoulli-Gaussian model. Roughly speaking, if we start searching for $[h_1, h_2]$ those maximize the posterior from a point near to $(0, 3)$, it is much more possible to converge to that local maximum in the Bernoulli-Gaussian model.

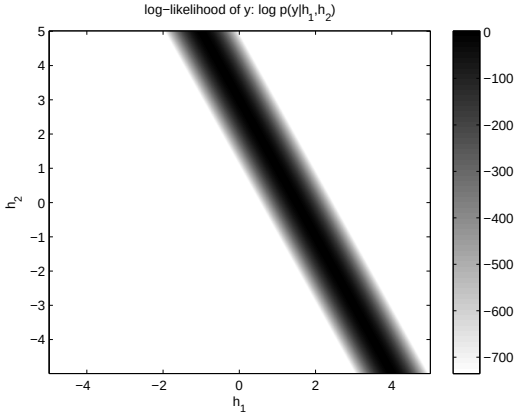
Problem Definition:. Having introduced the models, our problem can be defined: Given the observation data $\mathbf{y}$, we wish to estimate the unknown filter $\mathbf{h}$, as well as the source $\mathbf{s}$, and possibly the model parameters. For the Bernoulli-Gaussian model, the model parameters are $(\sigma_v^2, \sigma_0^2, \sigma_1^2, \lambda)$ and for the inverse-gamma model, they are $(\sigma_v^2, \alpha, \beta)$.



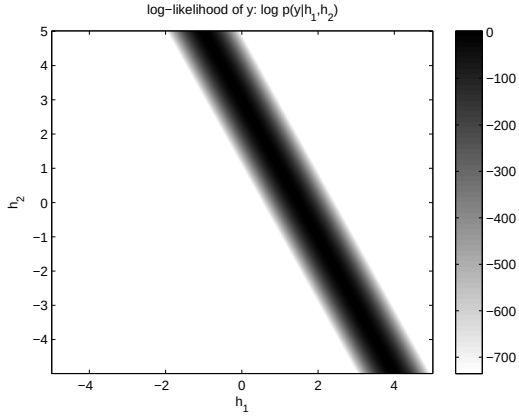
(a) Bernoulli-Gaussian log-prior



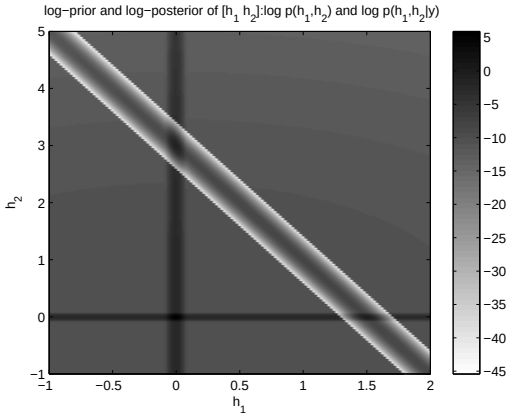
(b) Inverse-gamma prior



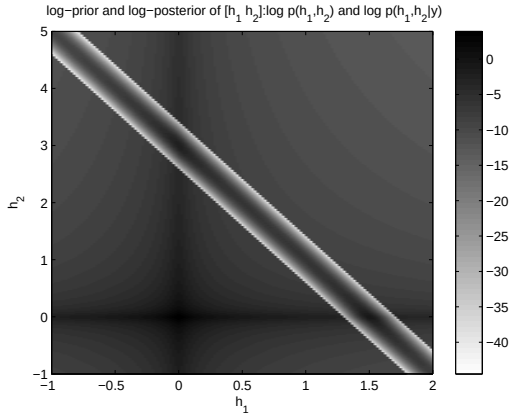
(c) Bernoulli-Gaussian log-likelihood



(d) Inverse-gamma log-likelihood



(e) Bernoulli-Gaussian log-posterior



(f) Inverse-gamma log-posterior

Figure 3.3. An illustration of the relaxation effect of the inverse-gamma model: log-prior, log-likelihood, and log-posterior (superimposed on the log-prior) are shown for both models with respect to $y = 2h_1 + h_2 + \sigma_v v$. ($y = 3$, $\sigma_0^2 = 0.0004$, $\sigma_1^2 = 4$, $\alpha = 0.5$, $\beta = 0.0001$, and $\sigma_v^2 = 0.05$.)

3.2. Methodology

To solve our inference and parameter estimation problem, we use the EM-type algorithms. As it was explained in Chapter 2, the algorithms contain two steps: Calculating the posterior of the statistical latent variables given the current set of parameters, i.e, *state inference*, and update of the parameters under the calculated posterior distribution, which we call *parameter estimation*. Performing these steps, we finally find *the posterior of the latent variables for the maximum likely parameter set*. In this part, we will investigate state inference and parameter estimation methods for both models.

3.2.1. Bernoulli-Gaussian Model

For the Bernoulli-Gaussian model, we have:

- Latent statistical variables: $\mathbf{h}$ and $\mathbf{c} \equiv c_{1:n}$,
- Deterministic parameters: $\Theta = (\mathbf{s}, \sigma_v^2, \sigma_1^2, \sigma_0^2, \lambda)$.

Now the EM steps are:

- E step: Requires sufficient statistics, i.e. inference of the posterior

$$\Omega^{(i)} = \langle \log p(\mathbf{y}, \mathbf{h}, \mathbf{c}|\Theta) \rangle_{p(\mathbf{h},\mathbf{c}|\mathbf{y},\Theta^{(i-1)})}. \tag{3.2}$$

- M step: Parameter estimation: Maximize $\Omega^{(i)}$ over Θ:

$$\Theta^{(i)} = \arg\max_{\Theta} \Omega^{(i)}. \tag{3.3}$$

3.2.1.1. State Inference for the Bernoulli-Gaussian Model. We have to calculate the posterior in (3.2), i.e. $p(\mathbf{h}, \mathbf{c}|\mathbf{y}, \Theta)$. The joint posterior distribution of $\mathbf{h}$ and $\mathbf{c}$ is

$$p(\mathbf{h}, \mathbf{c}|\mathbf{y}, \Theta) = p(\mathbf{c}|\mathbf{y}, \Theta) p(\mathbf{h}|\mathbf{c}, \mathbf{y}, \Theta). \tag{3.4}$$

The factors on the right hand side of (3.4) can be evaluated separately: The first factor can be written as

$$p(\mathbf{c}|\mathbf{y},\Theta) \propto p(\mathbf{c})p(\mathbf{y}|\mathbf{c},\Theta). \tag{3.5}$$

The first factor in (3.5) is easy to calculate: $p(\mathbf{c}) = \lambda^{n_1^{\mathbf{c}}}(1-\lambda)^{n_0^{\mathbf{c}}}$ with $n_1^{\mathbf{c}} = \sum_{t=1}^{T} c_t$, and $n_0^{\mathbf{c}} = T - n_1^{\mathbf{c}}$. The conditional density $p(\mathbf{y}|\mathbf{c},\Theta)$, which is the *likelihood* of the data, can easily be found as in the following:

$$p(\mathbf{h}|\mathbf{c},\Theta) = \mathcal{N}(\mathbf{h};\mathbf{0},\mathbf{C}),\ p(\mathbf{y}|\mathbf{h},\Theta) = \mathcal{N}(\mathbf{y};\mathbf{Sh},\sigma_v^2\mathbf{I}_T),$$

where $\mathbf{C}$ is the diagonal covariance matrix of size $T \times T$ with $\mathbf{C}_{ii} = \sigma_{c_i}^2$, and $\mathbf{I}_T$ is the identity matrix of size $T \times T$. Thus,

$$\begin{aligned} p(\mathbf{y}|\mathbf{c},\Theta) &= \int_h p(\mathbf{h}|\mathbf{c},\Theta)p(\mathbf{y}|\mathbf{h},\Theta)d\mathbf{h} = \langle p(\mathbf{y}|\mathbf{h},\Theta)\rangle_{p(\mathbf{h}|\mathbf{c},\Theta)} \\ &= \mathcal{N}(\mathbf{y};\mathbf{0},\mathbf{SCS}^\top + \sigma_v^2\mathbf{I}_T), \end{aligned} \tag{3.6}$$

where $\mathbf{S}$ is the $T \times T$ convolution matrix corresponding to $\mathbf{s}$ such that

$$\mathbf{S}_{ij} = \begin{cases} s_{i-j+1} & \text{if } 0 \le i-j \\ 0 & \text{else.} \end{cases} \tag{3.7}$$

The conditional density $p(\mathbf{y}|\mathbf{c},\Theta)$ can also be found by running a Kalman filter [11]. First, we shall write down our convolution system for the Bernoulli-Gaussian model as a *conditionally Gaussian linear state-space model* (CGLSSM) [50] as in the following:

$$\mathbf{h}_{k+1} = \mathbf{A}\mathbf{h}_k + \mathbf{U}_{c_{k+1}}e_{k+1} \tag{3.8}$$

$$y_k = \mathbf{s}^\top\mathbf{h}_k + v_k \tag{3.9}$$

where $e_{k+1} \sim \mathcal{N}(e_{k+1}, 0, 1)$, $\mathbf{h}_k = \begin{bmatrix} h_k & h_{k-1} & \dots & h_{k-T_s+1} \end{bmatrix}^\top$, $\mathbf{A}$ is the one-tap delay matrix of size $T_s \times T_s$, and $\mathbf{U}_{c_k} = \begin{bmatrix} \sigma_{c_k} & 0 & \dots & 0 \end{bmatrix}^\top$. Note that when $\mathbf{c}$ and Θ are given, all of the state-space parameters in (3.8) and (3.9) are known. Then, this problem can be solved by a Kalman filter. For the Kalman filter solution for this CGLSSM, see Appendix B.1.

To compute $p(\mathbf{y}|\mathbf{c}, \Theta)$ using (3.6), we have to compute the inverse of a $T \times T$ matrix at once. However, if we run Kalman filter, we deal T times with the inverses of $T_s \times T_s$ matrices. Therefore, it is advantageous to use the Kalman filter when T_s is small, but the batch method should be preferred when T_s is close to T.

Evaluation of the second factor in (3.4) is straightforward as well. Observe that

$$\begin{aligned} p(\mathbf{h}|\mathbf{y}, \mathbf{c}, \Theta) &\propto p(\mathbf{h}|\mathbf{c}, \Theta) p(\mathbf{y}|\mathbf{h}, \mathbf{c}, \Theta) \\ &= \mathcal{N}(\mathbf{h}; \mathbf{0}, \mathbf{C}) \mathcal{N}(\mathbf{y}; \mathbf{S}\mathbf{h}, \sigma_v^2 \mathbf{I}_T) \end{aligned}$$

Recall from the previous chapter that, when the prior and the likelihood densities are Gaussian, the posterior is another Gaussian, and it can be shown that

$$p(\mathbf{h}|\mathbf{y}, \mathbf{c}, \Theta) = \mathcal{N}(\mathbf{h}; \mathbf{m}_\mathbf{h}, \mathbf{\Sigma}_\mathbf{h}), \ \mathbf{\Sigma}_\mathbf{h} = \left(\mathbf{C}^{-1} + \frac{1}{\sigma_v^2} \mathbf{S}^\top \mathbf{S} \right)^{-1} \text{ and } \mathbf{m}_\mathbf{h} = \mathbf{\Sigma}_\mathbf{h} \mathbf{S}^\top \mathbf{y}.$$

An alternative way to calculate the posterior density $p(\mathbf{h}|\mathbf{y}, \mathbf{c}, \Theta)$ is to make use of the CGLSSM. Given $\mathbf{y}$ with all of the state-space parameters, namely Θ and $\mathbf{c}$, each $\mathbf{h}_k$ is Gaussian with mean $\hat{\mathbf{h}}_{k|T}$ and covariance $\hat{\mathbf{\Sigma}}_{k|T}$, and these moments can be evaluated by using one of the smoothing algorithms for Gaussian linear state-space systems, such as *disturbance smoothing* [54], *Rauch-Tung-Striebel smoothing* [55], and *forward-backward smoothing using two filter formula* [56]. Disturbance smoothing algorithm has a special feature, such that, referring to the CGLSSM defined in (3.8) and (3.9), it operates only on the disturbance vectors w_k rather than on the states $\mathbf{h}_k$. This form of smoothing is more efficient when the dimension of the state noise vector is smaller than

that of the state vector. This is the case in our model, where w_k is a scalar whereas $\mathbf{h}_k$ is an $L \times 1$ vector. Therefore, we shall use disturbance smoothing algorithm to make exact inference for the states. For the disturbance smoothing algorithm modified for our problem, see Appendix B.2.

The argument we have made for calculation of the likelihood is also true for choosing either the batch method or disturbance smoothing algorithm to calculate the posterior density of $\mathbf{h}$: It is advantageous to use the disturbance smoothing when T_s is small, on the other hand the batch method should be used when T_s is close to T.

$p(\mathbf{h}, \mathbf{c}|\mathbf{y}, \Theta)$ *is intractable:* We have shown that, for a given configuration of $\mathbf{c}$, we can evaluate $p(\mathbf{h}, \mathbf{c}|\mathbf{y}, \Theta)$ However, the main difficulty is due to the fact that there are 2^T distinct configurations of $\mathbf{c}$. Hence, we can not practically evaluate (3.4) for every configuration of $\mathbf{c}$, unless T is very small. Therefore the exact joint posterior $p(\mathbf{h}, \mathbf{c}|\mathbf{y}, \Theta)$ is intractable. To overcome this problem, we introduce two different MCMC sampling methods to draw samples from the joint posterior distribution.

MCMC 1: *Sample indicators* $\mathbf{c}$*, perform exact inference for* $\mathbf{h}$*:*

In this type of MCMC sampling, we first estimate the best sequence of indicators $\hat{\mathbf{c}}$ using Gibbs sampling, then use this indicator sequence for exact inference of $\mathbf{h}$. That is, if we use the notation $\mathbf{c}_{-k}^{(i)} = \left(\mathbf{c}_{1:k-1}^{(i)}, \quad \mathbf{c}_{k+1:T}^{(i-1)}\right)$, the procedure is:

- Start with the initial indicator values $\mathbf{c}^{(0)}$.
- for $i = 1, 2, \ldots, N$
  - for $k = 1, 2, \ldots, T$
    Draw $c_k^{(i)}$ from $p(c_k^{(i)}|\mathbf{c}_{-k}^{(i)}, \mathbf{y}, \Theta)$.
  - Evaluate the posterior $p_h^{(i)} = p(\mathbf{h}|\mathbf{y}, \mathbf{c}^{(i)}, \Theta)$ using a smoothing method.
  - Keep $\mathbf{c}^{(i)}$ and $p_h^{(i)}$ as the i^{th} *sample* set.

Note that we treat the densities $p_h^{(i)}$ as samples since they depend on the samples of indicators, $\mathbf{c}^{(i)}$. In fact, in this MCMC method, we draw samples for $\mathbf{c}$, and *sufficient*

statistics for $\mathbf{h}$. This type of sampling is also referred to as *Rao-Blackwellized Gibbs sampling* [57].

There are, again, two possible ways to sample from $p(c_k^{(i)}|\mathbf{c}_{-k}^{(i)}, \mathbf{y}, \Theta)$. The first way is to make use of (3.6). Denoting the covariance matrix corresponding to the current indicator sequence as $\mathbf{C}_k^{(i)} = \left(\mathbf{c}_{1:k-1}^{(i)}, \quad c_k^{(i)}, \quad \mathbf{c}_{k+1:T}^{(i-1)}\right)$, and using (3.6), then we can write

$$\begin{aligned} p(c_k^{(i)}|\mathbf{c}_{-k}^{(i)}, \mathbf{y}, \Theta) &\propto p(c_k^{(i)}, \mathbf{c}_{-k}^{(i)}|\mathbf{y}, \Theta) \propto p(y, c_k^{(i)}\mathbf{c}_{-k}^{(i)}|\Theta) \\ &= p(y|c_k^{(i)}\mathbf{c}_{-k}^{(i)}, \Theta)p(c_k^{(i)}\mathbf{c}_{-k}^{(i)}|\Theta) \propto p(y|c_k^{(i)}\mathbf{c}_{-k}^{(i)}, \Theta)p(c_k^{(i)}|\Theta) \\ &= \mathcal{N}(\mathbf{y}; \mathbf{0}, \mathbf{S}\mathbf{C}_k^{(i)}\mathbf{S}^\top + \sigma_v^2\mathbf{I}_T)p(c_k^{(i)}|\Theta) \end{aligned}$$

where the proportionalities do not depend on $c_k^{(i)}$. Therefore we can draw $c_k^{(i)}$ from a Bernoulli distribution such that

$$\begin{aligned} Pr(c_k^{(i)} = 1|\mathbf{c}_{-k}^{(i)}, \mathbf{y}, \Theta) &= \tfrac{p_1\lambda}{p_1\lambda + p_0(1-\lambda)}, \\ Pr(c_k^{(i)} = 0|\mathbf{c}_{-k}^{(i)}, \mathbf{y}, \Theta) &= 1 - Pr(c_k^{(i)} = 1|\mathbf{c}_{-k}, \mathbf{y}, \Theta), \end{aligned}$$

where $p_i = \mathcal{N}(\mathbf{y}; \mathbf{0}, \mathbf{S}\mathbf{C}_{k=1}^{(i)}\mathbf{S}^\top + \sigma_v\mathbf{I}_T),\ i = 0, 1.$

The second way to sample from $p(c_k^{(i)}|\mathbf{c}_{-k}^{(i)}, \mathbf{y}, \Theta)$ is, again, to make use of the CGLSSM's. CGLSSM's have the useful property that, when the filtering, and smoothing moments as well as the innovation and innovation covariance at time k for $c_k = c$ are available, the likelihood of observations $p(\mathbf{y}|\mathbf{c}, \Theta)$ can be calculated up to a proportionality which does not depend on the value c. What Gibbs sampler does is to combine this property with the use of forward-backward smoothing. At iteration i and time k, the algorithm has forward variables calculated using the updated indicator values $\mathbf{c}_{0:k-1}^{(i)}$, and backward parameters calculated using the previous indicators $\mathbf{c}_{k+1:n}^{(i-1)}$ are available. Using this property, at iteration i and time k, we can sample $c_k^{(i)}$ from the full conditional posterior distribution $p(c_k^{(i)}|\mathbf{c}_{-k}^{(i)}, \mathbf{y}, \Theta)$ [50]. This symmetric scanning order of the simulation sites reduce the computational load if T_s is small. For the details of this Gibbs sampling algorithm, see Appendix B.3.

MCMC 2: *Use a Gibbs sampler to sample both c_k and h_k jointly:*

In the second MCMC sampling method, we use the fact that, given all the filter values of $\mathbf{h}$ other than h_k, denoted by, $\mathbf{h}_{-k}$ one can obtain and sample from the posterior distribution of (h_k, c_k) [27]. First notice that

$$p(h_k, c_k|\mathbf{h}_{-k}, \mathbf{y}, \Theta) = p(c_k|\mathbf{h}_{-k}, \mathbf{y}, \Theta)p(h_k|c_k, \mathbf{h}_{-k}, \mathbf{y}, \Theta)$$

Then, it can be shown that

$$p(c_k = 1|\mathbf{h}_{-k}, \mathbf{y}, \Theta) = \left[1 + \frac{1-\lambda}{\lambda}\sqrt{\frac{V_0\sigma_1^2}{V_1\sigma_0^2}}\exp(\frac{m_0^2}{2V_0} - \frac{m_1^2}{2V_1})\right]^{-1}$$

where $V_{c_k} = \left(\frac{1}{V_v} + \frac{1}{\sigma_{c_k}^2}\right)^{-1}$, $m_{c_k} = \frac{V_{c_k}}{V_v}\frac{1}{E}\sum_{i=1}^{T_s} s_i\left[y_{k-i+1} - \sum_{j=1,j\neq i}^{T_s} s_j h_{i+k-j}\right]$, $V_v = \frac{\sigma_v^2}{E}$, and $E = \mathbf{s}^\top\mathbf{s}$; and that

$$p(h_k|c_k, \mathbf{h}_{-k}, \mathbf{y}, \Theta) \sim \mathcal{N}(h_k; m_{c_k}, V_{c_k})$$

So, the Gibbs sampling algorithm for joint simulation of c_k and h_k for I iterations is as follows:

- Start with the initial filter values $\mathbf{h}^{(0)}$.
- for $i = 1, ..., N$, for $k = 1, ..., T$:
  - Construct $\mathbf{h}_{-k}^{(i)} = \left(h_{1:k-1}^{(i)}, h_{k+1:T}^{(i-1)}\right)$,
  - Sample $c_k^{(i)}$ from the Bernoulli distribution $p(c_k|\mathbf{h}_{-k}^{(i)}, \mathbf{y}, \Theta)$,
  - Sample $h_k^{(i)}$ from the Gaussian density $p(h_k|c_k^{(i)}, \mathbf{h}_{-k}^{(i)}, \mathbf{y}, \Theta) = \mathcal{N}(m_{c_k^{(i)}}, V_{c_k^{(i)}})$
  - Keep $h_k^{(i)}$ and $c_k^{(i)}$ as the i^{th} sample set.

For the detailed derivation of the Gibbs sampling algorithm to sample h_k and c_k jointly, one can consult [27].

3.2.1.2. Parameter Estimation for the Bernoulli-Gaussian Model. Having analyzed the methods for state inference of the filter coefficients and the indicators, we can now skip to the parameter estimation part, which is the maximization of the expectation of $\log p(\mathbf{y}, \mathbf{h}, \mathbf{c}|\Theta)$ under the distribution of $p(\mathbf{h}, \mathbf{c}|\mathbf{y}, \Theta)$ over the parameters in Θ. At the i^{th} iteration, the EM quantity is

$$\Omega^{(i)} = \left\langle \log p(\mathbf{y}, \mathbf{h}, \mathbf{c}|\Theta) \right\rangle_{p(\mathbf{h},\mathbf{c}|\mathbf{y},\Theta^{(i-1)})} \tag{3.10}$$

We can write the term in the expectation as

$$\begin{aligned}
\log p(\mathbf{y}, \mathbf{h}, \mathbf{c}|\Theta) &= \log p(\mathbf{c}) + \log p(\mathbf{h}|\mathbf{c}) + \log p(\mathbf{y}|\mathbf{h}) \\
&= n_0^{\mathbf{c}} \log(1-\lambda) + n_1^{\mathbf{c}} \log \lambda - 0.5 \left[T \log 2\pi + \sum_{k=1}^{T} \left(\frac{h_k^2}{\sigma_{c_k}^2} + \log \sigma_{c_k}^2 \right) \right] \\
&- 0.5 \left[T \log 2\pi + T \log \sigma_v^2 + \frac{1}{\sigma_v^2} (\mathbf{y} - \mathbf{H}\mathbf{s})^\top (\mathbf{y} - \mathbf{H}\mathbf{s}) \right] \\
&= n_0^{\mathbf{c}} \log(1-\lambda) + n_1^{\mathbf{c}} \log \lambda \\
&- 0.5 \left[n_0^{\mathbf{c}} \log \sigma_0^2 + \sum_{k:c_k=0}^{T} h_k^2 \right] - 0.5 \left[n_1^{\mathbf{c}} \log \sigma_1^2 + \sum_{k:c_k=1}^{T} h_k^2 \right] \\
&- 0.5 \left[T \log \sigma_v^2 + \frac{1}{\sigma_v^2} \left(\mathbf{y}^\top \mathbf{y} + \mathbf{s}^\top \mathbf{H}^\top \mathbf{H} \mathbf{s} - 2\mathbf{y}^\top \mathbf{H} \mathbf{s} \right) \right] - T \log 2\pi, \quad (3.11)
\end{aligned}$$

where $n_1^{\mathbf{c}} = \sum_{t=1}^{T} c_t$, $n_0^{\mathbf{c}} = T - n_1^{\mathbf{c}}$, and $\mathbf{H}$ is the convolution matrix of size $T \times T_s$ corresponding to $\mathbf{h}$, constructed as in (3.7). To perform maximization, we take the gradient of (3.10) with respect to the parameters in Θ using (3.11) and equate it to zero to obtain the ML solutions for $\Theta^{(i)}$:

$$\hat{\mathbf{s}}^{(i)} = \left(\left\langle \mathbf{H}^\top \mathbf{H} \right\rangle \right)^{-1} \left\langle \mathbf{H} \right\rangle^\top \mathbf{y}, \tag{3.12}$$

$$\hat{\sigma}_v^{2(i)} = \tfrac{1}{T} \left(\mathbf{y}^\top \mathbf{y} + \hat{\mathbf{s}}^{(i)\top} \left\langle \mathbf{H}^\top \mathbf{H} \right\rangle \hat{\mathbf{s}}^{(i)} - 2\mathbf{y}^\top \left\langle \mathbf{H} \right\rangle \hat{\mathbf{s}}^{(i)} \right), \tag{3.13}$$

$$\hat{\sigma}_j^{2(i)} = \tfrac{1}{\left\langle n_j^{\mathbf{c}} \right\rangle} \left\langle \textstyle\sum_{k:c_k=j}^{T} h_k^2 \right\rangle, \; j = 0, 1, \tag{3.14}$$

$$\hat{\lambda}^{(i)} = \tfrac{\left\langle n_1^{\mathbf{c}} \right\rangle}{T}. \tag{3.15}$$

Here, we used a short hand notation such that by $\langle\cdot\rangle$, we mean the expectation under the posterior $p(\mathbf{h},\mathbf{q}|\mathbf{y},\Theta^{(i-1)})$, $\langle\cdot\rangle_{p(\mathbf{h},\mathbf{q}|\mathbf{y},\Theta^{(i-1)})}$.

As discussed in the previous part, we can not obtain the sufficient statistics in equations (3.12) to (3.15), since this requires taking expectations under the intractable posterior distribution $p(\mathbf{h},\mathbf{c}|\mathbf{y},\Theta^{(i-1)})$. To overcome this problem, we can apply Monte-Carlo related EM methods which are based on using (several replications of) simulations to approximate the values of concern. Below, we will recall MCEM, SEM, and SAEM. The general receipt for these methods is as follows:

- Start with the initial $\hat{\Omega}^{(0)}$.
- for $i = 1, 2, \ldots$
  - *Simulation:* Draw N_i samples for $(\mathbf{h},\mathbf{c})$; $(\hat{\mathbf{h}}^{i,1},\hat{\mathbf{c}}^{(i,1)}),...,(\hat{\mathbf{h}}^{(i,N_i)},\hat{\mathbf{c}}^{(i,N_i)})$ from $p(\mathbf{h},\mathbf{c}|\mathbf{y};\Theta^{(i-1)})$.
  - *Maximization:* Compute $\Theta^{(i)}$ that maximizes $\hat{\Omega}^{(i)}$, where

$$\hat{\Omega}^{(i)} = \nu_i \frac{1}{N_i}\sum_{j=1}^{N_i} \log p(\mathbf{y},\hat{\mathbf{h}}^{(i,j)},\hat{\mathbf{c}}^{(i,j)}|\Theta^{(i-1)}) + (1-\nu_i)\hat{\Omega}^{(i-1)}.$$

Recall that, we have:

- SAEM: $\{\nu_i\}_{i\geq 1} > 0$, $\nu_1 \neq 1$, and $\nu_i \leq \nu_{i+1} \leq 1$.
- MCEM: $\nu_i = 1$ and N_i increases with time.
- SEM: $\nu_i = 1$ and N_i is constant.

Note that the expectations in the update equations (3.12) to (3.15) will now be replaced by the weighted average of the average of their simulated values from the posterior at time i and their previous estimates at time $i-1$. That is, the approximation for any sufficient statistics of $(\mathbf{c},\mathbf{h})$, $S_f^{(i)} = \langle f(\mathbf{c},\mathbf{h})\rangle_{p(\mathbf{h},\mathbf{c}|\mathbf{y};\Theta^{(i-1)})}$ is

$$\hat{S}_f^{(i)} = (1-\nu_i)\hat{S}_f^{(i-1)} + \nu_i \frac{1}{N_i}\sum_{j=1}^{N_i} f(\mathbf{c}^{(i,j)},\mathbf{h}^{(i,j)}).$$

3.2.2. Inverse-Gamma Model

For the inverse-gamma model, we have:

- Latent statistical variables: $\mathbf{h}$ and $\boldsymbol{\sigma}_{\mathbf{h}}^{\mathbf{2}} \equiv \sigma_{h,(1:T)}^2$
- Deterministic parameters: $\Theta = (\mathbf{s}, \alpha, \beta, \sigma_v^2)$.

Thus, the EM steps are:

- E step: Requires inference of the posterior to compute the EM quantity:

$$\Omega^{(i)} = \langle \log p(\mathbf{y}, \mathbf{h}, \boldsymbol{\sigma}_{\mathbf{h}}^{\mathbf{2}} | \Theta) \rangle_{p(\mathbf{h}, \boldsymbol{\sigma}_{\mathbf{h}}^{\mathbf{2}} | \mathbf{y}, \Theta^{(i-1)})}. \tag{3.16}$$

- M step: Parameter estimation: Maximize $\Omega^{(i)}$ over Θ:

$$\Theta^{(i)} = \arg\max_{\Theta} \Omega^{(i)}.$$

3.2.2.1. State Inference for the Inverse-Gamma Model. We have to calculate the posterior in (3.16), i.e. $p(\mathbf{h}, \boldsymbol{\sigma}_{\mathbf{h}}^{\mathbf{2}} | \mathbf{y}, \Theta)$. It can be shown that the posterior distribution $p(\mathbf{h}, \boldsymbol{\sigma}_{\mathbf{h}}^{\mathbf{2}} | \mathbf{y}, \mathbf{s}, \sigma_v^2)$ is intractable. However, the full conditional posteriors $p(\mathbf{h} | \boldsymbol{\sigma}_{\mathbf{h}}^{\mathbf{2}}, \mathbf{y}, \Theta)$, and $p(\boldsymbol{\sigma}_{\mathbf{h}}^{\mathbf{2}} | \mathbf{h}, \mathbf{y}, \Theta)$ can easily be calculated. Motivated with this fact, we can use Gibbs sampling and variational Bayes methods in order to approximate the exact posterior of $(\mathbf{h}, \boldsymbol{\sigma}_{\mathbf{h}}^{\mathbf{2}})$. Recall that, the iterations of two methods are very similar to derive: One uses samples of the latent variables whereas the other one travels through their sufficient statistics.

Gibbs Sampling for $\mathbf{h}$ *and* $\boldsymbol{\sigma}_{\mathbf{h}}^{\mathbf{2}}$: We can write the logarithm of the unnormalized

posterior, which is the joint density of $\mathbf{y}$, $\mathbf{h}$, and $\sigma^{\mathbf{2}}_{\mathbf{h}}$, as in the following:

$$\begin{aligned}
\log & p(\mathbf{h}, \sigma^{\mathbf{2}}_{\mathbf{h}}, \mathbf{y}|\Theta) = \log p(\sigma^{\mathbf{2}}_{\mathbf{h}}) + \log p(\mathbf{h}|\sigma^{\mathbf{2}}_{\mathbf{h}}) + \log p(\mathbf{y}|\mathbf{h}, \sigma^{\mathbf{2}}_{\mathbf{h}}, \Theta) \\
&= \left[\sum_{k=1}^{T} \alpha \log \beta - (\alpha+1)\log \sigma^2_{h,k} - \beta/\sigma^2_{h,k} - \log\Gamma(\alpha)\right] \\
&- 0.5\left[\sum_{k=1}^{T} \log \sigma^2_{h,k} + \frac{h_k^2}{\sigma^2_{h,k}} + T\log\sigma_v^2 + \frac{1}{\sigma_v^2}(\mathbf{y}-\mathbf{Sh})^\top(\mathbf{y}-\mathbf{Sh})\right] + T\log 2\pi \\
&= \sum_{k=1}^{T}\left[\alpha\log\beta - (\alpha+1.5)\log(\sigma^2_{h,k}) - \frac{\beta+0.5h_k^2}{\sigma^2_{h,k}} - \log\Gamma(\alpha)\right] \\
&- 0.5\left[T\log\sigma_v^2 + \frac{1}{\sigma_v^2}\left(\mathbf{y}^\top\mathbf{y} + \mathbf{h}^\top\mathbf{S}^\top\mathbf{S}\mathbf{h} - 2\mathbf{h}^\top\mathbf{S}^\top\mathbf{y}\right)\right] + T\log 2\pi. \qquad (3.17)
\end{aligned}$$

From (3.17), we can deduce the full conditional posteriors of $\mathbf{h}$, and $\sigma^2_{h,k}$. The posterior of $\mathbf{h}$ is given by

$$p(\mathbf{h}|\sigma^{\mathbf{2}}_{\mathbf{h}}, \mathbf{y}, \Theta) = \mathcal{N}\left(\mathbf{h}; \mathbf{m_h}, \mathbf{\Sigma_h}\right), \qquad (3.18)$$

with $\mathbf{\Sigma_h} = \left(\mathbf{D_h}^{-1} + \frac{1}{\sigma_v^2}\mathbf{S}^\top\mathbf{S}\right)^{-1}$ and $\mathbf{m_h} = \mathbf{C_h}\frac{1}{\sigma_v^2}\mathbf{S}^\top\mathbf{y}$. where $\mathbf{D_h}$ is the diagonal matrix of size $T \times T$ with $\mathbf{D_h}_{k,k} = \sigma^2_{h,k}$.

The full conditional posterior of $\sigma^{\mathbf{2}}_{\mathbf{h}}$ is in fact conditioned on only the coefficients of the sparse filter $\mathbf{h}$. It can be shown from (3.17) that,

$$p(\mathbf{h}, \sigma^{\mathbf{2}}_{\mathbf{h}}) \propto \exp\left[\sum_{k=1}^{T} -(\alpha+1.5)\log(\sigma^2_{h,k}) - \frac{\beta+0.5h_k^2}{\sigma^2_{h,k}}\right]. \qquad (3.19)$$

Since the posterior in (3.19) is factorized, the posterior distributions of the variances can be found separately for each k as:

$$p(\sigma^{\mathbf{2}}_{\mathbf{h}}|\mathbf{h}, \mathbf{y}, \Theta) = p(\sigma^{\mathbf{2}}_{\mathbf{h}}|\mathbf{h}) = \prod_{k=1}^{T} p(\sigma^2_{h,k}|h_k) = \prod_{k=1}^{T} \mathcal{IG}(\sigma^2_{h,k}; \alpha+0.5, \beta+0.5h_k^2). \qquad (3.20)$$

As we can see from (3.18) and (3.20), a Gibbs sampler can be used to draw samples from the joint posterior distribution.

Variational Bayes Algorithm for $\mathbf{h}$ *and* $\sigma_{\mathbf{h}}^{\mathbf{2}}$*:* We have shown that the full conditional posteriors of $\mathbf{h}$, and $\sigma_{\mathbf{h}}^{\mathbf{2}}$ can easily be calculated. Moreover, since they are in a closed form in terms of known distributions, and the (required) sufficient statistics of the variables can be calculated from these distributions, we can use a variational Bayes algorithm with mean fild approximation to infer the joint posterior. Recall that, in the mean field approximation, we assume a factorized form for the full joint posterior as

$$p(\mathbf{h}, \sigma_{\mathbf{h}}^{\mathbf{2}}|\mathbf{y}, \Theta) = q(\mathbf{h})q(\sigma_{\mathbf{h}}^{\mathbf{2}});$$

and the update rules for this factorized distribution are as follows:

$$q(\mathbf{h})^{(n+1)} \propto \exp\left(\langle \log p(\mathbf{h}, \sigma_{\mathbf{h}}^{\mathbf{2}}, \mathbf{y}|\Theta)\rangle_{q(\sigma_{\mathbf{h}}^{\mathbf{2}})^{(n)}}\right)$$
$$q(\sigma_{\mathbf{h}}^{\mathbf{2}})^{(n+1)} \propto \exp\left(\langle \log p(\mathbf{h}, \sigma_{\mathbf{h}}^{\mathbf{2}}, \mathbf{y}|\Theta)\rangle_{q(\mathbf{h})^{(n+1)}}\right)$$

Using similar steps to the ones used for deriving the Gibbs sampling iterations, we can show that

$$q(\mathbf{h})^{(n+1)} = \mathcal{N}\left(\mathbf{h}; \mathbf{m_h}^{(n+1)}, \mathbf{\Sigma_h}^{(n+1)}\right), \tag{3.21}$$

with $\mathbf{\Sigma_h}^{(n+1)} = \left(\langle \mathbf{D_h}^{-1}\rangle_{q(\sigma_{\mathbf{h}}^{\mathbf{2}})^{(n)}} + \frac{1}{\sigma_v^2}\mathbf{S}^\top\mathbf{S}\right)^{-1}$ and $\mathbf{m_h}^{(n+1)} = \frac{1}{\sigma_v^2}\mathbf{\Sigma_h}^{(n+1)}\mathbf{S}^\top\mathbf{y}$; and

$$q(\sigma_{\mathbf{h}}^{\mathbf{2}})^{(n+1)} = \prod_{k=1}^{T} \mathcal{IG}(\sigma_{h,k}^2; \alpha_h^{(n+1)}, \beta_{h,k}^{(n+1)}) \tag{3.22}$$

with $\alpha_h = \alpha + 0.5$ and $\beta_{h,k} = \beta + 0.5\langle h_k^2\rangle_{q(\mathbf{h})^{(n+1)}}$. The sufficient statistics in equations (3.21) and (3.22) are easy to derive. Since $q(\sigma_{h,k}^2)^{(n)}$ is an inverse-gamma with scale and shape parameters α_h and $\beta_{h,k}$, $\frac{1}{\sigma_{h,k}^2}$ is a gamma random variable with shape and scale parameters $(\alpha_h, \frac{1}{\beta_{h,k}})$. It is well known that the mean of a gamma random variable is

the product of its shape and scale parameters, therefore we have

$$\langle \frac{1}{\sigma_{h,k}^2} \rangle_{q(\sigma_{h,k}^2)^{(n)}} = \frac{\alpha_h}{\beta_{h,k}}$$

For (3.22), we only need to know the sufficient statistics $\langle h_k^2 \rangle_{q(\mathbf{h})^{(n+1)}}$ for each k. Since $q(h)^{(n+1)}$ is a Gaussian with mean $\mathbf{m_h}^{(n+1)}$ and covariance $\mathbf{\Sigma_h}^{(n+1)}$,

$$\langle h_k^2 \rangle_{q(\mathbf{h})^{(n+1)}} = \mathbf{m_h}_k^{2(n+1)} + \mathbf{\Sigma_h}_{k,k}^{(n+1)}.$$

One can also calculate the variational bound $\mathcal{VB}_Q$ at every iteration. Recall that the marginal likelihood $\log Z = \log p(\mathbf{y}|\Theta)$ is lower bounded by the variational bound $\mathcal{VB}_Q = \langle \log \phi \rangle_Q + H(Q)$ where $H(\cdot)$ is the differential entropy. It can be verified that, the variational bound for $\log Z$ is given below (see Appendix):

$$\begin{aligned}\mathcal{VB}_Q = &\sum_{k=1}^{T} \log \frac{\beta^\alpha \Gamma(\alpha_h)}{\beta_{h,k}^{\alpha_h} \Gamma(\alpha)} - 0.5 \left(T \log(2\pi\sigma_v^2) - T - \log |\mathbf{\Sigma_h}| \right) \\ &- 0.5 \left[\frac{1}{\sigma_v^2} \left(\mathbf{y}^\top \mathbf{y} - 2\mathbf{m_h}^\top \mathbf{S}^\top \mathbf{y} + tr\left[\mathbf{S}^\top \mathbf{S} (\mathbf{m_h} \mathbf{m_h}^\top + \mathbf{\Sigma_h}) \right] \right) \right]\end{aligned}$$

3.2.2.2. Parameter Estimation for the Inverse-Gamma Model. Recall that, estimation of $\Theta^{(i)}$ is the maximization of the EM quantity at time i in (3.16). In (3.17), we have already found the term in the expectation in (3.16). It can easily be seen that the update rules for $\hat{\mathbf{s}}^{(i)}$ and $\hat{\sigma_v}^{2(i)}$ are the same as those found in (3.12) and (3.13) for the Bernoulli-Gaussian model. For $\hat{\alpha}^{(i)}$ and $\hat{\beta}^{(i)}$, notice from (3.17) that

$$\Omega^{(i)} = T\alpha \log \beta - \alpha \sum_{k=1}^{T} \left\langle \log \sigma_{h,k}^2 \right\rangle - \beta \sum_{k=1}^{T} \left\langle \frac{1}{\sigma_{h,k}^2} \right\rangle - \log \Gamma(\alpha) + C \tag{3.23}$$

where C does not depend on $\hat{\alpha}^{(i)}$ and $\hat{\beta}^{(i)}$, and by $\langle \cdot \rangle$ we mean $\langle \cdot \rangle_{p(\mathbf{h}, \sigma_{\mathbf{h}}^2 | \mathbf{y}; \Theta^{(i-1)})}$. A closed form expression for $\hat{\alpha}^{(i)}$ and $\hat{\beta}^{(i)}$ that maximize (3.23) is not available. However, for a

given estimate of $\hat{\alpha}^{(i)}$, we have

$$\hat{\beta}^{(i)} = \frac{T\hat{\alpha}^{(i)}}{\sum_{k=1}^{T} \left\langle \frac{1}{\sigma^2_{h,k}} \right\rangle}. \tag{3.24}$$

Substituting (3.24) into (3.23), one can obtain a function which only depends on α. Hopefully, this unction is known to be well behaved and maximization algorithms such as grid based methods, gradient descent [58], or Newton's method [59], work well to determine $\hat{\alpha}^{(i)}$.

The general EM procedure applies when we choose the variational Bayes method to approximate $p(\mathbf{h}, \sigma^{\mathbf{2}}_{\mathbf{h}}|\mathbf{y}; \Theta^{(i-1)})$, because we have a closed form expression for it, and we can evaluate the sufficient statistics needed for maximization. However, if we choose Gibbs sampling for the state inference, then we do not have a closed form expression but samples. In this case, we perform Monte-Carlo related EM methods as for the Bernoulli-Gaussian model:

- Start with the initial $\hat{\Omega}^{(0)}$.
- for $i = 1, 2, \ldots,$
  - *Simulation:* Draw N_i samples for $(\mathbf{h}, \sigma^{\mathbf{2}}_{\mathbf{h}})$; $(\hat{\mathbf{h}}^{(i,1)}, \hat{\sigma}^{\mathbf{2}(i,1)}_{\mathbf{h}}), ..., (\hat{\mathbf{h}}^{(i,N_i)}, \hat{\sigma}^{\mathbf{2}(i,N_i)}_{\mathbf{h}})$ from $p(\mathbf{h}, \sigma^{\mathbf{2}}_{\mathbf{h}}|\mathbf{y}; \Theta^{(i-1)})$.
  - *Maximization:* Compute $\Theta^{(i)}$ that maximizes $\hat{\Omega}^{(i)}$, where

$$\hat{\Omega}^{(i)} = \nu_i \frac{1}{N_i} \sum_{j=1}^{N_i} \log p(\mathbf{y}, \hat{\sigma}^{\mathbf{2}i,j}_{\mathbf{h}}), \hat{\sigma^{\mathbf{2}}_{\mathbf{h}}}^{i,j}|\Theta^{(i-1)}) + (1 - \nu_i)\hat{\Omega}^{(i-1)}.$$

3.3. Experiments and Results

3.3.1. Simulation Experiments for Blind Deconvolution

In this part, we run simulations in order to compare the performances of the two models, the Bernoulli-Gaussian model and the inverse-gamma model, in estimating the coefficients of the sparse processes and the source vector. Furthermore, for each model

we compare all the techniques elaborated in inference and parameter estimation steps in Section 3.2. We investigate the performances by observing the results averaged over 50 different simulations. In each simulation, we generated a sparse trace using the Bernoulli-Gaussian model, with the parameters set as the following:

- Length of the sparse process: $T = 200$,
- Length of the source vector: $T_s = 10$,
- Generation of source vector: $\mathbf{s} \sim \mathcal{N}(\mathbf{s}; \mathbf{0}, \mathbf{I}_{T_s})$,
- Density of spikes: $\lambda = 0.05$,
- Lower and higher variances of the sparse process: $\sigma_0^2 = 0.0001, \sigma_1^2 = 4$,
- Observation noise variance: $\sigma_v^2 = 0.01$.

Recall that, both the Bernoulli-Gaussian and inverse-gamma models can generate sparse processes which are quantitatively very similar (see Figures 3.1 and 3.2). However, in the Bernoulli-Gaussian model, the locations of spikes are given by $\mathbf{c}$. This property not only makes this model more intuitive, but also enables us to measure the spike detection rate-number of false alarms in an automatic way. That is why in our tests we generate sparse processes using this model. In this way, we will also realize from the results that, the inverse-gamma model acts as a relaxation of the Bernoulli model, which was argued in Chapter 2.

For the Bernoulli-Gaussian model, we have 3 different parameter estimation methods, namely MCEM, SAEM and SEM. two different Gibbs sampling techniques for state inference, MCMC1 and MCMC2. Therefore, we have 6 different combinations for this model. In the inverse-gamma model, if we use Gibbs sampling for inference of the posterior, we have again three different techniques: MCEM, SAEM, SEM. If we use variational Bayes, then the classical EM can be used because we have the posterior approximated by closed form expressions. Therefore, we have 4 different categories for the inverse-gamma model. For all these categories, algorithmic parameters such as the number of EM iterations, number of variational Bayes iterations in every E step of EM, etc. are chosen sufficiently large to let the parameter estimations or distribution approximations converge.

In order to compare the results, we look at the *normalized mean squared errors* (nMSE) in the mean estimates of $\mathbf{h}$ and $\mathbf{s}$. The nMSE for an estimate $\hat{x}$ of x is given by

$$\text{nMSE}_x(\hat{x}) = \frac{\sum_i (x_i - \hat{x}_i)^2}{\sum_i x_i^2} \tag{3.25}$$

We use nMSE, because MSE values do not give a reliable result unless you know the powers of $\mathbf{h}$, and $\mathbf{s}$. Since we generate different realizations in every simulations, the power values are not constant.

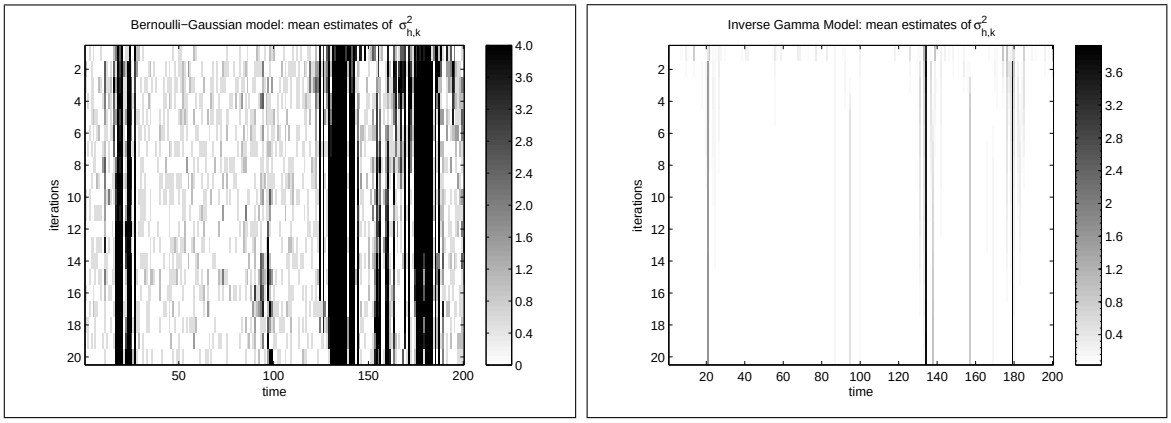
Scale invariance problem: It is important to note that, there is a scale invariance between the estimates of $\mathbf{h}$ and $\mathbf{s}$ by the nature of our model formulations and the methods we use. Assume that $\mathbf{y} = \mathbf{h} * \mathbf{s} + \mathbf{v}$ and $h_k \sim \lambda\mathcal{N}(h_k; 0, \sigma_0^2) + (1-\lambda)\mathcal{N}(h_k; 0, \sigma_1^2)$. As long as the parameters σ_0^2, σ_1^2 and $\mathbf{s}$ are treated to be unknown and fully deterministic, there is an invariance between the estimates $\Theta = (\mathbf{s}/k, k^2\sigma_0^2, k^2\sigma_1^2)$ for all k's. Similarly for the inverse-gamma model, there is no difference for the maximum likelihood estimator between the parameter sets $\Theta = (\mathbf{s}/k, \alpha, k\beta)$, for any k. Therefore, it is unnecessary to set all the parameters free. That is why we assume that σ_0^2 and β are known and fix them to certain values. As a consequence, for an estimate $\hat{\mathbf{h}}$ of $\mathbf{h}$, , we calculate the nMSE's for $k\hat{\mathbf{h}}$ and $\hat{\mathbf{s}}/k$ where $k = \hat{\mathbf{h}}^\top \mathbf{h} / \hat{\mathbf{h}}^\top \hat{\mathbf{h}}$.

Table 3.1 shows the nMSE values in estimation of $\mathbf{s}$ and $\mathbf{h}$ for all the combinations of the models and the methods. We can see that, in every combination of methods, the inverse-gamma model gives much better results than the Bernoulli-Gaussian model. We also observe that, in the inverse-gamma case, Gibbs sampling works better than variational Bayes in the posterior inference step. Another result is that, though the performances of MCEM, SAEM and SEM are very close to each other, SEM is slightly worse. Finally, even though the performances of all of the combinations in the Bernoulli-Gaussin model are not acceptable for a good estimation, we can observe that MCMC1 Gibbs sampling type gives better results.

Table 3.1. Performances of the models and methods. nMSE-h/nMSE-s: nMSE of estimates for $\mathbf{h}/\mathbf{s}$

Model	Par.est.	St.inf	nMSE-h	nMSE-s
Bernoulli-Gaussian	MCMC1	MCEM	0.3674 ± 0.4127	0.3510 ± 0.4090
		SEM	0.3547 ± 0.4394	0.3591 ± 0.4100
		SAEM	0.1839 ± 0.1837	0.2166 ± 0.1880
	MCMC2	MCEM	0.4238 ± 0.3686	0.4115 ± 0.4478
		SEM	0.3854 ± 0.3178	0.5281 ± 0.3384
		SAEM	0.2698 ± 0.1977	0.2270 ± 0.1802
Inverse-Gamma	Gibbs	MCEM	0.0028 ± 0.0022	0.0005 ± 0.0006
		SEM	0.0067 ± 0.0366	0.0015 ± 0.0087
		SAEM	0.0024 ± 0.0016	0.0004 ± 0.0004
	VB	varEM	0.0142 ± 0.0548	0.0063 ± 0.0375

The reason of this significant difference between the performance of the models was explained intuitively in section 3.1, where our argument was illustrated by Figure 3.3. Here we can explain the reason with another illustration. Figure 3.4 shows the estimated means of the variances of the sparse process over the first 20 iterations. For the Bernoulli-Gaussian model, MCEM with MCMC1 is used; whereas MCEM with Gibbs sampling is used for the inverse-gamma model. In the Bernoulli-Gaussian model, the effect of initial densities shows itself during iterations, and the algorithm can not move to finer solutions. It is also observed that the algorithm *feels* an attraction in the locations where there are spikes, but can not tune itself to a sharper estimate. Moreover, because of the discreteness of the model, that is, because the variances are assumed to be one of the two distant variances, the transitions of the estimates are not smooth. However, we can see that in the inverse-gamma model the algorithms can easily get rid of their initial points and can move to more likely estimates. Also, the estimates are more tuned to the exact locations compared to the other model; that is, the inverse-gamma model can detect the exact location of the spikes, unlike in the Bernoulli-Gaussian model.



(a) (b)

Figure 3.4. Mean estimates of the variances $\sigma^2_{h,k}$. (a) Bernoulli-Gaussian model (MCEM with MCMC1 is used), (b) inverse-gamma model MCEM with Gibbs sampling is used.

The effect of this difference in the variance estimates surely affects the waveform estimates for both $\mathbf{h}$, and $\mathbf{s}$. Figure 3.5 shows a typical example of how the estimates for $\mathbf{s}$ and $\mathbf{h}$ look like in both models. Notice that, whereas the Bernoulli-Gaussian estimates are not successful, the estimates of inverse-gamma model are almost exactly the same with the true ones, with a scale difference the reason of which we explained above.

We have shown that the inverse-gamma model is much more suitable for statistical inference and parameter estimation in cases containing sparsity. That is why we will continue with inverse-gamma model in the rest of our work. However, this does not mean that the Bernoulli-Gaussian model is useless at all. As we have stated above, it is more intuitive than the inverse-gamma model since it gives direct information for the locations of spikes by its construction. Moreover, Yıldırım [32] showed that the Bernoulli-Gaussian model gives satisfactory results when initialized properly. Therefore, we can still use this model when we have strong prior belief on the unknown variables/parameters.

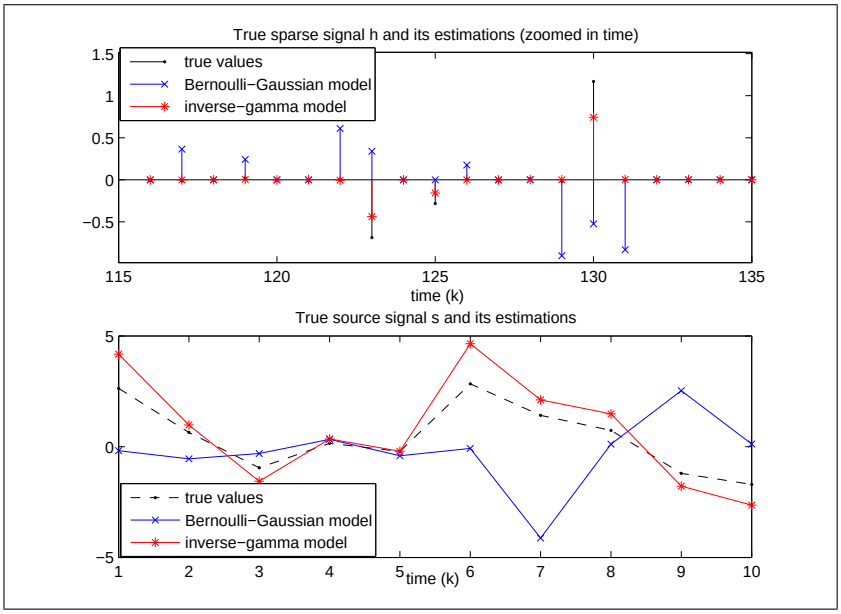


Figure 3.5. True and estimated waveforms for **h** and **s**. For the Bernoulli-Gaussian model, MCEM with MCMC1 is used, for the inverse-gamma model, MCEM with Gibbs sampling is used.

3.3.2. Estimation of Receiver Functions from Earthquake Records

As stated in Chapter 1, in earthquake seismology, receiver functions are used to analyze the relative response of earth structure. Via a three component station, the vertical and radial components of motion are available. The vertical component corresponds to the source signal, whereas the radial component is the observation, which is the convolution of the source and the receiver function of the earth structure near the receiver. Some additive noise in the observation records is present and assumed to be white Gaussian. Therefore, to estimate the receiver function, a deconvolution operation must be performed to the noisy observation using the source. For the sake of convenience with the notation, we will denote the source by **s**, the observation by **y**, and the noise component by **v**.

Gaussian Filter for the Receiver Function. To clean up high-frequency noise in the receiver functions, the receiver function is assumed to be a sparse trace filtered

with a low-pass Gaussian filter, which is given by [9]

$$G(\omega) = \exp(-\frac{\omega^2}{4a^2}) \tag{3.26}$$

where $\omega = 2\pi f$ denotes the angular frequency. Note that the frequency content is controlled by the Gaussian filter-width parameter, a. and the filter is a unit area pulse (i.e., the filter gain at $\omega = 0$ is unity). It is usual in seismology to quantify the filter by the frequency at which $G(2\pi f)$ has a value of 0.1 [9].

Let $\mathbf{g}$ be the time domain vector of non-zero coefficients of the Gaussian filter. Then, the receiver function $\mathbf{r}$ can be expressed by

$$\mathbf{r} = \mathbf{g} * \mathbf{h} \tag{3.27}$$

Since the convolution operation is associative, using (3.27) we can write

$$\mathbf{y} = \mathbf{s} * \mathbf{r} + \mathbf{v} = \mathbf{s} * (\mathbf{g} * \mathbf{h}) + \mathbf{v} = (\mathbf{s} * \mathbf{g}) * \mathbf{h} + \mathbf{v}. \tag{3.28}$$

Notice that $\mathbf{x} \equiv \mathbf{s}*\mathbf{g}$ in (3.28) is given to us. Hence, we can use our Bayesian methods for blind deconvolution in order to estimate the reflectivity function. The only difference is that $\mathbf{s}$ is replaced by $\mathbf{x}$, and $\mathbf{x}$ is known. So, we shall skip the parameter update step for the source vector $\mathbf{s}$ in the M step of the methods. To summarize, given $\mathbf{y}$ and $\mathbf{s}$, we shall follow the steps below for Bayesian deconvolution of the receiver function $\mathbf{r}$:

- Calculate $\mathbf{x} = \mathbf{s} * \mathbf{g}$.
- Given $\mathbf{y}$ and $\mathbf{x}$, use the Bayesian methods for deconvolution of $\mathbf{h}$.
- Calculate the estimated receiver function $\mathbf{r} = \mathbf{h} * \mathbf{g}$.

3.3.2.1. Experiments with Simulated Data. We already mentioned about the disadvantages of the iterative deconvolution method in chapter 1, and claimed that a Bayesian approach might improve the estimations in the cases where there is noise

and the whole convolution is not available. Before dealing with real data, we will now support those theoretical arguments with simulation experiments.

Recall that, one of the disadvantages is that, in iterative deconvolution no observation noise is assumed, which may lead to wrong estimates in the noisy convolution case. However, because of the observation noise assumption, we expect better results from the Bayesian deconvolution. In our first experiment, we aim to compare the performances of iterative deconvolution and Bayesian deconvolution under noise. To do this, we evaluated the nMSE values for the two types of deconvolution using different signal-to-noise (SNR) ratios. By SNR, we mean the ratio obtained by the power of $\mathbf{r}$ divided by the power of the observation noise, which is σ_v^2 in our case. We averaged our results over 100 simulations. For each simulation:

- We generate a sparse process of length $T = 50$ using the Bernoulli-Gaussian model with parameters $\sigma_0^2 = 0.01$, $\sigma_1^2 = 1$, and $\lambda = 0.05$.
- The source signal $\mathbf{s}$ of length $T_s = 50$ was generated by an AR process of order 5, where AR parameters are chosen randomly.
- For SNR values $-10, -9, \ldots, 50$:
  - We generated the observation noise with power corresponding to the current SNR, and add it to the convolution to obtain the noisy convolution $\mathbf{y}$,
  - We perform Bayesian deconvolution and iterative deconvolution separately for the estimation of $\mathbf{r}$,
  - For each result, we calculated the nMSE in the estimation of $\mathbf{r}$.

Figure 3.6 shows the nMSE's versus the SNR values for both deconvolution types. It can be observed that, the iteration deconvolution method exhibits larger errors while SNR decreases. Bayesian deconvolution suffers from noise, too, however, it is still more reliable than iterative deconvolution. Also, Bayesian deconvolution performs even better when there is very small, almost no noise. That is due to the fact that in Bayesian deconvolution we can introduce prior knowledge about $\mathbf{h}$, whereas in iterative deconvolution, the algorithm does not take the statistical properties of $\mathbf{h}$ into account.

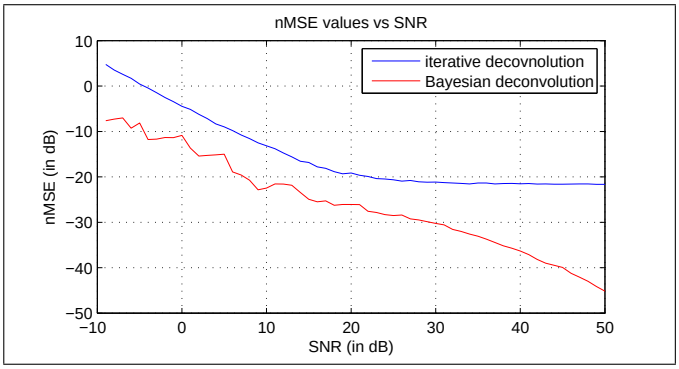


Figure 3.6. nMSE vs SNR values for iterative deconvolution and Bayesian deconvolution

Another disadvantage of iterative deconvolution is that it cannot handle missing data correctly. The missing data problem in deconvolution usually occurs when the convolution is truncated, that is, length of the convolution less than the sum of the lengths of the convolving components. Indeed, we have such a problem in our earthquake data, where the source signal, observation signal and the receiver functions are very close in length, say, 500, 500, and 450, respectively. This means that almost half of the convolution data is missing. This is a serious problem for iterative deconvolution. However, Bayesian deconvolution method is based on a generative model, and the missing data case can be introduced to the model by suitable choices of the vector lengths $\mathbf{s}$, $\mathbf{h}$ and $\mathbf{y}$. We can show this fact with a simulation experiment, where we obtain nMSE values versus the number of truncated samples in the convolution. The steps and the system parameters of the experiment are the same as the one above, the only difference is that we truncate the observation signal instead of adding noise to it. Figure 3.7 shows the result of the experiment. It is obvious that Bayesian deconvolution keeps estimating $\mathbf{h}$ with an acceptable error even if the observation length is equal to the source length, whereas in the iterative deconvolution method nMSE gets close to 1, even greater than 1, while the observation signal is getting shorter.

3.3.2.2. Experiments with Real Earthquake Data. We have three sets of seismological observation and source signals, recorded and supplied by Boğaziçi University Kandilli Observatory and Earthquake Research Institute. Each set was recorded in a different station and contains over 100 vertical-radial component couples. The stations are

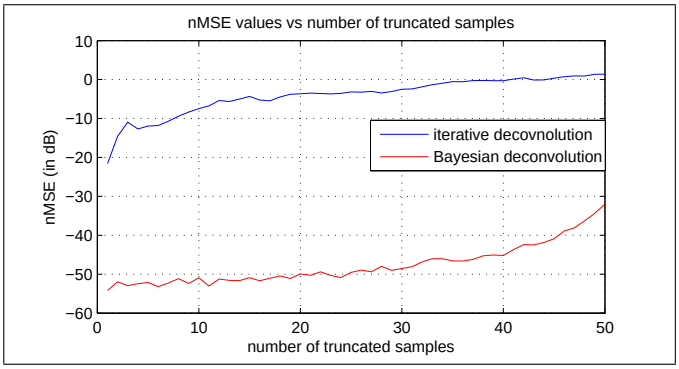


Figure 3.7. nMSE vs number of truncated samples from the observation signal for iterative deconvolution and Bayesian deconvolution

labeled as 'lod0', 'bnn0' and 'kars'. The frequency at which the gaussian filter $G(2\pi f)$ has a value of 0.1 is determined as $f = 1.2$Hz for this data, which can be realized by selecting $a = 2.5$ in (3.26). Then, the corresponding essential width of the Gaussian filter is approximately 1 second. The original data sampling rate is 50Hz, but since it is known that no information is contained in the frequencies higher than 5Hz, we resample the data to 5Hz before applying Bayesian deconvolution. Therefore, the length D of the Gaussian filter is chosen 5.

The receiver functions corresponding to the data we have were already estimated by Özakın [23], who applied iterative deconvolution to the records. We apply Bayesian deconvolution to this data using the inverse-gamma model and compare our results with the ones obtained in [23].

For comparison, the obtained results for each station are plotted as follows: First, the records are sorted with respect to their back azimuth values, which are the angles of direction from the source of the earthquake to the receiver [60]. Then all of the estimated receiver functions belonging to that station are placed in a top-to-down order. For the sake of visuality, the positive and negative parts of each receiver function estimate are shaded with different colors. Also, all of the receiver estimates are scaled by a constant to increase the visual quality of the images.

In fact, there is almost no analytical criteria which can be used to determine and compare the practical quality of our results. In ideal situation, where all earthquake

occurs in a fix location on earth, all reflectivity waveforms are expected to be equal; and when plotted with the usual procedure, all reflectivity pulses (main pulses as well as reflected pulses) are expected to align in a vertical line. However; in real situation, where earthquakes are scattered around the globe, the vertical alignment can not be observed. The deviation from verticality is due to the non-horizontality of layers which produce different reflection arrivals depending on the azimuth of the incoming wave. In such situation, when reflectivity waveforms are plotted in the order of azimuth angle, the vertical alignment of the pulses do not show a straight line any more, but a gently undulating curve. This undulation is in turn used for the estimation of the dipping angle for the subsurface layers. However, in Turkey, for the last 5 recent years, most of the recorded distant earthquakes came from two given geographical location, the Kurile-Japan subduction zone and the Java-Sumatra subduction zone, which can be assumed to have a fix azimuth and a fix distance. We therefore expect that, whatever the horizontality of the subsurface layers, the reflectivity sequence from a seismic station in Turkey should show at least two clear sections where vertical alignment is close to a straight line. An example for such a straight line can be seen in the left of Figure 3.10, at about 25 (x0.2) seconds. If these lines are angled, this indicates that the layers are not fully horizontal, but they have an angle with respect to the ground. We can observe such a situation on the upper-left side of Figure 3.12, in the top few receiver estimates at about 50 (x0.2) seconds. A start and end of a line indicates the starting and ending of a new subsurface layer. A simple illustration of all these cases are given in Figure 3.8.

Figures 3.9 to 3.14 show the results of iteraive deconvolution and Bayesian deconvolution methods for the three stations. The first thing to say about the figures is that the results of Bayesian deconvolution are consistent with the ones of iterative deconvolution. This is an important and promising result for us, because further seismological analysis Özakın derived out of these results have been found to be meaningful. Furthermore, the Bayesian deconvolution results have some advantages in some aspects. Firstly, some new vertical lines which are not apparent in the iterative deconvolution results are detected in the Bayesian deconvolution results. For example, the third blue line in the upper half of Figure 3.10 at around 40 (x0.2) seconds is not apparent in

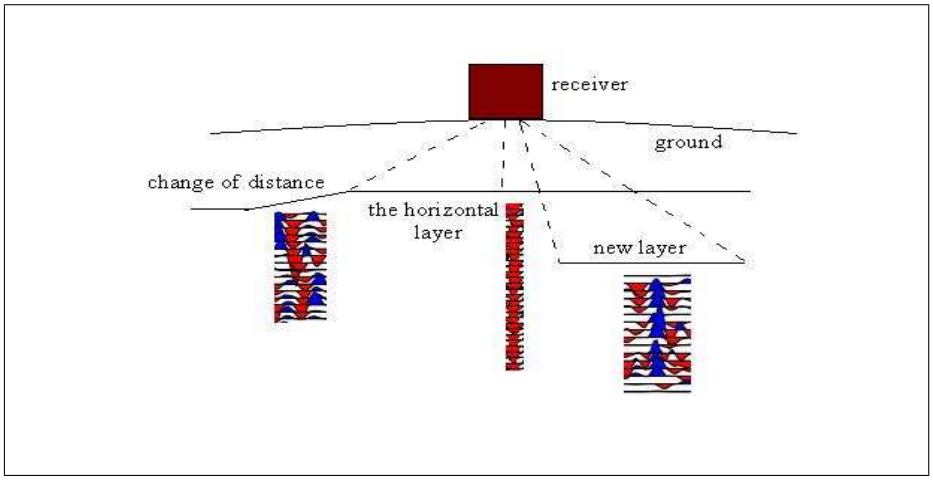


Figure 3.8. Correspondence of lines in an image of receiver functions with the localization of subsurface layers.

Figure 3.9. Similar situations can be observed in the image belonging to 'bnn0' station. Secondly, the vertical lines in the images of Bayesian deconvolution are more straight than the ones belonging to the iterative deconvolution. The examples of this difference can easily be seen in the figures for the stations 'lod0' and 'kars'. Moreover, it is observed that Bayesian deconvolution can separate two close pulses as separate more successfully than the iterative deconvolution. This can be clearly seen at the first 10 (x0.2) seconds of the upper halves of 'kars' images (Figures 3.13 and 3.14).

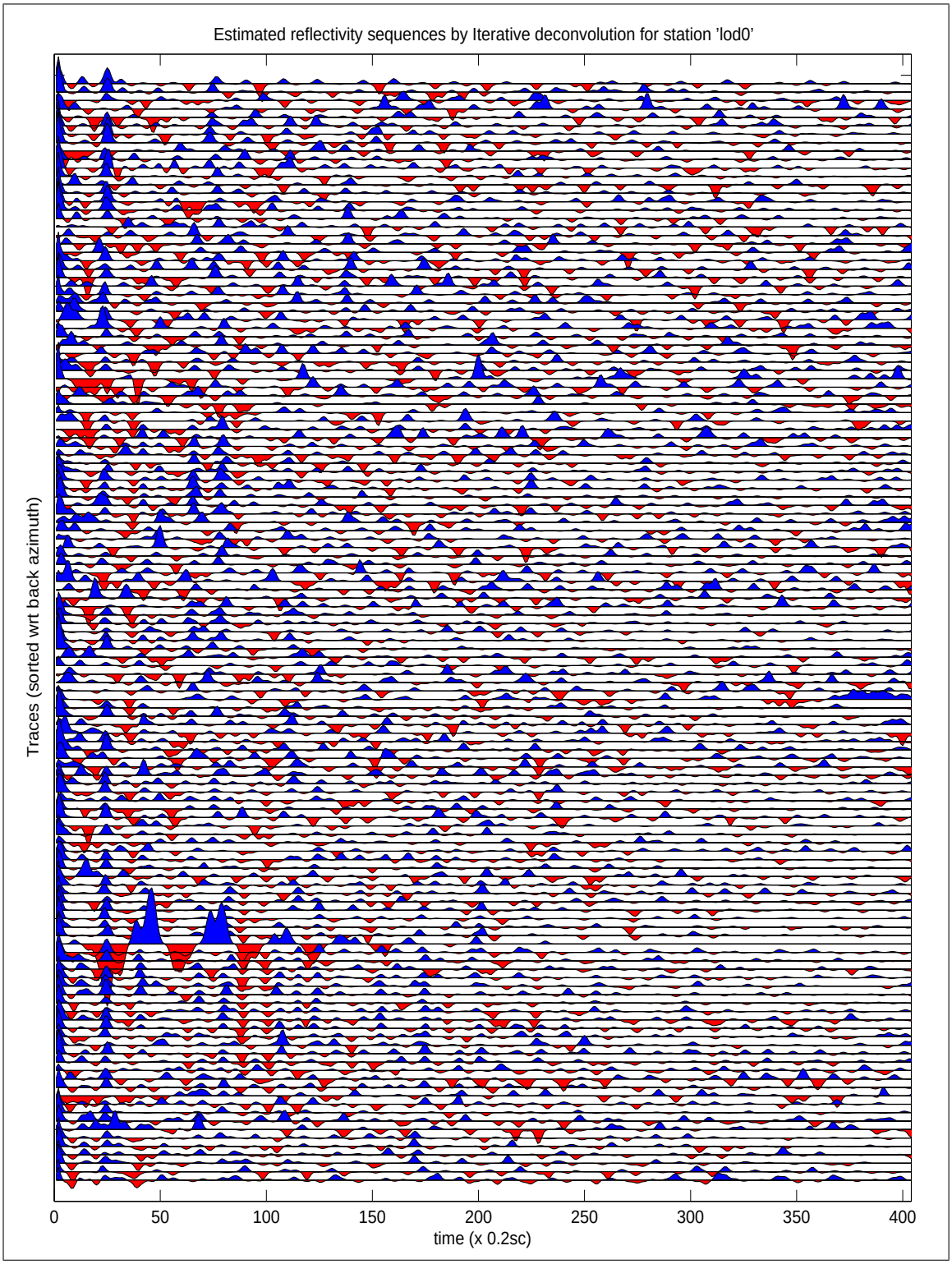


Figure 3.9. Deconvolution results for station 'lod0' by Iterative deconvolution method.

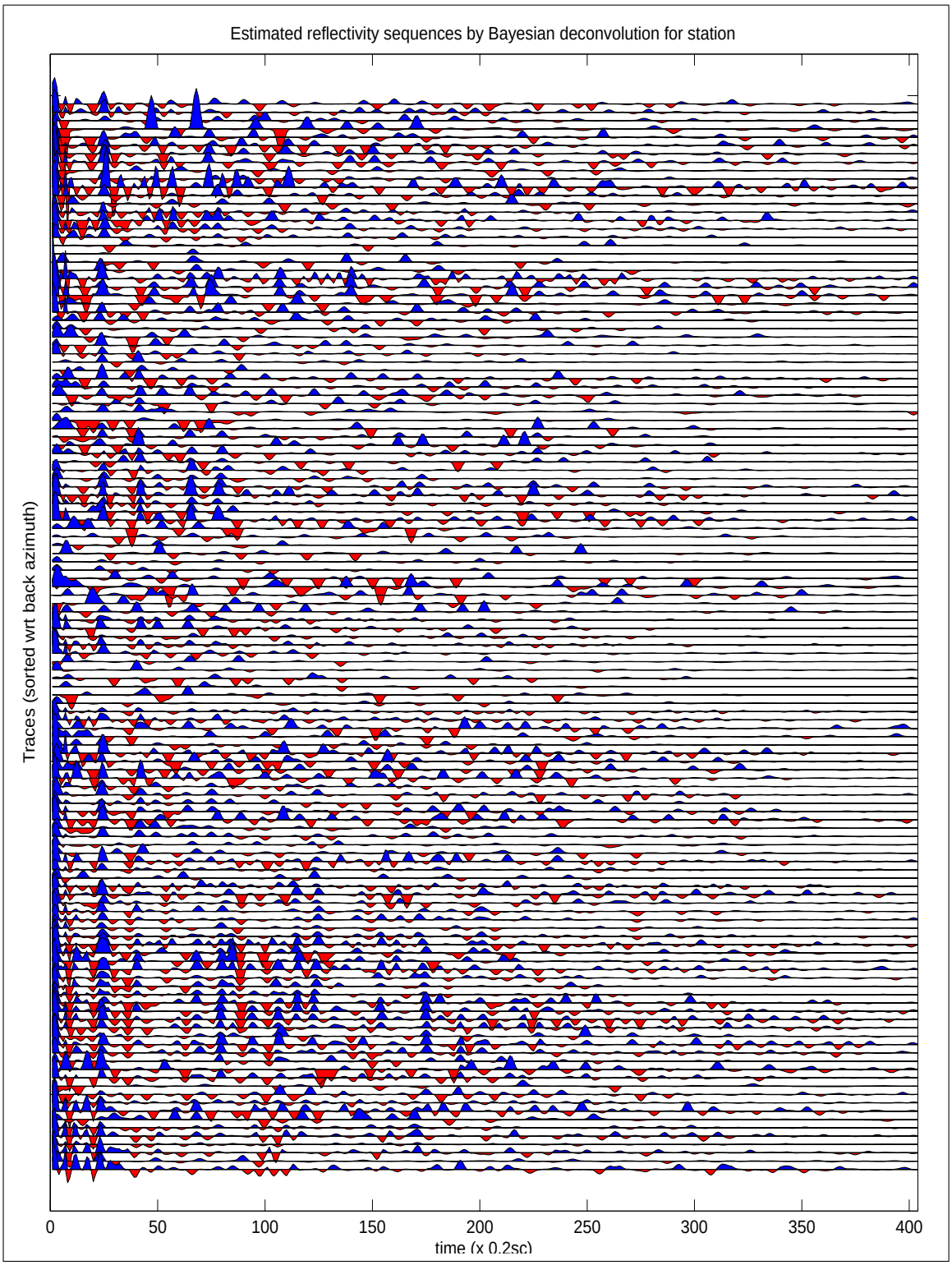


Figure 3.10. Deconvolution results for station 'lod0' by Bayesian deconvolution method.

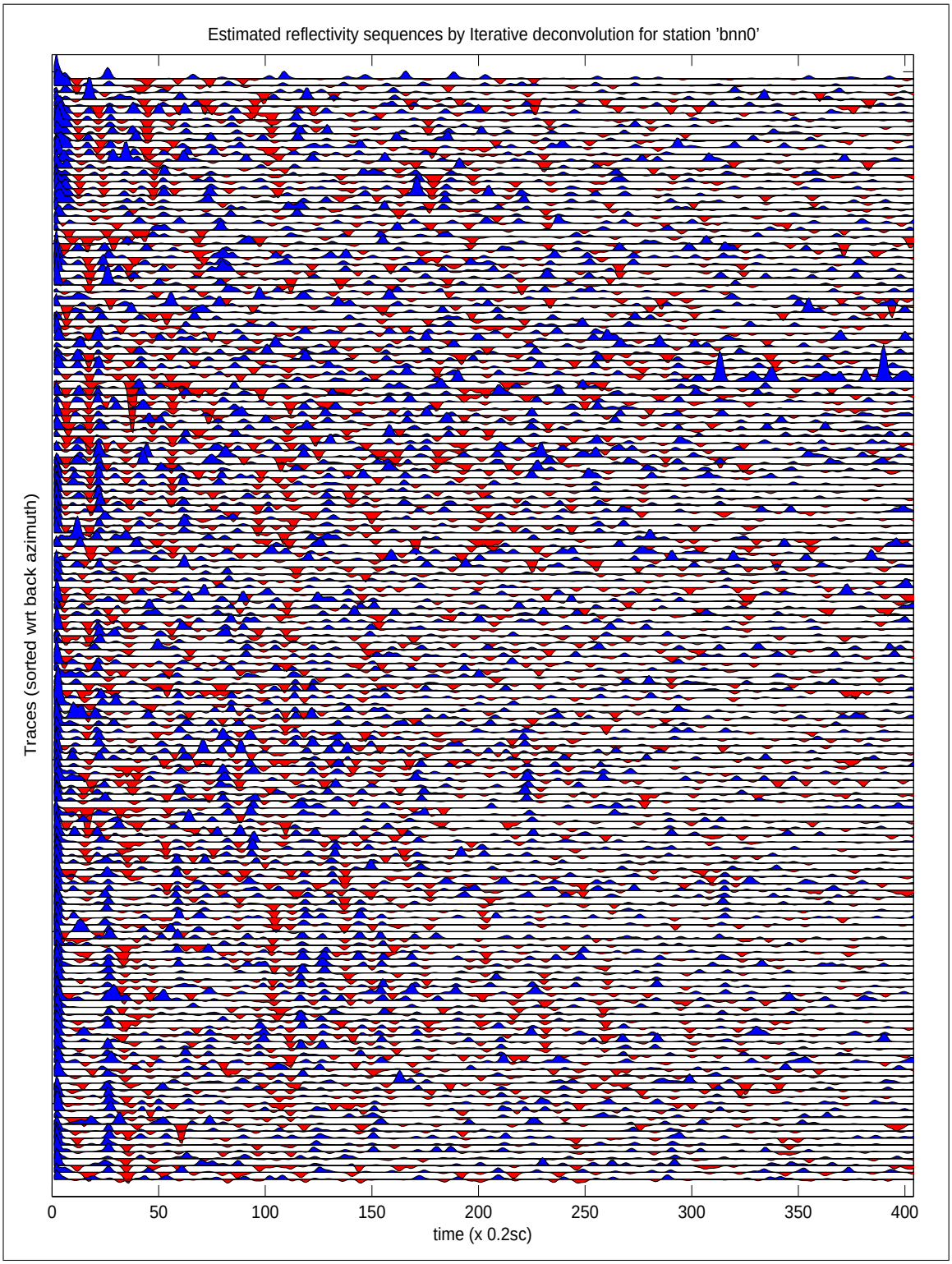


Figure 3.11. Deconvolution results for station 'bnn0' by Iterative deconvolution method.

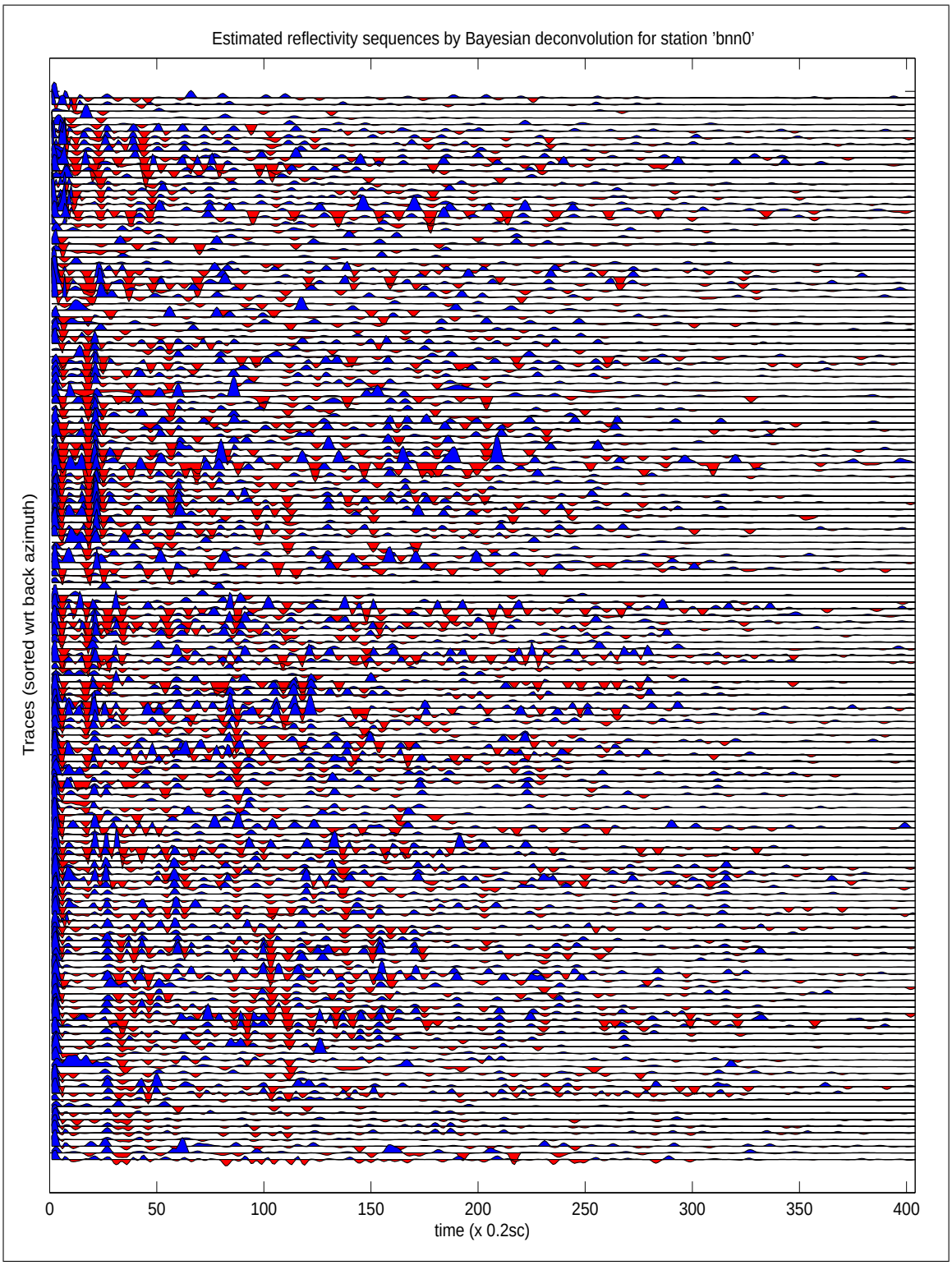


Figure 3.12. Deconvolution results for station 'bnn0' by Bayesian deconvolution method.

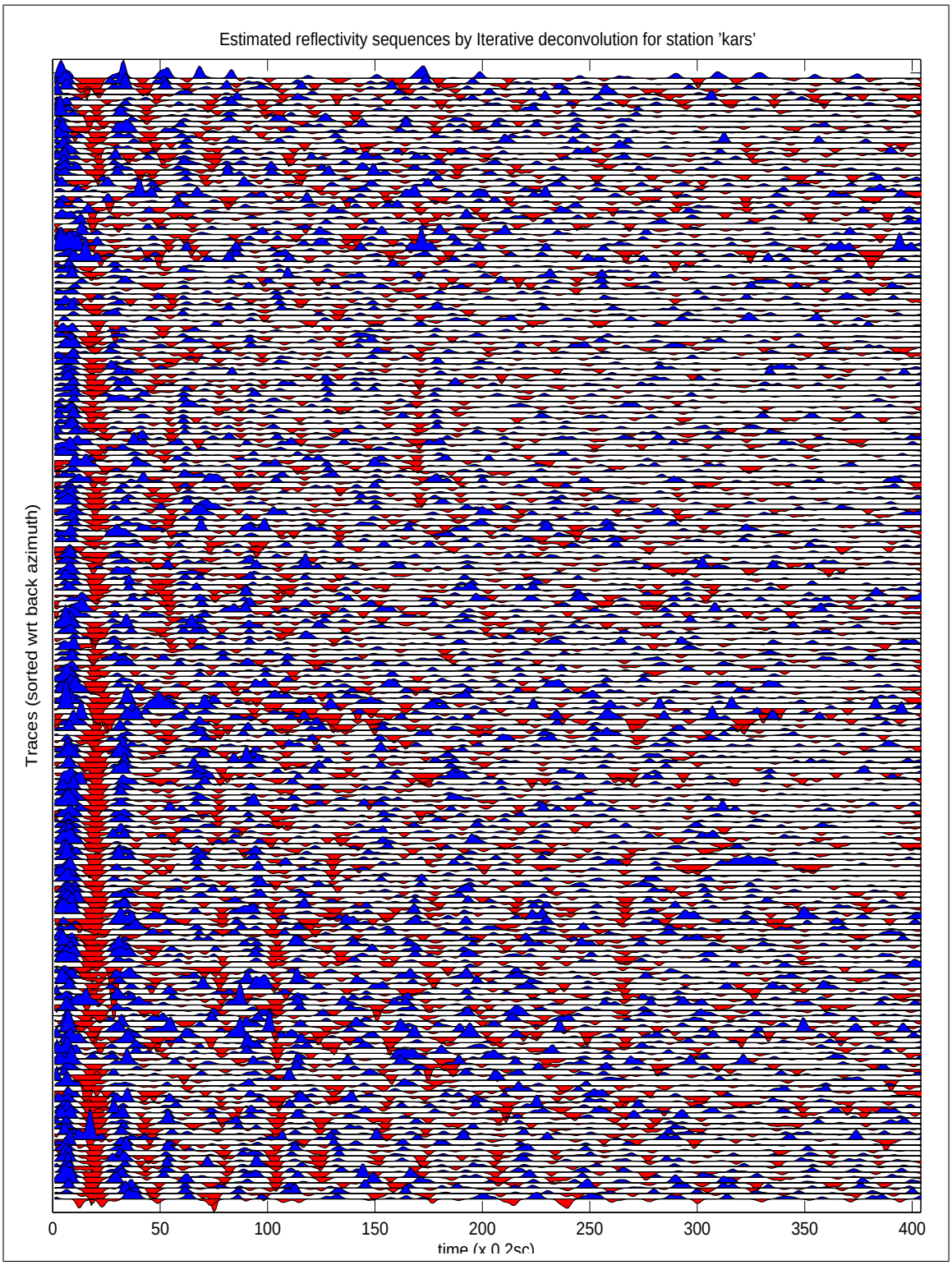


Figure 3.13. Deconvolution results for station 'kars' by Iterative deconvolution method.

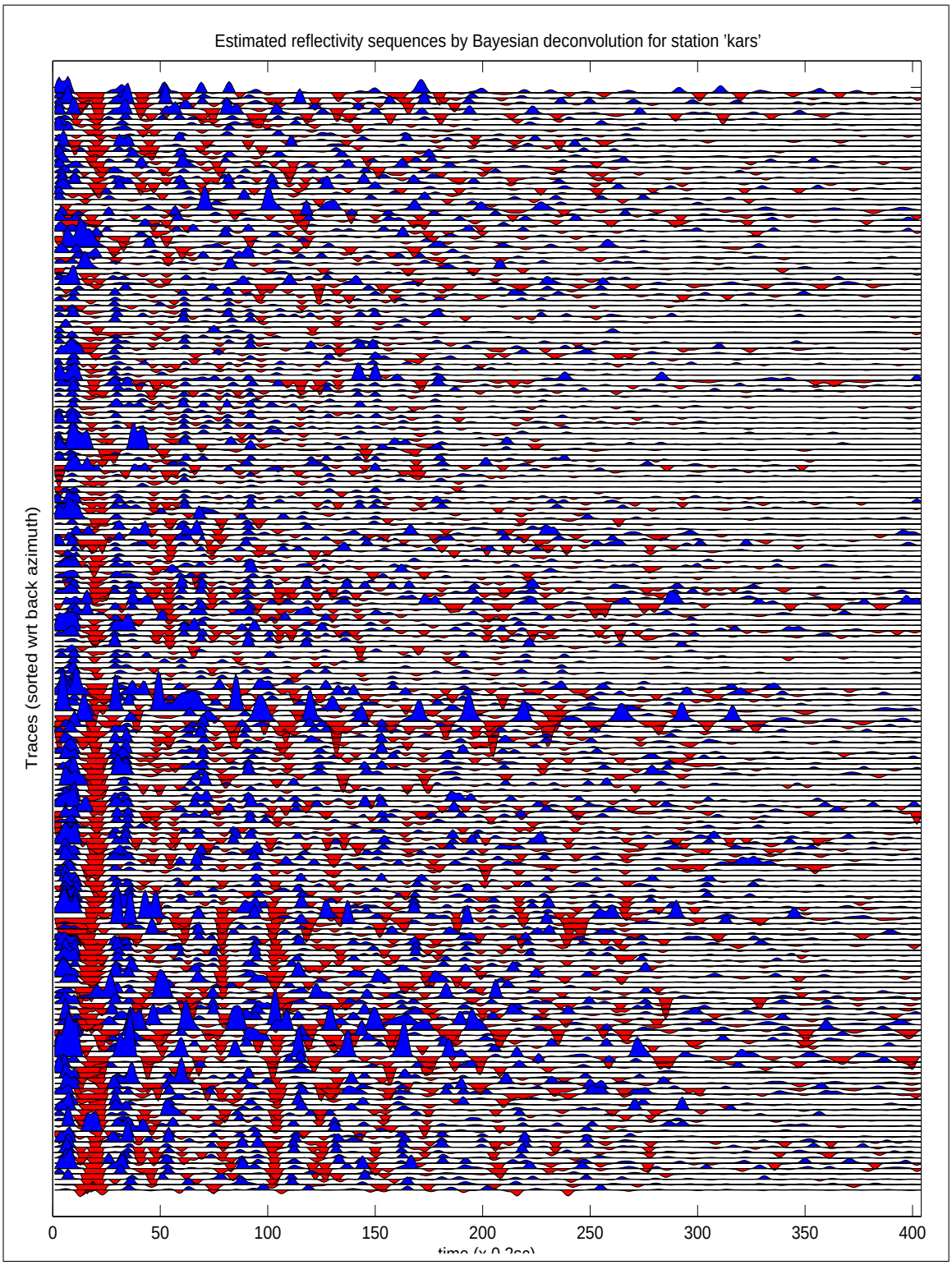


Figure 3.14. Deconvolution results for station 'kars' by Bayesian deconvolution method.

4. BLIND DECONVOLUTION OF SPARSE FILTERS WHEN SOURCE IS AUTOREGRESSIVE

In the previous chapter, we have dealt with the problems when the source signal is known, or it is unknown with no prior assumption. However, this fully parametric approach for the unknown source signal fails when the source signal is very long. In this case, instead of treating the source signal as a parameter vector which has no prior, assigning a statistical model to it may help us handle this difficulty. In this chapter, we introduce a *non-complete* solution to the problem of blindly estimating a sparse filter from a noisy convolution when the source is an autoregressive (AR) signal. This problem is of our interest, because the seismological source signals are known to be suitable for being modeled as AR processes. Unlike big earthquakes, the daily seismological records usually do not provide a reliable information for the source signal [23]. Therefore, the solution to this problem could be a crucial contribution to the seismological signal processing area. In this work, we will introduce Bayesian approaches for this blind deconvolution problem. It will be shown that the problem is a difficult one, and the performance strongly depends on the initial values, even if we know some of the system parameters, such as the AR parameters and the observation noise variance.

4.1. Noisy Convolution Model and Problem Statement

The convolution model is very similar to the one in (3.1):

$$y_t = \sum_{l=0}^{T_h-1} s_{t-l} r_l + v_t \qquad t = 1, 2, \ldots, T_y \tag{4.1}$$

Here, $\mathbf{s}$ is modeled by an autoregressive (AR) process of order M and length T_s, and the process is characterized by the AR coefficients $\rho = \begin{bmatrix} \rho_1 & \rho_2 & \ldots & \rho_M \end{bmatrix}^\top$, and the driving noise variance σ_w^2. r_l's are the coefficients of the filter, which is a low-pass filtered sparse waveform $\mathbf{r}$, and the filter is assumed to be the convolution of a sparse

waveform $\mathbf{h}$ and a known gaussian pulse $\mathbf{g} = \begin{bmatrix} g_{-d} & g_{-d+1} & \dots & g_{d-1} & g_d \end{bmatrix}^\top$, of length D, just as the case in the seismic deconvolution problem in section 3.3.2. We use the inverse-gamma model for the coefficients of $\mathbf{h}$, i.e.,

$$h_k \sim \mathcal{N}(h_k; 0, \sigma_{h,k}^2),\ \sigma_{h,k}^2 \sim \mathcal{IG}(\sigma_{h,k}^2; \alpha, \beta),\ k = 1, \dots, T_h.$$

Finally, v_k's are i.i.d. Gaussian with zero mean and variance σ_v^2.

Note that the convolution in (4.1) can be expressed in a matrix form either as $\mathbf{y} = \mathbf{H}\mathbf{G_s}\mathbf{s} + \mathbf{v}$ or as $\mathbf{y} = \mathbf{S}\mathbf{G_h}\mathbf{h} + \mathbf{v}$, where $\mathbf{H}$, $\mathbf{S}$, $\mathbf{G_s}$, and $\mathbf{G_h}$ are the suitable convolution matrices of sizes $T_y \times (T_s + D - 1)$, $T_y \times (T_h + D - 1)$, $(T_s + D - 1) \times T_s$, and $(T_h + D - 1) \times T_h$, respectively.

Matrix-vector representation of an AR process: Recall that any AR process can be expressed by a linear regression equation as

$$s_t = \sum_{k=1}^{M} s_{t-k}\rho_k + w_t,\ t = 1, 2, \dots, T_s, \tag{4.2}$$

where w_k's are the samples of the driving Gaussian noise. It can be shown from (4.2) that, the process can be written by a linear transformation of the noise vector obtained by gathering the noise samples in the order of time, and that the transformation matrix is constructed by using the AR coefficients ρ. That is, we can write

$$\mathbf{s} = \mathbf{R}\mathbf{w}, \tag{4.3}$$

where $\mathbf{s} = \begin{bmatrix} s_1 & s_2 & \dots & s_{T_s} \end{bmatrix}^\top$, $\mathbf{w} = \begin{bmatrix} w_1 & w_2 & \dots & w_{T_s} \end{bmatrix}^\top$, and the rows of the $T \times T$ matrix $\mathbf{R} = \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \dots & \mathbf{r}_{T_s} \end{bmatrix}^\top$, are constructed as follows:

$$\mathbf{r_1} = \begin{bmatrix} 1 & 0 & \dots & 0 \end{bmatrix},\ \mathbf{r_t} = \sum_{k=1}^{M} \rho_k \mathbf{r_{t-k}} + \mathbf{o_t},\ \mathbf{o_t}_i = \begin{cases} 0 & \text{if } i \neq t \\ 1 & \text{if } i = t \end{cases} \tag{4.4}$$

Thus, using (4.3) we can write that $p(\mathbf{s}) \sim \mathcal{N}(\mathbf{s}; \mathbf{0}, \sigma_w^2 \mathbf{R}\mathbf{R}^\top)$.

Blind Deconvolution Problem: In the blind deconvolution problem, we are given the observation $\mathbf{y}$; and we are required to estimate $\mathbf{r}$, the source signal $\mathbf{s}$, and the parameters σ_v^2, ρ, and σ_w^2.

4.2. Methodology

To solve our inference and parameter estimation problem, we use the EM type algorithms, as in chapter 3. In this problem, our parameter set is $\Theta = (\sigma_v^2, \alpha, \beta, \sigma_w^2, \rho)$ and the statistical latent variables are $(\mathbf{s}, \mathbf{h}, \boldsymbol{\sigma}_\mathbf{h}^\mathbf{2})$. Therefore, the EM steps are as follows:

- E step: Requires inference of the posterior to compute the EM quantity:

$$\Omega^{(i)} = \langle \log p(\mathbf{y}, \mathbf{s}, \mathbf{h}, \boldsymbol{\sigma}_\mathbf{h}^\mathbf{2} | \Theta) \rangle_{p(\mathbf{h}, \mathbf{s}, \boldsymbol{\sigma}_\mathbf{h}^\mathbf{2} | \mathbf{y}, \Theta^{(i-1)})}. \tag{4.5}$$

- M step: Parameter estimation: Maximize $\Omega^{(i)}$ over Θ:

$$\Theta^{(i)} = \arg\max_{\Theta} \Omega^{(i)}.$$

4.2.1. Inference for $(\mathbf{s}, \mathbf{h}, \boldsymbol{\sigma}_\mathbf{h}^\mathbf{2})$: Expectation

In the expectation step, we have to calculate the posterior distribution in (4.5). The unnormalized posterior ϕ can be expressed as:

$$\begin{aligned}
\log \phi &= \log p(\mathbf{h}, \boldsymbol{\sigma}_\mathbf{h}^\mathbf{2}, \mathbf{s}, \mathbf{y} | \Theta) = \log p(\boldsymbol{\sigma}_\mathbf{h}^\mathbf{2}) + \log p(\mathbf{h} | \boldsymbol{\sigma}_\mathbf{h}^\mathbf{2}) + \log p(\mathbf{s}) + \log p(\mathbf{y} | \mathbf{s}, \mathbf{h}) \\
&= \sum_{k=1}^{T_h} \left[\alpha \log \beta - (\alpha + 1.5) \log \sigma_{h,k}^2 - \frac{\beta + 0.5 h_k^2}{\sigma_{h,k}^2} - \log \Gamma(\alpha) \right] \\
&- 0.5 \Bigg[T_s \log \sigma_w^2 + \log \left| \mathbf{R}\mathbf{R}^\top \right| + (T_s + T_y + T_h) \log(2\pi) + T_y \log \sigma_v^2 \\
&+ \frac{1}{\sigma_v^2} \left(\mathbf{y}^\top \mathbf{y} - 2(\mathbf{S}\mathbf{G}_\mathbf{h}\mathbf{h})^\top \mathbf{y} + (\mathbf{S}\mathbf{G}_\mathbf{h}\mathbf{h})^\top \mathbf{S}\mathbf{G}_\mathbf{h}\mathbf{h} \right) + \frac{\mathbf{s}^\top \left(\mathbf{R}\mathbf{R}^\top \right)^{-1} \mathbf{s}}{\sigma_w^2} \Bigg]
\end{aligned} \tag{4.6}$$

It can be seen from (4.6) that the exact posterior can not be obtained exactly. Therefore, we have to apply methods for approximating it. Subsequently, we will show how Gibbs sampling and variational Bayes methods can be used to approximate the posterior.

4.2.1.1. Gibbs Sampling. As we have stated before, the Gibbs sampling algorithm suggests sampling from the full conditional posterior distributions in an iterative manner so that as time goes to infinity, the samples are guaranteed to be drawn from the exact posterior. The way to calculate the full conditional posterior for any of the latent variables is to write (4.6) as a proportionality in terms of that latent variable. By doing so, it can be verified that the sampling scheme is as follows:

$$\mathbf{s}^{(i+1)} \sim p(\mathbf{s}|\mathbf{y}, \mathbf{h}^{(i)}, \boldsymbol{\sigma_{\mathbf{h}}^{2}}^{(i)}) = \mathcal{N}(\mathbf{s}; \mathbf{m_s}^{(i+1)}, \boldsymbol{\Sigma_s}^{(i+1)})$$
$$\mathbf{h}^{(i+1)} \sim p(\mathbf{h}|\mathbf{y}, \mathbf{s}^{(i+1)}, \boldsymbol{\sigma_{\mathbf{h}}^{2}}^{(i)}) = \mathcal{N}(\mathbf{h}; \mathbf{m_h}^{(i+1)}, \boldsymbol{\Sigma_h}^{(i+1)})$$
$$\boldsymbol{\sigma_{\mathbf{h}}^{2}}^{(i+1)} \sim p(\boldsymbol{\sigma_{\mathbf{h}}^{2}}|\mathbf{y}\mathbf{h}^{(i+1)}, \mathbf{s}^{(i+1)}) = \prod_{k=1}^{T_h} \mathcal{IG}(\sigma_{h,k}^2; \alpha_{h,k}^{(i+1)}, \beta_{h,k}^{(i+1)})$$

where

$$\boldsymbol{\Sigma_s}^{(i+1)} = \mathbf{R}\left(\frac{\mathbf{I}}{\sigma_w^2} + \frac{\mathbf{R}^\top \mathbf{G_s}^\top \mathbf{H}^{\top(i)} \mathbf{H}^{(i)} \mathbf{G_s} \mathbf{R}}{\sigma_v^2}\right)^{-1} \mathbf{R}^\top,$$
$$\mathbf{m_s}^{(i+1)} = \frac{1}{\sigma_v^2} \boldsymbol{\Sigma_s}^{(i+1)} \mathbf{G_s}^\top \mathbf{H}^{\top(i)} \mathbf{y},$$
$$\boldsymbol{\Sigma_h}^{(i+1)} = \left(\boldsymbol{\Sigma_h}^{-1(i)} + \frac{\mathbf{G_h}^\top \mathbf{S}^{\top(i+1)} \mathbf{S}^{(i+1)} \mathbf{G_h}}{\sigma_v^2}\right)^{-1},$$
$$\mathbf{m_h}^{(i+1)} = \frac{1}{\sigma_v^2} \boldsymbol{\Sigma_h}^{(i+1)} \mathbf{G_h}^\top \mathbf{S}^{\top(i+1)} \mathbf{y},$$
$$\alpha_{h,k}^{(i+1)} = \alpha_h^{(i+1)} = \alpha + 0.5, \ \beta_{h,k}^{(i+1)} = \beta + 0.5 h_k^{2(i+1)}.$$

4.2.1.2. Variational Bayes. Recall that, in variational Bayes algorithm, we approximate the posterior with the factorized distribution:

$$Q = q(\mathbf{s}) q(\mathbf{h}) q(\boldsymbol{\sigma_{\mathbf{h}}^{2}}) \tag{4.7}$$

It can be found that the iterations between the factors in (4.7) can be found in a similar manner as in Gibbs sampling, as the following:

$$q(\mathbf{s})^{(i+1)} = \mathcal{N}(\mathbf{s}; \mathbf{m_s}^{(i+1)}, \mathbf{\Sigma_s}^{(i+1)})$$
$$q(\mathbf{h})^{(i+1)} = \mathcal{N}(\mathbf{h}; \mathbf{m_h}^{(i+1)}, \mathbf{\Sigma_h}^{(i+1)})$$
$$q(\sigma_\mathbf{h})^{\mathbf{2}(i+1)} = \prod_{k=1}^{T_h} \mathcal{IG}(\sigma_{h,k}^2; \alpha_{h,k}^{(i+1)}, \beta_{h,k}^{(i+1)})$$

where

$$\mathbf{\Sigma_s}^{(i+1)} = \mathbf{R}\left(\frac{\mathbf{I}}{\sigma_w^2} + \frac{\mathbf{R}^\top \mathbf{G_s}^\top \langle \mathbf{H}^\top \mathbf{H} \rangle_{q(\mathbf{h})^{(i)}} \mathbf{G_s} \mathbf{R}}{\sigma_v^2}\right)^{-1} \mathbf{R}^\top,$$
$$\mathbf{m_s}^{(i+1)} = \tfrac{1}{\sigma_v^2} \mathbf{\Sigma_s}^{(i+1)} \mathbf{G_s}^\top \langle \mathbf{H}^\top \rangle_{q(\mathbf{h})^{(i)}} \mathbf{y}$$
$$\mathbf{\Sigma_h}^{(i+1)} = \left(\langle \mathbf{\Sigma_h}^{-1} \rangle_{q(\sigma_\mathbf{h}^\mathbf{2})^{(n)}} + \frac{\mathbf{G_h}^\top \langle \mathbf{S}^\top \mathbf{S} \rangle_{q(\mathbf{s})^{(i+1)}} \mathbf{G_h}}{\sigma_v^2}\right)^{-1},$$
$$\mathbf{m_h}^{(i+1)} = \tfrac{1}{\sigma_v^2} \mathbf{\Sigma_h}^{(i+1)} \mathbf{G_h}^\top \langle \mathbf{S}^\top \rangle_{q(\mathbf{s})^{(i+1)}} \mathbf{y}$$
$$\alpha_{h,k} = \alpha_h = \alpha + 0.5, \qquad \beta_{h,k} = \beta + 0.5 \langle h_k^2 \rangle_{p(\mathbf{h})^{(i+1)}}$$

One can also calculate the variational bound $\mathcal{VB}_Q$ at every iteration. It can be verified that, the variational bound for $\log Z = \log p(\mathbf{y}|\Theta)$ is given below (see Appendix):

$$\mathcal{VB}_Q = \sum_{k=1}^{T_h} \log \frac{\beta^\alpha \Gamma(\alpha_h)}{\beta_{h,k}^{\alpha_h} \Gamma(\alpha)} - 0.5 \left(T_y \log(2\pi\sigma_v^2) - (T_s + T_h) - \log|\mathbf{\Sigma_h}\mathbf{\Sigma_s}|\right)$$
$$- 0.5 \left[\frac{1}{\sigma_v^2}\left(\mathbf{y}^\top \mathbf{y} - 2(\langle \mathbf{S} \rangle_{q(\mathbf{s})} \mathbf{G_h} \mathbf{m_h})^\top \mathbf{y} + tr\left[\mathbf{G_h}^\top \langle \mathbf{S}^\top \mathbf{S} \rangle_{q(\mathbf{s})} \mathbf{G_h} (\mathbf{m_h} \mathbf{m_h}^\top + \mathbf{\Sigma_h})\right]\right)\right]$$
$$- 0.5 \left[T_s \log \sigma_w^2 + \log\left|\mathbf{R}\mathbf{R}^\top\right| + tr\left[\frac{1}{\sigma_w^2}(\mathbf{R}\mathbf{R}^\top)^{-1}(\mathbf{m_s}\mathbf{m_s}^\top + \mathbf{\Sigma_s})\right]\right]$$

For the calculations above, we need the sufficient statistics of the convolution matrices $\mathbf{H}$ and $\mathbf{S}$, which are $\langle \mathbf{S} \rangle_{q(\mathbf{s})}$, $\langle \mathbf{H} \rangle_{q(\mathbf{h})}$, $\mathbf{X_s} = \langle \mathbf{S}^\top \mathbf{S} \rangle_{q(\mathbf{s})}$ and $\mathbf{X_h} = \langle \mathbf{H}^\top \mathbf{H} \rangle_{q(\mathbf{h})}$.

The first two are easy, so the second two are. For $\mathbf{X_s}$, we can write

$$\begin{aligned}\mathbf{X_s}_{i,j} &= \left\langle \sum_{k=1}^{T_y} \mathbf{S}_{i,k}^{\top}\mathbf{S}_{k,j} \right\rangle_{q(\mathbf{s})} = \left\langle \sum_{k=1}^{T_y} \mathbf{S}_{k,i}\mathbf{S}_{k,j} \right\rangle_{q(\mathbf{s})} = \left\langle \sum_{k=\max(i,j)}^{T_s+\min(i,j)-1} \mathbf{S}_{k,i}\mathbf{S}_{k,j} \right\rangle_{q(\mathbf{s})} \\ &= \left\langle \sum_{k=\max(i,j)}^{T_s+\min(i,j)-1} s_{k-i+1}s_{k-j+1} \right\rangle_{q(\mathbf{s})} = \sum_{k=\max(i,j)}^{T_s+\min(i,j)-1} \left\langle s_{k-i+1}s_{k-j+1} \right\rangle_{q(\mathbf{s})} \\ &= \sum_{k=\max(i,j)}^{T_s+\min(i,j)-1} \mathbf{m}_{\mathbf{s}(k-i+1)}\mathbf{m}_{\mathbf{s}(k-j+1)} + \mathbf{\Sigma}_{\mathbf{s}(k-i+1,k-j+1)} \qquad i,j = 1,\dots T_h.\end{aligned}$$

In a similar way, we can write

$$\mathbf{X_h}_{i,j} = \sum_{k=\max(i,j)}^{T_h+\min(i,j)-1} \mathbf{m}_{\mathbf{h}(k-i+1)}\mathbf{m}_{\mathbf{h}(k-j+1)} + \mathbf{\Sigma}_{\mathbf{h}(k-i+1,k-j+1)} \qquad i,j = 1,\dots T_s$$

4.2.1.3. A Modified Order of Iterations. We know that, when $\mathbf{s}$ is given, we can make a very good estimate for $\mathbf{h}$ after a few Gibbs or variational Bayes iterations between $\mathbf{h}$ and $\sigma_{\mathbf{h}}^{\mathbf{2}}$. But, in the classical version of these algorithm above, for a given $\mathbf{s}$, or $q(\mathbf{s})$ we allow only one cycle between them. We can modify the algorithms as follows: For a given sample of $\mathbf{s}$, or $q(\mathbf{s})$, iterate more than once between $\mathbf{h}$ and $\sigma_{\mathbf{h}}^{\mathbf{2}}$, to have better samples. This approach is equivalent to assuming $\mathbf{h}$ and $\sigma_{\mathbf{h}}^{\mathbf{2}}$ as a single variable, which is reasonable because the variances are used just to assist the samples of the sparse signal which are of our main interest. In this way, we have an *inner* Gibbs sampler as a sampling step of our main Gibbs sampling algorithm, or, for variational Bayes, we have an inner variational Bayes as a step of the main iteration. We refer to these methods as *modified Gibbs sampling* and *modified variational Bayes*, respectively.

4.2.2. Parameter Estimation for Θ: Maximization

Once we have calculated the posterior $p(\mathbf{h}, \mathbf{s}, \sigma_{\mathbf{h}}^{\mathbf{2}}|\mathbf{y}, \Theta^{(i-1)})$, we can calculate the EM quantity $\Omega^{(i)}$ taking the expectation of (4.6) under this posterior. It can be derived that, maximization of $\Omega^{(i)}$ over Θ gives us the following equations: It can be found

that, once we have found the factorized densities of interest, we can perform parameter estimation for σ_w^2, σ_v^2, and ρ as follows:

$$\hat{\sigma}_v^2 = \frac{1}{T_y}\left(\mathbf{y}^\top\mathbf{y} - 2\mathbf{y}^\top\langle\mathbf{H}\rangle_{q(\mathbf{h})}\mathbf{G}_\mathbf{s}\langle\mathbf{s}\rangle_{q(\mathbf{s})}\right) + tr\left(\mathbf{G}_\mathbf{s}{}^\top\langle\mathbf{H}^\top\mathbf{H}\rangle_{q(\mathbf{h})}\mathbf{G}_\mathbf{s}\langle\mathbf{s}\mathbf{s}^\top\rangle_{q(\mathbf{s})}\right),$$
$$\hat{\sigma}_w^2 = \frac{1}{T_s}tr\left((\mathbf{R}\mathbf{R}^\top)^{-1}\langle\mathbf{s}\mathbf{s}^\top\rangle_{q(\mathbf{s})}\right),$$
$$\hat{\rho} = \left(\langle\mathbf{S}_\rho{}^\top\mathbf{S}_\rho\rangle_{q(\mathbf{s})}\right)^{-1}\langle\mathbf{S}_\rho^\top\rangle_{p(\mathbf{s})}$$

where $\mathbf{S}_\rho$ is the truncated convolution matrix of $\mathbf{s}$ with size $T_s \times M$, and its corresponding sufficient statistics, $\langle\mathbf{S}_\rho\rangle_{p(\mathbf{s})}$ and $\mathbf{X}_\rho = \langle\mathbf{S}_\rho{}^\top\mathbf{S}_\rho\rangle_{q(\mathbf{s})}$ are found in a similar way with the ones of $\mathbf{H}$ and $\mathbf{S}$:

$$\mathbf{X}_{\rho_{i,j}} = \sum_{k=\max(i,j)}^{T_s+\min(i,j)-1} \mathbf{m}_{\mathbf{s}(k-i+1)}\mathbf{m}_{\mathbf{s}(k-j+1)} + \mathbf{\Sigma}_{\mathbf{s}(k-i+1,k-j+1)}, \qquad i,j = 1,\dots M$$

4.3. Experiments and Results

In the previous part we have analyzed the very generalized problem of statistical inference of $\mathbf{s}$ and $\mathbf{h}$ and estimation of deterministic system parameters; and then developed methods considering the case when $\mathbf{s}$, $\mathbf{h}$, and all of the parameters are unknown. However, it is known by experience that the system is highly multi-modal, and when everything is unknown, the offered methods can converge to completely different states from the true ones. Especially, the AR parameters are observed to be very hard to estimate and crucial for correctly estimating other unknown variables. For this reason, we limit our experiments for performance evaluations of the inference methods, i.e, Gibbs sampling, variational Bayes, and their modified versions, as described in section 4.2.1.3. Hence, we assume that the parameter set Θ is known. Even in this case, it will be apparent from the results that initialization is very important for reliable estimation of the processes $\mathbf{s}$ and $\mathbf{h}$ and that $\mathbf{s}$ is much harder to estimate compared to $\mathbf{h}$.

In the following experiment, we compared the performances of Gibbs sampling, variational Bayes, and their modified versions, as described in section 4.2.1.3. The

simulated data are generated using the following parameters:

- AR Source, sparse filter and observation lengths: $T_s = 100$, $T_h = 50$, $T_y = 140$;
- Length of the Gaussian pulse: $D = 5$;
- Observation noise variance $\sigma_v^2 = 0.1$;
- AR order and driving noise variance: $M = 5$, $\sigma_w^2 = 2$;
- The inverse-gamma parameters $\alpha = 0.5$, $\beta = 0.0001$

As we said above, we assume we know the parameters $\Theta = (\alpha, \beta, \sigma_v^2, \sigma_w^2, \rho)$, and we want to estimate $\mathbf{s}$ and $\mathbf{h}$. We tested our inference methods in cases of 'good' initial and 'bad' initial for $\mathbf{s}$. To assign a 'good' initial, we add a small noise to s such that the SNR is 5 dB. To assign a 'bad' initial, we just generate a random AR process with parameters ρ and σ_w^2. Number of iterations for Gibbs sampling and variational methods are taken 1000. The number of iterations for the inner Gibbs sampling or inner variational Bayes in the modified version of methods are set to 10. These number of iterations are chosen by trial and error.

The criteria we use for the performance of the methods are again the nMSE values in their mean estimates of $\mathbf{s}$ and $\mathbf{h}$, and the nMSE values are averaged over 50 simulations.

Table 4.1. Performances of the inference methods in case of good and bad initializations. nMSE-s/nMSE-h: nMSE for the estimates of $\mathbf{h}/\mathbf{s}$

Initialization	Inference method	nMSE-s	nMSE-h
Good Initial	Gibbs Sampling	0.9294 ± 0.7436	0.6694 ± 0.7230
	Modified Gibbs Sampling	0.1936 ± 0.1390	0.0554 ± 0.1928
	Variational Bayes	0.4719 ± 0.5446	0.1303 ± 0.1892
	Modified Variational Bayes	0.1913 ± 0.1110	0.0492 ± 0.2002
Bad Initial	Gibbs Sampling	2.3468 ± 1.0839	2.1624 ± 1.0020
	Modified Gibbs Sampling	2.1763 ± 1.1082	2.2514 ± 1.2193
	Variational Bayes	2.3829 ± 1.1988	2.1584 ± 1.1880
	Modified Variational Bayes	2.0468 ± 0.9905	2.4551 ± 1.2770

Table 4.1 shows the results of the methods in case of 'good' and 'bad' initializations of **s**. We can see from the results that the initialization is a crucial factor for successful estimations. When we initialize **s** randomly, we can not make correct estimations at all. Another point is that estimating **s** is much more challenging than estimating **h**. The nMSE in estimation of **s** in the best case is about 0.2, which is still not a very satisfactory result. On the other hand, error in estimation of **h** can be acceptably small. Another conclusion that can be derived from the results is that, when the estimate of **s** is initialized close to the true **s**, the modified versions of Gibbs sampling and variational Bayes outperform their general versions, as expected. However, in bad initialization, there is not a considerable difference. The last thing to say about the results is that the variational Bayes methods give slightly better results than the Gibbs sampling methods.

5. CONCLUSIONS AND DISCUSSIONS

In this work, we aimed to investigate the deconvolution and blind deconvolution problems in the Bayesian signal processing framework, where we use our prior knowledge about processes. We have particularly dealt with sparse processes, and our application was deconvolution of seismological signals. We tested our methods on simulated data as well as seismological data. In all the methods we developed, we have shown that, once we include our prior knowledge into the solution of our problem, both deconvolution and blind deconvolution operations can be mapped to inference and parameter estimation methods.

One of the contributions in this work is that we have proposed the well known inverse-gamma model for (blindly) deconvolving sparse signals, as alternative to the Bernoulli-Gaussian model. It has been shown by experimental evidence that the performance of the inverse-gamma model is much better than the Bernoulli-Gaussian model, even if the sparse process is a Bernoulli-Gaussian process. We claim that this is an expected result of a general fact: Search algorithms work better in smoother surfaces. Since we use EM, which searches for a maximum point over a likelihood surface, this fact is valid for our problem we solved in this thesis. In this line of thinking, it can be said that the inverse-gamma model has been proposed as a successful relaxation of the Bernoulli-Gaussian model for sparsity.

Another main result of our study is that; in deconvolution problem, if we have prior knowledge about the signal which is to be deconvolved, then it is advantageous to use Bayesian deconvolution rather than algorithmic deconvolution methods such as iterative deconvolution. In fact, this is a special result of a general idea: If you have prior knowledge, include it into the solution of your problem. We have verified the correctness of this idea by simulation experiments which compare the performances of iterative and Bayesian deconvolution methods. Furthermore, using Bayesian deconvolution on real seismological data, we have obtained receiver function estimations which have some superiorities over the results of iterative deconvolution, though more reli-

able comparison can be done after further seismological analysis of the results. This result is very promising for us, because it validates our sparsity assumption on receiver functions.

The promising results in deconvolution of seismological signal opens the door for further research on receiver function analysis in a Bayesian framework, such as blind deconvolution of receiver functions when the source is not known. This was indeed the further goal of our study in this work. However, the results of Chapter 4 reveal that blind deconvolution is not an easy task when the source signal is too long to be considered as a parameter vector, hence the methods in Chapter 3 are not applicable any more. Furthermore, the prior knowledge about the source, being an AR process, does not provide a significant improvement to the solution of our problem when we use classical inference and parameter estimation methods, such as Gibbs sampling, variational Bayes, EM, etc. Even if we know all of the AR parameters, the performance of the methods are strongly dependent on initialization. This difficulty is a sign of high multi-modality in the problem. Moreover, the scenario where we know the AR coefficients, AR order, AR driving noise variance etc. is not a realistic one. When those parameters are unknown as well, the problem becomes even harder. However, despite of all those difficulties, our work can be considered valuable as it is a first step to the solution of this hard problem.

5.1. Future Work

Even though we have shown that the inverse-gamma model is more successful in blind deconvolution of sparse processes, it would be valuable to test its performance in real life applications. In this concept, a possible future work is to apply the methods developed for inverse-gamma model to a real life problem, such as blind deconvolution of *marine seismic* data [3].

In receiver function analysis, we have stated that we expect to see vertical straight lines of pulses when the receiver functions are aligned in an up-down order with respect to their back azimuth values. Hence, it is easy to come up with the idea that, given a

set of receiver functions ordered with respect to their back azimuth values, successive receiver functions should look alike. In our future work, we aim to exploit this similarity by using our current receiver function estimate in the initialization of the next receiver function estimate.

In literature, receiver functions are estimated by using the records of big earthquakes. However, there is always an activation under the surface and these activations are recorded daily. Since their power are not as large as the ones of big earthquakes, the SNR in the observation is low, hence these daily records are not used for receiver function estimation. Due to the robustness of Bayesian deconvolution to noise, as a future work, we can apply Bayesian deconvolution on daily seismological data for receiver function estimation. We aim to do so in an early future time. If meaningful results are obtained, this will be a novel contribution to the area of receiver function analysis by means of reducing economical costs significantly and enabling the use of more data.

An even further step of this work would be not to trust the noisy vertical component in the daily records (recall that the vertical component is taken as the source), and seek for blindly estimating the receiver function from the observation. We have already started to deal with this problem in Chapter 4, and seen how it is challenging. To gain improvement in the solution of this problem, more sophisticated inference methods or more intelligent sampling schemes could be investigated. Moreover, ways of suitable initialization could be searched. For example, the receiver estimates obtained from big earthquakes could be used for reliable initializations, or the noisy vertical component could be assumed as the source in noise and thus inserted into the methods in that way.

APPENDIX A: CONJUGATE PRIORS

A.1. Likelihood is Gaussian, Prior is Gaussian

We have $p(\mathbf{x}) \sim \mathcal{N}(\mathbf{x}; \mathbf{m_x}, \mathbf{\Sigma_x})$, and $p(\mathbf{y}|\mathbf{x}) \sim \mathcal{N}(\mathbf{y}; \mathbf{Ax}, \mathbf{\Sigma_{y|x}})$, where

$$\mathcal{N}(\mathbf{z}; \mathbf{m}, \mathbf{\Sigma}) = \frac{1}{\sqrt{|2\pi\mathbf{\Sigma}|}} \exp\left[-0.5\,(\mathbf{z}-\mathbf{m})^\top\,\mathbf{\Sigma}^{-1}\,(\mathbf{z}-\mathbf{m})^\top\right]. \tag{A.1}$$

Note that the exponential in (A.1) is in a quadratic form:

$$\mathcal{N}(\mathbf{z}; \mathbf{m}, \mathbf{\Sigma}) = \frac{1}{\sqrt{|2\pi\mathbf{\Sigma}|}} \exp\left[-0.5\left(\mathbf{z}^\top\mathbf{\Sigma}^{-1}\mathbf{z} - 2\mathbf{z}^\top\mathbf{\Sigma}^{-1}\mathbf{m} + \mathbf{m}^\top\mathbf{\Sigma}^{-1}\mathbf{m}\right)\right]. \tag{A.2}$$

By the Bayes rule:

$$\begin{aligned}
p(\mathbf{x}|\mathbf{y}) &= \frac{p(\mathbf{x})p(\mathbf{y}|\mathbf{x})}{p(\mathbf{y})} \propto p(\mathbf{x})p(\mathbf{y}|\mathbf{x}) \\
&= \frac{1}{\sqrt{2\pi|\mathbf{\Sigma_x}|}} \exp\left[-0.5\,(\mathbf{x}-\mathbf{m_x})^\top\,\mathbf{\Sigma_x}^{-1}\,(\mathbf{x}-\mathbf{m_x})\right] \\
&\frac{1}{\sqrt{2\pi|\mathbf{\Sigma_{y|x}}|}} \exp\left[-0.5\,(\mathbf{y}-\mathbf{Ax})^\top\,\mathbf{\Sigma_{y|x}}^{-1}\,(\mathbf{y}-\mathbf{Ax})\right] \\
&= \frac{1}{2\pi\sqrt{|\mathbf{\Sigma_x}\mathbf{\Sigma_{y|x}}|}} \exp\left[-0.5\left(\mathbf{x}^\top\mathbf{\Sigma_x}^{-1}\mathbf{x} + \mathbf{x}^\top\mathbf{A}^\top\mathbf{\Sigma_{y|x}}^{-1}\mathbf{Ax}\right.\right. \\
&\left.\left.-2\mathbf{x}^\top\mathbf{\Sigma_x}^{-1}\mathbf{m_x} - 2\mathbf{x}^\top\mathbf{A}^\top\mathbf{\Sigma_{y|x}}^{-1}\mathbf{y} + \mathbf{m_x}^\top\mathbf{\Sigma_x}^{-1}\mathbf{m_x} + \mathbf{y}^\top\mathbf{\Sigma_{y|x}}^{-1}\mathbf{y}\right)\right] \\
&\propto \exp\left[-0.5\left(\mathbf{x}^\top\left(\mathbf{\Sigma_x}^{-1} + \mathbf{A}^\top\mathbf{\Sigma_{y|x}}^{-1}\mathbf{A}\right)\mathbf{x} - 2\mathbf{x}^\top\left(\mathbf{\Sigma_x}^{-1}\mathbf{m_x} + \mathbf{A}^\top\mathbf{\Sigma_{y|x}}\mathbf{y}\right)\right)\right]
\end{aligned} \tag{A.3}$$

where the proportionalities do not depend on $\mathbf{x}$. Since the expression in the expression (A.3) is in the same quadratic form with the one in a Gaussian distribution, we know that $p(\mathbf{x}|\mathbf{y})$ is a Gaussian with a mean $\mathbf{m_x}$ and a covariance $\mathbf{\Sigma_{x|y}}$. Using the analogy between (A.2) and (A.3), we can write $\mathbf{\Sigma_{x|y}} = \left(\mathbf{\Sigma_x}^{-1} + \mathbf{A}^\top\mathbf{\Sigma_{y|x}}^{-1}\mathbf{A}\right)^{-1}$, and

$\boldsymbol{\Sigma}_{\mathbf{x}|\mathbf{y}}{}^{-1}\mathbf{m}_{\mathbf{x}|\mathbf{y}} = \mathbf{A}^\top \boldsymbol{\Sigma}_{\mathbf{y}|\mathbf{x}}{}^{-1}\mathbf{y} + \boldsymbol{\Sigma}_{\mathbf{x}}{}^{-1}\mathbf{m}_{\mathbf{x}}$. Hence, we have

$$p(\mathbf{x}|\mathbf{y}) = \mathcal{N}(\mathbf{x}; \mathbf{m}_{\mathbf{x}|\mathbf{y}}, \boldsymbol{\Sigma}_{\mathbf{x}|\mathbf{y}}),$$

with $\boldsymbol{\Sigma}_{\mathbf{x}|\mathbf{y}} = \left(\boldsymbol{\Sigma}_{\mathbf{x}}{}^{-1} + \mathbf{A}^\top \boldsymbol{\Sigma}_{\mathbf{y}|\mathbf{x}}{}^{-1}\mathbf{A}\right)^{-1}$, and $\mathbf{m}_{\mathbf{x}|\mathbf{y}} = \boldsymbol{\Sigma}_{\mathbf{x}|\mathbf{y}}\left(\mathbf{A}^\top \boldsymbol{\Sigma}_{\mathbf{y}|\mathbf{x}}{}^{-1}\mathbf{y} + \boldsymbol{\Sigma}_{\mathbf{x}}{}^{-1}\mathbf{m}\right)$.

A.2. Likelihood is Gaussian, Prior is inverse-gamma:

We have $p(x) \sim \mathcal{IG}(x, \alpha, \beta)$ and $p(y|x) \sim \mathcal{N}(y; 0, x)$. where

$$\mathcal{IG}(x; \alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)}\left(\frac{1}{x}\right)^{\alpha+1}\exp\left(-\frac{\beta}{x}\right), \tag{A.4}$$

By the Bayes rule, we have

$$\begin{aligned} p(x|y) = \frac{p(x)p(y|x)}{p(y)} &\propto p(x)p(y|x) \\ &\frac{\beta^\alpha}{\Gamma(\alpha)}\left(\frac{1}{x}\right)^{\alpha+1}\exp\left(-\frac{\beta}{x}\right)\frac{1}{\sqrt{2\pi x}}\exp\left(\frac{-0.5y^2}{x}\right) \\ &\propto \left(\frac{1}{x}\right)^{\alpha+1.5}\exp\left(\frac{\beta + 0.5y^2}{x}\right) \end{aligned} \tag{A.5}$$

Since the expression above is in the form of an inverse-gamma distribution for x, and the proportionalities do not depend on x, $p(x|y)$ is an inverse-gamma distribution with a scale parameter α' and a scale parameter β'. By the analogy between (A.4) and (A.5) it can be shown that

$$p(x|y) \sim \mathcal{IG}(x, \alpha', \beta'). \text{ with } \alpha' = \alpha + 0.5, \beta' = \beta + 0.5y^2$$

APPENDIX B: ALGORITHMS FOR CGLSSM

We have the CGLSSM:

$$\mathbf{h}_{k+1} = \mathbf{A}\mathbf{h}_k + \mathbf{U}_{c_{k+1}} e_{k+1}$$
$$y_k = \mathbf{s}^\top \mathbf{h}_k + v_k$$

- $e_{k+1} \sim \mathcal{N}(e_{k+1}, 0, 1)$, $v_k \sim \mathcal{N}(v_k; 0, \sigma_v^2)$
- $\mathbf{h}_k = \begin{bmatrix} h_k & h_{k-1} & \dots & h_{k-T_s+1} \end{bmatrix}^\top$,
- $\mathbf{A}$ is the one-tap delay matrix of size $T_s \times T_s$,
- $\mathbf{U}_{c_k} = \begin{bmatrix} \sigma_{c_k} & 0 & \dots & 0 \end{bmatrix}^\top$, $c_k \in \{0, 1\}$

For the algorithms, we will use the following notation:

$$\hat{\mathbf{h}}_{i|j} = \langle \mathbf{h}_i \rangle_{p(\mathbf{h}_i|y_{1:j})} \text{ and } \mathbf{\Sigma}_{i|j} = \left\langle \left(\mathbf{h}_i - \hat{\mathbf{h}}_{i|j}\right)\left(\mathbf{h}_i - \hat{\mathbf{h}}_{i|j}\right)^\top \right\rangle_{p(\mathbf{h}_i|y_{1:j})}$$

B.1. Kalman Filter for the CGLSSM

- For an initial covariance matrix $\mathbf{\Sigma}_\nu$, set $\hat{\mathbf{h}}_{1|0} = 0$, $\mathbf{\Sigma}_{1|0} = \mathbf{\Sigma}_\nu$, and $l_0(\mathbf{y}) = 1$.
- Compute for $k = 1, 2, ..., T$

$\epsilon_k = y_k - \mathbf{s}^\top \hat{\mathbf{h}}_{k\|k-1}$	innovation
$\gamma_k = \mathbf{s}^\top \hat{\mathbf{h}}_{k\|k-1}\mathbf{s} + \sigma_v^2$	innovation variance
$l_k(\mathbf{y}) = l_{k-1}(\mathbf{y})\mathcal{N}(\epsilon_k; 0, \gamma_k)$	Likelihood of data
$\mathbf{K}_k = \mathbf{\Sigma}_{k\|k-1}\mathbf{s}/\gamma_k$	Kalman filtering gain
$\mathbf{P}_k = \mathbf{A}\mathbf{\Sigma}_{k\|k-1}\mathbf{s}/\gamma_k$	Kalman prediction gain
$\hat{\mathbf{h}}_{k\|k} = \hat{\mathbf{h}}_{k\|k-1} + \mathbf{K}_k\epsilon_k$	filtering state estimation
$\mathbf{\Sigma}_{k\|k} = \mathbf{\Sigma}_{k\|k-1} + \mathbf{K}_k\mathbf{s}\mathbf{\Sigma}_{k\|k-1}$	filtering error covariance
$\hat{\mathbf{h}}_{k+1\|k} = \mathbf{A}\hat{\mathbf{h}}_{k\|k}$	predicted state estimation
$\mathbf{\Sigma}_{k+1\|k} = \mathbf{A}\hat{\mathbf{h}}_{k\|k}\mathbf{A}^\top + \mathbf{U}_{c_k}\mathbf{U}_{c_k}^\top$	predicted error covariance

B.2. Disturbance Smoother for the CGLSSM

- Forward filtering: Run the Kalman filter in Appendix B.1 and store for $k = 1, \ldots, T$ the innovation ϵ_k, the inverse innovation variance γ_k^{-1}, the state prediction error covariance $\mathbf{\Sigma}_{k+1|k}$, and

$$\Lambda_k = \mathbf{A} - \mathbf{P}_k \mathbf{s}^\top$$

- Backward smoothing: For $k = T-1, \ldots, 1$, compute

$$p_k = \begin{cases} \mathbf{s}\epsilon_T/\gamma_T & \text{if } k = T-1 \\ \mathbf{s}\epsilon_{k+1}/\gamma_{k+1} + \Lambda_{k+1}^\top p_{k+1} & \text{otherwise,} \end{cases}$$

$$B_k = \begin{cases} \mathbf{s}\mathbf{s}^\top/\gamma_T & \text{if } k = T-1 \\ \mathbf{s}\mathbf{s}^\top/\gamma_{k+1} + \Lambda_{k+1}^\top B_{k+1}\Lambda_{k+1} & \text{otherwise,} \end{cases}$$

$$\hat{e}_{k|T} = \mathbf{U}_{c_k}^\top p_k$$

$$\varepsilon_{k|T} = 1 - \mathbf{U}_{c_k}^\top B_k \mathbf{U}_{c_k}$$

- Smoothed State Estimator: Compute

$$\hat{\mathbf{h}}_{1|T} = \mathbf{\Sigma}_\nu \left(\mathbf{s}\epsilon_1/\gamma_1 + \Lambda_1^\top p_1\right)$$

$$\mathbf{\Sigma}_{1|T} = \mathbf{\Sigma}_\nu - \mathbf{\Sigma}_\nu \left(\mathbf{s}\mathbf{s}^\top/\gamma_1 + \Lambda_1^\top B_1 \Lambda_1\right) \mathbf{\Sigma}_\nu$$

- Smoothed State Estimator: For $k = 1, \ldots, T-1$

$$\hat{\mathbf{h}}_{k+1|T} = \mathbf{A}\hat{\mathbf{h}}_{k|T} + \mathbf{U}_{c_t}\hat{e}_{k|T}$$

$$\mathbf{D} = \mathbf{A}\mathbf{\Sigma}_{k|k-1}\Lambda_k^\top B_k \mathbf{U}_{c_k}\mathbf{U}_{c_k}^\top$$

$$\mathbf{\Sigma}_{k+1|T} = \mathbf{A}\mathbf{\Sigma}_{k|T}\mathbf{A}^\top + \varepsilon_{k|T}\mathbf{U}_{c_k}\mathbf{U}_{c_k}^\top - \mathbf{D} - \mathbf{D}^\top$$

(Modified from the algorithm in [50])

B.3. MCMC 1 Gibbs Sampling for CGLSSM

Assuming that a current simulated sequence of indicators $\mathbf{c}^i$ is available, draw $\mathbf{c}^{i+1}$ as follows:

- Backward Recursion: Set $\kappa_{T|T} = 0$ and $\Pi_{T|T} = 0$. For $k = T, \ldots, 1$

$$\begin{aligned}
\hat{\kappa}_{k+1|T} &= \mathbf{s}y_{k+1}/\sigma_v^2 + \kappa_{k+1|T} \\
\hat{\Pi_{k+1|T}} &= \mathbf{s}\mathbf{s}^\top/\sigma_v^2 + \Pi_{k+1|T} \\
\kappa_{k|T} &= \mathbf{A}^\top \left(\mathbf{I}_{T_s} + \hat{\Pi_{k+1|T}}\mathbf{U}_{c_k^i}\mathbf{U}_{c_k^i}^\top\right)^{-1} \hat{\kappa_{k+1|T}} \\
\Pi_{k|T} &= \mathbf{A}^\top \left(\mathbf{I}_{T_s} + \hat{\Pi_{k+1|T}}\mathbf{U}_{c_k^i}\mathbf{U}_{c_k^i}^\top\right)^{-1} \hat{\Pi_{k+1|T}}\mathbf{A}
\end{aligned}$$

- Initial State: For q = 0,1, compute

$$\begin{aligned}
\epsilon_1 &= y_1 \\
\gamma_1(q) &= \mathbf{s}^\top \mathbf{\Sigma}_\nu(q)\mathbf{s} + \sigma_v^2 \\
\hat{\mathbf{h}}_{1|1}(q) &= \mathbf{\Sigma}_\nu(q)\mathbf{s}\epsilon_1/\gamma_1(q) \\
\mathbf{\Sigma}_{1|1}(q) &= \mathbf{\Sigma}_\nu(q) - \mathbf{\Sigma}_\nu(q)\mathbf{s}\mathbf{s}^\top\mathbf{\Sigma}_\nu(q)/\gamma_1(q) \\
l_1(q) &= -\left[\log\gamma_1(q) + \epsilon_1^2/\gamma_1(q)\right]/2 \\
\hat{\mathbf{h}}_{1|T}(q) &= \hat{\mathbf{h}}_{1|1}(q) + \mathbf{\Sigma}_{1|1}(q)\left(\mathbf{I} + \Pi_{1|T}\mathbf{\Sigma}_{1|1}(q)\right)^{-1}\left(\kappa_{1|T} - \Pi_{1|T}\hat{\mathbf{h}}_{1|1}(q)\right) \\
\mathbf{\Sigma}_{1|T}(q) &= \mathbf{\Sigma}_{1|1}(q) - \mathbf{\Sigma}_{1|1}(q)\left(\mathbf{I} + \Pi_{1|T}\mathbf{\Sigma}_{1|1}(q)\right)^{-1}\mathbf{\Sigma}_{1|1}(q) \\
m_1(q) &= -\left[\log|\mathbf{\Sigma}_{1|1}(q)| + \hat{\mathbf{h}}_{1|1}(q)^\top\mathbf{\Sigma}_{1|1}(q)^{-1}\hat{\mathbf{h}}_{1|1}(q)\right]/2 \\
&\quad + \left[\log|\mathbf{\Sigma}_{1|T}(q)| + \hat{\mathbf{h}}_{1|T}(q)^\top\mathbf{\Sigma}_{1|T}(q)^{-1}\hat{\mathbf{h}}_{1|T}(q)\right]/2 \\
p_1(q) &= \exp\left[l_1(q) + m_1(q)\right] Pr(c_1 = q)
\end{aligned}$$

Normalize the vector p_1 by $\bar{p}_1(q) = p_1(q)/(p_1(0) + p_1(1))$, and sample c_1^{i+1} from the probability distribution $\bar{p}_1$. Then store $\hat{\mathbf{h}}_{1|1}(c_1^{i+1})$ and $\mathbf{\Sigma}_{1|1}(c_1^{i+1})$ for the next iteration.

- For $k = 2, \ldots, T$, and $q = 0, 1$, compute

$$\hat{\mathbf{h}}_{k|k-1}(q) = \mathbf{A}\hat{\mathbf{h}}_{k-1|k-1}(c_{k-1}^{i+1})$$
$$\mathbf{\Sigma}_{k|k-1}(q) = \mathbf{A}\mathbf{\Sigma}_{k-1|k-1}(c_{k-1}^{i+1})\mathbf{A} + \mathbf{U}_q\mathbf{U}_q^\top$$
$$\epsilon_k = y_k - \mathbf{s}\hat{\mathbf{h}}_{k|k-1}(q)$$
$$\gamma_k(q) = \mathbf{s}^\top \Sigma_{k|k-1}(q)\mathbf{s} + \sigma_v^2$$
$$\hat{\mathbf{h}}_{k|k}(q) = \hat{\mathbf{h}}_{k|k-1}(q) + \mathbf{\Sigma}_{k|k-1}(q)\mathbf{s}\epsilon_k/\gamma_k(q)$$
$$\mathbf{\Sigma}_{k|k}(q) = \mathbf{\Sigma}_{k|k-1}(q) - \mathbf{\Sigma}_{k|k-1}(q)\mathbf{s}\mathbf{s}^\top\mathbf{\Sigma}_{k|k-1}(q)/\gamma_k(q)$$
$$l_k(q) = -\left[\log\gamma_k(q) + \epsilon_k^2/\gamma_k(q)\right]/2$$
$$\hat{\mathbf{h}}_{k|T}(q) = \hat{\mathbf{h}}_{k|k}(q) + \mathbf{\Sigma}_{k|k}(q)\left(\mathbf{I} + \Pi_{k|T}\mathbf{\Sigma}_{k|k}(q)\right)^{-1}\left(\kappa_{k|T} - \Pi_{k|T}\hat{\mathbf{h}}_{k|k}(q)\right)$$
$$\mathbf{\Sigma}_{k|T}(q) = \mathbf{\Sigma}_{k|k}(q) - \mathbf{\Sigma}_{k|k}(q)\left(\mathbf{I} + \Pi_{k|T}\mathbf{\Sigma}_{k|k}(q)\right)^{-1}\mathbf{\Sigma}_{k|k}(q)$$
$$m_k(q) = -\left[\log|\mathbf{\Sigma}_{k|k}(q)| + \hat{\mathbf{h}}_{k|k}(q)^\top\mathbf{\Sigma}_{k|k}(q)^{-1}\hat{\mathbf{h}}_{k|k}(q)\right]/2 + \left[\log|\mathbf{\Sigma}_{k|T}(q)| + \hat{\mathbf{h}}_{k|T}(q)^\top\mathbf{\Sigma}_{k|T}(q)^{-1}\hat{\mathbf{h}}_{k|T}(q)\right]/2$$
$$p_k(q) = \exp\left[l_k(q) + m_k(q)\right]Pr(c_k = q)$$

Normalize the vector p_k by $\bar{p}_k(q) = p_k(q)/(p_k(0) + p_k(1))$, and sample c_k^{i+1} from the probability distribution $\bar{p}_k$. Then store $\hat{\mathbf{h}}_{k|k}(c_k^{i+1})$ and $\mathbf{\Sigma}_{k|k}(c_k^{i+1})$ for the next iteration.

(Modified from the algorithm in [50])

APPENDIX C: VARIATIONAL BOUNDS

C.1. Variational Bound in Chapter 3

$$
\begin{aligned}
\log \phi = \log p(\mathbf{h}, \boldsymbol{\sigma}_{\mathbf{h}}^{\mathbf{2}}, \mathbf{y}|\Theta) &= \log p(\boldsymbol{\sigma}_{\mathbf{h}}^{\mathbf{2}}) p(\mathbf{h}|\boldsymbol{\sigma}_{\mathbf{h}}^{\mathbf{2}}) p(\mathbf{y}|\mathbf{h}) \\
&= \sum_{k=1}^{T} \left[\alpha \log \beta - (\alpha + 1) \log \sigma_{h,k}^2 - \frac{\beta}{\sigma_{h,k}^2} - \log \Gamma(\alpha) \right] \\
&- 0.5 \left[T \log(2\pi) + \sum_{i=1}^{T} \log \sigma_{h,k}^2 + \mathbf{h}^\top {\boldsymbol{\Sigma}_{\mathbf{h}}}^{-1} \mathbf{h} \right] \\
&- 0.5 \left[T_y \log(2\pi) + T_y \log \sigma_v^2 + \frac{1}{\sigma_v^2} (\mathbf{y} - \mathbf{S}\mathbf{h})^\top (\mathbf{y} - \mathbf{S}\mathbf{h}) \right] \\
&= \sum_{k=1}^{T} \left[\alpha \log \beta - (\alpha + 1.5) \log \sigma_{h,k}^2 - \frac{\beta}{\sigma_{h,k}^2} - \log \Gamma(\alpha) \right] \\
&- 0.5 \left[(T_y + T) \log(2\pi) + T_y \log \sigma_v^2 \right] \\
&- 0.5 \left[\mathbf{h}^\top {\boldsymbol{\Sigma}_{\mathbf{h}}}^{-1} \mathbf{h} + \frac{1}{\sigma_v^2} \left[\mathbf{y}^\top \mathbf{y} - 2\mathbf{h}^\top \mathbf{S}^\top \mathbf{y} + \mathbf{h}^\top \mathbf{S}^\top \mathbf{S} \mathbf{h} \right] \right]
\end{aligned}
$$

We approximate the posterior with the factorized distribution: $Q = q(\mathbf{h}) q(\boldsymbol{\sigma}_{\mathbf{h}}^{\mathbf{2}})$ with $q(\mathbf{h}) = \mathcal{N}(\mathbf{h}; \mathbf{m}_{\mathbf{h}}, \boldsymbol{\Sigma}_{\mathbf{h}})$, and $q(\boldsymbol{\sigma}_{\mathbf{h}}^{\mathbf{2}}) = \prod_{k=1}^{T} \mathcal{IG}(\sigma_{h,k}^2; \alpha_{h,k}, \beta_{h,k})$, where

$$
\begin{aligned}
\boldsymbol{\Sigma}_{\mathbf{h}} = \left(\langle {\boldsymbol{\Sigma}_{\mathbf{h}}}^{-1} \rangle_{q(\boldsymbol{\sigma}_{\mathbf{h}}^{\mathbf{2}})} + \tfrac{1}{\sigma_v^2} \mathbf{S}^\top \mathbf{S} \right)^{-1}, \ \mathbf{m}_{\mathbf{h}} &= \tfrac{1}{\sigma_v^2} \boldsymbol{\Sigma}_{\mathbf{h}} \langle \mathbf{S}^\top \rangle_{q(\mathbf{s})} \mathbf{y} \\
\alpha_{h,k} = \alpha_h = \alpha + 0.5, \ \beta_{h,k} &= \beta + 0.5 \langle h_k^2 \rangle_{p(\mathbf{h})}
\end{aligned} \tag{C.1}
$$

$$
\mathcal{B}_{VB} = \langle \log \phi \rangle_Q + H[Q] \tag{C.2}
$$

$$
\begin{aligned}
\langle \log \phi \rangle_Q = & \sum_{k=1}^{T} \left[\alpha \log \beta + (\alpha + 1.5) \left\langle \log \frac{1}{\sigma_{h,k}^2} \right\rangle_{q(\sigma_{\mathbf{h}}^{\mathbf{2}})} - \beta \left\langle \frac{1}{\sigma_{h,k}^2} \right\rangle_{q(\sigma_{\mathbf{h}}^{\mathbf{2}})} - \log \Gamma(\alpha) \right] \\
& - 0.5 \left[(T_y + T) \log(2\pi) + T_y \log \sigma_v^2 + \frac{1}{\sigma_v^2} \mathbf{y}^\top \mathbf{y} \right] \\
& - 0.5 \left[\sum_{k=1}^{T} \left\langle h_k^2 \right\rangle_{q(\mathbf{h})} \left\langle \frac{1}{\sigma_h^2} \right\rangle_{q(\sigma_{\mathbf{h}}^{\mathbf{2}})} - \frac{2}{\sigma_v^2} \langle \mathbf{h} \rangle_{q(\mathbf{h})}^\top \mathbf{S}^\top \mathbf{y} + tr \left[\frac{1}{\sigma_v^2} \mathbf{S}^\top \mathbf{S} \langle \mathbf{h}\mathbf{h}^\top \rangle_{q(\mathbf{h})} \right] \right] \\
& = \sum_{k=1}^{T} [\alpha \log \beta + (\alpha + 1.5)(\log \beta_{h,k} - \Psi(\alpha_{h,k})) \\
& - \left(\beta + 0.5 \left\langle h_k^2 \right\rangle_{q(\mathbf{h})} \right) \frac{\alpha_{h,k}}{\beta_{h,k}} - \log \Gamma(\alpha) \Big] \\
& - 0.5 \left[(T_y + T) \log(2\pi) + T_y \log \sigma_v^2 + \frac{1}{\sigma_v^2} \mathbf{y}^\top \mathbf{y} \right] \\
& - 0.5 \left[-\frac{2}{\sigma_v^2} \mathbf{m_h}^\top \mathbf{S}^\top \mathbf{y} + tr \left[\frac{1}{\sigma_v^2} \mathbf{S}^\top \mathbf{S} (\mathbf{m_h} \mathbf{m_h}^\top + \mathbf{\Sigma_h}) \right] \right] \qquad \text{(C.3)}
\end{aligned}
$$

$$
\begin{aligned}
H[Q] = -\langle \log Q \rangle_Q = & \, 0.5 \log \left[(2\pi e)^T |\mathbf{\Sigma_h}| \right] \\
& - \sum_{k=1}^{T} \left[\alpha_{h,k} \log \beta_{h,k} + (\alpha_{h,k} + 1) \left\langle \log \frac{1}{\sigma_{h,k}^2} \right\rangle_{q(\sigma_{\mathbf{h}}^{\mathbf{2}})} - \beta_{h,k} \left\langle \frac{1}{\sigma_{h,k}^2} \right\rangle_{q(\sigma_{\mathbf{h}}^{\mathbf{2}})} - \log \Gamma(\alpha_{h,k}) \right] \\
& = 0.5 \log \left[(2\pi e)^T |\mathbf{\Sigma_h}| \right] \\
& - \sum_{k=1}^{T} [\alpha_{h,k} \log \beta_{h,k} + (\alpha_{h,k} + 1)(\log \beta_{h,k} - \Psi(\alpha_{h,k})) - \alpha_{h,k} - \log \Gamma(\alpha_{h,k})] \qquad \text{(C.4)}
\end{aligned}
$$

Substituting (C.3) and (C.4) into (C.2), we get

$$
\begin{aligned}
\mathcal{B}_{VB} = & \sum_{k=1}^{T}\left[\alpha\log\beta+(\alpha+1.5)(\log\beta_{h,k}-\Psi(\alpha_{h,k}))-(\beta+0.5\left\langle h_k^2\right\rangle_{q(\mathbf{h})})\frac{\alpha_{h,k}}{\beta_{h,k}}-\log\Gamma(\alpha)\right] \\
& -0.5\left[(T_y+T)\log(2\pi)+T_y\log\sigma_v^2+\frac{1}{\sigma_v^2}\mathbf{y}^\top\mathbf{y}\right] \\
& -0.5\left[-\frac{2}{\sigma_v^2}\mathbf{m_h}^\top\mathbf{S}^\top\mathbf{y}+tr\left[\frac{1}{\sigma_v^2}\mathbf{S}^\top\mathbf{S}(\mathbf{m_h}\mathbf{m_h}^\top+\mathbf{\Sigma_h})\right]\right] \\
& +0.5\log\left[(2\pi e)^T|\mathbf{\Sigma_h}|\right] \\
& -\sum_{k=1}^{T}\left[\alpha_{h,k}\log\beta_{h,k}+(\alpha_{h,k}+1)(\log\beta_{h,k}-\Psi(\alpha_{h,k}))-\alpha_{h,k}-\log\Gamma(\alpha_{h,k})\right] \\
& =\sum_{k=1}^{T}\left[\alpha\log\beta-\alpha_{h,k}\log\beta_{h,k}+(\alpha+0.5-\alpha_{h,k})(\log\beta_{h,k}-\Psi(\alpha_{h,k}))\right. \\
& \left.-\alpha_{h,k}\left(\frac{\beta+0.5\left\langle h_k^2\right\rangle_{q(\mathbf{h})}}{\beta_{h,k}}-1\right)-\log\frac{\Gamma(\alpha)}{\Gamma(\alpha_{h,k})}\right] \\
& -0.5\left[T_y\log(2\pi)+T_y\log\sigma_v^2-T-\log\left[|\mathbf{\Sigma_h}|\right]\right] \\
& -0.5\left[\frac{1}{\sigma_v^2}\left(\mathbf{y}^\top\mathbf{y}-2\mathbf{m_h}^\top\mathbf{S}^\top\mathbf{y}+tr\left[\mathbf{S}^\top\mathbf{S}(\mathbf{m_h}\mathbf{m_h}^\top+\mathbf{\Sigma_h})\right]\right)\right] \qquad \text{(C.5)}
\end{aligned}
$$

Substituting the updates for $\alpha_{h,k}$ and $\beta_{h,k}$ in (C.1) into (C.5), we get a the final expression:

$$
\begin{aligned}
\mathcal{VB}_Q = & \sum_{k=1}^{T}\log\frac{\beta^\alpha\Gamma(\alpha_h)}{\beta_{h,k}^{\alpha_h}\Gamma(\alpha)}-0.5\left(T_y\log(2\pi\sigma_v^2)-T-\log|\mathbf{\Sigma_h}|\right) \\
& -0.5\left[\frac{1}{\sigma_v^2}\left(\mathbf{y}^\top\mathbf{y}-2\mathbf{m_h}^\top\mathbf{S}^\top\mathbf{y}+tr\left[\mathbf{S}^\top\mathbf{S}(\mathbf{m_h}\mathbf{m_h}^\top+\mathbf{\Sigma_h})\right]\right)\right]
\end{aligned}
$$

C.2. Variational Bound in Chapter 4

The unnormalized posterior:

$$
\begin{aligned}
\log\phi &= \log p(\mathbf{h}, \boldsymbol{\sigma}_{\mathbf{h}}^{\mathbf{2}}, \mathbf{s}, \mathbf{y}|\Theta) = \log p(\boldsymbol{\sigma}_{\mathbf{h}}^{\mathbf{2}}) p(\mathbf{h}|\boldsymbol{\sigma}_{\mathbf{h}}^{\mathbf{2}}) p(\mathbf{s}) p(\mathbf{y}|\mathbf{s}, \mathbf{h}) \\
&= \sum_{k=1}^{T_h} \left[\alpha \log\beta - (\alpha+1)\log\sigma_{h,k}^2 - \frac{\beta}{\sigma_{h,k}^2} - \log\Gamma(\alpha) \right] \\
&- 0.5 \left[T_h \log(2\pi) + \sum_{i=1}^{T_h} \log\sigma_{h,k}^2 + \mathbf{h}^\top {\boldsymbol{\Sigma}_{\mathbf{h}}}^{-1} \mathbf{h} \right] \\
&- 0.5 \left[T_s \log(2\pi) + T_s \log\sigma_w^2 + \log\left|\mathbf{R}\mathbf{R}^\top\right| + \frac{\mathbf{s}^\top \left(\mathbf{R}\mathbf{R}^\top\right)^{-1} \mathbf{s}}{\sigma_w^2} \right] \\
&- 0.5 \left[T_y \log(2\pi) + T_y \log\sigma_v^2 + \frac{1}{\sigma_v^2} \left(\mathbf{y} - \mathbf{S}\mathbf{G}_{\mathbf{h}}\mathbf{h}\right)^\top \left(\mathbf{y} - \mathbf{S}\mathbf{G}_{\mathbf{h}}\mathbf{h}\right) \right] \\
&= \sum_{k=1}^{T_h} \left[\alpha \log\beta - (\alpha+1.5)\log\sigma_{h,k}^2 - \frac{\beta}{\sigma_{h,k}^2} - \log\Gamma(\alpha) \right] \\
&- 0.5 \left[T_s \log\sigma_w^2 + \log\left|\mathbf{R}\mathbf{R}^\top\right| + (T_s + T_y + T_h)\log(2\pi) + T_y \log\sigma_v^2 + \mathbf{h}^\top {\boldsymbol{\Sigma}_{\mathbf{h}}}^{-1} \mathbf{h} \right] \\
&- 0.5 \left[\frac{1}{\sigma_v^2} \left[\mathbf{y}^\top\mathbf{y} - 2(\mathbf{S}\mathbf{G}_{\mathbf{h}}\mathbf{h})^\top \mathbf{y} + (\mathbf{S}\mathbf{G}_{\mathbf{h}}\mathbf{h})^\top \mathbf{S}\mathbf{G}_{\mathbf{h}}\mathbf{h} \right] + \frac{\mathbf{s}^\top \left(\mathbf{R}\mathbf{R}^\top\right)^{-1} \mathbf{s}}{\sigma_w^2} \right]
\end{aligned}
$$

We approximate the posterior with the factorized distribution: $Q = q(\mathbf{s})q(\mathbf{h})q(\boldsymbol{\sigma}_{\mathbf{h}}^{\mathbf{2}})$ with $q(\mathbf{s}) = \mathcal{N}(\mathbf{s}; \mathbf{m}_{\mathbf{s}}, \boldsymbol{\Sigma}_{\mathbf{s}})$, $q(\mathbf{h}) = \mathcal{N}(\mathbf{h}; \mathbf{m}_{\mathbf{h}}, \boldsymbol{\Sigma}_{\mathbf{h}})$, and $q(\boldsymbol{\sigma}_{\mathbf{h}}^{\mathbf{2}}) = \prod_{k=1}^{T_h} \mathcal{IG}(\sigma_{h,k}^2; \alpha_{h,k}, \beta_{h,k})$, where

$$
\begin{aligned}
\boldsymbol{\Sigma}_{\mathbf{s}} &= \mathbf{R}\left(\frac{\mathbf{I}}{\sigma_w^2} + \frac{\mathbf{R}^\top {\mathbf{G}_{\mathbf{s}}}^\top \langle\mathbf{H}^\top\mathbf{H}\rangle_{q(\mathbf{h})} \mathbf{G}_{\mathbf{s}}\mathbf{R}}{\sigma_v^2}\right)^{-1} \mathbf{R}^\top,\ \mathbf{m}_{\mathbf{s}} = \tfrac{1}{\sigma_v^2} \boldsymbol{\Sigma}_{\mathbf{s}} {\mathbf{G}_{\mathbf{s}}}^\top \langle\mathbf{H}^\top\rangle_{q(\mathbf{h})} \mathbf{y} \\
\boldsymbol{\Sigma}_{\mathbf{h}} &= \left(\langle {\boldsymbol{\Sigma}_{\mathbf{h}}}^{-1} \rangle_{q(\boldsymbol{\sigma}_{\mathbf{h}}^{\mathbf{2}})} + \frac{{\mathbf{G}_{\mathbf{h}}}^\top \langle\mathbf{S}^\top\mathbf{S}\rangle_{q(\mathbf{s})} \mathbf{G}_{\mathbf{h}}}{\sigma_v^2}\right)^{-1},\ \mathbf{m}_{\mathbf{h}} = \tfrac{1}{\sigma_v^2} \boldsymbol{\Sigma}_{\mathbf{h}} {\mathbf{G}_{\mathbf{h}}}^\top \langle\mathbf{S}^\top\rangle_{q(\mathbf{s})} \mathbf{y} \\
\alpha_{h,k} &= \alpha_h = \alpha + 0.5,\ \beta_{h,k} = \beta + 0.5\langle h_k^2 \rangle_{p(\mathbf{h})}
\end{aligned} \tag{C.6}
$$

$$
\mathcal{B}_{VB} = \langle \log\phi \rangle_Q + H[Q] \tag{C.7}
$$

$$
\begin{aligned}
\langle \log \phi \rangle_Q = & \sum_{k=1}^{T_h} \left[\alpha \log \beta + (\alpha + 1.5) \left\langle \log \frac{1}{\sigma_{h,k}^2} \right\rangle_{q(\sigma_{\mathbf{h}}^{\mathbf{2}})} - \beta \left\langle \frac{1}{\sigma_{h,k}^2} \right\rangle_{q(\sigma_{\mathbf{h}}^{\mathbf{2}})} - \log \Gamma(\alpha) \right] \\
& - 0.5 \left[T_s \log \sigma_w^2 + \log \left| \mathbf{R}\mathbf{R}^\top \right| + (T_s + T_y + T_h) \log(2\pi) + T_y \log \sigma_v^2 + \frac{1}{\sigma_v^2} \mathbf{y}^\top \mathbf{y} \right] \\
& - 0.5 \left[\sum_{k=1}^{T_h} \left\langle h_k^2 \right\rangle_{q(\mathbf{h})} \left\langle \frac{1}{\sigma_h^2} \right\rangle_{q(\sigma_{\mathbf{h}}^{\mathbf{2}})} - \frac{2}{\sigma_v^2} (\langle \mathbf{S} \rangle_{q(\mathbf{s})} \mathbf{G_h} \langle \mathbf{h} \rangle_{q(\mathbf{h})})^\top \mathbf{y} \right. \\
& \left. + tr \left[\frac{1}{\sigma_v^2} \mathbf{G_h}^\top \langle \mathbf{S}^\top \mathbf{S} \rangle_{q(\mathbf{s})} \mathbf{G_h} \langle \mathbf{h}\mathbf{h}^\top \rangle_{q(\mathbf{h})} \right] + tr \left[\frac{1}{\sigma_w^2} (\mathbf{R}\mathbf{R}^\top)^{-1} \langle \mathbf{s}\mathbf{s}^\top \rangle_{q(\mathbf{s})} \right] \right] \\
& = \sum_{k=1}^{T_h} [\alpha \log \beta + (\alpha + 1.5)(\log \beta_{h,k} - \Psi(\alpha_{h,k})) \\
& - \left(\beta + 0.5 \left\langle h_k^2 \right\rangle_{q(\mathbf{h})} \right) \frac{\alpha_{h,k}}{\beta_{h,k}} - \log \Gamma(\alpha) \Big] \\
& - 0.5 \left[T_s \log \sigma_w^2 + \log \left| \mathbf{R}\mathbf{R}^\top \right| + (T_y + T_s + T_h) \log(2\pi) + T_y \log \sigma_v^2 + \frac{1}{\sigma_v^2} \mathbf{y}^\top \mathbf{y} \right] \\
& - 0.5 \left[-\frac{2}{\sigma_v^2} (\langle \mathbf{S} \rangle_{q(\mathbf{s})} \mathbf{G_h} \mathbf{m_h})^\top \mathbf{y} + tr \left[\frac{1}{\sigma_v^2} \mathbf{G_h}^\top \langle \mathbf{S}^\top \mathbf{S} \rangle_{q(\mathbf{s})} \mathbf{G_h} (\mathbf{m_h} \mathbf{m_h}^\top + \mathbf{\Sigma_h}) \right] \right. \\
& \left. + tr \left[\frac{1}{\sigma_w^2} (\mathbf{R}\mathbf{R}^\top)^{-1} (\mathbf{m_s} \mathbf{m_s}^\top + \mathbf{\Sigma_s}) \right] \right] \qquad \text{(C.8)}
\end{aligned}
$$

$$
\begin{aligned}
H[Q] = -\langle \log Q \rangle_Q = & \, 0.5 \log \left[(2\pi e)^{T_h} |\mathbf{\Sigma_h}| \right] + 0.5 \log \left[(2\pi e)^{T_s} |\mathbf{\Sigma_s}| \right] \\
& - \sum_{k=1}^{T_h} \left[\alpha_{h,k} \log \beta_{h,k} + (\alpha_{h,k} + 1) \left\langle \log \frac{1}{\sigma_{h,k}^2} \right\rangle_{q(\sigma_{\mathbf{h}}^{\mathbf{2}})} - \beta_{h,k} \left\langle \frac{1}{\sigma_{h,k}^2} \right\rangle_{q(\sigma_{\mathbf{h}}^{\mathbf{2}})} - \log \Gamma(\alpha_{h,k}) \right] \\
& = 0.5 \log \left[(2\pi e)^{T_s + T_h} |\mathbf{\Sigma_h}| |\mathbf{\Sigma_s}| \right] \\
& - \sum_{k=1}^{T_h} [\alpha_{h,k} \log \beta_{h,k} + (\alpha_{h,k} + 1)(\log \beta_{h,k} - \Psi(\alpha_{h,k})) - \alpha_{h,k} - \log \Gamma(\alpha_{h,k})] \qquad \text{(C.9)}
\end{aligned}
$$

Substituting (C.8) and (C.9) into (C.7), we get

$$
\begin{aligned}
\mathcal{B}_{VB} = & \sum_{k=1}^{T_h} \left[\alpha \log \beta + (\alpha + 1.5)(\log \beta_{h,k} - \Psi(\alpha_{h,k})) - (\beta + 0.5 \left\langle h_k^2 \right\rangle_{q(\mathbf{h})}) \frac{\alpha_{h,k}}{\beta_{h,k}} - \log \Gamma(\alpha) \right] \\
& - 0.5 \left[T_s \log \sigma_w^2 + \log \left| \mathbf{R}\mathbf{R}^\top \right| + (T_s + T_y + T_h) \log(2\pi) + T_y \log \sigma_v^2 + \frac{1}{\sigma_v^2} \mathbf{y}^\top \mathbf{y} \right] \\
& - 0.5 \left[-\frac{2}{\sigma_v^2} (\langle \mathbf{S} \rangle_{q(\mathbf{s})} \mathbf{G_h} \mathbf{m_h})^\top \mathbf{y} + tr \left[\frac{1}{\sigma_v^2} \mathbf{G_h}^\top \langle \mathbf{S}^\top \mathbf{S} \rangle_{q(\mathbf{s})} \mathbf{G_h} (\mathbf{m_h} \mathbf{m_h}^\top + \mathbf{\Sigma_h}) \right] \right. \\
& \left. + tr \left[\frac{1}{\sigma_w^2} (\mathbf{R}\mathbf{R}^\top)^{-1} (\mathbf{m_s} \mathbf{m_s}^\top + \mathbf{\Sigma_s}) \right] \right] + 0.5 \log \left[(2\pi e)^{T_s + T_h} |\mathbf{\Sigma_h}| |\mathbf{\Sigma_s}| \right] \\
& - \sum_{k=1}^{T_h} \left[\alpha_{h,k} \log \beta_{h,k} + (\alpha_{h,k} + 1)(\log \beta_{h,k} - \Psi(\alpha_{h,k})) - \alpha_{h,k} - \log \Gamma(\alpha_{h,k}) \right] \\
= & \sum_{k=1}^{T_h} \left[\alpha \log \beta - \alpha_{h,k} \log \beta_{h,k} + (\alpha + 0.5 - \alpha_{h,k})(\log \beta_{h,k} - \Psi(\alpha_{h,k})) \right. \\
& \left. - \alpha_{h,k} \left(\frac{\beta + 0.5 \left\langle h_k^2 \right\rangle_{q(\mathbf{h})}}{\beta_{h,k}} - 1 \right) - \log \frac{\Gamma(\alpha)}{\Gamma(\alpha_{h,k})} \right] \\
& - 0.5 \left[T_s \log \sigma_w^2 + \log \left| \mathbf{R}\mathbf{R}^\top \right| + T_y \log(2\pi) + T_y \log \sigma_v^2 - (T_s + T_h) - \log \left[|\mathbf{\Sigma_h} \mathbf{\Sigma_s}| \right] \right] \\
& - 0.5 \left[\frac{1}{\sigma_v^2} \left(\mathbf{y}^\top \mathbf{y} - 2(\langle \mathbf{S} \rangle_{q(\mathbf{s})} \mathbf{G_h} \mathbf{m_h})^\top \mathbf{y} + tr \left[\mathbf{G_h}^\top \langle \mathbf{S}^\top \mathbf{S} \rangle_{q(\mathbf{s})} \mathbf{G_h} (\mathbf{m_h} \mathbf{m_h}^\top + \mathbf{\Sigma_h}) \right] \right) \right. \\
& \left. + tr \left[\frac{1}{\sigma_w^2} (\mathbf{R}\mathbf{R}^\top)^{-1} (\mathbf{m_s} \mathbf{m_s}^\top + \mathbf{\Sigma_s}) \right] \right] \qquad \text{(C.10)}
\end{aligned}
$$

Substituting the updates for $\alpha_{h,k}$ and $\beta_{h,k}$ in (C.6) into (C.10), we get a the final expression:

$$
\begin{aligned}
\mathcal{VB}_Q = & \sum_{k=1}^{T_h} \log \frac{\beta^\alpha \Gamma(\alpha_h)}{\beta_{h,k}^{\alpha_h} \Gamma(\alpha)} - 0.5 \left(T_y \log(2\pi\sigma_v^2) - (T_s + T_h) - \log |\mathbf{\Sigma_h} \mathbf{\Sigma_s}| \right) \\
& - 0.5 \left[\frac{1}{\sigma_v^2} \left(\mathbf{y}^\top \mathbf{y} - 2(\langle \mathbf{S} \rangle_{q(\mathbf{s})} \mathbf{G_h} \mathbf{m_h})^\top \mathbf{y} + tr \left[\mathbf{G_h}^\top \langle \mathbf{S}^\top \mathbf{S} \rangle_{q(\mathbf{s})} \mathbf{G_h} (\mathbf{m_h} \mathbf{m_h}^\top + \mathbf{\Sigma_h}) \right] \right) \right] \\
& - 0.5 \left[T_s \log \sigma_w^2 + \log \left| \mathbf{R}\mathbf{R}^\top \right| + tr \left[\frac{1}{\sigma_w^2} (\mathbf{R}\mathbf{R}^\top)^{-1} (\mathbf{m_s} \mathbf{m_s}^\top + \mathbf{\Sigma_s}) \right] \right]
\end{aligned}
$$

REFERENCES

1. Mendel, J. M., *Optimum Seismic Deconvolution*, Academic Press (New York), 1983.

2. Cheng, Q., R. Chen, and T.-H. Li, "Simultaneous wavelet estimation and deconvolution of reflection seismic signals", *Geoscience and Remote Sensing, IEEE Transactions on*, Vol. 34, No. 2, pp. 377–384, Mar 1996.

3. Rosec, O., J. M. Boucher, B. Nsiri, and T. Chonavel, "Blind Marine Seismic Deconvolution Using Statistical MCMC Methods", *IEEE Oceanic Engineering*, Vol. 28, pp. 502–512, 2003.

4. Lin, Y., J. Chen, Y. Kim, and D. D. Lee, "Blind Sparse-Nonnegative (BSN) Channel Identification for Acoustic Time-Difference-of-Arrival Estimation", *2007 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pp. 106–109, Oct. 2007.

5. Liu, J. S. and R. Chen, "Blind Deconvolution via Sequential Imputations", *Journal of the American Statistical Association*, Vol. 90, No. 430, pp. 567–576, 1995.

6. Phinney, R. A., "Structure of the Earth's Crust from Spectral Behavior of Long-Period Body Waves", *J. Geophys. Res.*, Vol. 69, No. 14, pp. 2997–3017, 1964.

7. Langston, C. A., "Structure under Mount Rainier, Washington, inferred from teleseismic body waves", *J. Geophys. Res.*, Vol. 84, pp. 4749–4762, 1979.

8. Kiknchi, M. and H. Kanamori, "Inversion of complex body waves", *Bull. Seism. Soe. Am.*, Vol. 72, pp. 491–506, 1982.

9. Ligorra, J. P. and C. J. Ammon, "Iterative deconvolution and receiver-function estimation", *Bull. Seism. Soe. Am.*, Vol. 89, pp. 1395–1400, 1999.

10. Winener, N., *The Extrapolation, Interpolation, and Smoothing of Stationary Time Series with Engineering Applications*, Wiley, New York, 1949.

11. Haykin, S., *Adaptive Filter Theory*, Prentice Hall, 2002.

12. Bell, A. J. and T. J. Sejnowski, "An information-maximization approach to blind separation and blind deconvolution.", *Neural Comput*, Vol. 7, No. 6, November 1995, `http://view.ncbi.nlm.nih.gov/pubmed/7584893`.

13. Amari, S., S. C. Douglas, A. Cichocki, and H. H. Yang, "Novel On-line Adaptive Learning Algorithms for Blind Deconvolution using the Natural Gradient Approach", *in Proc. SYSID, Kitakyushu*, pp. 1057–1062, 1997.

14. Tong, L., G. Xu, and T. Kailath, "Blind Identification and Equalization Based on Second-Order Statistics: A Time Domain Approach", *IEEE Trans. Inform. Theory*, Vol. 40, pp. 340–349, 1994.

15. Hua, Y., "Fast maximum likelihood for blind identification of multiple FIR channels", *IEEE Transactions on Signal Processing*, Vol. 44, No. 3, pp. 661–672, 1996.

16. Shalvi, O. and E. Weinstein, "New criteria for blind deconvolution of nonminimum phase systems (channels)", *IEEE Transactions on Information Theory*, Vol. 36, No. 2, pp. 312–321, 1990.

17. Bellini, S., "Bussgang techniques for blind equalization", *IEEE Global Telecommunications Conference (GLOBECOM 86)*, pp. 1634–1640, Dec. 1986.

18. Li, Y., A. Cichocki, and L. Zhang, "Blind source estimation of FIR channels for binary sources: a grouping decision approach", *Signal Processing*, Vol. 84, No. 12, pp. 2245–2263, 2004.

19. Amari, S., A. Cichocki, and H. H. Yang, "A New Learning Algorithm for Blind Signal Separation", *Advances in Neural Information Processing Systems*, pp. 757–763, MIT Press, 1996.

20. J., L., O. Macchi, and C. Laot, "Adaptive decision feedback equalization: can you skip the training period?", *IEEE Transactions on Communications*, Vol. 46, No. 7, pp. 921–930, 1998.

21. Xia, B. and L. Zhang, "Blind deconvolution in nonminimum phase systems using cascade structure", *EURASIP J. Appl. Signal Process.*, Vol. 2007, No. 1, pp. 85–85, 2007.

22. Hurley, N. and S. Rickard, "Comparing measures of sparsity", *IEEE Workshop on Machine Learning for Signal Processing, 2008. MLSP 2008.*, pp. 55–60, Oct. 2008.

23. Özakın, Y., *Crustal structure of Southwestern Anatolia using p-receiver function analysis*, Master's thesis, Boğaziçi University, 2008.

24. Gelman, A., J. Carlin, H. S. Stern, and D. B. Rubin, *Bayesian Data Analysis*, "Chapman & Hall", 1995.

25. Labat, C. and J. Idier, "Sparse Blind Deconvolution Accounting for Time-Shift Ambiguity", *2006 IEEE International Conference*, 2006.

26. Cappé, O., A. Doucet, M. Lavielle, and E. Moulines, "Simulation Based Methods for Blind Maximum Likelihood Filter Identification", *Signal Processing*, Vol. 73, No. 1, pp. 3–25(23), 1999.

27. Lavielle, M., "A stochastic algorithm for parametric and non-parametric estimation in the case of incomplete data", *Signal Processing*, Vol. 42, pp. 3–17, 1995.

28. Cemgil, A. T., C. Fevotte, and S. J. Godsill, "Variational and Stochastic Inference for Bayesian Source Separation", *Digital Signal Processing*, Vol. 17, No. 5, pp. 891–913, 2007.

29. Cemgil, A. T., C. Fevotte, and S. J. Godsill, "Blind Separation of Sparse Sources using Variational EM", *13th European Signal Processing Conference*, EURASIP, Antalya/Turkey, 2005.

30. Tzikas, D., A. Likas, and N. Galatsanos, "Variational Bayesian Blind Image Deconvolution with Student-T Priors", *IEEE International Conference*, ICIP, 2007.

31. Fevotte, C. and S. Godsill, "A Bayesian approach for blind separation of sparse sources", *IEEE Transactions on Audio, Speech and Language Processing*, Vol. 14, No. 6, pp. 2174–2188, 2006.

32. Yıldırım, S., A. T. Cemgil, and A. B. Ertüzün, "A hybrid method for deconvolution of Bernoulli-Gaussian processes", *IEEE International Conference on Audio, Speech, and Signal Processing*, ICASSP, 2009.

33. Kirkpatrick, S., C. D. Gelatt, and M. P. Vecchi, "Optimization by Simulated Annealing", *Science*, Vol. 220, No. 4598, pp. 671–680, 1983.

34. Cerny, V., "A thermodynamical approach to the travelling salesman problem: an efficient simulation algorithm", *Journal of Optimization Theory and Applications*, Vol. 45, pp. 41–51, 1985.

35. Xu, C. and J. Prince, "Gradient vector flow: a new external force for snakes", *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 66–71, Jun 1997.

36. Oppenheim, A. and A. Willsky, *Signals and Systems*, Prentice Hall, 1997.

37. Box, G. E. P. and G. C. Tiao, *Bayesian Inference in Statistical Analysis*, Addison-Wesley Publishing Company, 1973.

38. Sivia, D. S., *Data Analysis: A Bayesian Tutorial*, Oxford University Press, NY, 1998.

39. Everitt, D. S., *The Cambridge Dictionary of Statistics*, Cambridge University Press, 2006.

40. Caflisch, R. E., *Monte Carlo and quasi-Monte Carlo methods*, Cambridge Univer-

sity Press, 1998.

41. MacKay, D., *Information Theory, Inference and Learning Algorithms*, Cambridge University Press, 2003.

42. Tanner, M., *Tools for statistical inference: methods for the exploration of posterior distributions and likelihood functions*, Springer, 1996.

43. Robert, C. and G. Casella, *Monte Carlo Statistical Methods*, Springer, 1999.

44. Metropolis, N., A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equations of state calculations by fast computing machines", *Journal of Chemical Physics*, Vol. 21, pp. 1087–1091, 1963.

45. Gilks, W. R., S. Richardson, and D. J. Spiegelhalter, *Markov Chain Monte Carlo in Practice*, Chapman & Hall/CRC, 1996.

46. Geman, S. and D. Geman, "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 6, No. 6, pp. 721–741, 1984.

47. Beal, M., *Variational Algorithms for Approximate Bayesian Inference*, Ph.D. thesis, Gatsby Computational Neuroscience Unit, University College London, UK, 2003.

48. Cover, T. and J. Thomas, *Elements of Information Theory*, Wiley, 1991.

49. Dempster, A., N. Laird, and D. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm", *Journal of the Royal Statistical Society*, Vol. 39, No. 1, pp. 1–38, 1977.

50. Cappé, O., E. Moulines, and T. Rydén, *Inference in Hidden Markov Models*, Springer, 2005.

51. Wei, G. and M. Tanner, "A Monte-Carlo implementation of the EM algorithm and the Poor's Man's data augmentation algorithm", *J. Amer. Statist. Assoc*, Vol. 85, No. 1, pp. 699–704, 1990.

52. Delyon, B., M. Lavielle, and E. Moulines, "Convergence of a stochastic approximation version of the EM algorithm", *Annals of Statistics*, Vol. 27, No. 1, pp. 94–128, 1999.

53. Celeux, G. and J. Diebolt, "The SEM algorithm: A probabilistic teacher algorithm derived from the EM algorithm for the mixture problem", *Computational Statistics Quarterly*, Vol. 2, pp. 73–82, 1985.

54. Durbin, J. and S. J. Koopman, *Time series analysis by state space methods*, Oxford University Press, 2001.

55. Rauch, H. E., F. Tung, and C. T. Striebel, "Maximum likelihood estimates of linear dynamic systems", *AIAA Journal*, Vol. 3, No. 8, pp. 1445–1450, 1965.

56. Mayne, Q. D., "A solution of the smoothing problem for linear dynamic systems", *Automatica*, Vol. 4, pp. 73–92, 1966.

57. Cemgil, A. T., "Strategies for sequential inference in factorial switching state space models", *Proc. of IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP 07)*, pp. 513–516, Honolulu, Hawaii, 2007.

58. Snyman, J. A., *Practical Mathematical Optimization: An Introduction to Basic Optimization Theory and Classical and New Gradient-Based Algorithms*, Springer, 2005.

59. Ypma, T. J., "Historical development of the Newton-Raphson method", *SIAM Review*, Vol. 37, No. 4, pp. 531–551, 1995.

60. Stein, S. and M. Wysession, *An Introduction to Seismology, Earthquakes, and Earth Structure*, Wiley-Blackwell, 2003.