

Polyphonic Correlated Source Separation
(*PCSS*)

Eliseo Steve Rodríguez Rodríguez

Master of Science
School of Artificial Intelligence
Division of Informatics
University of Edinburgh
2002

Abstract

This thesis presents a learning-based approach that considers temporal dependencies for doing *Polyphonic Correlated Source Separation*. We use prior knowledge about the physics of the instrument notes such as information from their frequencies and decay rate using time-frequency analysis (*Short Time Fourier Transform*) and fit autoregressive models to the instrument notes. Then, we use linear dynamical systems for modelling the observations, and switches for including the models of the instrument notes. The inference concerning the presence of the sounds for each time step and their contents (for the signal separation) is achieved by using a variational approximation that maximizes a lower bound on the log likelihood. Algorithms for smoothing and filtering are derived and implemented, with the latter giving good results in our tests.

Acknowledgments

I would like to express my gratitude to my first and second supervisors, Dr. David Barber and Felix Agakov, for letting me learn from them and being such a good guides in these first steps towards probability and machine learning.

In addition, I appreciate Luchepe's non-stopping worries and support, and Aaron's and Maggie's efforts for correcting many horrible mistakes related to my use of the English language. I thank *Mexico's National Council for Science and Technology (CONACYT)* for the funding that allowed me to study in Edinburgh. Finally, I express my gratitude to all the people I met in here who helped me in modelling my perspective about the essence of life, and to the many neurons and "fairies" that died in the process.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(Eliseo Steve Rodríguez Rodríguez)

Table of Contents

1	INTRODUCTION	1
1.1	PCSS	1
1.2	OBJECTIVE	2
1.3	LIMITATIONS	4
1.4	MOTIVATION	5
1.5	SCOPE.....	7
1.6	NOTATION	7
1.7	ORGANIZATION OF THE DOCUMENT	8
2	LITERATURE REVIEW	9
2.1	SIGNALS AND ACOUSTICS FROM MUSICAL INSTRUMENTS.....	9
2.1.1	Properties of Signals	9
2.1.2	Properties of Acoustics from Musical Instruments	13
2.2	AUTOREGRESSIVE MODELS	15
2.3	BLIND SOURCE SEPARATION	17
2.4	DYNAMICAL MODELS.....	19
2.4.1	Hidden Markov Models.....	19
2.4.2	State Space Models and Linear Dynamical Systems.....	20
3	BUILDING A PROBABILISTIC MODEL	25
3.1	GENERAL IDEA.....	25
3.1.1	Sound model	26
3.1.2	Unifying model	28
3.2	MSE AS AN OPTIMALITY MEASURE.....	30
4	DERIVING THE MODEL	33
4.1	GLOBAL STRUCTURE	33
4.1.1	Defining a probability distribution for each switch	34
4.1.2	Learning an autoregressive model for each sound.....	37
4.2	INFERENCE	39
4.2.1	Q distribution for the linear models	42

4.2.2	Q distribution for the switches	43
4.2.3	Equations for Inference.....	44
4.2.4	Generalization	46
5	IMPLEMENTING AND TESTING THE MODEL	49
5.1	IMPLEMENTATION AND DATA ACQUISITION	49
5.2	TESTING THE AR MODELS	50
5.3	TESTING THE INFERENCE MODELS.....	54
5.3.1	Selection of Parameters	55
5.3.2	Tests with two models.....	58
5.3.3	Tests with different instruments	61
5.3.4	More complex tests	62
5.3.5	Generalization tests	63
6	RESULTS, FUTURE DIRECTIONS AND CONCLUSIONS	65
6.1	RESULTS.....	65
6.1.1	Use of temporal Information.....	65
6.2	FUTURE DIRECTIONS.....	66
6.2.1	Solve Problems.....	66
6.2.2	Start loosening some assumptions.....	68
6.2.3	Conclusions.....	70
A	PCSS DERIVATIONS	71
A.1	MEANING OF VARIABLES.....	71
A.2	OPTIMAL ML PARAMETERS (AR MODEL).....	72
A.2.1	Optimal mean.....	72
A.2.2	Optimal variance.....	73
A.2.3	Optimal autoregressive matrix	74
A.3	OPTIMAL ML PARAMETERS (Q DENSITIES)	75
A.3.1	Optimal $q(\mathbf{f}^{(s)}(t))$	77
A.3.2	Optimal $q(\mathbf{C}(t))$	81
<hr/>		
	REFERENCES	85

Table of Figures

FIGURE 1.1: SIGNAL SEPARATION	1
FIGURE 1.2: A SIMPLE POLYPHONIC SEPARATION	2
FIGURE 1.3: A COMPLEX POLYPHONIC SEPARATION	3
FIGURE 1.4: GRAPHICAL MODEL OF A MIXED OBSERVATION	4
FIGURE 2.1: SIMPLE SINE SIGNAL	10
FIGURE 2.2: TWO SIMPLE SIGNALS AND A COMPLEX ONE	11
FIGURE 2.3: <i>FOURIER</i> COMPONENTS OF A COMPLEX SIGNAL	12
FIGURE 2.4: WAVEFORMS OF A GUITAR AND PIANO PLAYING A G4.....	14
FIGURE 2.5: HARMONICS OF A GUITAR AND PIANO PLAYING A G4.....	15
FIGURE 2.6: FIRST EIGHT HARMONICS OF G2 (98 Hz) (ROEDERER 1995).....	15
FIGURE 2.7: SOURCES AND OBSERVATIONS FOR <i>BSS</i>	17
FIGURE 2.8: GRAPHICAL MODEL OF A HMM.....	19
FIGURE 2.9: GRAPHICAL MODEL OF A SSM.....	21
FIGURE 3.1: INSTANT MIXING OF DIFFERENT SOURCES.....	25
FIGURE 3.2: EVOLUTION OF AN INSTRUMENT NOTE AND OF THE REAL COMPONENTS OF ITS FIRST THREE HARMONICS.....	27
FIGURE 3.3: SWITCHING LINEAR DYNAMICAL SYSTEM.....	29
FIGURE 3.4: DEPHASED SIGNALS	31
FIGURE 4.1: GRAPHICAL MODEL OF TWO MIXED SOURCES WITH BOOLEAN SWITCHES.....	33
FIGURE 4.2: EXPONENTIAL DISTRIBUTION OF A STATIONARY SOUND	35
FIGURE 4.3: APPROXIMATIONS TO AN EXPONENTIAL DISTRIBUTION OF A STATIONARY SOUND.....	37
FIGURE 4.4: GRAPHICAL MODEL OF S MIXED SOURCES WITH BOOLEAN SWITCHES	46
FIGURE 4.5: GRAPHICAL MODEL FOR AN ORDER O <i>MARKOVIAN</i> AUTOREGRESSION.....	48
FIGURE 5.1: WINDOWING A SIGNAL	53
FIGURE 5.2: AUTOREGRESSIVE SOUND OF A PIANO PLAYING EB6.....	53
FIGURE 5.3: AUTOREGRESSIVE SOUNDS OF A PIANO PLAYING EB4, USING RAW AND PROCESSED DATA.....	54

FIGURE 5.4: ERRORS FOR DIFFERENT SIGMAS FOR AR MODELS OF ORDER 1, 2, AND 3	57
FIGURE 5.5: ERRORS FOR DIFFERENT GAMMAS FOR AR MODELS OF ORDER 1, 2, AND 3	58
FIGURE 5.6: EXAMPLE OF THE RESULTS OF A COMPLEX TEST FOR SEPARATING A MIXTURE OF TWO SOURCES.....	59
FIGURE 5.7: CONFUSION OF ONE MODEL WITH THE REST OF THE MODELS	60
FIGURE 5.8: CONFUSION OF THE REST OF THE MODELS WITH A SPECIFIC MODEL.....	60
FIGURE 5.9: COMPARISON OF THE HARMONICS OF THE SOURCES	61
FIGURE 5.10: CONFUSION OF EACH MODEL WITH THE REST OF THE MODELS USING DIFFERENT INSTRUMENTS	62
FIGURE 5.11: COMPLEX RECONSTRUCTION	63
FIGURE 5.12: TESTS WITH SAMPLES FROM A REAL INSTRUMENT	64
FIGURE 5.13: GENERALIZATION TO A NOTE DIFFERENT FROM THE MODELS.....	64

Chapter 1

Introduction

1.1 *PCSS*

The problem of *Source Separation* (SS), as we can infer from its name, is related to recovering different signals (or “sources”) from observed mixtures. For example, consider Figure 1.1. In it, two sources (left part) are combined so that they produce another signal that in some way contains the information about them (middle part). Given the mixed signal but not the original sources, the problem is to recover the source signals such that, if not identical, they resemble the true sources as faithfully as possible (right part).¹

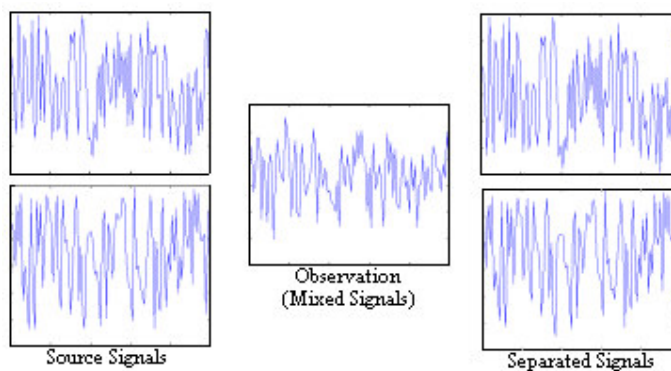


Figure 1.1: Signal separation

Considering SS as a general problem without restrictions complicates this task. The sources could be arbitrary, such as images, electro-magnetic signals, or random noise. In the same regard, the signals could be mixed in virtually an infinite number of combinations. Therefore, this problem has to be delimited in some specific ways.

¹ From here to the rest of this document, all those graphics with no labels written in their axis can be treated similarly. In them, the horizontal axis is related to time while the vertical axis is related to amplitude.

Polyphonic Correlated Source Separation (PCSS) is related to *SS*, taking into account that the source signals originated from musical instruments (basically being sounds with information about a particular note). Since the notes/instruments are correlated given a mixed polyphonic observation, the tasks are to identify which sources are present in the given signal, where, and to separate them. Figure 1.2 shows a particularization of the above example when using sounds instead of unknown signals (a silence was included in the second source to exemplify that the sound signals could be anywhere in the mixture).

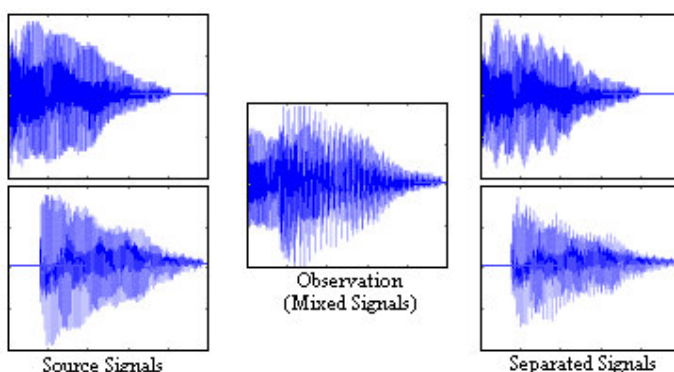


Figure 1.2: A simple polyphonic separation

Using sounds containing notes played by instruments draws up the boundaries of the broad problem covered by *SS*, as our uncertainty regarding the content of the sources diminishes. In broad terms, by considering each combination of instrument-note as one source, we simplify the *SS* problem by delimiting the instruments and notes that could be playing.

1.2 Objective

The objective of the present work is to derive and implement a probabilistic model for separating different sound sources from a polyphonic observation, by exploiting the temporal information of the signals. In principle, our model has to identify the sources and separate them from a simple mixing case (as that already presented in Figure 1.2), to a more complicated and useful auditory environment, such as the one shown in Figure 1.3. In this new example, a piano is playing a score and the output is used as a mixture

(upper part). What ideally has to be the output of the separation process for each note is shown below the score. The interesting issue about the score is its complexity for the separation task. In it, we have different notes with different durations playing alone and in chords. The chords involve the mixture of different notes (Eb4-D5-F5, Eb4-D5, A4-C5, Eb4-G4, Eb4-G4-D5, Eb4-D5), similar notes with different octaves (D5-D6, Eb5-Eb6, F5-F6) and similar chords with slight changes (Eb4-G4, Eb4-G4-D5). In addition, there are notes that are played on their own and as a chord with different notes (such as Eb4), while other notes are played only once and as a chord (Eb5-Eb6).

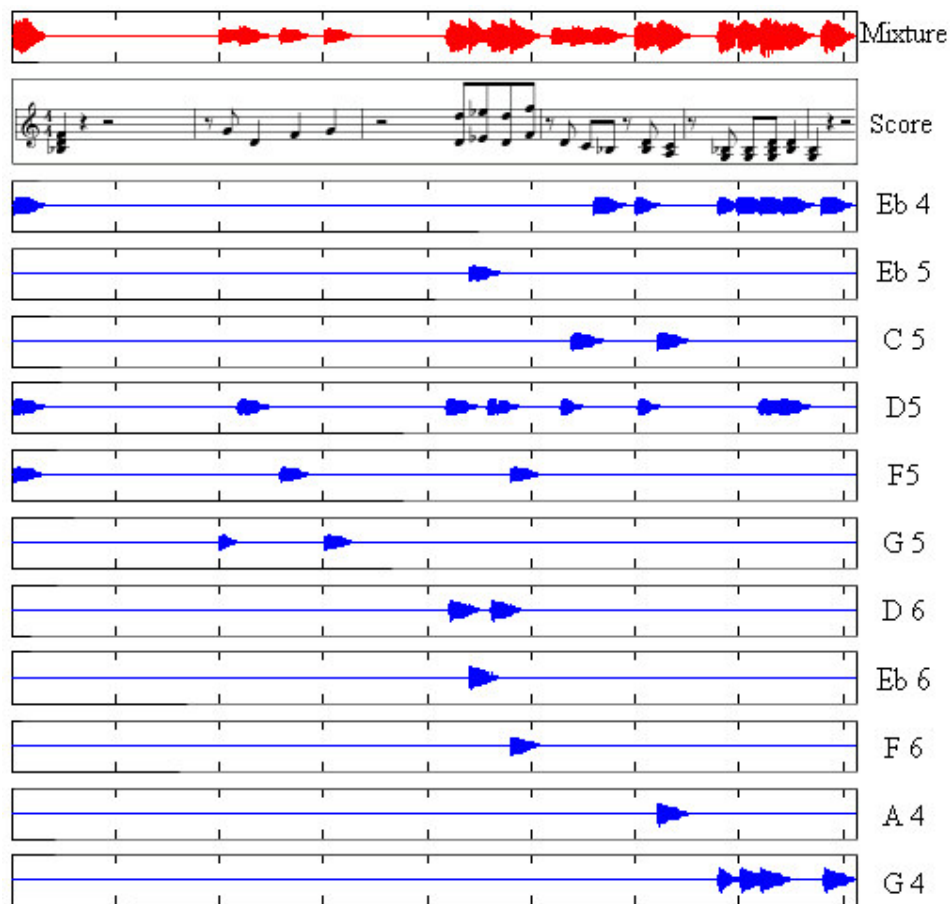


Figure 1.3: A complex polyphonic separation

In order to achieve this kind of separations, we adopt a posture that focuses on two points: exploiting the characteristic of each note/instrument, and using probability to infer its presence/content given a mixed observation.

Related to the first point, we can observe from the figure above that there are two primary characteristics in acoustic signals that we could exploit to differentiate and separate diverse instruments and notes: their dependence and evolution in time, and the frequencies they use (from the frequency components we could determine the form of the signal, from its evolution in time we define, for example, its amplitude). Concerning the second point, we will be given a mixed polyphonic observation where we don't know which signals it contain, so we are interested in extracting the unobserved instrumental information. This can be represented by the graphical model shown in Figure 1.4, leading to the use of probabilistic frameworks. In the figure, \mathbf{Y} represents the mixed observation (shaded), while each \mathbf{f} stands for a couple instrument-note (non shaded).

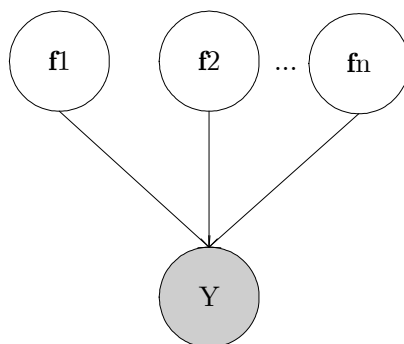


Figure 1.4: Graphical model of a mixed observation

1.3 Limitations

As a result of trying to approach the *PCSS* problem by using probabilistic models with observed/unobserved elements (1.2), our model must make use of hidden (or latent) variables for extracting the unobserved information. In addition, since we are interested in exploiting the temporal information of the instrument notes, our model should employ temporal dependencies.

To simplify both the problem and the model, we consider the following restrictions:

- Use of instrument notes without effects. This limits the possible characteristics that could be added to a sound such as reverberation, flange and chorus (to name some), so the sounds will be as natural as possible.
- Use of instantaneous mixing of the sources. The sources will be mixed directly, without considering other effects that may appear in nature (such as and echoes and delays), therefore excluding convolutive mixtures.
- Use of *a priori* knowledge of the characteristics of the instrument notes. Some probabilistic approaches for doing Source Separation use online learning for the distributions of the possible sources (2.3). We will relax this assumption by learning the distributions of all the sounds (which may or may not be part of a mixed observation) in an offline way.

1.4 Motivation

The motivation for this project basically lies in three aspects: the concern of understanding and manipulating in a better way some probabilistic models, the interest in the problem of signal separation including temporal information, and the will to support technologies for musical applications.

Regarding the first and second points, this project will focus on the derivation and the implementation a probabilistic model for doing *Source Separation* using temporal information. First of all, we need to generalize between similar sources and observations, hoping that our solution could be extended for real-life situations. In this regard, the use of probabilistic models is a natural way to estimate uncertainty with strict objective criteria. They also have other inherent benefits such as allowing evolution and being naturally extensible. For example, if we were to model some probabilistic behaviour regarding musical observations, our initial models could also be simplified by relaxing some assumptions and dependencies and later replaced with more realistic ones (such as changing independent relations for dependent ones, or using more complex probabilistic models). Finally,

temporal dependencies have been commonly ignored for identifying/separating sources (Cardoso 1998), and when considering sound signals, the methodologies have concentrated to a large extent on basically using the frequency content (Plumbley, Abdallah et al. in press). Hence, this project could have some relevance concerning the use temporal information for *Source Separation*.

With respect to the support for musical applications, *PCSS* could have different uses such as musical instruction and analysis of sources,² whereas undoubtedly the most obvious application is *Automatic Music Transcription (AMT)*. *AMT* deals with transforming an acoustic signal into musical notation for the sounds that constitute the piece (a symbolic representation which of musical events and their parameters). Clearly, after separating a mixed observation into its constitutive instrument notes such as shown in Figure 1.3, the polyphonic transcription gets considerably simplified (as the task of identifying which note and which instrument was playing at every specific time and for how long can be saved just by detecting the onsets and offsets).³

Despite of its relevance for music, this project may be generally applicable in areas where the need for recognizing elements/objects could be achieved by identifying related signals with temporarily dependent patterns. Some possible applications may concern radars, surveillance, and acoustics.

² Most people, even musicians, do not have the ability to name the pitch of a note heard in isolation, and even less to say what a chord is. More drastically, other people cannot say what two different instruments are playing if they do it at the same time (a common case is related to the bass and a guitar, where some listeners cannot distinguish the former when both instruments are playing). Only after being trained, they are able to perform musical transcription (and the richer is the complexity of a musical composition, the more experience is needed in musical ear training, instruments involved and in music theory) (Klapuri 2001; Plumbley, Abdallah et al. in press).

³ Since transcription is out of the scope of this project, perhaps the easiest way to achieve would be to parse the information from the separated signals and save it as a midi file (since vast research has already been done for it) (Klapuri 1998).

1.5 Scope

The scope for this project is limited to the following points:

- Derivation and implementation of a probabilistic model of sound signals to be used for the prior knowledge of the sources with their underlying dynamics (*n*-variate *Autoregressive model*).
- Derivation and implementation of a probabilistic model that, using the prior knowledge of the signals, could be used to perform inference about the presence and the contents of two sources given a mixed observation (*Switching Linear Dynamical System*).
- Generalization of the previous model for supporting any number of sources.

1.6 Notation

We will use some particular notation for the rest of the document, so it is recommended that the reader be aware of it for a better understanding. Some notational conventions are: scalars in italic lower case (x) and arrays (matrices or vectors) in boldface (\mathbf{x}).⁴ The i -th component of a vector \mathbf{x} is denoted as x_i , and the i,j -th element of a matrix is referenced by \mathbf{X}_{ij} . Elements/parameters of different models are addressed with indices, so the vector \mathbf{x} related to the s -th model is $\mathbf{x}^{(s)}$. The expectation of \mathbf{x} with respect to another variable \mathbf{f} is denoted by $\langle \mathbf{x} \rangle_{\mathbf{f}}$. Finally, \mathbf{T} denotes the transpose of \mathbf{x} when we have $\mathbf{x}^{\mathbf{T}}$.

We also use graphical models (such as the one in Figure 1.4) to represent the probabilistic models. As usual with graphical models, a graph consists of nodes and edges. A shaded circle node corresponds to an observed variable; an open circle node corresponds to an unobserved variable. An edge (directed arrow) makes reference to a probabilistic conditional dependence of the node at the arrow's head on the node at its tail.

⁴ The vector/matrix particularization will be expressed when required.

1.7 Organization of the document

This document is organized in six chapters. In Chapter 2, we present the literature review covering basic properties of sound signals and *Fourier* transforms (elementary for manipulating our data from auditory signals), *Autoregressive models* (for learning from temporarily dependent data and simulating outputs), the *Blind Source Separation* problem and some approaches for solving it, *Dynamical Models* (for modelling temporal-dependent distributions) and the *Kalman Filter* (for performing inference). In Chapter 3 we present in a higher level the idea and explanation of the model to be built. Chapter 4 is dedicated to deriving all the mathematics required for the model. In Chapter 5 we present the results of the implementation and tests. Finally, we present our results, future directions, and conclusions in Chapter 6.

Chapter 2

Literature Review

The objective of this chapter is to present concepts relevant to *PCSS*, limiting us to those where their inherent difficulty is not to be considered out of the scope of this project.

2.1 Signals and Acoustics from Musical Instruments

This section presents some of the characteristics of acoustic signals that have to be taken into consideration for building our model. We start by describing some of the characteristics of signals, and then we relate them with the ones of acoustic signals from musical instruments.

2.1.1 Properties of Signals

A signal is a measure that varies in time, and therefore is a function of it. Depending on the way signals vary, they can be classified as periodic or aperiodic. One signal is said to be periodic if it repeats over fixed intervals of time i (such that $s(t+i) = s(t)$, $i > 0$ for all time steps) and aperiodic otherwise (where in this case they may have a completely random waveform or just a pattern that, although similar, doesn't repeat) (Howard and Angus 1996; Johnson 1997).

Periodic signals, because of their composition, are subdivided into simple and complex signals. Simple periodic signals are the ones that are merely characterized by a sine wave, and to define them we need to know three properties (Figure 2.1). The first property is the frequency, which relates to

the number of repetitions of the period (T_p) per second.¹ It is measured in Hertz (Hz) and it is given by $f_0 = 1/T_p$ (where f_0 stands for the *fundamental frequency*). The second property of periodic signals is the amplitude, which refers to the instantaneous value of a signal at any time. Finally, the third property is the phase and it refers to the relative position in time within a single period of a signal (measured in degrees) (Johnson 1997).

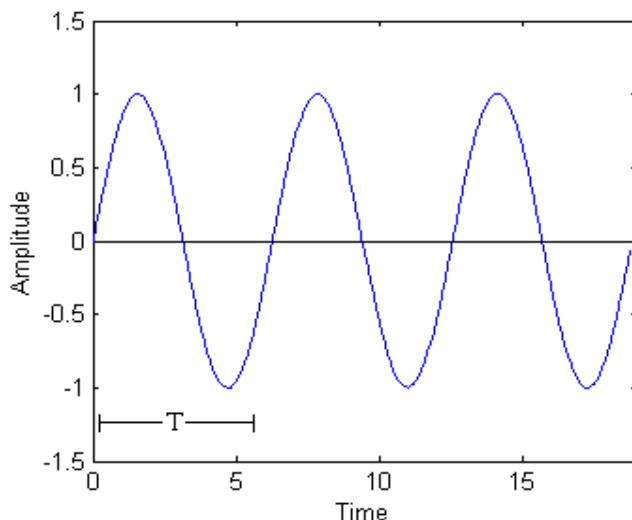


Figure 2.1: Simple sine signal

Complex periodic signals are composed of at least two sine waves. From Figure 2.2 we exemplify this by adding two simple signals (where the second has four times the frequency of the first one). As we can see, complex periodic signals are like simple periodic ones in that they involve a repeating waveform pattern and thus have cycles. The rate at which the complex pattern repeats is f_0 (which in this example, is related to the frequency of the first sinusoidal) and in general, if there are m frequencies to be mixed, f_0 will be the greatest common divisor of them (with an amplitude of zero if it is not in the set of m frequencies) (Taylor 1965; Johnson 1997).

¹ An index p was included in the commonly used symbol of the period for not confusing the reader with T , a symbol that we will introduce later which refers to the total number of time steps in a sequence.

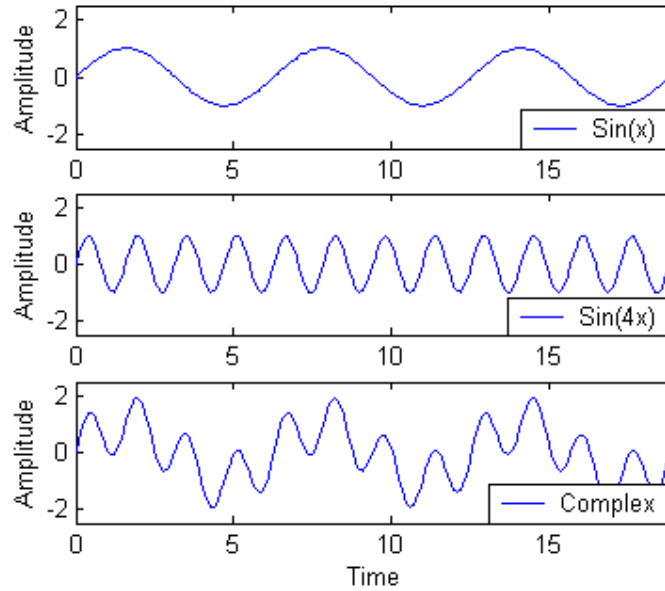


Figure 2.2: Two simple signals and a complex one

2.1.1.1 *Fourier Transform*

Inversely to creating complex signals by the addition of simple ones, a related property of complex signals is that they can be analysed in terms of sine wave components. This is, with the *Fourier Transform (FT)*, any complex waveform (periodic or aperiodic) can be decomposed into a set of sine waves having particular frequencies, amplitudes and phase relations. By considering $x(t)$ as a given waveform and having a time segment of duration T , the *Fourier series* are then given by

$$\begin{aligned}
 x(t) &= \sum_{k=-\infty}^{\infty} (\hat{q}_k \exp\{-i_k 2\pi k f_0 t\}) \\
 &= \text{Re} \left(\sum_{k=0}^{\infty} (\hat{q}_k \exp\{-i_k 2\pi k f_0 t\}) \right) \\
 &= a_0 + \sum_{k=1}^{\infty} (a_k \cos(2\pi k f_0 t) + b_k \sin(2\pi k f_0 t)) \quad (2.1.1)
 \end{aligned}$$

where $\hat{q}_k = a_k + b_k i$. Therefore, the job of a *FT* is to figure out all the a_k and b_k values to produce the series, given the base frequency and the function $x(t)$ (Pierce 1994). The value of each component \hat{q}_k is then

$$\hat{q}_k = \frac{1}{T} \int x(t) \exp(i 2\pi k f_0 t) dt. \quad (2.1.2)$$

The result of the *FT* is two distributions existing different “spaces” in which dimensions are reciprocally related (Taylor 1965). In Figure 2.3 we illustrate

this with our complex signal and the reciprocal representations (which were mapped to negative and positive areas). If we were to consider just one dimension, we square the product \hat{q}_k and its conjugate to obtain a *power spectrum* (an amplitude versus frequency plot) (Taylor 1965; Johnson 1997).

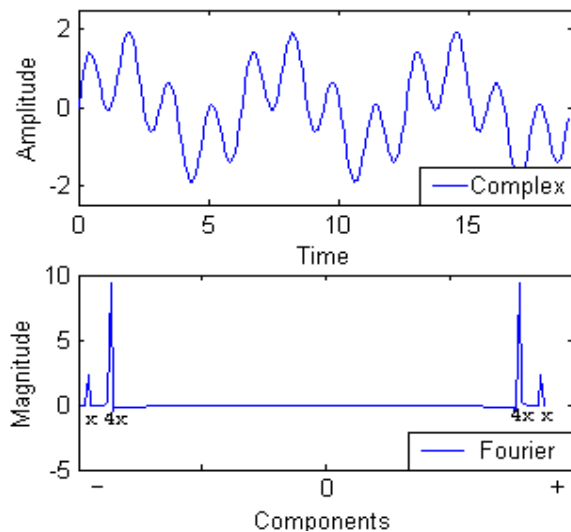


Figure 2.3: *Fourier* components of a complex signal

One important property of *FT* is that if two functions are added in one space, then their transforms add vectorially in the other space (Taylor 1965). In other words, if we add together two different complex signals (giving a mixed complex signal), the *Fourier transform* of the mixture equals the addition of the individual transforms corresponding to each original complex signal, therefore being a linear operator.

2.1.1.2 Short Time *Fourier* Transform

If the spectral content of a signal does not change over time (such when considering periodicity), the *Fourier transform* gives appropriate information about the contents in general. However, if the signal to be analysed has considerable changes in time (e.g. when signals are appended/added indifferent positions, e.g., one after the other), then, as the *FT* provides how much of each frequency is present in a signal, it will not give more information than the one contained in the whole signal (Gerhard 2000).

The *Short-Time Fourier Transform (STFT)* is an attempt to fix the lack of time resolution in the *FT*. The input data is broken into many small sequential pieces called frames or windows, and the *FT* is applied to each one in succession (being a trade-off between window size and frequency resolution, where spectral details are better resolved when the analysis windows is long). As a result, a time dependent representation is produced, showing the changes in the spectrum as the signal processes (Gerhard 2000).

2.1.2 Properties of Acoustics from Musical Instruments

An acoustic waveform is a record of sound producing pressure fluctuations over time, which repeat regularly (Howard and Angus 1996; Johnson 1997). Regarding sounds from musical instruments, there are three primary characteristics associated with them: pitch (corresponding to the “altitude” or “height” of a tone), loudness (the “strength” or “intensity” it has), and timbre (the quality that enables us to distinguish among different instruments). The assignment of pitch, loudness, and timbre to a musical sound is the result of processing operations in the ear and in the brain. However, each one of these primary sensations can be associated to a well-defined physical quantity of the original stimulus that can be measured. By relating instrument sounds to some of the properties of signals already presented in 2.1.1, we'll find that pitch is primarily associated to the *fundamental frequency* (the repetition rate of the vibration pattern), loudness to the amplitude of the signal, and timbre to the higher frequencies (harmonics) that accompany f_0 (Roederer 1995).

As we can imagine, the most elementary kind of musical sound is a plain, steady single note of constant pitch and intensity such as the one shown in Figure 2.1 (Taylor 1965). However, not all sounds are simple as that and, in general, they have more complex compositions and other interesting characteristics. For example, consider Figure 2.4 where two acoustic waveforms produced by a guitar and a piano playing a same note.

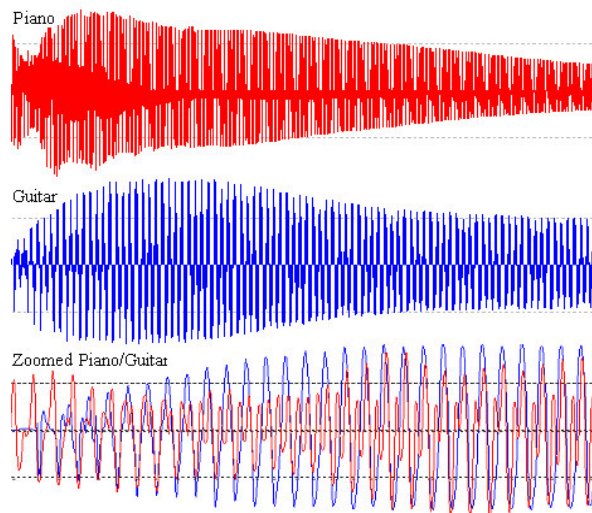


Figure 2.4: Waveforms of a guitar and piano playing a G4

As we can observe, even though they are not periodic signals, they do have some patterns that repeat (so we could think of them of pseudo-periodic). Clearly, the frequency of both signals is well defined (characterizing the fact that they playing the same note G4) but their shapes do change through time (in general, we could characterize them with a particular form that has its major changes at the beginning and then stabilizes at the end, kept in an envelope with exponential decay). The perceived pitch therefore will be the note they are playing and the distinctive characteristics of the instrument, where the latter map into their different harmonics.

We can treat each sound of the above example as a complex signal and obtain its constitutive sine frequencies using *Fourier*. By looking at Figure 2.5, we can notice that both signals make use of similar frequencies with different magnitudes while other frequencies are not shared. We can also see that the form of their envelope and their initial composition are different, therefore providing us with several ways to differentiate the instruments.² In

² For example, when we strike a key of a piano, it produces a highly characteristic burst that decays as the harmonics from the both the note and physical characteristics of the instrument get present. Other instrument, let's say a harmonica, produces sounds without the burst and with a more periodic evolution in time (a decay rate won't be there unless we change the amount of air).

general, some harmonics can be shared between similar notes of different instruments, while different notes of a specific instrument share other harmonics. Also, sounds depend on their harmonics in a different way. Some instruments may depend on several harmonics, while the sound of other instruments is based on just a few harmonics (such is the case of pianos, which depend heavily on the fundamental).

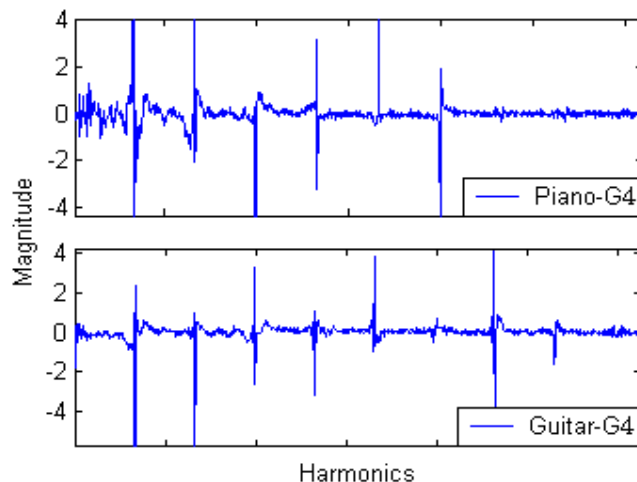


Figure 2.5: Harmonics of a guitar and piano playing a G4

In general, notes can share different frequencies (*upper harmonics* or *overtones*). The *upper harmonics* of a tone are H1 (the equivalent to f_0), H2 ($2 * f_0$), H3 ($3 * f_0$), etc. The relations of the harmonics and notes are the following: H2 is the upper octave of H1, H3 is the twelfth, H4 is the fifteenth, etc. This is illustrated in Figure 2.6 (Roederer 1995).



Figure 2.6: First eight harmonics of G2 (98 Hz) (Roederer 1995)

2.2 Autoregressive Models

Dynamical characteristics of a complex system can be often be inferred from the analysis of stochastic time series model fitted to observations of the

system (Neumaier and Schneider 2001). A standard technique for doing time series modelling is *Linear Prediction (LP)*, which is highly used for modelling audio, speech, and other applications for analysis and synthesis of signals. In general, they allow modelling the periodicity of a sound and changes such as the decay rate when certain restrictions apply. For example, conventional *LP* techniques utilize *window-based autoregressive spectral* modelling and consider that the signal is stationary within the analysis frame (Harma, Juntunen et al. 2001). By considering that a signal $x(t)$ can be expressed as a linear combination of the previous samples, we have an autoregressive model of order o described as

$$\begin{aligned} x(t) &= m_1x(t-1) + m_2x(t-2) + \dots + m_o x(t-o) + b \\ &= b + \sum_{k=1}^o m_k x(t-k), \quad \forall t = \{o+1, \dots, T\} \end{aligned} \quad (2.2.1)$$

where m_k is a set of o coefficients to be estimated from a signal with equally spaced samples till time T , and b is a prediction error that is included to allow for a nonzero mean of the time series. When having n elements for each time step instead of just one, this naturally extends to an n -variate autoregressive model of order o such that

$$\mathbf{x}(t) = \mathbf{b} + \sum_{k=1}^o \mathbf{M}^{(k)} \mathbf{x}(t-k) \quad (2.2.2)$$

being both $\mathbf{x}(t)$ and \mathbf{b} vectors of size $n \times 1$ and having \mathbf{M} as matrix with $n \times n$ components for each one of the k autoregressive time steps till order o (Neumaier and Schneider 2001).

Depending on the nature of the problem, we can modify and complicate (2.2.2) such as by adding noise and having autoregressive components that are also dependent on time (useful when having signals with complex non-stationarities) (Harma, Juntunen et al. 2001). Consequently, diverse methods for estimating the coefficients could be employed for different models.

2.3 Blind Source Separation

A very popular problem in contemporary literature regarding *Source Separation* is *Blind Source Separation (BSS)*,³ which is related to the separation of several sound sources based on their different locations in an auditory scene. In it, several recordings of the same scene are obtained using different “microphones” where each one is placed in a different position (Figure 2.7), so different combinations of the source signals can be used for the separation (Rowe 1999).

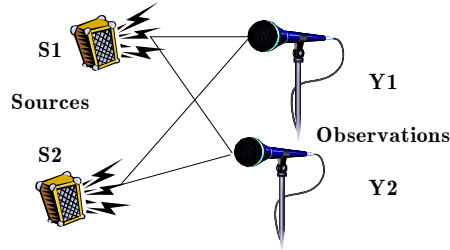


Figure 2.7: Sources and observations for *BSS*

In *BSS*, the “blind” qualification refers to the fact the sources are not observable and nothing is known about their properties, the process used to mix them, nor possible noise. This means, no *a priori* information is specified. However, to simplify the problem, an assumption regarding mutual independence between the signals is taken into consideration. Also, the mixture is commonly believed to be made of a linear combination of the sources (Cardoso 1998).

By considering n sources and m observations, the underlying model is that of n statistically independent signals $s_1(t) \dots s_n(t)$ where m (possibly noisy) generally correlated variables $y_i(t)$ are observed, such that

$$y_i(t) = \sum_{j=1}^n M_{ij} s_j(t) + u_i(t), \quad i = \{1 \dots m\}$$

$$\mathbf{y}(t) = \mathbf{M}\mathbf{s}(t) + \mathbf{n}(t) \tag{2.3.1}$$

³ Other names for *BSS* are *Blind Array Processing*, *Signal Copy*, *Waveform Preserving Estimation*, and *Cocktail Party Problem* (where with the latter it clearly refers to the ability of humans for separating various conversations going on in a cocktail party, concentrating just on one).

where $y_i(t)$ depends on the linear combinations of all the sources $s_j(t)$ with the constant coefficients M_{ij} (the mixing matrix), and the probabilistic nature of this dependence is modelled by the m additive noise signals $u_i(t)$. Given $\mathbf{y}(t)$, $\mathbf{s}(t)$ has to be recovered by finding a “separating” matrix \mathbf{B} , so $\mathbf{B}\mathbf{y}(t)$ will give an estimate of the source signals. By using prior information about the probability distribution of the inputs, this can be obtained by optimising a contrast function, for example entropy, mutual independence, high-order decorrelations, or divergence between the joint distribution of \mathbf{y} and some model (Cardoso 1998; Attias 1999).

BSS primarily exploits the “spatial diversity”, since different sensors receive different mixtures of the sources. Therefore, it depends heavily on the outputs of the sensors, and that task complicates as the number of sensors is reduced.⁴ Other diversities, if they exist, could also be exploited but the approach of *BSS* is essentially spatial. Structures that are not commonly exploited and that could help for cleaner source reconstructions are time structures, spectral diversity, and cyclostationary properties (Cardoso 1998).

Actually, there are different approaches for doing *BSS* such as *Independent Component Analysis (ICA)* (Roberts and Everson 2001), *Bayesian BSS* (Rowe 1999), and *Independent Factor Analysis (IFA)* (Attias 1999), to name a few. The most popular one is the former, which is restricted by several assumptions and therefore is considered as one of the “simplest” methods for doing *BSS*. It requires equal number of sensors and observed mixtures, at most one *Gaussian* source and noiseless data. However, mixing in realistic situations generally includes noise and different numbers of sources and sensors, and using *non-Gaussian* distributions can complicate the derivations. The sources are commonly described by a temporal independent density

⁴ For example, if only one sensor would be used for the separation of the sources considered in Figure 1.3, probably it would be able to learn the distribution of those notes that appear on their own or mixed with different sources. However, the case where notes Eb6 and Eb6 are played as a chord and only once, only one distribution would be learned and therefore *BSS* wont be able to achieve that separation.

model (so as in the normal *BSS* problem) and the temporal statistics of the data are not modelled. In consequence, the model learned would not be affected by permuting the time indices. In addition, we cannot tell what amplitude of the original sources were since the estimated sources subject to an unknown scaling and perturbation (Plumbley, Abdallah et al. in press).

2.4 Dynamical models

One important feature of the model we want to build is that it is to be able to consider temporal dependencies. Most commonly used probabilistic models of time series are descendants of either *Hidden Markov Models (HMMs)* or *State-Space Models (SSM)* (Ghahramani and Hinton 1998). In this section, we present them showing their main characteristics.

2.4.1 Hidden Markov Models

A *HMM* is a model appropriate for dealing with sequential data, therefore having its common applications in modelling molecular biological sequences, identifying musical transitions or recognizing speech (to name some). It represents information about the past using sequences of *Markovian* states $s^{(n)}(t)$, where t relates to a time transition, and n to one of the possible number of different states that we may have. The states are hidden, (hence its name) and only outputs related to each state are observed ($y^{(n)}(t)$). The graphical model of a *HMM* is described by:

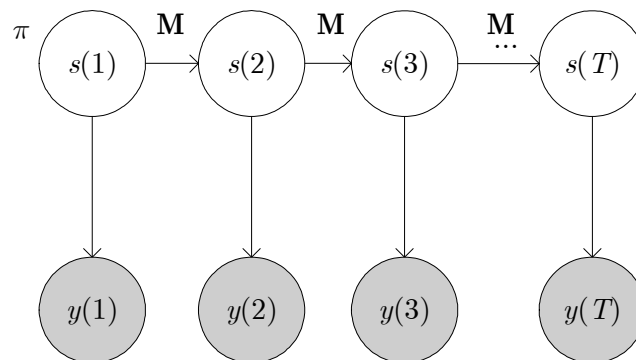


Figure 2.8: Graphical model of a HMM

where $\pi^{(n)}$ is the probability of starting the sequence in state n , and \mathbf{M} is a 1st-order *Markovian* state transition matrix with dimensions $n \times n$ (in which

the element i, j indicates the probability of the transition from state i to state j). As a result of the *Markovian* assumption, by conditioning on a specific state makes its past and future observations statistically independent. The joint probability of a sequence and a specific observation is given by

$$P(y(1)...y(T), s(1)...s(T)) = \pi^{(s(1))} P(y(1) | s(1)) \prod_{t=2}^T \mathbf{M}_{s(t-1)s(t)} P(y(t) | s(t)) \quad (2.4.1)$$

from which we can derive the inference of a state or a sequence of states given the observations ($P(s(1)...s(T) | y(1)...y(T))$). Algorithms for obtaining the sequence that maximizes this conditional probability already exist, being *Viterbi* an efficient one (Forney 1973).

As we can observe, one of the main characteristics of *HMMs* is that they have discrete states and discrete outputs. By using more states, continuous dynamics could be approximated but at the cost of having an exponential growth in the number of calculations. The generalization towards a continuous state exists, leading to *State Space Models*.

2.4.2 State Space Models and Linear Dynamical Systems

The dynamical generalization of *Hidden Markov Models* (2.4.1) directs to *State Space Models* (*SSM*), having an identical graphical structure but the type of nodes are real valued vectors. Therefore, a *SSM* has exactly the same *Markov* properties as a *HMM*, and its states are hidden in exactly the same way. The dependency between the present state vector and the previous state vector is specified through the dynamic equations of the system and the noise model. When these equations are linear and the noise model is *Gaussian*, the state-space model is known as a *Linear Dynamical Systems* (*LDS*) (Ghahramani and Hinton 1998). Its graphical representation is similar to the previously shown, and is illustrated in Figure 2.9.

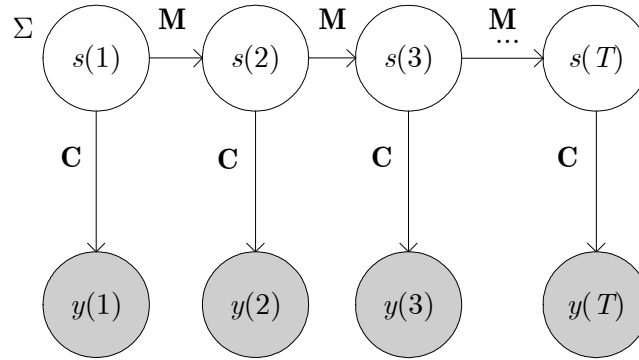


Figure 2.9: Graphical model of a SSM

The model is given by

$$\mathbf{s}(t+1) = \mathbf{M}\mathbf{s}(t) + \mathbf{G}\mathbf{v}(t) \quad (2.4.2)$$

and

$$\mathbf{y}(t) = \mathbf{C}\mathbf{s}(t) + \mathbf{w}(t) \quad (2.4.3)$$

where the transition and output functions are commonly time invariant. In these equations, $\mathbf{s}(t)$ refers to the hidden state variable at time t , $\mathbf{y}(t)$ is the observation at time t , $\mathbf{v}(t) \sim N(0, \mathbf{Q})$ and $\mathbf{w}(t) \sim N(0, \mathbf{R})$, and \mathbf{M} , \mathbf{G} and \mathbf{C} are matrices (commonly squared related to the size of the vector $\mathbf{s}(t)$). Also, for the initial conditions, $P(\mathbf{s}(1)) \sim N(0, \Sigma)$.

The use of a linear models and *Gaussian* distributions is justifiable for a number of reasons. Often such a model is adequate for the purpose at hand, and when non-linearities do exist, the typical engineering approach is to linearize about some nominal point or trajectory, achieving a perturbation model or error model. *Linear systems* are desirable in that they are more easily manipulated with engineering tools, and linear system (or differential equation) theory is much more complete and practical than non-linear. In regards to the use of *Gaussian* distributions, it turns out that the mathematics involved in possible derivations get vastly simplified and tractable. (Maybeck 1979). Besides these reasons, an efficient algorithm called the *Kalman Filter* can be used for performing inference regarding $P(\mathbf{s}(t) | y(1) \dots y(t))$ (Murphy 1998).

2.4.2.1 Kalman Filter

The *Kalman filter* is the general solution to produce an estimate $\tilde{\mathbf{s}}(t)$ of the state model $\mathbf{s}(t)$ using measurements until time t , as to minimize the mean square error between the estimate and the state (where states refer to any quantities of interest involved in a dynamic process). It does not assume wide-sense stationary signals but ones with certain structure, dealing with simple irregularities. In consequence, the *Kalman filter* has been used extensively for many diverse applications such as in navigational and guidance systems, radar tracking, sonar ranging, and satellite orbit determination (Leon-Garcia 1993; Doyle 1995).

Within the class of linear estimators, the *Kalman Filter* gives a linear, unbiased, and minimum error variance recursive algorithm to estimate the unknown states of a dynamic process from noisy data taken at discrete real-time intervals. For *Gaussian* random variables with white noise,⁵ the *Kalman filter* is the optimal linear predictor-estimator (and for variables of forms other than *Gaussian* the estimator is the best only within the class of linear estimators).⁶ It processes all available measurements, regardless of their precision, to estimate the current value of the variables of interest. Also, it makes use of knowledge of the system and measurement device dynamics, the statistical description of the system noises, measurements errors and uncertainty in the dynamics models, and any available information about the initial conditions of the variables of interest (Maybeck 1979; Doyle 1995).

The algorithm for performing the filtering is well-known and can be consulted in (Leon-Garcia 1993) while detailed derivations of the whole theory are in

⁵ Whiteness implies that the noise value is not correlated in time and that has equal power at all frequencies.

⁶ For example, possible optimal estimates may be: the mean (the centre of probability mass estimate), the mode (the value of \mathbf{s} that has the highest probability, locating the peak of the density) or the median (the value of \mathbf{s} such that half of the probability weight lies to the left and half to the right) (Maybeck 1979).

(Chui and Chen 1987). It requires knowledge of the second-order statistics of the noise of process being observed and of the measurement noise in order to provide the solution that minimises the mean square error between the true state and the estimate of state (Leon-Garcia 1993; Doyle 1995). In the absence of any higher order statistics, there is no better form to assume than the *Gaussian* density. The first and second order statistics completely determine a *Gaussian* density, unlike most densities that require an endless number of orders to specify their shape entirely (Maybeck 1979).

Chapter 3

Building a probabilistic model

3.1 General Idea

We know that simple instrument notes are signals with different frequency content which varies and evolves through time, and that their physical attributes can be used to differentiate them (2.1.2). We also know that instant polyphonic observations can be produced by adding the frequency contents of individual sounds for different time steps (Figure 3.1, where each \mathbf{f} represents the frequencies of sound 1 till sound S). Given these facts, the idea is to create a model for doing the inverse procedure: use the characteristics of the sources to separate them from a polyphonic observation (this is, define which instrument notes are active at every time step, and which frequencies from the observation correspond to each).

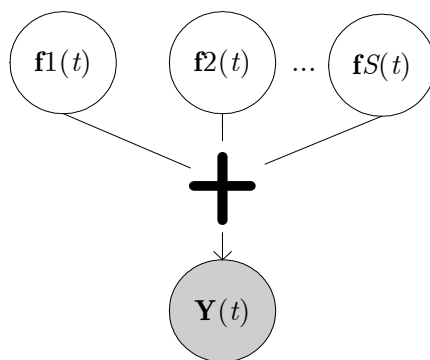


Figure 3.1: Instant mixing of different sources

If we were able to create a probabilistic model of the above process, by using Bayesian probability we could infer the inverse situation. However, we only know that we will have the polyphonic observations, but not the

sources (whose densities are required for the model). If some models of the sources would exist (which may or may not be present in the observation, and are required to have some points of reference for the comparisons), our model would then be complete. From this situation, we derive the constitutive elements of our probabilistic model: a part for modelling each source (which we will call the “*Sound Model*”), and a part for integrating the sources and modelling the observation (which we'll refer as the “*Unifying Model*”). The following sections describe these constitutive elements with their corresponding form.

3.1.1 Sound model

The *Sound Model* has to exploit the characteristics of each instrument note to emulate its behaviour. In a signal, these characteristics can be primarily described in terms its constitutive harmonics and their corresponding evolution in time.

As the probabilistic component for creating the *Sound Model*, an *AR* model (2.2) was chosen. This selection was considered appropriate since *AR* models have considerable benefits such as learning periodic characteristics (while the instrument notes can be considered semi-periodic, 2.1.2) and their decay rates (which normally exist in simple sounds), and are also capable to model real-valued outputs (which are the values that amplitude of the sound signals can take).

In principle, an *AR* would be enough to characterize an instrument note. However, we may want to make our model more powerful by not using information about the form of the signal, but instead about the constitutive harmonics of the source and model how it is that they evolve in time (which is also in a semi-periodic way, Figure 3.2). To do this, the *Sound Model* involves the use of the *Short Term Fourier Transform (STFT)* to obtain the *Fourier* components of the signal through different time steps (2.1.1.2). If we consider the use of a small window and assume that the signal is stationary within the analysis frame (which can be quite reasonable since our signals are

semi-periodic), then this turns out to be a *window-based autoregressive spectral model* (2.2).

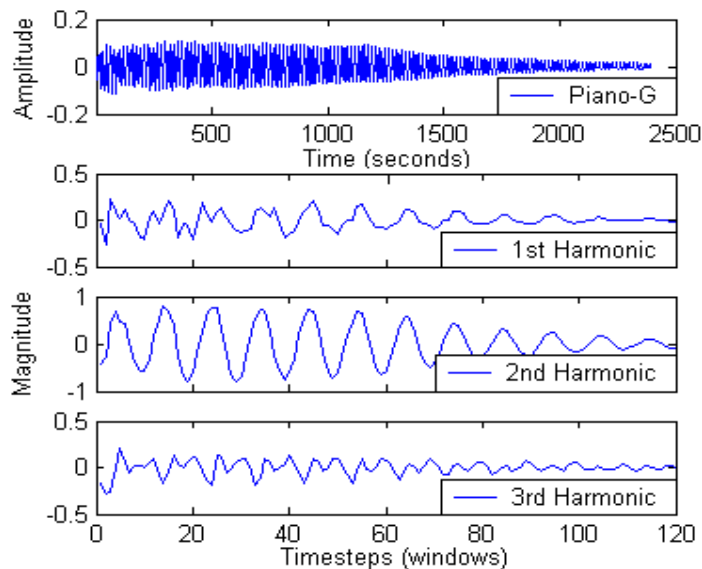


Figure 3.2: Evolution of an instrument note and of the real components of its first three harmonics¹

For the simplicity of our model, we did not consider the use of *Temporal AR models* for this task, but of n -variate *AR* models of order o (2.2). Undoubtedly, difficulties were thought to happen at the beginning of the note (where the most complex changes occur, 2.1.2). However, the n -variate *AR* models were considered powerful enough for modelling the non-complex stationarities that are major in the rest of the signal. Also, they have other benefits besides being simpler than the *Temporal AR* models, such as allowing us to control the amount of information to be used for inferring present values by changing their *Markovian* order (and, when mixed with the *STFT*, we can also modify the size of the window to condense more information per each *AR* time step). In addition, n -variate *AR* models have no restriction about the number of elements to be considered for the size of the vectors, which is also beneficial for generalizing the model to learn data from either normal sounds or n -dimensional vectors that result of the application of the *STFT*.

¹ The imaginary parts of the components also have a semi-periodic behaviour.

3.1.2 Unifying model

The model to be built has to unify several *Sound Models* (using them as prior knowledge of the sources), and to perform inference about the presence and the contents of each source signal for each time step given a polyphonic observation.

For the *Unifying model*, we chose to use a *Linear Dynamical System* (2.4.2). Although it has practical benefits such as being one of the simplest models for modelling continuous temporal behaviour, having tractability benefits regarding the use of *Gaussian* distributions, and the fact that inference with the *Kalman Filter* is known to work in many applications (2.4.2.1), it also fits to some extent with the description of our problem. Polyphonic observations, when thought as instant mixtures, are also linear combinations of the sources. This happens both for considering the signals without any processing (such a complex signal is the addition of two other signals, 2.1.1), and when processing them previously with the *Fourier Transform* (since the *FT* of a mixture equals the addition of the individual transforms, 2.1.1.1).

However, the use of a *LDS* is not instantaneous since we need to modify it for being capable of supporting two or more *AR* models. Also, it would be desirable to exploit not only the information regarding autoregressive contents of order one, but of order o . The latter case is solved directly by using the n -variate *AR* models of order o , however, the former case requires more structural changes.

The approach that was considered for this problem is related to the use of hybrid models (models with discrete transitions with linear dynamics) with the “switching” perspective of Ghahramani and Murphy. The idea is to have a bank of S different *Sound Models*, and switch between them. For doing this, we consider the inclusion of a hidden switching variable c , which is related to each linear model. By assigning it as Boolean, this allows us to say whether a sound is active or not (on/off) in the linear mixture $\mathbf{Y}(t)$, allowing different permutations of the models to be applied for each time step.

Therefore, if we would know that c is, then we would be able to apply the corresponding linear model for each time step. In Figure 3.3 we show the corresponding graphical model, where $\mathbf{C}(t)$ is a vector that contains all the corresponding switches.²

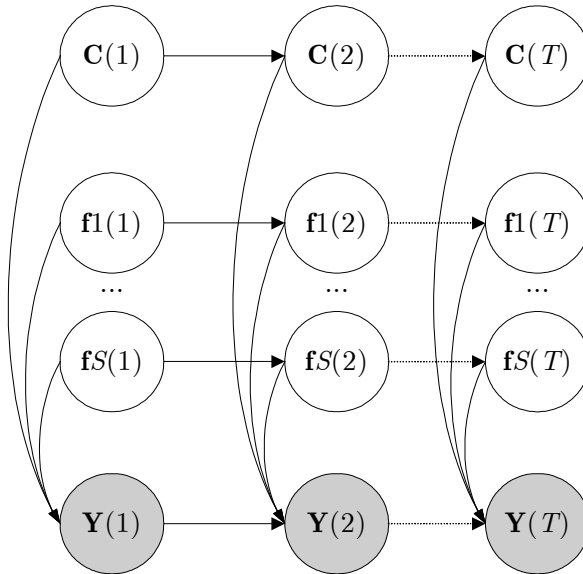


Figure 3.3: Switching Linear Dynamical System

Mathematically, this is described by

$$\begin{aligned} \mathbf{Y}(t) &= c^{(1)}(t)\mathbf{f}^{(1)}(t) + c^{(2)}(t)\mathbf{f}^{(2)}(t) + \dots + c^{(S)}(t)\mathbf{f}^{(S)}(t) + \mathbf{w}(t) \\ &= \mathbf{C}(t)\mathbf{F}(t) + \mathbf{w}(t) \end{aligned} \quad (3.1.1)$$

where $\mathbf{w}(t) \sim N(0, \sigma^2)$ is an independent *Gaussian* noise source, $\mathbf{F}(t)$ is a set of S different linear models ($\mathbf{F}(t) \in \mathfrak{R}^{N \times S}$, N standing for the size of each vector \mathbf{f}), $\mathbf{C}(t)$ is a vector of size S that contains all the corresponding Boolean switches, and $\mathbf{Y}(t)$ is the observed signal at that time t ($\mathbf{Y}(t) \in \mathfrak{R}^N$).

From equation (3.1.1) we can clearly see how this is in a way related to *BSS* problem (2.3). If we were to make inference regarding the activation of each linear model for each time step, we would need to find what the mixing vector $\mathbf{C}(t)$ is. Clearly, for one time step this would not look as complicated as in *ICA* approaches where we would have a mixing matrix with real coefficients. In here, we have a Boolean vector whose contents are 2^S possible

² This model was initially proposed by (Ghahramani and Hinton 1998).

combinations (and maybe we could even think on doing a direct exploration if S is small). However, the real problem is not that trivial since the mixing matrix is not constant as in *ICA*, but now depends on the time until time step T . This means that, in theory, we could have $(2^S)^T$ different activation sequences and exhaustive searches are prohibitive.³

3.2 MSE as an optimality measure

Kalman Filtering uses the *Mean Square Error (MSE)* as an optimality measure between the modelled time series and the observed time series. If we were to derive our equations using then *MSE*, we should be careful about one point in particular: dephase problems. Sounds are phase-less to the human ear, but not to the *MSE*. To exemplify, consider Figure 3.4 where we present two complex signals with a different phase (“Signal 1” and “Dephased Signal 1”), and a different signal which has the same phase as the first one (“Signal 2”). Depending on the amount of dephase, the *MSE* between the similar cases could be greater than the one between the distinct cases (in both the raw signals and in their harmonics).

This could be a serious problem if we were to compare the signals deterministically, but not if we have a probabilistic model of the sound (the *AR* model). As we already stated, the *AR* models have one special characteristic: given the past (as much as we consider to include), they will provide an output for the present. Therefore, if we were to sample the original signals into smaller time steps, then the output should be “on phase”. This closely depends on the size of the window, and we would expect that as we make the window smaller, the dephase problems would be minimized. Cases regarding sounds with different volume levels might fall in a similar situation, where the *AR* model should provide the corresponding output depending on the past (so if the amplitude was high, then it would preserve that aspect in the output).

³ For example, by considering just 3 instruments that could be playing or not in 10 time steps, this would lead to 1,073,741,824 possible different activation sequences.

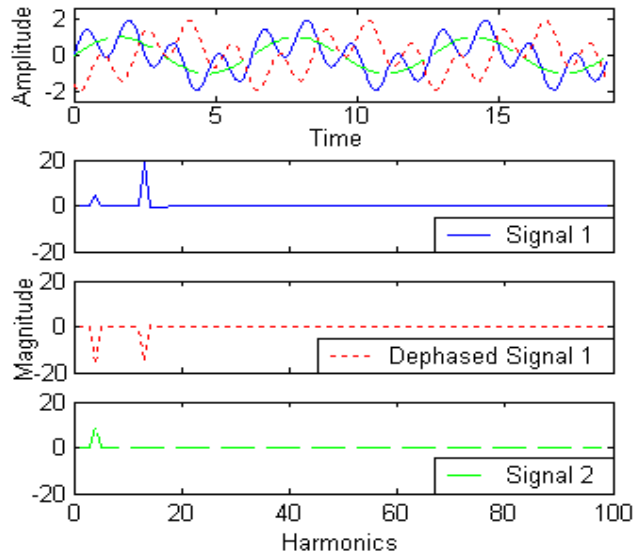


Figure 3.4: Dephased signals

As we can see, the measure of the distances for the comparisons could be justified by the behaviour of the AR , and a MSE could be used as our optimality measure. However, we still have to bear in mind that our main problem is the separation of the sources, and this is coupled with performance of the AR models. This means that if we are not able to separate the information of one time step, then the outputs of the linear models will not be good enough for next time steps.

Chapter 4

Deriving the model

4.1 Global Structure

As detailed in the previous chapter, our model for *PCSS* comprises *Switching Linear Dynamical Systems*, where n -variate autoregressive models of order o are used. In this chapter, we present the mathematical derivation of the model, including the inference relations. It is worth to add that this model started with the simplest assumptions, and an incremental procedure was followed for allowing the addition of dependencies (therefore making it more real and more complex). We present the final model from which all our previous simpler models can be obtained.

The probabilistic model to be derived, initially can be diagrammatically simplified by the graphical model shown in Figure 1.1.¹

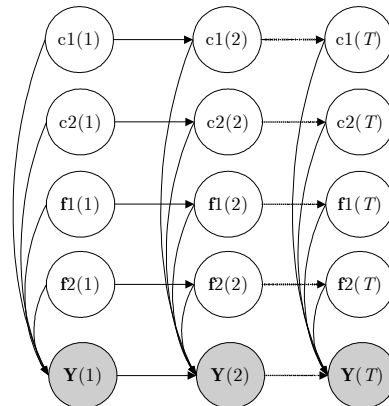


Figure 4.1: Graphical model of two mixed sources with Boolean switches

¹ A table describing the variables and the notation used throughout this section is provided in the Appendix A.1

By considering only two linear models for the source signals and linear mappings towards the observation,¹ we have the following equation

$$\mathbf{Y}(t) = c^{(1)}(t)\mathbf{f}^{(1)}(t) + c^{(2)}(t)\mathbf{f}^{(2)}(t) + \mathbf{w}(t) \quad (4.1.1)$$

where $c^{(s)}(t)$ is a Boolean switching variable, $\mathbf{f}^{(s)}(t)$ is a linear model of a source with $\mathbf{f}^{(s)}(t) \in \Re^N$ (4.1.2), $\mathbf{w}(t) \sim N(0, \sigma^2)$ is an independent *Gaussian* noise source with spherical (isotropic) covariance, t is a specific time step from the range $\{1, 2, \dots, T\}$, s is a sound from $\{1, 2\}$, and $\mathbf{Y}(t)$ is the observed signal at that each instant and has the same size as $\mathbf{f}^{(s)}(t)$.²

$$\mathbf{f}^{(i)}(t) = \begin{bmatrix} f_1^{(i)}(t) \\ f_2^{(i)}(t) \\ \dots \\ f_N^{(i)}(t) \end{bmatrix} \quad (4.1.2)$$

Given that $\mathbf{Y}(t)$ is modelled as a *Gaussian*, $P(\mathbf{Y}(t) | \mathbf{C}(t), \mathbf{f}^{(1)}(t), \mathbf{f}^{(2)}(t))$ is given by:

$$P(\mathbf{Y}(t) | \mathbf{C}(t), \mathbf{f}^{(1)}(t), \mathbf{f}^{(2)}(t)) \propto \exp\left\{-\frac{1}{2\sigma^2}[\mathbf{Y}(t) - c^{(1)}(t)\mathbf{f}^{(1)}(t) - c^{(2)}(t)\mathbf{f}^{(2)}(t)]^2\right\} \quad (4.1.3)$$

Each linear model $\mathbf{f}^{(s)}$ is approximated by an autoregressive model and each switch by a switching probability $P(c^{(s)}(t))$. The next section describes how they can be parameterised.

4.1.1 Defining a probability distribution for each switch

As already described, each switching variable $c^{(s)}(t)$ tells if the corresponding sound $\mathbf{f}^{(s)}$ is playing at time t . In order to create a probability distribution for each switch, we can use some *a priori* knowledge regarding the behaviour of c .

Without considering that a note is normally played for some specific intervals of a unit (for example 1/16, 1/8, 1/4, 1/2 and a whole), one certain fact regarding the probability distribution of a sound for a specific time step is that it is highly correlated with the one of its immediate previous time steps.

¹ The use of two linear models is just for simplicity, since later we will generalize this situation. However, (4.1.1) is our start point for all derivations.

² In general, s refers to a sound from $\{1, 2, S\}$.

This means, if an instrument was playing some time steps ago, it is highly probable that it will be still playing (and more probable if our measure for the time steps is quite small), and similarly for the case when it was not playing. For example, consider that we know that a specific instrument started playing 2.5 ms ago, and we want to know whether it will still be active for the next 2.5 ms. By taking into account that our time division is very small, the most probable thing is that it will keep on playing for more time steps, but being less probable to still be active as time goes on (Figure 4.2). In fact, without considering any prior knowledge of note intervals, we could approximate the conditional probability $P(c^{(s)}(t) = 1 \mid c^{(s)}(t-1) = 1, \dots, c^{(s)}(t-o) = 1)$ of a stationary sound by an exponential distribution and learn the decay rate by focusing on our time measures and the average of the time it plays.

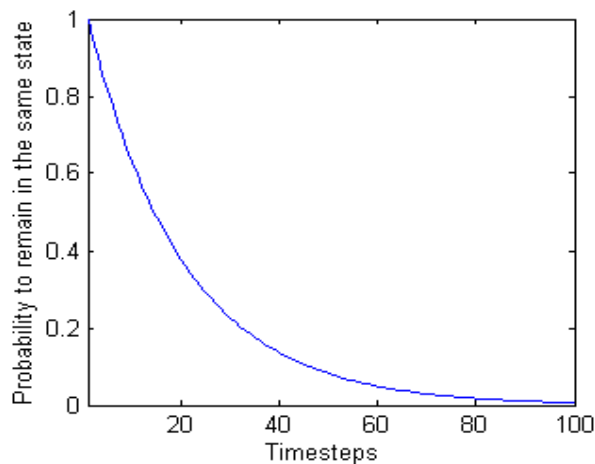


Figure 4.2: Exponential distribution of a stationary sound

The stationary principle is important to be able to make this approximation. This means, we won't expect a sound to be active for one time step, inactive for the other, and so on, but to be active for some consecutive time steps, and inactive for successive ones. Of course, in observations from the real world this assumption can be justified as we make our time window smaller.

Let $P(c(t) \mid c(t-1), c(t-2), \dots, c(t-O))$ be the probability of the switching activity for a sound (on/off) at time t given the status of the previous time

steps. As an example, consider a *conditional probability table* (*CPT*) for an order 3 Markovian model, such as

$c(t-3)$	$c(t-2)$	$c(t-1)$	$P(c(t)=on Past)$	$P(c(t)=off Past)$
off	off	off	$1-\gamma^3$	γ^3
off	off	on	γ	$1-\gamma$
off	on	on	γ^2	$1-\gamma^2$
on	off	off	$1-\gamma^2$	γ^2
on	on	off	$1-\gamma$	γ
on	on	on	γ^3	$1-\gamma^3$

Table 4.1: CPT of stationary cases for an order 3 Markovian model

where the probability of remaining in a particular status is determined by a fixed constant γ ($\gamma \in \{0,1\}$), while the probability of changing is therefore $1-\gamma$. The non-stationary cases such as on-off-on, and off-on-off clearly complicate things. To be in some accordance with our example, we can give them values like

$c(t-3)$	$c(t-2)$	$c(t-1)$	$P(c(t)=on Past)$	$P(c(t)=off Past)$
off	on	off	$1-\gamma$	γ^r
on	off	on	γ^r	$1-\gamma$

Table 4.2: CPT of non-stationary cases for an order 3 Markovian model

Hence, by considering the values of the *CPT* and grouping them, then we could characterize the probabilities of changing and not changing of state at time t given the status of the previous states by

$$P(\text{change}(t) | c(t-1), c(t-2), \dots, c(t-O)) = 1-\gamma^r \quad (4.1.4)$$

and

$$P(\sim \text{change}(t) | c(t-1), c(t-2), \dots, c(t-O)) = \gamma^r \quad (4.1.5)$$

where r is the number of the last time steps where the states until time step $t-1$ were the same. Depending on the value γ , ($\gamma \in [0,1]$), we can get distributions such as the one shown in Figure 4.2. In Figure 4.3 we exemplify this by considering different values for γ , and an order 1 Markovian model.

Undoubtedly, the non-stationary cases were defined to have to a simplification of the probability densities. However, the values for those

cases can be difficult to parameterise in a higher-order *AR* since outliers obviously can exist. For the sake of simplicity, an initial *AR* model of order 1 with the Markovian assumption for maintaining/changing the status can be defined as

$$P(c^{(s)}(t) | c^{(s)}(t-1)) = \begin{cases} \gamma & \text{for } c^{(s)}(t) = c^{(s)}(t-1) \\ 1 - \gamma & \text{for } c^{(s)}(t) \neq c^{(s)}(t-1) \end{cases} \quad (4.1.6)$$

From (4.1.6) we can see that as γ approaches 0.5, changing from on to off would be equivalent to tossing an unbiased coin. However, as γ approaches one, the sound will remain on the same state, having some agreement with the stationary assumptions already exposed.

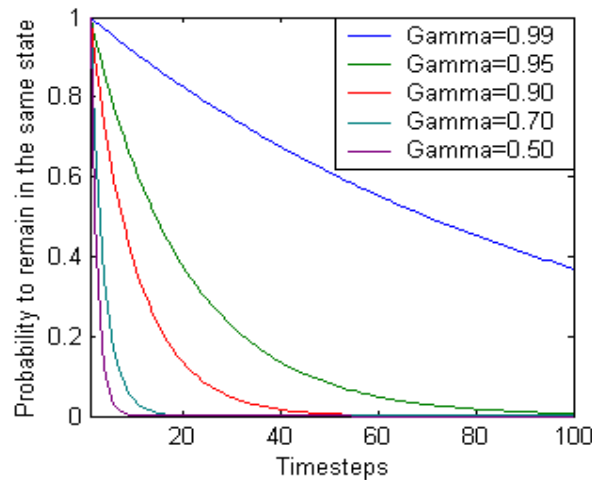


Figure 4.3: Approximations to an exponential distribution of a stationary sound

4.1.2 Learning an autoregressive model for each sound

Let the elements of each vector \mathbf{f} for time step t be approximated by

$$\mathbf{f}(t) = \mathbf{M}\mathbf{g}(t) + \mathbf{w}(t) \quad (4.1.7)$$

where $\mathbf{w}(t) \sim N(\mathbf{0}, \tau^2)$ is an independent *Gaussian* noise source with spherical covariance and bias $\mathbf{0}$ (required to take the approximation towards a zero mean), and \mathbf{g} is a vector that contains all the N components of \mathbf{f} from its previous time step till order O (where $O < T$).³

³ For clearness, indices $s \in \{1, 2, \dots, S\}$ for c , \mathbf{g} , \mathbf{f} , \mathbf{M} , \mathbf{b} and τ^2 will be dropped until required later (by now, all parameters make reference to just one specific sound s).

$$\mathbf{g}(t) = [f_1(t-1) \dots f_N(t-1) \ f_1(t-2) \dots f_N(t-2) \dots f_1(t-O) \dots f_N(t-O)]^T \quad (4.1.8)$$

For a clearer explanation of some future derivations, imaginary smaller vector till the order O of the autoregressive model will delimit vector \mathbf{g} , such that:

$$\mathbf{g}(t) = [\mathbf{g}^{(o1)}(t) \ \mathbf{g}^{(o2)}(t) \ \dots \ \mathbf{g}^{(oO)}(t)]^T = [\mathbf{f}(t-1) \ \mathbf{f}(t-1) \ \dots \ \mathbf{f}(t-O)]^T. \quad (4.1.9)$$

Regarding \mathbf{M} , it is a $N \times ON$ matrix whose size also depends on the order of our regressive model. Similarly, imaginary smaller matrices until the order O will be used to draw up its boundaries.⁴

$$\mathbf{M} = [\mathbf{M}^{(o1)} \mathbf{M}^{(o2)} \dots \mathbf{M}^{(oO)}] \quad (4.1.10)$$

By modelling \mathbf{f} as a *Gaussian*, we have that $P(\mathbf{f}(t) | \mathbf{g}(t))$ is

$$P(\mathbf{f}(t) | \mathbf{g}(t)) = \frac{1}{(2\pi\tau^2)^{|\mathbf{f}|/2}} \exp\left\{-\frac{1}{2\tau^2}[\mathbf{f}(t) - \mathbf{M}\mathbf{g}(t) - \mathbf{b}]^2\right\} \quad (4.1.11)$$

where the optimal \mathbf{b} , τ^2 and \mathbf{M} had to be determined. Our approach for this was to get the *Maximum Likelihood Estimate* for each parameter. For the interested reader, a detailed derivation of each parameter can be found in the appendix (A.1).

The optimal \mathbf{b} , τ^2 and \mathbf{M} are given by

$$\mathbf{b} = \frac{1}{T-1} \sum_{t=2}^T [\mathbf{f}(t) - \mathbf{M}\mathbf{g}(t)] = \tilde{\mathbf{f}} - \mathbf{M}\tilde{\mathbf{g}} \quad (4.1.12)$$

$$\tau^2 = \frac{1}{T|\mathbf{f}|} \left([\mathbf{f}(1) - \langle \mathbf{f}(1) \rangle]^2 + \sum_{t=2}^T [(\mathbf{f}(t) - \tilde{\mathbf{f}}) - \mathbf{M}(\mathbf{g}(t) - \tilde{\mathbf{g}})]^2 \right) \quad (4.1.13)$$

$$\mathbf{M} = \left(\sum_{t=2}^T (\mathbf{f}(t) - \tilde{\mathbf{f}})(\mathbf{g}(t) - \tilde{\mathbf{g}})^T \right) \left(\sum_{t=2}^T (\mathbf{g}(t) - \tilde{\mathbf{g}})(\mathbf{g}(t) - \tilde{\mathbf{g}})^T \right)^{-1} \quad (4.1.14)$$

where the average of the constant vectors $\mathbf{f}(t)$ and $\mathbf{g}(t)$ are, respectively, $\tilde{\mathbf{f}}$ and $\tilde{\mathbf{g}}$.⁵ Hence, each linear model $\mathbf{f}^{(s)}$ has $\mathbf{M}^{(s)}$, $\mathbf{b}^{(s)}$ and $\tau^{(s)2}$ has its corresponding parameters, and they decouple when $\mathbf{M}^{(s)}$ is calculated first.

⁴ As an example, when we only use a first order autoregressive model, we will have that $\mathbf{g}(t)$ equals $\mathbf{g}^{(o1)}(t)$ (or $\mathbf{f}(t-1)$), and that \mathbf{M} is $\mathbf{M}^{(o1)}$.

⁵ Intuitively, \mathbf{b} is the difference between the real components and the predicted ones (by using the autoregressive matrix), averaged by the number of time steps in which the autoregressive model is used. The variance τ^2 has a similar explanation, but now decrementing the difference by including the estimated mean values given by \mathbf{b} to second power, and then averaged trough all time steps and

4.2 Inference

Given a sequence of observation vectors $\mathbf{Y}(t)$, we want to make inference regarding the presence or absence of each sound s for a specific time step (this is, we want to know if each $c^{(s)}(t)$ is 0 or 1, where 0 stands for absence and 1 for presence of sound s). Also, we are interested in separating each signal by determining what $\mathbf{f}^{(s)}(t)$ is. In probabilistic terms, these inferences can be expressed, respectively, as $P(\mathbf{C}(t) | \mathbf{Y}(1), \dots, \mathbf{Y}(t))$ and $P(\mathbf{f}^{(s)}(t) | \mathbf{Y}(1), \dots, \mathbf{Y}(t))$.

The joint posterior probability conditioned on the observations is

$$J = P(\mathbf{C}(1), \dots, \mathbf{C}(T), \mathbf{f}^{(1)}(1), \dots, \mathbf{f}^{(1)}(T), \mathbf{f}^{(2)}(1), \dots, \mathbf{f}^{(2)}(T) | \mathbf{Y}(1), \dots, \mathbf{Y}(T)) \quad (4.2.1)$$

and by using *Bayes theorem*, it can be re-expressed as

$$J = \frac{P(\mathbf{Y} | \mathbf{C}, \mathbf{f}^{(1)}, \mathbf{f}^{(2)})P(\mathbf{C}, \mathbf{f}^{(1)}, \mathbf{f}^{(2)})}{P(\mathbf{Y})} \quad (4.2.2)$$

where $\mathbf{C} = \mathbf{C}(1), \dots, \mathbf{C}(T)$, $\mathbf{f}^{(1)} = \mathbf{f}^{(1)}(1), \dots, \mathbf{f}^{(1)}(T)$, $\mathbf{f}^{(2)} = \mathbf{f}^{(2)}(1), \dots, \mathbf{f}^{(2)}(T)$, and $\mathbf{Y} = \mathbf{Y}(1), \dots, \mathbf{Y}(T)$.

Basically, from (4.2.2) we can get all the quantities in which we are interested. For example, the probability of a sequence $\mathbf{C}(1), \dots, \mathbf{C}(T)$, would be therefore given by

$$P(\mathbf{C} | \mathbf{Y}) = \int_{\mathbf{f}^{(1)}, \mathbf{f}^{(2)}} J \, d\mathbf{f}^{(1)} d\mathbf{f}^{(2)} \quad (4.2.3)$$

whilst the probability of a sequence $\mathbf{f}^{(1)}(1), \dots, \mathbf{f}^{(1)}(t)$, by

$$P(\mathbf{f}^{(1)} | \mathbf{Y}) = \int_{\mathbf{C}, \mathbf{f}^{(2)}} J \, d\mathbf{C} d\mathbf{f}^{(2)} \quad (4.2.4)$$

(and analogously for $\mathbf{f}^{(2)}(1), \dots, \mathbf{f}^{(2)}(t)$ but integrating with respect to $\mathbf{f}^{(1)}$).

If we would not be using hidden nodes, then the required computations could be performed analytically. However, in the presence of hidden nodes, the

the number of components of \mathbf{f} . Finally, \mathbf{M} is related to the real components and the regressive ones (where again, \mathbf{b} has a decrementing effect).

integrals (marginals over $\mathbf{f}^{(s)}(1), \mathbf{f}^{(s)}(T)$) are computationally non-trivial and there does not exist a known exact tractable algorithm for carrying out inference. This is due to the fact that, even the sources are independent, when conditioned on $\mathbf{Y}(1), \dots, \mathbf{Y}(T)$ they correlate and then an exponential number of hidden configurations appear (resulting into a *Gaussian* mixture with S^T components) (Ghahramani and Hinton 1998; Murphy 1998).

Mean field theory (Parisi 1998) provides an alternative on the inference problem. In its simplest form, it assumes that nodes fluctuate independently around their mean values.⁶ This makes possible a tractable approximation to the posterior probability $P(Q|J)$, where J makes reference to the observed variables and Q to the unobserved or hidden ones.

$$P(Q|J) = \prod_i P(Q_i | J) \quad (4.2.5)$$

In this regard, the probability of the unobserved variables given the observations is therefore

$$P(Q|J) = P(\mathbf{C}|J)P(\mathbf{f}^{(1)}|J)P(\mathbf{f}^{(2)}|J) \quad (4.2.6)$$

that, just to associate it with variational methods, is rewritten in terms of a simplified factorised distribution Q as

$$\begin{aligned} Q &= q(\mathbf{C})q(\mathbf{f}^{(1)})q(\mathbf{f}^{(2)}) \\ &\approx \prod_t q(\mathbf{C}(t))q(\mathbf{f}^{(1)}(t))q(\mathbf{f}^{(2)}(t)) \end{aligned} \quad (4.2.7)$$

An approximation J of Q can be obtained by minimizing the *Kullback-Leibler divergence* (KL) (Cover and Thomas 1991). This function is a measure of the distance between two probability densities and approaches to zero when they are more similar ($KL(Q(x) || J(x)) \geq 0$, with equality when $Q=J$). Then, the KL function regarding both distributions is

⁶ The intuition behind mean field methods is that in dense graphs each node is subject to influences from many other nodes, thus to the extent that each influence is weak and in the total influence is roughly additive. In consequence, each node can be approximated by its mean value (particularly, in situations where the law of the large numbers can be applied) (Jaakkola and Jordan 1999).

$$\begin{aligned}
KL(Q(x) || J(x)) &= \int Q(x) \log \left(\frac{Q(x)}{J(x)} \right) dx \\
&= \langle \log(Q(x)) \rangle_{Q(x)} - \langle \log(J(x)) \rangle_{Q(x)} \quad (4.2.8)
\end{aligned}$$

where

$$\langle \log(Q(x)) \rangle_{Q(x)} = \left\langle \log \left(\prod_{t=1}^T q(\mathbf{C}(t)) q(\mathbf{f}^{(1)}(t)) q(\mathbf{f}^{(2)}(t)) \right) \right\rangle_{Q(x)} \quad (4.2.9)$$

and

$$\begin{aligned}
\langle \log(J(x)) \rangle_{Q(x)} &= \left\langle \log \left(\frac{P(\mathbf{Y} | \mathbf{C}, \mathbf{f}^{(1)}, \mathbf{f}^{(2)}) P(\mathbf{C}, \mathbf{f}^{(1)}, \mathbf{f}^{(2)})}{P(\mathbf{Y})} \right) \right\rangle_{Q(x)} \\
&\propto \left\langle \log \left(P(\mathbf{Y} | \mathbf{C}, \mathbf{f}^{(1)}, \mathbf{f}^{(2)}) P(\mathbf{C}) P(\mathbf{f}^{(1)}) P(\mathbf{f}^{(2)}) \right) \right\rangle_{Q(x)} \quad (4.2.10)
\end{aligned}$$

To estimate the optimal form of $q(\mathbf{C}(t))$, $q(\mathbf{f}^{(1)}(t))$ and $q(\mathbf{f}^{(2)}(t))$, we used again *Maximum Likelihood* (the corresponding derivations are given in the appendix, A.3). The partial optimal equations for autoregression of order 1 are:

$$q(\mathbf{f}^{(s)}(t)) \alpha \exp \left[\begin{aligned} &\left\langle \log \left(P(\mathbf{Y}(t) | \mathbf{C}(t), \mathbf{f}^{(s)}(t), \mathbf{f}^{(k)}(t)) \right) \right\rangle_{q(\mathbf{C}(t))q(\mathbf{f}^{(k)}(t))} + \left\langle \log \left(P(\mathbf{f}^{(s)}(t) | \mathbf{g}^{(s)}(t)) \right) \right\rangle_{q(\mathbf{g}^{(s)}(t))} \\ &+ \left\langle \log \left(P(\mathbf{f}^{(s)}(t+1) | \mathbf{g}^{(s)}(t+1)) \right) \right\rangle_{q(\mathbf{f}^{(s)}(t+1))q(\mathbf{f}^{(s)}(t-1)) \dots q(\mathbf{f}^{(s)}(t-o+1))} \end{aligned} \right] \quad (4.2.11)$$

$$q(\mathbf{C}(t)) \alpha \exp \left[\begin{aligned} &\left\langle \log \left(P(\mathbf{Y}(t) | \mathbf{C}(t), \mathbf{f}^{(1)}(t), \mathbf{f}^{(2)}(t)) \right) \right\rangle_{q(\mathbf{f}^{(1)}(t))q(\mathbf{f}^{(2)}(t))} + \left\langle \log \left(P(\mathbf{C}(t) | \mathbf{C}(t-1)) \right) \right\rangle_{q(\mathbf{C}(t-1))} \\ &+ \left\langle \log \left(P(\mathbf{C}(t+1) | \mathbf{C}(t)) \right) \right\rangle_{q(\mathbf{C}(t+1))} \end{aligned} \right] \quad (4.2.12)$$

where in (4.2.11), by using (s, k) we make reference to the two possible combinations $\{(1,2), (2,1)\}$ for $\mathbf{f}(t)$ (so we will only write one formula).

It is worth to comment that temporal dependencies of order 1 and order O were used respectively for $P(\mathbf{C}(t))$ and $P(\mathbf{f}^{(s)}(t))$. As we can observe from (4.2.12), the equations derived by using *Maximum Likelihood* are smoothing equations whose contents are related to the autoregressive order. We have to add that these equations are not completely true for the beginning and the end of a time sequence. When t equals 1 we have that $P(\mathbf{C}(t) | \mathbf{C}(t-1)) = P(\mathbf{C}(t))$, and that $P(\mathbf{f}^{(s)}(t) | \mathbf{g}^{(s)}(t)) = P(\mathbf{f}^{(s)}(t))$. Besides,

when t equals T we will also have that $P(\mathbf{C}(t+1) | \mathbf{C}(t)) = 0$ and that $P(\mathbf{f}^{(s)}(t+1) | \mathbf{g}^{(s)}(t+1)) = 0$ for each component of $\mathbf{f}^{(s)}$. Therefore, we have to consider those specific time steps separately with the corresponding changes.

4.2.1 Q distribution for the linear models

To obtain the distribution $q(\mathbf{f}^{(s)}(t))$, we make use of equations (4.1.3), (4.1.11) and (4.2.11).⁷ After factorising and solving, we get

$$q(f_i^{(s)}(t)) \propto \exp \left[-\frac{1}{2\sigma^2} \left(\begin{aligned} & -2 \langle c^{(s)}(t) \rangle_{q(c^{(s)}(t))} y_i(t) f_i^{(s)}(t) \\ & + 2 \langle c^{(s)}(t) c^{(k)}(t) \rangle_{q(\mathbf{C}(t))} f_i^{(s)}(t) \langle f_i^{(k)}(t) \rangle_{q(f_i^{(k)}(t))} \\ & + \langle c^{(s)}(t) \rangle_{q(c^{(s)}(t))} f_i^{(s)}(t)^2 \end{aligned} \right) \right. \\ \left. -\frac{1}{2\tau^2} \left(\begin{aligned} & f_i^{(s)}(t)^2 - 2f_i^{(s)}(t) \sum_j M_{ij}^{(s)} \langle g_j^{(s)}(t) \rangle_{q(g_j^{(s)}(t))} - 2f_i^{(s)}(t) b_i^{(s)} \\ & -2f_i^{(s)}(t) \sum_j M_{ji}^{(o1)(s)} \langle f_j^{(s)}(t+1) \rangle_{q(\mathbf{f}^{(s)}(t+1))} \\ & + 2f_i^{(s)}(t) \sum_j M_{ji}^{(o1)(s)} b_j^{(s)} \\ & + 2 \langle \mathbf{g}^{(s)\top}(t+1) \rangle_{q(\mathbf{f}^{(s)}(t) \dots q(\mathbf{f}^{(s)}(t-o))} \mathbf{M}^{(s)\top} \mathbf{M}^{(s)} Z_i^{(ON)} f_i^{(s)}(t) \\ & - f_i^{(s)}(t)^2 Z_i^{(ON)\top} \mathbf{M}^{(s)\top} \mathbf{M}^{(s)} Z_i^{(ON)} \end{aligned} \right) \right] \quad (4.2.13)$$

which makes use of the order-one data in \mathbf{M} (4.1.10). Also, an operator $Z_i^{(ON)}$ that creates a vector with ON Boolean values where all but i are zero was included to simplify the last elements of the equation.

Consequently, each factor is parameterised by a *Gaussian* with variance

$$\left\langle (f_i^{(s)}(t))^2 \right\rangle - \langle f_i^{(s)}(t) \rangle^2 = \frac{\sigma^2 \tau^{(s)2}}{\tau^{(s)2} \langle c^{(s)}(t) \rangle_{q(c^{(s)}(t))} + \sigma^2 \left(1 + [\mathbf{M}^{(1)\top} \mathbf{M}^{(1)}]_{ii} \right)} \quad (4.2.14)$$

and mean

⁷ Detailed derivations for this can be found in the Appendix A.3.1.

$$\langle f_i^{(s)}(t) \rangle = \frac{\tau^{(s)2} \langle c^{(s)}(t) \rangle_{q(c^{(s)}(t))} \left(\begin{array}{l} +y_i(t) \\ -\langle c^{(k)}(t) \rangle_{q(c^{(k)}(t))} \langle f_i^{(k)}(t) \rangle_{q(f_i^{(k)}(t))} \end{array} \right) + \sigma^2 \left[\begin{array}{l} \sum_j M_{ij}^{(s)} \langle g_j^{(s)}(t) \rangle_{q(g_j^{(s)}(t))} + b_i^{(s)} \\ + \sum_j M_{ji}^{(o1)(s)} \langle f_j^{(s)}(t+1) \rangle_{q(f_j^{(s)}(t+1))} - \sum_j M_{ji}^{(o1)(s)} b_j^{(s)} \\ - \langle \mathbf{g}^{(s)\text{T}}(t+1) \rangle_{q(\mathbf{r}^{(s)}(t)) \dots q(\mathbf{r}^{(s)}(t-o))} \mathbf{M}^{(s)\text{T}} \mathbf{M}^{(s)} \mathbf{Z}_i^{(ON)} \\ + \mathbf{Z}_i^{(ON)\text{T}} \mathbf{M}^{(s)\text{T}} \mathbf{M}^{(s)} \mathbf{Z}_i^{(ON)} f_i^{(s)}(t) \end{array} \right]}{\tau^{(s)2} \langle c^{(s)}(t) \rangle_{q(c^{(s)}(t))} + \sigma^2 \left(1 + [\mathbf{M}^{(1)\text{T}} \mathbf{M}^{(1)}]_{ii} \right)} \quad (4.2.15)$$

At this point, by using equation (4.2.15), we still cannot separate the component $f_i^{(s)}(t)$ from the observation $y_i(t)$ since it is coupled with the mean of $c^{(k)}(t)$. In the next section we derivate what the mean is, so we use it for all the N components to determine $\mathbf{f}^{(s)}(t)$.

4.2.2 Q distribution for the switches

By using the *mean field theory* (4.2.6), we approximate $P(\mathbf{C}|J)$ by $P(c^{(1)}|J)P(c^{(2)}|J)$ and rewrite $q(\mathbf{C}(t))$ as $q(c^{(1)}(t))q(c^{(2)}(t))$. So (4.2.12) turns to be

$$q(c^{(s)}(t)) \propto \exp \left(\begin{array}{l} \left\langle \log(P(\mathbf{Y}(t) | c^{(s)}(t), \mathbf{f}^{(s)}(t), \mathbf{f}^{(k)}(t))) \right\rangle_{q(\mathbf{r}^{(s)}(t))q(\mathbf{r}^{(k)}(t))} \\ + \left\langle \log(P(c^{(s)}(t) | c^{(s)}(t-1))) \right\rangle_{q(c^{(s)}(t-1))} + \left\langle \log(P(c^{(s)}(t+1) | c^{(s)}(t))) \right\rangle_{q(c^{(s)}(t+1))} \end{array} \right) \quad (4.2.16)$$

where again, by using (s, k) we make reference to the two possible combinations we are using $\{(1,2), (2,1)\}$ for $c(t)$.⁸

If we look at the exponential, the first term was already introduced in (4.2.11) (just averaged with respect to other variables), while we have not said what $P(c^{(s)}(t) | c^{(s)}(t-1))$ and $P(c^{(s)}(t+1) | c^{(s)}(t))$ are. By using (4.1.6), we get the following expressions:

$$P(c^{(s)}(t) | c^{(s)}(t-1)) = \left(\begin{array}{l} (1 - 2c^{(s)}(t) - 2c^{(s)}(t-1) + 4c^{(s)}(t)c^{(s)}(t-1))\gamma \\ -2c^{(s)}(t)c^{(s)}(t-1) + c^{(s)}(t) + c^{(s)}(t-1) \end{array} \right) \quad (4.2.17)$$

and

$$P(c^{(s)}(t+1) | c^{(s)}(t)) = \left(\begin{array}{l} (1 - 2c^{(s)}(t+1) - 2c^{(s)}(t) + 4c^{(s)}(t+1)c^{(s)}(t))\gamma \\ -2c^{(s)}(t+1)c^{(s)}(t) + c^{(s)}(t+1) + c^{(s)}(t) \end{array} \right) \quad (4.2.18)$$

⁸ Detailed derivations for this can be found in the Appendix A.3.2

So if the values for $c^{(s)}$ are equal in consecutive time steps, then we will have as output the logarithm of γ times the probability of the current state t of being on/off (and similarly for the other case, but having $1-\gamma$ instead of γ).

After considering (4.1.3), (4.2.17) and (4.2.18), we obtain that $q(c^{(s)}(t))$ has the following form:

$$q(c^{(s)}(t)) \propto \exp \left[\begin{aligned} & -\frac{1}{2\sigma^2} \left(-2c^{(s)}(t) \mathbf{Y}^T(t) \langle \mathbf{f}^{(s)}(t) \rangle_{q(\mathbf{f}^{(s)}(t))} \right. \\ & \quad + 2c^{(s)}(t) \langle c^{(k)}(t) \rangle_{q(c^{(k)}(t))} \langle \mathbf{f}^{(s)\top}(t) \rangle_{q(\mathbf{f}^{(s)}(t))} \langle \mathbf{f}^{(k)}(t) \rangle_{q(\mathbf{f}^{(k)}(t))} \\ & \quad \left. + c^{(s)}(t) \text{var}(\mathbf{f}^{(s)}(t)) + \langle \mathbf{f}^{(s)\top}(t) \rangle_{q(\mathbf{f}^{(s)}(t))} \langle \mathbf{f}^{(s)}(t) \rangle_{q(\mathbf{f}^{(s)}(t))} \right) \\ & + \left(1 - \langle c^{(s)}(t-1) \rangle_{q(c^{(s)}(t-1))} \right) \log \left((1 - 2c^{(s)}(t))\gamma + c^{(s)}(t) \right) \\ & + \left(\langle c^{(s)}(t-1) \rangle_{q(c^{(s)}(t-1))} \right) \log \left((2c^{(s)}(t) - 1)\gamma - c^{(s)}(t) + 1 \right) \\ & + \left(1 - \langle c^{(s)}(t+1) \rangle_{q(c^{(s)}(t+1))} \right) \log \left((1 - 2c^{(s)}(t))\gamma + c^{(s)}(t) \right) \\ & + \left(\langle c^{(s)}(t+1) \rangle_{q(c^{(s)}(t+1))} \right) \log \left((2c^{(s)}(t) - 1)\gamma - c^{(s)}(t) + 1 \right) \end{aligned} \right] \quad (4.2.19)$$

4.2.3 Equations for Inference

Given a sequence of observation vectors $\mathbf{Y}(t)$, we can do signal separation by using equation (4.2.15) to determine what each component i of $\mathbf{f}^{(s)}(t)$ is ($\forall i \in \{1..N\}$). However, to determine if $c^{(s)}(t)$ is 0 or 1, we need some extra calculations.

To obtain the probability $P(c^{(s)}(t) = l)$, $l \in \{0,1\}$ we will have to normalize (4.2.19), so we get

$$P(c^{(s)}(t) = l) = \frac{q(c^{(s)}(t) = l)}{q(c^{(s)}(t) = 0) + q(c^{(s)}(t) = 1)} \quad (4.2.20)$$

and the equation (4.2.19) splits into the following cases:

$$\begin{aligned}
q(c^{(s)}(t) = 0) &\propto \exp \left\{ \begin{aligned} &\left(1 - \langle c^{(s)}(t-1) \rangle_{q(c^{(s)}(t-1))} \right) \log(\gamma) + q(c^{(s)}(t-1) = 1) \log(1-\gamma) \\ &+ \left(1 - \langle c^{(s)}(t+1) \rangle_{q(c^{(s)}(t+1))} \right) \log(\gamma) + q(c^{(s)}(t+1) = 1) \log(1-\gamma) \end{aligned} \right\} \\
&= \exp \left\{ \left(\langle c^{(s)}(t-1) \rangle_{q(c^{(s)}(t-1))} + \langle c^{(s)}(t+1) \rangle_{q(c^{(s)}(t+1))} \right) \log \left(\frac{1-\gamma}{\gamma} \right) + 2 \log(\gamma) \right\}
\end{aligned} \tag{4.2.21}$$

and

$$\begin{aligned}
q(c^{(s)}(t) = 1) &\propto \exp \left\{ \begin{aligned} &\left(-2\mathbf{Y}^T(t) \langle \mathbf{f}^{(s)}(t) \rangle_{q(\mathbf{f}^{(s)}(t))} \right. \\ &-\frac{1}{2\sigma^2} \left. + 2 \langle c^{(k)}(t) \rangle_{q(c^{(k)}(t))} \langle \mathbf{f}^{(s)T}(t) \rangle_{q(\mathbf{f}^{(s)}(t))} \langle \mathbf{f}^{(k)}(t) \rangle_{q(\mathbf{f}^{(k)}(t))} \right. \\ &\left. + \text{var}(\mathbf{f}^{(s)}(t)) + \langle \mathbf{f}^{(s)T}(t) \rangle_{q(\mathbf{f}^{(s)}(t))} \langle \mathbf{f}^{(s)}(t) \rangle_{q(\mathbf{f}^{(s)}(t))} \right) \\ &\left(\langle c^{(s)}(t-1) \rangle_{q(c^{(s)}(t-1))} + \langle c^{(s)}(t+1) \rangle_{q(c^{(s)}(t+1))} \right) \log \left(\frac{\gamma}{1-\gamma} \right) + 2 \log(1-\gamma) \end{aligned} \right\}
\end{aligned} \tag{4.2.22}$$

where the variance and mean of $\mathbf{f}^{(s)}(t)$ are calculating using (4.2.14) and (4.2.15). The mean of $c^{(s)}(t)$ is given by

$$\langle c^{(s)}(t) \rangle_{q(c^{(s)}(t))} = P(c^{(s)}(t) = 1) \tag{4.2.23}$$

since

$$\langle c^{(s)}(t) \rangle_{q(c^{(s)}(t))} = P(c^{(s)}(t) = 0) \langle c^{(s)}(t) = 0 \rangle + P(c^{(s)}(t) = 1) \langle c^{(s)}(t) = 1 \rangle \tag{4.2.24}$$

and the parameters $\mathbf{b}^{(s)}$, $\tau^{(s)2}$ and $\mathbf{M}^{(s)}$ for $\mathbf{f}^{(s)}(t)$ can be obtained as it was respectively described in (4.1.12), (4.1.13) and (4.1.14).

As we can see, to determine if a model s is active for time step t , we use the equations corresponding to both $q(c^{(s)}(t) = 1)$ and $q(\mathbf{f}^{(s)}(t))$. These equations employ the results of other model k , therefore existing dependence between the equations for inference. As a result, an iterative procedure for solving these equations has to be taken into consideration (in general, this has to be done when using mean field approximations). Stopping criteria such as iterating for a maximum number of loops and/or waiting for the stabilization of the calculations (this is, when the calculations do not change after a specific number of loops) are proposed to limit the number of calculations.

4.2.4 Generalization

Consider the general graphical model shown in Figure 4.4.

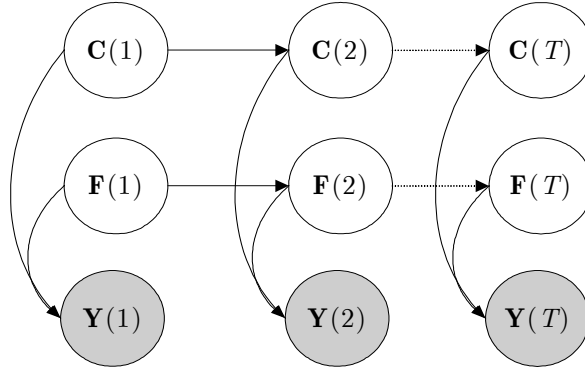


Figure 4.4: Graphical model of S mixed sources with Boolean switches

In general, the observation $\mathbf{Y}(t)$ is modelled by

$$\mathbf{Y}(t) = \mathbf{C}(t)\mathbf{F}(t) + \mathbf{w}(t) \quad (4.2.25)$$

where $\mathbf{C}(t)$ is a vector that contains S switching variables that indicate whether an assigned linear model will be used or not for time step t ; $\mathbf{F}(t)$ is a set of S different linear models ($\mathbf{F}(t) \in \Re^{N \times S}$); $\mathbf{w}(t) \sim \mathcal{N}(0, \sigma^2)$ is an independent *Gaussian* noise source; and t is a specific time step from the range $\{1, 2, \dots, T\}$. Regarding $\mathbf{C}(t)$ and $\mathbf{F}(t)$, we have

$$\mathbf{C}(t) = \begin{bmatrix} c^{(1)}(t) \\ c^{(2)}(t) \\ \dots \\ c^{(S)}(t) \end{bmatrix}, \mathbf{F}(t) = \begin{bmatrix} \mathbf{f}^{(1)\top}(t) & \mathbf{f}^{(2)\top}(t) & \dots & \mathbf{f}^{(S)\top}(t) \end{bmatrix}, \text{ and } \mathbf{f}^{(s)}(t) = \begin{bmatrix} f_1^{(s)}(t) \\ f_2^{(s)}(t) \\ \dots \\ f_N^{(s)}(t) \end{bmatrix} \quad (4.2.26)$$

By using results from 4.2.3, generalizing the equations becomes a straightforward procedure where summations for all k models have to be included in the formulas of s that include k . Most of the equations remain the same as in 4.2.3, and the only ones that change the q function of $c^{(s)}(t)$ conditioned on being active

$$q(c^{(s)}(t) = 1) \propto \exp \left[\begin{aligned} & -\frac{1}{2\sigma^2} \left(-2\mathbf{Y}^T(t) \langle \mathbf{f}^{(s)}(t) \rangle_{q(\mathbf{f}^{(s)}(t))} \right. \\ & \left. + \sum_{k \neq s} 2 \langle c^{(k)}(t) \rangle_{q(c^{(k)}(t))} \langle \mathbf{f}^{(s)T}(t) \rangle_{q(\mathbf{f}^{(1)}(t))} \langle \mathbf{f}^{(k)}(t) \rangle_{q(\mathbf{f}^{(k)}(t))} \right. \\ & \left. + \text{var}(\mathbf{f}^{(s)}(t)) + \langle \mathbf{f}^{(s)T}(t) \rangle_{q(\mathbf{f}^{(1)}(t))} \langle \mathbf{f}^{(s)}(t) \rangle_{q(\mathbf{f}^{(1)}(t))} \right) \\ & + \left(1 - \langle c^{(s)}(t-1) \rangle_{q(c^{(s)}(t-1))} \right) \log(1-\gamma) + \left(\langle c^{(s)}(t-1) \rangle_{q(c^{(s)}(t-1))} \right) \log(\gamma) \\ & + \left(1 - \langle c^{(s)}(t+1) \rangle_{q(c^{(s)}(t+1))} \right) \log(1-\gamma) + \left(\langle c^{(s)}(t+1) \rangle_{q(c^{(s)}(t+1))} \right) \log(\gamma) \end{aligned} \right] \quad (4.2.27)$$

and the factorised mean of $\langle \mathbf{f}^{(s)}(t) \rangle$

$$\langle f_i^{(s)}(t) \rangle_{q(f^{(s)}(t))} = \frac{\tau^{(s)2} \langle c^{(s)}(t) \rangle_{q(c^{(s)}(t))} \left(\begin{aligned} & +y_i(t) \\ & - \sum_{k \neq s} \langle c^{(k)}(t) \rangle_{q(c^{(k)}(t))} \langle f_i^{(k)}(t) \rangle_{q(f^{(k)}(t))} \end{aligned} \right) + \sigma^2 \left(\begin{aligned} & \sum_j M_{ij}^{(s)} \langle g_j^{(s)}(t) \rangle_{q(g^{(s)}(t))} + b_i^{(s)} \\ & + \sum_j M_{js}^{(s)} \langle f_j^{(s)}(t+1) \rangle_{q(f^{(s)}(t+1))} - \sum_j M_{js}^{(s)} b_j^{(s)} \\ & - \langle \mathbf{g}^{(s)T}(t+1) \rangle_{q(\mathbf{f}^{(s)}(t) \dots q(\mathbf{f}^{(s)}(t-o))} \mathbf{M}^{(s)T} \mathbf{M}^{(s)} \mathbf{Z}_i^{(ON)} \\ & + \mathbf{Z}_i^{(ON)T} \mathbf{M}^{(s)T} \mathbf{M}^{(s)} \mathbf{Z}_i^{(ON)} f_i^{(s)}(t) \end{aligned} \right)}{\tau^{(s)2} \langle c^{(s)}(t) \rangle_{q(c^{(s)}(t))} + \sigma^2 (1 + [\mathbf{M}^{(1)T} \mathbf{M}^{(1)}]_{ii})} \quad (4.2.28)$$

As we already stated, all these equations are related to smoothing and, even though they are written in a general form, they were derived for an order 1 *Markovian* regression. Generalization is a straightforward procedure (basically summations in the terms that look into the future have to be included), but the formulas turn messier. Filtering equations can be derived from the smoothing equations (just by dropping all the terms that look into the future), and generalization to an order O *Markovian* regression can be expressed in a cleaner way. The equations for filtering a model such as the one shown in Figure 4.5 are the same as the ones derived throughout this section, but with changes in: the q functions of $c^{(s)}(t)$

$$q(c^{(s)}(t) = 0) \propto \exp \left\{ \left(1 - \langle c^{(s)}(t-1) \rangle_{q(c^{(s)}(t-1))} \right) \log(\gamma) + q(c^{(s)}(t-1) = 1) \log(1-\gamma) \right\} \quad (4.2.29)$$

$$q(c^{(s)}(t) = 1) \propto \exp \left\{ \begin{aligned} & -2\mathbf{Y}^T(t) \langle \mathbf{f}^{(s)}(t) \rangle_{q(\mathbf{f}^{(s)}(t))} \\ & - \frac{1}{2\sigma^2} \left[\sum_{k \neq s} 2 \langle c^{(k)}(t) \rangle_{q(c^{(k)}(t))} \langle \mathbf{f}^{(s)T}(t) \rangle_{q(\mathbf{f}^{(1)}(t))} \langle \mathbf{f}^{(k)}(t) \rangle_{q(\mathbf{f}^{(k)}(t))} \right. \\ & \quad \left. + \text{var}(\mathbf{f}^{(s)}(t)) + \langle \mathbf{f}^{(s)T}(t) \rangle_{q(\mathbf{f}^{(1)}(t))} \langle \mathbf{f}^{(s)}(t) \rangle_{q(\mathbf{f}^{(1)}(t))} \right] \\ & + \left[1 - \langle c^{(s)}(t-1) \rangle_{q(c^{(s)}(t-1))} \right] \log(1-\gamma) + \left[\langle c^{(s)}(t-1) \rangle_{q(c^{(s)}(t-1))} \right] \log(\gamma) \end{aligned} \right\} \quad (4.2.30)$$

the factorised variance of $\langle \mathbf{f}^{(s)}(t) \rangle$

$$\langle (f_i^{(s)}(t))^2 \rangle - \langle f_i^{(s)}(t) \rangle^2 = \frac{\sigma^2 \tau^{(s)2}}{\tau^{(s)2} \langle c^{(s)}(t) \rangle_{q(c^{(s)}(t))} + \sigma^2} \quad (4.2.31)$$

and the factorised mean of $\langle \mathbf{f}^{(s)}(t) \rangle$

$$\langle f_i^{(s)}(t) \rangle_{q(f_i^{(s)}(t))} = \frac{\tau^{(s)2} \langle c^{(s)}(t) \rangle_{q(c^{(s)}(t))} \left(+y_i(t) - \sum_{k \neq s} \langle c^{(k)}(t) \rangle_{q(c^{(k)}(t))} \langle f_i^{(k)}(t) \rangle_{q(f_i^{(k)}(t))} \right) + \sigma^2 \left(\sum_j M_{ij}^{(s)} \langle g_j^{(s)}(t) \rangle_{q(g_j^{(s)}(t))} + b_i^{(s)} \right)}{\tau^{(s)2} \langle c^{(s)}(t) \rangle_{q(c^{(s)}(t))} + \sigma^2}. \quad (4.2.32)$$

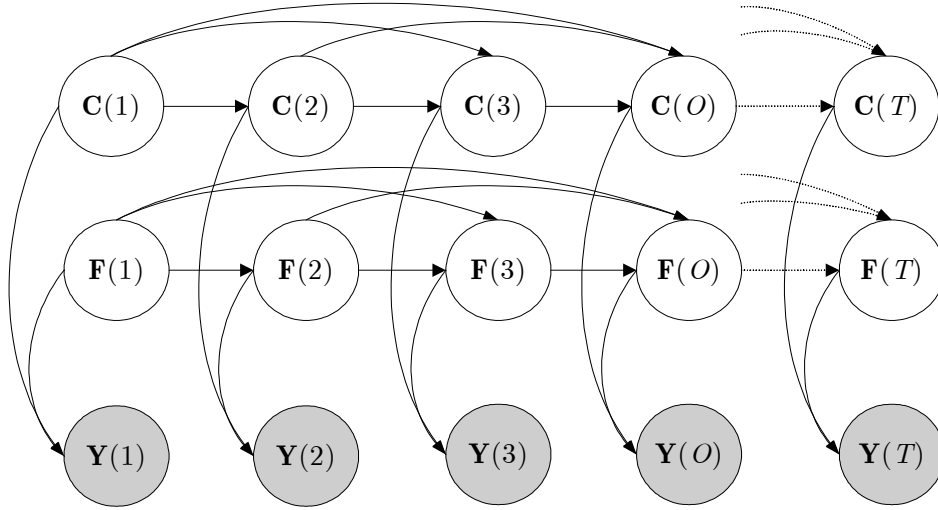


Figure 4.5: Graphical model for an order O Markovian autoregression

Chapter 5

Implementing and testing the model

In this chapter, we present tests related to the *AR* models and the *Switching Linear Dynamical System* that was derived. Some parameters are learned directly from the data (\mathbf{M} , \mathbf{b} and τ^2), while others had to be found by maximizing the likelihood of our training data (σ^2 and γ). First we introduce the characteristics of our implementation procedure, then we present how is that our *AR* models behave, and finally we conclude with different tests concerning *Polyphonic Correlated Source Separation*.

5.1 Implementation and Data acquisition

Matlab 6.1 was used for the implementation of our model. It is a widely use tool in areas related to engineering, statistics and machine learning that allows fast prototyping and an efficient use of vectors/matrices (so was convenient for the present work).

Instrument notes were obtained from midi instruments (piano, guitar and bass) and from a real instrument (a harmonica, which was selected for convenience). Several notes were recorded from the piano for doing general tests (A4, Bb4, G4, C5, D5, Eb5, F5, G5, D6, Eb6 and F6), while just a few from the other instruments to see if we could generalize our results (C5, G5 from the guitar, C3, G3 from the bass, and B4 and E4 from the harmonica).¹ Every sound was recorded as monophonic using a sampling rate of 8Khz and

¹ The midi sounds were quarter notes.

the same volume (or similar) and then saved in wav format (which is handled by Matlab).¹

General modules were built for allowing different tests to be done. Simple midi-like structures were implemented (basically specifying a note to be played and, in seconds, the onset and offset points) and an automated procedure for converting midi outputs to our midi-like structures was created (so we could test the model with more realistic musical cases). However, since the recording/cleaning of each instrument and its notes was not automated, the midi contents had to refer only to those sounds that were previously recorded.

5.2 Testing the *AR* models

Learning an *AR* model basically reduces to learning the coefficients of the autoregressive matrix \mathbf{M} (after the mean and the variance are obtained, 4.1.2). We can use data from two sources to obtain this matrix: a wav sound (which is represented by a vector of samples of size $1 \times T$ and that we will make reference as "raw data"), or its *Fourier* coefficients (which are $N \times T$ dimensional vectors that result from the *Short Time Fourier Transform* (2.1.1.2), and that we will refer as "processed data").² Consequently, we have one free parameter for training the models when using the raw data (the *AR* order), and three free parameters when using processed data (the *AR* order, the size of the *STFT* window, and use either the real part of the *Fourier* coefficients or both reals and imaginaries as learning data).³

¹ Initial sampling considered 44.1 KHz for high quality. However, the amount of data related to samples with that quality was massive (a recording of 0.4 seconds translated into a vector with thousands of components), and processing it derived a very expensive use of computational resources. Our choice was to drop down the sampling of the sounds to 8Kz. Although this affected the quality of the of the tests since at least there is a trade-off with the *Fourier* analysis (2.1.1.2), we thought of it as convenient for having a faster performance.

² Obviously, T stands for a final number of time steps and not for a measure of time. It is clear that the length of a wav sound is different from the one of its *STFT* (since the latter is windowed).

³ If we drop the imaginaries, we still can reconstruct signals that sound very similar to the original ones. However, since the use of imaginaries led to better results in

In case of using raw data, no further processing has to be done for learning \mathbf{M} than using the wav files (we train with amplitudes per sample, we get the same sort of data). However, if we decide to use the *STFT*, we still have the option of whether we want to use only its real output or both the real and imaginary parts. For doing this we need to drop the reciprocal dimension (2.1.1.1) (this is required for not having linear combinations of the data, as we will explain in the next paragraph), separate the components into reals/imaginaries, and then arrange them as the N elements of the vector \mathbf{f} (where areas corresponding to reals and imaginaries have to be delimited for a later reconstruction). After using the *AR* model for generating a new sequence of vectors, we reconstruct a signal by mapping the elements of these vectors into their corresponding complex values (mixing reals and imaginaries). Finally, the conjugate that was dropped (concerning the reciprocal dimension) needs to be built again and the inverse of the *Fourier Transform* needs to be applied.

To define each \mathbf{M} , we make reference to the optimal derived equation (4.1.14), which is rewritten below.

$$\mathbf{M} = \left(\sum_{t=2}^T (\mathbf{f}(t) - \tilde{\mathbf{f}})(\mathbf{g}(t) - \tilde{\mathbf{g}})^T \right) \left(\sum_{t=2}^T (\mathbf{g}(t) - \tilde{\mathbf{g}})(\mathbf{g}(t) - \tilde{\mathbf{g}})^T \right)^{-1}$$

As we can observe from this equation, we need the matrix that results from multiplying the autoregressive vectors to be invertible (right part). This is related to using data that is not a linear combination of each other per training vector, and consequently has to do with controlling the amount of data to be considered for each time step by modifying the size of each window (making it smaller will be necessary if we are including linear dependencies). As well, this relates to employing an autoregressive order which filters non-monotonous information (if our signal is completely

our tests and were necessary to reconstruct the signal with its full characteristics, they were maintained.

periodic, adding information of a higher order would be redundant, resulting into the inclusion of linear dependencies too).⁴

In addition, we also have a problem related the window-frequency trade-off. If we were to use big windows for the *STFT*, we have to remember that the resolution increases as the window gets bigger (more harmonics are calculated), and the use of different frequencies turns more selective (some *Fourier* coefficients increase while others decrease). As a result, some harmonics can completely unused for all time steps (which translates into zero rows), and then we will have non-invertible matrix after the autoregressive vectors multiply.

To deal with the above situation, data was split into time divisions smaller than the ones that in general are involved in one period for all the sounds (Figure 2.9) and the regressive order was chosen with an inverse relation regarding the degree of "periodicity". In consequence, this had an impact on the possible values of the parameters concerning the size of the *STFT* window (for processed data), and the order of the autoregression (for raw and processed data). For our sounds, the maximum value for time splitting that was found was around 4 ms (for both the raw and the processed data) allowing the use of *AR* models of order 1. If we apply a smaller windowing, then we can use higher autoregressive orders, but we have to bear in mind the window-frequency trade-off. With windows smaller than 1.5 ms, reconstruction was not good.

Using a window division of 2.5 ms (5.3.1), *STFT* data with both reals and imaginaries, and a maximum possible use of an *AR* order of 3 *AR* model (taking to the limit the use of information of one period to 1.5 periods) we

⁴ Clearly, we will have different situations depending on how "periodic" is each sound, and if it has minor changes through time (being almost periodic), less information is required for calculating the **M** matrix.

had the following reconstructions when feeding the model with the original data for the initial 1, 2 or 3 time steps (with the addition of noise).⁵

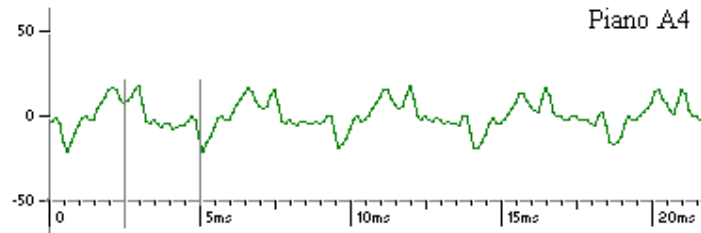


Figure 5.1: Windowing a signal

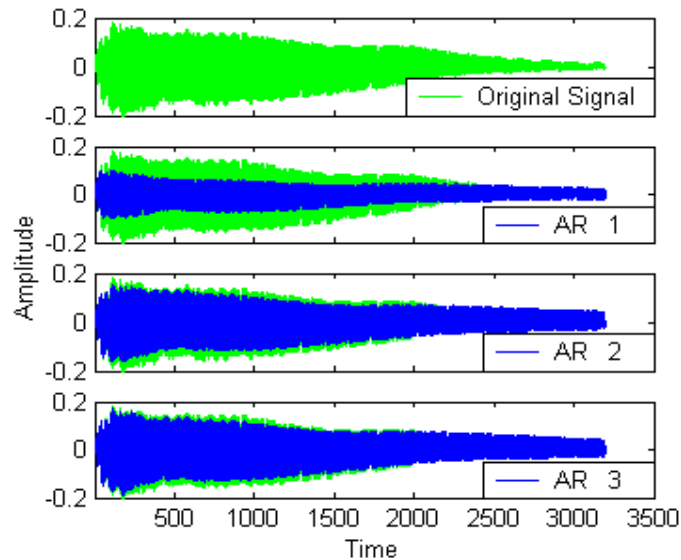


Figure 5.2: Autoregressive sound of a piano playing Eb6

Without surprise, the use of a higher autoregressive order resulted into a better reconstruction of the sound signals. However, not all signals were reproduced as faithfully as this one (which fits almost perfectly the initial part of the sound and then maps into its general form). In other cases, such as in our worst case (Figure 5.3), the signals had an initial "bursting" part which was more complex and their envelope was not described by a simple linear/exponential decay as in our previous illustration (so the reconstruction

⁵ Other initialisation data that was used was random data and the mean of the sounds, however we were not able to produce good reconstructions with them. The mean has less power than the initial part of the sound (so similar sounds were built but with less volume), and random initialisation derived into noise.

was not as good). Still, they captured the essential characteristics of the acoustic signal.

In Figure 5.3 we compare the results of a *Fourier*-based model and a raw-based model in similar circumstances. Training considered 3 *STFT* windows containing 20 samples each (therefore 60 samples in total) and 60 samples for the raw data (using both reals and imaginaries). The initialisation conditions were also similar, being the first 3 *Fourier* components for the first model, and the 60 ones for the normal one. The *Mean Square Error (MSE)* was inferior when using *Fourier*.

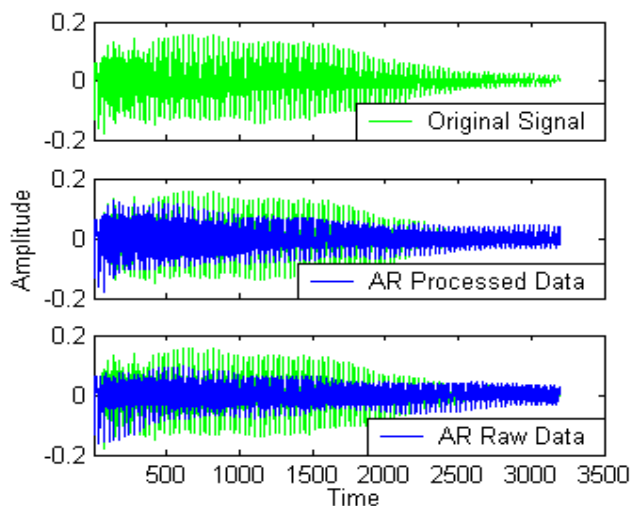


Figure 5.3: Autoregressive sounds of a piano playing Eb4, using raw and processed data

5.3 Testing the inference models

A general module was built for smoothing/filtering, where the initial conditions are random. It uses the inference equations related to $q(c^{(s)}(t))$ and $q(\mathbf{f}^{(s)}(t))$ to determine their real states (4.2.3), and cycles until a maximum number of loops is reached or when the mean of every linear model stabilizes for a specific number of time steps (they stabilize only if the switches stabilize).

Initial tests involved either raw data or processed data (using the *STFT*), and the latter had better performance in both classification and separation

(so they were chosen for the final tests).⁶ In addition, the tests that are presented are based on the filtering equations, since the smoothing equations had worse performance.⁷

5.3.1 Selection of Parameters

The parameters were chosen to be those that maximized the likelihood of our data (this is, those that minimized the number of errors). The form of the tests consisted in using a wav file for building each model. Each wav was windowed, vectorized (splitting the real and the and the imaginary components), processed with the *STFT*, and the parameters of the *AR* model were learned from that data. Then, some wavs were selected to construct the observation (knowing where each sound was played for further evaluations), and this observation was windowed and processed in the same way as the data that was used for the *AR* models. Inference of the contents of each signal and the values of the switches was performed using the *AR* models and the processed observation. Consequently, the separated components were reconstructed using the inverse procedure for creating the observations. Finally, the error measure was the numbers of times where the activations of the switches of each model were different from the original ones (that were previously recorded when building the observation), averaged for all those time steps, then averaged for all the models that were tested in the observation.

⁶ There are other reasons to choose *Fourier* data instead of raw data. The first one is processing speed. Our algorithm heavily uses matrices and vectors, but also has several cycles (for the inference about each time step and for convergence towards a result), where Matlab is not good. When using the *STFT* our number of time steps reduces to $T/(\text{size of the window})$ and, even though we process vectors, this turns to be much faster than when processing T elements from the raw data, and saves time when using several loops (in practice, we had a faster convergence also). In addition, *AR* models that learned from processed data resulted to be more accurate than the ones that learned from raw data (considering the same initial conditions, 5.2).

⁷ Essentially, the smoothing equations are very sensitive to the use of parameters that were not learned (σ^2 and γ) and, even though achieved some separation, this was not as good as the one we had with the filter.

Clearly, several parameters had to be calculated for this procedure. The size of the window was constrained by the calculation of the \mathbf{M} matrix, where *ipso facto* restrictions regarding the periodicity of the sounds limited the window to be smaller than 4 ms (5.2). Finding a good value for it concerned testing different window sizes and calculating the performance of the filtering algorithm. Bigger windows had a good reconstruction but bad classification results. Windows smaller than 1.5 ms had bad reconstruction (because of the window-frequency trade-off), and bad classification results. Both issues were maximized when a window of approximately 2.5 ms was used.⁸ This makes also sense with our initial assumptions regarding the need of smaller windows (3.2).⁹

Concerning the selection of σ^2 and (σ^2 and γ), our initial tests gave considerable good results when σ^2 was small (in the range 0.1 to 0.001) and γ was high (around 0.9). This value of σ^2 was found to be dependent on the order of the *AR* model, being smaller when a higher order was used. By fixing γ to 0.9, several values of σ^2 were calculated, where some optimal values were found to be around 0.01, 0.003, and 0.001 for *AR* models of order 1, 2, and 3, respectively (Figure 5.4). This error measure is related to the test procedure realized in 5.3.2 for all the 11 piano sounds, averaging the number of errors (the test consists on creating an observation of two sounds, see how they confuse with each other, and doing this for all the possible combinations of sounds).

Since comparing all the sounds takes took considerable time (and more when we do it for *AR* models with different orders and testing several σ^2 's), the

⁸ By using a sampling rate of 8000 samples/second and windows with 2.5 ms, we have an amount of 20 samples per window. Our base frequency is then 400Hz, therefore relating our *Fourier* coefficients with frequencies 400, 800, 1200, and so on until 8000 Hz.

⁹ It is interesting to comment that tests for understanding the data and see if we could classify vectors without considering temporal information were carried out at the beginning of this project. In them, a window of 2.5 ms maximized the results of classifiers that resulted from using *Logistic Regression*, *K-Nearest Neighbours*, and fitting a *Gaussian/Mixture of Gaussians* to each sound.

maximum number of loops was set to 10 for every AR order, and complete stabilization was used for 3 time steps.¹⁰ In general, higher order models require more loops than 10 to have better results and then they stabilize and do better than models with a lower order (but they take longer to process since use more data). However, experience with the models told us that with 10 loops we were going to have a reasonable idea of the behaviour each model was going to reach.

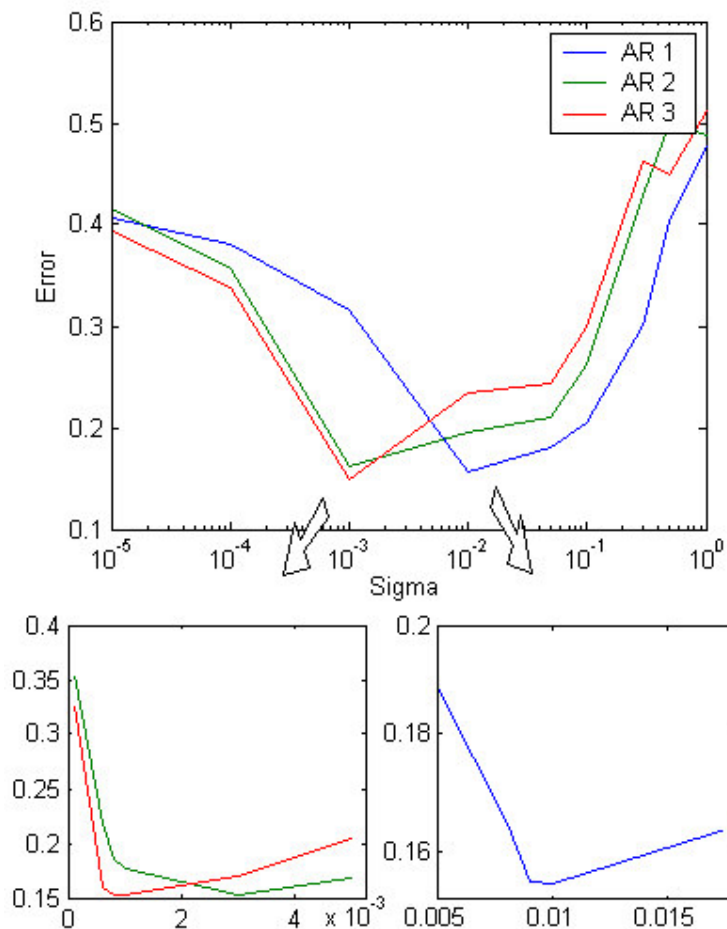


Figure 5.4: Errors for different sigmas for AR models of order 1, 2, and 3 (when γ is fixed to 0.9)

By fixing the values of σ^2 to the optimal values that were found (0.01, 0.003 and 0.001) for each AR order, we did a similar procedure for obtaining γ .

¹⁰ For complete stabilization for 3 time steps we mean that the difference between the means computed for a previous time step and the actual step was zero, and happened for 3 consecutive times.

When γ was near one, different results were commonly obtained (since it influences too much the probability of changing/maintaining its state and our starting means for the contents of each sound and of the switches are initialised at random, as we described at the beginning of this chapter). Optimal values can be considered to be from 0.6 to 0.8, and the *AR* model of order 2 had the minimum classification error.

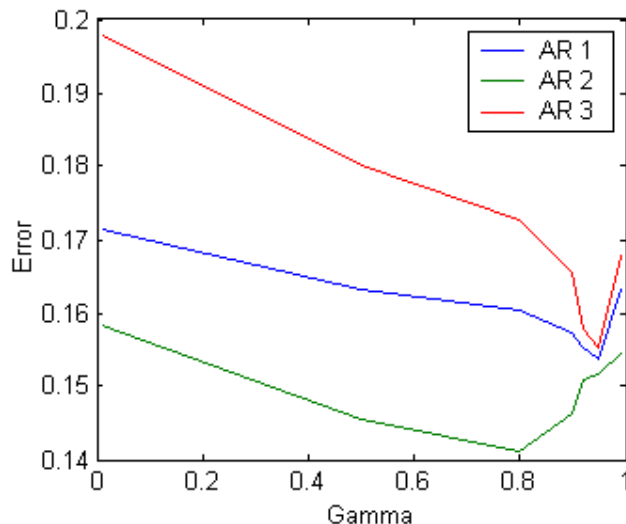


Figure 5.5: Errors for different gammas for *AR* models of order 1, 2, and 3 (when sigmas are fixed to 0.01, 0.003 and 0.001 respectively)

5.3.2 Tests with two models

The idea of comparing two sources is to see which models get more confused with each other, which are the ones that confuse the most and which confuse least. Several initial tests between two models were carried out (such as identifying one sound, playing one and then playing the other, and having overlapping intervals to test a chords), and in the simplest ones, we had remarkably good results. At the end, we selected a more difficult test to train our models, which comprises all the previously mentioned situations.

The test that was built consists on playing one source, stopping it and then the other (which is also cut). Afterwards, a silence is added, then both sources play as a chord, one makes a small silence to continue later, and finally both end at different times. This is illustrated with Figure 5.6 (left).

The lines make reference to the real activations and the crosses to the value that was found for each switch (left), and are dephased up/down for an easier observation. The sources are extracted from the observation (right), and then extracted sources are added to compare what was extracted and the original observation (labelled as “reconstructed observation”).

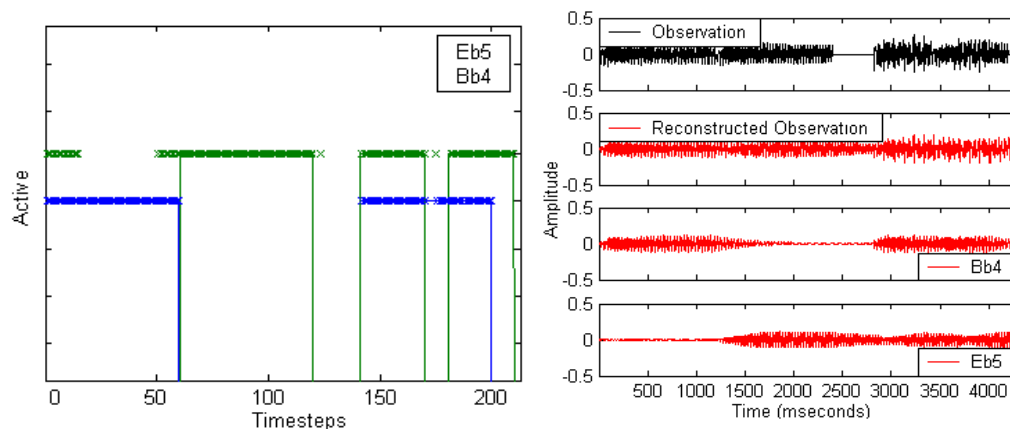


Figure 5.6: Results of a complex test for separating a mixture of two sources

There are some interesting things to note from this example. First of all, we can see that the switches can be found to be on while there is no considerable activation in the contents of the sources (see Eb5 at the figure from left, which is active at the beginning, but from the figure of the right there is no considerable activation).¹¹ For each switch to be on, we decided that it had to be above the threshold of 0.5. We would expect that these switches were in some undeterminable value as 0.6-0.7, but no, the results are commonly polarised to the Boolean values (in fact, the activation of these cases was around 0.998). This happens normally if there are just some harmonics that are similar, where sounds start “picking” up harmonics that are shared. Besides, we can find that in first silence we have the opposite situation. There are no switches active for Eb5 (and there are no contents in the original signal), and the mean values are active. This also happens when we stop playing Eb4 at the beginning. In general, this is one of the effects of

¹¹ This is not a very problematic issue for calculating the optimal value of our parameters, since the areas of the minimum error related to the switches are correlated with the quality of the extraction.

considering temporal dependencies. A model is learned to have a decay rate, and, when it is suddenly “cut out”, the *AR* completes the observation.

After comparing all possible combinations of the models, we came out with some predictable results regarding our optimality measure, the *Mean Square Error*. Those sounds that confuse most with each other are those that share the same note but a different octave, since the harmonics of a higher note are self contained in the lower octave (2.1.2) (Figure 5.7 and Figure 5.8). In general, there is confusion with sounds that share several harmonics (Figure 5.9).

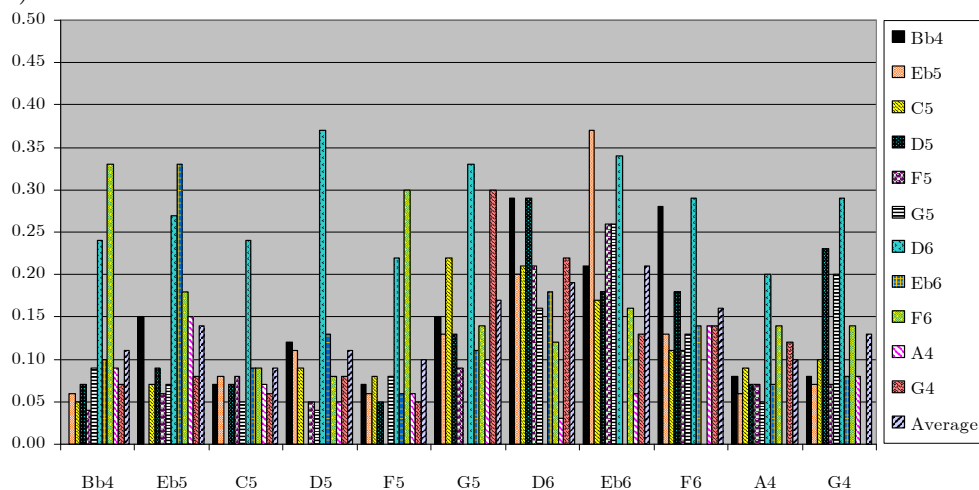


Figure 5.7: Confusion of one model with the rest of the models (error percentage)

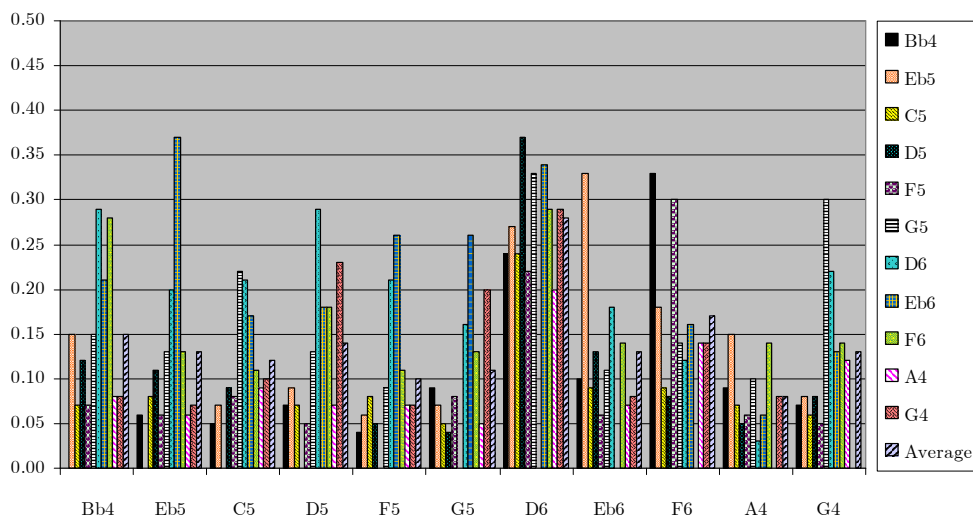


Figure 5.8: Confusion of the rest of the models with a specific model (error percentage)

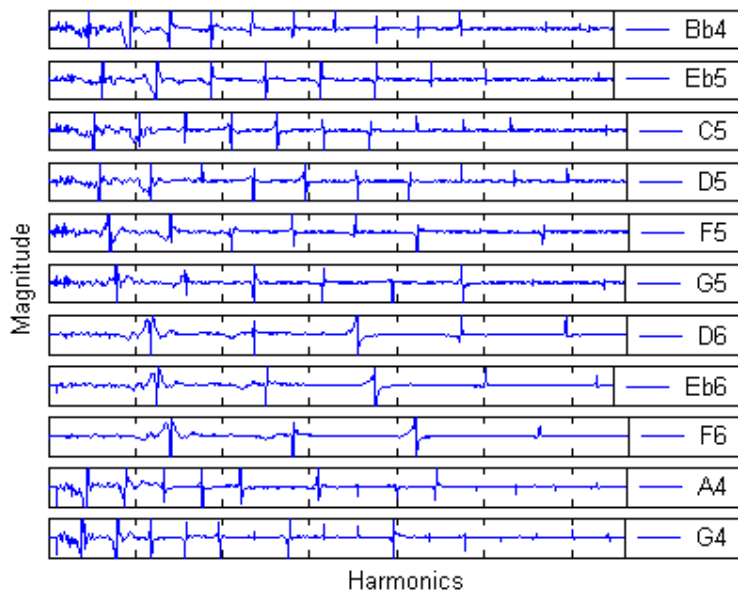


Figure 5.9: Comparison of the harmonics of the sources

5.3.3 Tests with different instruments

In general, the basic differences between notes of different instruments rely on the beginning of the note (where the physical characteristics of the instrument sometimes are more present, 2.1.2). By using the same parameters as in 5.3.2, our model had problems recognizing some notes of different instruments (Figure 5.10), which was expected since notes of different instruments share several harmonics. However, the interesting thing to note here is the dependence of the results related to σ^2 and γ . By changing σ^2 from 0.003 to 0.001, the confusion between the notes C5 played by the piano and the guitar disappeared, and now confusion turned towards the bass. This implies that our results are highly dependent on free parameters and have to be learned from the data.

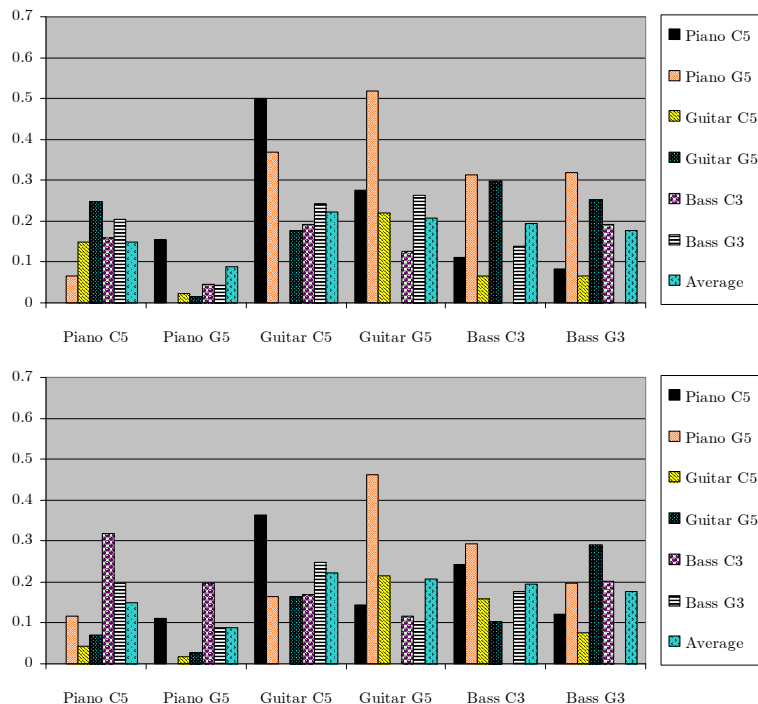


Figure 5.10: Confusion of each model with the rest of the models using different instruments (sigma equals 0.003 in the upper figure, and it is 0.001 in the lower one)

5.3.4 More complex tests

Different tests were carried out, such as distinguishing a melody with few instruments (six) with no overlapping and with no common notes with different octaves. In that case, the filter did quite well. However, as the number of notes were increased and there was overlapping, the performance dropped. More cycles were required to undo the confusion of the models, and the “clarity” of the sources vanished as other sources started “sharing” some of the harmonics that were in the signal and that did not belong to them. Nevertheless, results are not that bad. In Figure 5.11 we test what we wanted to achieve in the introduction. We can distinguish the tones very clearly, but the crisp characteristics of the piano are lost.

For this test, since there are many silences in it and commonly we get good classification results where silences are, we chose to select only those areas that were active and filtered them individually (that is why we can see

regions where the separation looks somehow “chopped”).¹² In addition, there is a zoom factor of 2x in the separated sources for better visibility.

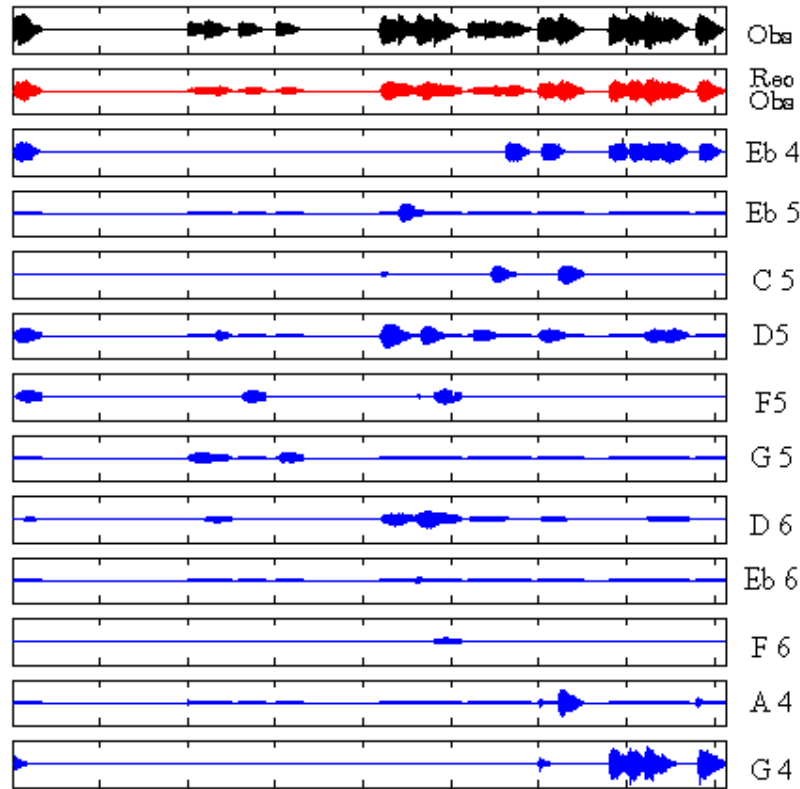


Figure 5.11: Complex reconstruction

Our error results were of 17% considering the average of all the activations of the switches (only where there was not any silence). Maybe this is not the best way to evaluate the models, and if we were to detect some threshold in the amplitude of the extracted signals, we would have better results.

5.3.5 Generalization tests

Creation of more complex tests was not possible since lack of time. However, we ran some tests that allow us to think that our filter could be generalized to very simple real-life situations. The first experiment is related to sampling

¹² We raised the number of loops to 30 for this experiment, and we had to wait more than 5 hours to get the results (the observation lasted for 10 seconds, and translated into a vector with more than 80,000 elements). If we had not “chopped” the silences, we would have waited for about 12 hours (as it happened with other previous experiments).

notes from a real harmonica (Figure 5.12), where separation of a simple situation was possible. It is interesting to remark that we were only able to use an *AR* model of order 1 for this case, since the signals are almost periodic (5.2).

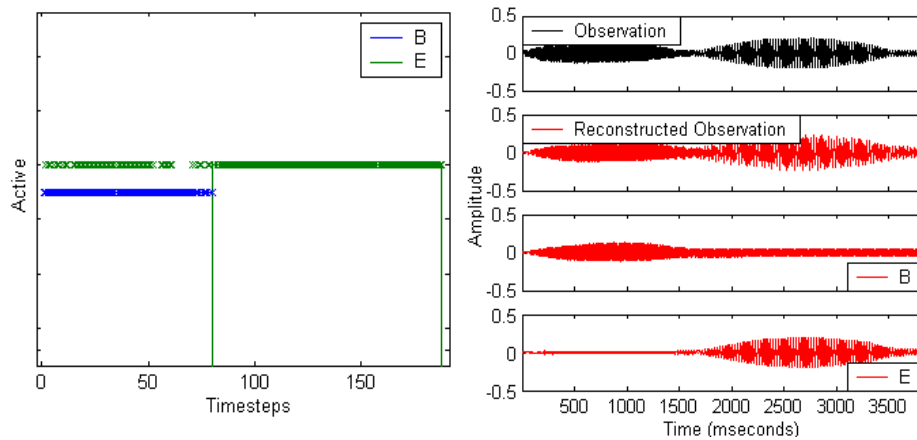


Figure 5.12: Tests with samples from a real instrument

The last experiment we present has to do with using our 11 quarter notes from the piano and testing the models previously built models with a different version of F5 (instead of a quarter the wav contained two linked wholes). The results are shown in Figure 5.13. We can see again the same problem with the switches, which there are many active (basically F5 and its higher octave, F6). However, if we look at the separation we can see a different situation.

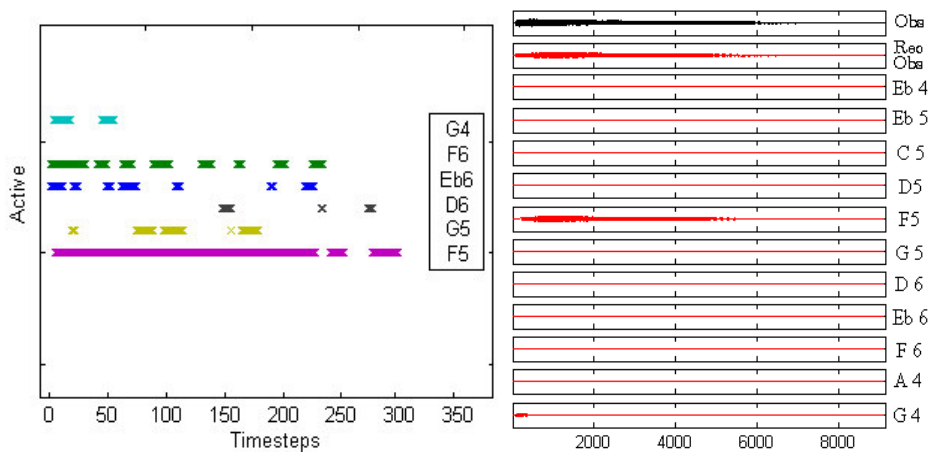


Figure 5.13: Generalization to a note different from the models

Chapter 6

Results, Future Directions and Conclusions

6.1 Results

Derivation and implementation of the model presented in the previous chapters followed an incremental procedure for allowing the addition of dependencies, therefore allowing the model to be more realistic and more complex. The initial model considered only two sources, autoregression of order one, no temporal dependencies on the switches, and filtering for inference. As progress was reached concerning the derivation and the implementation results, the model evolved by including temporal dependencies of order one in the switches, was generalized to filter S sources, dependencies of order O in the AR models were added, and smoothing was also included.

6.1.1 Use of temporal Information

Temporal information mixed with spectral analysis can be used for source separation. In our experiments, models with a higher Markovian order commonly had better performance than those with a lower order (when they did not over fit the data). However, convergence is slower for models with a higher autoregressive order and takes more processing time (since it is required to include more autoregressive information for each calculation).

The possibility to create models with a higher AR order depends on the periodicity of the sounds. For complex semi-periodic sounds, we can make use of more temporal information, while in periodic sounds we just cannot (such is the case with the harmonica, where it can be modelled only by an

AR model of order 1). Reconstruction with an *AR* model will be good if we use most of the data that the model can learn (for both complex and simple cases), and differencing an original sound from its reconstruction one can be difficult for the human ear in some situations.

6.2 Future Directions

The future directions basically have to do with focusing on short-term and long term situations: solving problems and start loosening some assumptions that were considered in the model.

6.2.1 Solve Problems

Several issues have to be considered in the short term before addressing other concerns. They are:

- Need to learn the free parameters. As already discussed, one of the main problems of the model is that its performance, related to both the correct classification of the switches and the quality of the separation, depends heavily on the values of σ^2 and γ (so they have to be learned from the data). The difficulty we will face is that these values are not trivial to learn. Maybe the simplest to deal with is γ , where the real distribution of $P(\mathbf{C})$ has to be learned. Some ideas were already developed in 4.1.1, however we have to be careful about the non-stationary cases (which could occur). For σ^2 this can be more complicated. In theory, it should be zero, since our observation is exactly the addition of the sources, with no noise. However, if we were to learn the rest of the free parameters, σ^2 would be easier to find searching its corresponding dimension.
- Smoothing algorithm. As with the filter, our smoothing algorithm is very dependent on the values assigned for σ^2 and γ . Even though it achieves some separation when the noise is small, it seems that it cannot overcome the random initialisation. We would expect that if we were to create a real distribution for $P(\mathbf{C})$, we would get better

results. By now, a constant γ hardly helps in smoothing non-stationary sequences such as 010 or 101, which do appear in our random initialisation.

- Convergence to a solution. Initialisation is also an important factor that we have to face in a better way. The model sometimes depends on the initial random values for convergence towards the real solution, and cases where it stays in a local minimum, therefore giving different results, are not exceptional (specially when considering several sources). Other interesting approaches to be thought about are realizing updates in parallel instead of using serial updates, or to update at random. Maybe we could find different types of behaviour.
- Evaluation of the separation. As we discussed in Chapter 5, the evaluation of quality of the separation is something that has to be reconsidered. Even though it can be correlated with the value of the switches, probably that is not the most effective way for comparing results from source separation. A better way for training and evaluating the model would be to compare each separated signal with the corresponding constitutive sources in the observation, and find the parameters that minimize the distance between the original signal and the reconstruction.
- Differentiate between notes that share harmonics. As we saw in (5.3.2), there is some conflict regarding the identification of notes that have a simple ratio between their fundamental frequencies. To be able to separate them, some researchers such as (Plumbley, Abdallah et al. in press) propose to use the energy that is expected in each harmonic for a particular instrument as extra information. Therefore, the inclusion of a new conditional term such as $P(\mathbf{f}(t)|\mathbf{E}(t))$ could be a way for approaching this problem.

6.2.2 Start loosening some assumptions

In *PCSS*, there are assumptions that were considered as primordial for simplifying our initial problem. Nevertheless, the stronger the assumptions, the narrower the applicability. Assumptions that have to be reconsidered are:

- Knowledge of the sources. We make the strong assumption that, in order to separate the information from a particular signal, we require knowing it (so we fit an *AR* model). This means that if we do not have a model of something that is present in the observation, we will not extract that information. Even though this can be useful to concentrate on the extraction of just some specific elements, and in music extraction this may be desired (to focus basically on one instrument, for example), it may not be the case in other applications. Another approach should concentrate on creating different models as different characteristics are extracted from data. This is commonly simplified by focusing merely on the frequency content (as *BSS* approaches do), and then maximizing some contrast function to build the probability distributions. However, it would be desirable to include temporal information also (since we have shown that it can be also important for the separation process).
- Convolutional mixing. We assumed that the sources contain information only of themselves and that they are correlated (derived from a common electrical source), allowing us to do instant mixing (Howard and Angus 1996). However, in real-world mixing environments, the sources could be uncorrelated and affected with considerable delay due to reflections (leading to effects such as echoes and reverberation), transforming the mixture into a convolutional one. These cases, since clearly complicate the assumptions, commonly are not considered in *BSS* approaches until the instant mixing works as desired. For *PCSS*, we did the same. Obviously, the incursion to convolutional mixtures

should be of interest, however not until issues such as the short-term future directions have been addressed.

- Use of more complex autoregressive models. There are cases when we get very good reconstructions, but there are cases when we do not (5.2). In general, models with complex aperiodicities are difficult to be modelled by the n -variate AR model that was derived, and a more powerful solution is required. The next step may be using a Temporal AR model for this issue.
- Use other methods to extract characteristics of the signals. The *Fourier transform* clearly is not the only spectral transform, but using a more complex one was considered out of the scope of this project. Even though it is relevant to the real world components of audio signals, it is not very useful at time localization (so that's why we used the STFT and AR models). Other approaches that could be considered are, for example, multi-resolution techniques. There has been a lot of research regarding *Computational Auditory Scene Analysis* (*CASA*) for creating computer systems for learn to recognize sound sources, so this could be another source of information.
- Addition of complexity and dependencies. In music, there are several cases where there are no-independent relations. For example, we can find correlation between notes, timing, and instruments (e.g., drums and bass). We could use extra information from a tablature or another source to create dependencies for restricting the outputs (e.g., if we know that some specific notes never play as a chord because they are dissonant, maybe we will not select them when we have related observations).
- Convert to Midi. One of the most interesting features of *PCSS* is the possibility to convert the output to a midi format. An onset/offset detector is required to determine length of notes, which should not be

complicated to build. Most musical applications work with this standard, and it may be the easiest way to extend other functionality that already exists (such as music transcription)

6.2.3 Conclusions

A model for *Polyphonic Correlated Source Separation* that uses both spectral information and temporal information for achieving the separation was built. There are still many things to add for making this model capable of separating simple real-life instrument notes, however for the scope of the project it has an acceptable performance.

The operation of the filtering model is good for simple and complex observations, being obviously better in the former case. It seems to have no problem to generalize: identifies different observations of a same note, and works for very simple real-life instrument notes (such as notes from a harmonica). In addition, this source separation approach can extract S different sources from only one observation, (in comparison with other source separation approaches that need S observations to separate S sources, as it is the case with *ICA*, 2.3) making it convenient for normal recording situations.

However, the actual model also has some deficiencies. It has many free parameters that have to be adjusted, and its performance is highly dependent on some of them. It requires a model of the sources to be built in order to achieve the separation and, as the number of sources increases, the quality of the separation decreases.

Appendix A

PCSS Derivations

A.1 Meaning of variables

We present a table that was created to help the reader to follow the meaning of every symbol used in the equations.

Symbol	Size	Description
Variables		
$\mathbf{Y}(t)$	$N \times 1$	Observation vector at time t .
$\mathbf{C}(t)$	$S \times 1$	Vector with the switching variables $c^{(s)}(t)$.
$c^{(s)}(t)$	1×1	Switching variable (Boolean) for model s at time t .
$\mathbf{F}(t)$	$N \times S$	Matrix with the contents of the linear models $\mathbf{f}^{(s)}$ at time t .
$\mathbf{f}^{(s)}(t)$	$N \times 1$	Vector related to the linear model s at time t .
$\mathbf{M}^{(s)}$	$N \times ON$	Matrix with the autoregressive coefficients of model s .
$\mathbf{b}^{(s)}$	$N \times 1$	Vector with the mean of the linear model $\mathbf{f}^{(s)}$.
$\tau^{(s)2}$	1×1	Variance of the linear model $\mathbf{f}^{(s)}$.
$\mathbf{g}(t)$	$ON \times 1$	Vector containing all the O regressive elements of $\mathbf{f}(t)$.
Dimensions		
N		Number of elements of each n -variate autoregressive model per time step (1 if taken from the raw signal, or N if taken from the <i>Fourier</i> components)
T		Number of time steps
O		Order of the autoregressive model
S		Number of sounds

A.2 Optimal ML Parameters (AR model)

Our approach to get the optimal \mathbf{M} , \mathbf{b} and τ^2 consists on obtaining the *Maximum Likelihood Estimate* for each parameter. Since the likelihood of a sequence \mathbf{f} ($\mathbf{f}(1), \mathbf{f}(2), \dots, \mathbf{f}(T)$) is

$$\begin{aligned} L(\mathbf{F}) &= P(\mathbf{f}(1))P(\mathbf{f}(2) | \mathbf{g}(2)) \dots P(\mathbf{f}(T) | \mathbf{g}(T)) \\ &= P(\mathbf{f}(1)) \prod_{t=2}^T P(\mathbf{f}(t) | \mathbf{g}(t)) \end{aligned} \quad (\text{A.2.1})$$

and given that we defined $P(\mathbf{f}(t) | \mathbf{g}(t))$ as

$$P(\mathbf{f}(t) | \mathbf{g}(t)) = \frac{1}{(2\pi\tau^2)^{|\mathbf{f}|/2}} \exp\left\{-\frac{1}{2\tau^2} [\mathbf{f}(t) - \mathbf{M}\mathbf{g}(t) - \mathbf{b}]^2\right\} \quad (\text{A.2.2})$$

therefore, the *log-likelihood* of \mathbf{f} given the parameters $L(\underline{\Theta}) \equiv P(\mathbf{f} | \underline{\Theta})$, where $\underline{\Theta} = (\mathbf{M}, \mathbf{b}, \tau^2)$, is given by

$$\begin{aligned} L(\underline{\Theta}) &= \log(P(\mathbf{f}(1))) + \sum_{t=2}^T \log(P(\mathbf{f}(t) | \mathbf{g}(t))) \\ &= \log\left\{\frac{1}{(2\pi\tau^2)^{|\mathbf{f}|/2}} \exp\left\{-\frac{1}{2\tau^2} [\mathbf{f}(1) - \langle \mathbf{f}(1) \rangle]^2\right\}\right\} + \sum_{t=2}^T \log\left\{\frac{1}{(2\pi\tau^2)^{|\mathbf{f}|/2}} \exp\left\{-\frac{1}{2\tau^2} [\mathbf{f}(t) - \mathbf{M}\mathbf{g}(t) - \mathbf{b}]^2\right\}\right\} \\ &= -\frac{|\mathbf{f}|}{2} \log(2\pi\tau^2) - \frac{1}{2\tau^2} [\mathbf{f}(1) - \langle \mathbf{f}(1) \rangle]^2 + \sum_{t=2}^T \left\{-\frac{|\mathbf{f}|}{2} \log(2\pi\tau^2) - \frac{1}{2\tau^2} [\mathbf{f}(t) - \mathbf{M}\mathbf{g}(t) - \mathbf{b}]^2\right\} \end{aligned} \quad (\text{A.2.3})$$

To obtain the *Maximum Likelihood Estimate* for each parameter, equation (A.2.3) has to be differentiated with respect to the corresponding parameter and then solved for it. This process for the mean \mathbf{b} , variance τ^2 and autoregressive matrix \mathbf{M} , is detailed as follows.

A.2.1 Optimal mean

By differentiating (A.2.3) with respect to \mathbf{b}_i , we get

$$\begin{aligned} \frac{\partial L(\underline{\Theta})}{\partial b_i} &= \frac{\partial}{\partial b_i} \sum_{t=2}^T -\frac{1}{2\tau^2} [\mathbf{f}(t) - \mathbf{M}\mathbf{g}(t) - \mathbf{b}]^2 \\ &= \frac{\partial}{\partial b_i} \sum_{t=2}^T -\frac{1}{2\tau^2} \sum_{k=1}^N \left[f_k(t) - \sum_l M_{kl} g_l(t) - b_k \right]^2 \\ &= \frac{1}{\tau^2} \sum_{t=2}^T \sum_{k=1}^N \left[f_k(t) - \sum_l M_{kl} g_l(t) - b_k \right] \frac{\partial b_k}{\partial b_i} \end{aligned} \quad (\text{A.2.4})$$

where the differential of component k with respect to i turns to be

$$\frac{\partial L(\Theta)}{\partial b_i} = \frac{1}{\tau^2} \sum_{t=2}^T \left[f_i(t) - \sum_l M_{il} g_l(t) - b_i \right]. \quad (\text{A.2.5})$$

By equating to zero and solving for b_i , we obtain

$$\begin{aligned} 0 &= \frac{1}{\tau^2} \sum_{t=2}^T \left[f_i(t) - \sum_l M_{il} g_l(t) - b_i \right] \\ \sum_{t=2}^T b_i &= \sum_{t=2}^T \left[f_i(t) - \sum_l M_{il} g_l(t) \right] \\ (T-1)b_i &= \sum_{t=2}^T \left[f_i(t) - \sum_l M_{il} g_l(t) \right] \\ b_i &= \frac{1}{T-1} \sum_{t=2}^T \left[f_i(t) - \sum_l M_{il} g_l(t) \right], \quad i \in \{1, \dots, N\}. \end{aligned} \quad (\text{A.2.6})$$

So the vector that minimizes the error towards a zero mean is

$$\begin{aligned} \mathbf{b} &= \frac{1}{T-1} \sum_{t=2}^T [\mathbf{f}(t) - \mathbf{M}\mathbf{g}(t)] \\ \mathbf{b} &= \tilde{\mathbf{f}} - \mathbf{M}\tilde{\mathbf{g}} \end{aligned} \quad (\text{A.2.7})$$

where the average of the constant vectors $\mathbf{f}(t)$ and $\mathbf{g}(t)$ are, respectively, $\tilde{\mathbf{f}}$ and $\tilde{\mathbf{g}}$. In addition, it is worth commenting that in these derivations we consider that $\langle \mathbf{f}(1) \rangle \neq \mathbf{b}$ (if this is not considered, clearly we would have some extra terms after differentiating).

A.2.2 Optimal variance

To obtain the optimal variance τ^2 let's use (A.2.3), consider a change of variable by making $\beta = \tau^2$, and then differentiate the equation with respect to β .

$$\begin{aligned} \frac{\partial L(\Theta)}{\partial \beta} &= \frac{\partial}{\partial \beta} \left\{ -\frac{|\mathbf{f}|}{2} \log(2\pi\beta) - \frac{\beta^{-1}}{2} [\mathbf{f}(1) - \langle \mathbf{f}(1) \rangle]^2 + \sum_{t=2}^T \left\{ -\frac{|\mathbf{f}|}{2} \log(2\pi\beta) - \frac{\beta^{-1}}{2} [\mathbf{f}(t) - \mathbf{M}\mathbf{g}(t) - \mathbf{b}]^2 \right\} \right\} \\ &= -\frac{|\mathbf{f}|}{2\beta} + \frac{\beta^{-2}}{2} [\mathbf{f}(1) - \langle \mathbf{f}(1) \rangle]^2 + \sum_{t=2}^T \left\{ -\frac{|\mathbf{f}|}{2\beta} + \frac{\beta^{-2}}{2} [\mathbf{f}(t) - \mathbf{M}\mathbf{g}(t) - \mathbf{b}]^2 \right\} \end{aligned} \quad (\text{A.2.8})$$

By substituting β for τ^2 , equating to zero and solving for τ^2 , we obtain:

$$\begin{aligned}
0 &= -\frac{|\mathbf{f}|}{2\tau^2} + \frac{1}{2\tau^4} [\mathbf{f}(1) - \langle \mathbf{f}(1) \rangle]^2 + \sum_{t=2}^T \left\{ -\frac{|\mathbf{f}|}{2\tau^2} + \frac{1}{2\tau^4} [\mathbf{f}(t) - \mathbf{M}\mathbf{g}(t) - \mathbf{b}]^2 \right\} \\
0 &= -\frac{|\mathbf{f}|}{2\tau^2} + \frac{1}{2\tau^4} [\mathbf{f}(1) - \langle \mathbf{f}(1) \rangle]^2 - \sum_{t=2}^T \frac{|\mathbf{f}|}{2\tau^2} + \frac{1}{2\tau^4} \sum_{t=2}^T [\mathbf{f}(t) - \mathbf{M}\mathbf{g}(t) - \mathbf{b}]^2 \\
\frac{T|\mathbf{f}|}{2\tau^2} &= \frac{1}{2\tau^4} \left([\mathbf{f}(1) - \langle \mathbf{f}(1) \rangle]^2 + \sum_{t=2}^T [\mathbf{f}(t) - \mathbf{M}\mathbf{g}(t) - \mathbf{b}]^2 \right) \\
\tau^2 &= \frac{1}{T|\mathbf{f}|} \left([\mathbf{f}(1) - \langle \mathbf{f}(1) \rangle]^2 + \sum_{t=2}^T [\mathbf{f}(t) - \mathbf{M}\mathbf{g}(t) - \mathbf{b}]^2 \right) \tag{A.2.9}
\end{aligned}$$

and by changing \mathbf{b} for its corresponding equation (A.2.7), we get the equivalent expression (A.2.10).

$$\tau^2 = \frac{1}{T|\mathbf{f}|} \left([\mathbf{f}(1) - \langle \mathbf{f}(1) \rangle]^2 + \sum_{t=2}^T [(\mathbf{f}(t) - \tilde{\mathbf{f}}) - \mathbf{M}(\mathbf{g}(t) - \tilde{\mathbf{g}})]^2 \right) \tag{A.2.10}$$

A.2.3 Optimal autoregressive matrix

Similarly as in A.2.2, let's use (A.2.3) to obtain the optimal autoregressive matrix \mathbf{M} . By changing \mathbf{b} for its equivalent expression (A.2.7), we get

$$L(\Theta) = -\frac{|\mathbf{f}|}{2} \log(2\pi\tau^2) - \frac{1}{2\tau^2} [\mathbf{f}(1) - \langle \mathbf{f}(1) \rangle]^2 + \sum_{t=2}^T \left\{ -\frac{|\mathbf{f}|}{2} \log(2\pi\tau^2) - \frac{1}{2\tau^2} [(\mathbf{f}(t) - \tilde{\mathbf{f}}) - \mathbf{M}(\mathbf{g}(t) - \tilde{\mathbf{g}})]^2 \right\} \tag{A.2.11}$$

To minimize the error contribution of \mathbf{M} , we start by differentiating (A.2.11) with respect to M_{ij} .

$$\begin{aligned}
\frac{\partial L(\Theta)}{\partial M_{ij}} &= \frac{\partial}{\partial M_{ij}} \sum_{t=2}^T -\frac{1}{2\tau^2} [(\mathbf{f}(t) - \tilde{\mathbf{f}}) - \mathbf{M}(\mathbf{g}(t) - \tilde{\mathbf{g}})]^2 \\
&= \frac{\partial}{\partial M_{ij}} \sum_{t=2}^T -\frac{1}{2\tau^2} \sum_{k=1}^N \left[(f_k(t) - \tilde{f}_k) - \sum_l M_{kl} (g(t)_l - \tilde{g}_l) \right]^2 \\
&= \frac{1}{\tau^2} \sum_{t=2}^T \sum_{k=1}^N \left[(f_k(t) - \tilde{f}_k) - \sum_l M_{kl} (g(t)_l - \tilde{g}_l) \right] \sum_{l'} (g(t)_{l'} - \tilde{g}_{l'}) \frac{\partial M_{kl'}}{\partial M_{ij}} \tag{A.2.12}
\end{aligned}$$

By differentiating the components k, l' with respect to i, j we find that

$$\frac{\partial L(\Theta)}{\partial M_{ij}} = \frac{1}{\tau^2} \sum_{t=2}^T \left[(f_i(t) - \tilde{f}_i) - \sum_l M_{il} (g(t)_l - \tilde{g}_l) \right] (g(t)_j - \tilde{g}_j) \tag{A.2.13}$$

and by equating to zero we get

$$\begin{aligned}
0 &= \frac{1}{\tau^2} \sum_{t=2}^T \left[(f_i(t) - \tilde{f}_i) - \sum_l M_{il} (g(t)_l - \tilde{g}_l) \right] (g(t)_j - \tilde{g}_j) \\
\sum_{t=2}^T \sum_l M_{il} (g(t)_l - \tilde{g}_l) (g(t)_j - \tilde{g}_j) &= \sum_{t=2}^T (f_i(t) - \tilde{f}_i) (g(t)_j - \tilde{g}_j). \quad (\text{A.2.14})
\end{aligned}$$

To simplify this expression, consider the example vectors \mathbf{f} and \mathbf{g} and the following multiplication:

$$\mathbf{fg}^T = \begin{pmatrix} f_1 \\ f_2 \\ \dots \\ f_N \end{pmatrix} \begin{pmatrix} g_1 & g_2 & \dots & g_{oN} \end{pmatrix} = \begin{pmatrix} f_1 g_1 & f_1 g_2 & \dots & f_1 g_{oN} \\ f_2 g_1 & f_2 g_2 & \dots & f_2 g_{oN} \\ \dots & \dots & \dots & \dots \\ f_N g_1 & f_N g_2 & \dots & f_N g_{oN} \end{pmatrix} \quad (\text{A.2.15})$$

where

$$[\mathbf{fg}^T]_{ij} = f_i g_j \quad (\text{A.2.16})$$

Finally, using (A.2.16) to re-express (A.2.14), we get

$$\sum_l M_{il} \sum_{t=2}^T (g(t)_l - \tilde{g}_l) (g(t)_j - \tilde{g}_j) = \sum_{t=2}^T (f_i(t) - \tilde{f}_i) (g(t)_j - \tilde{g}_j), \quad \forall i, l, j \quad (\text{A.2.17})$$

that in the form of the above matrix would correspond to

$$\left[\mathbf{M} \sum_{t=2}^T (g(t)_l - \tilde{g}_l) (g(t)_j - \tilde{g}_j)^T \right]_{ij} = \sum_{t=2}^T \left[(f_i(t) - \tilde{f}_i) (g(t)_j - \tilde{g}_j)^T \right]_{ij} \quad (\text{A.2.18})$$

Therefore, we have

$$\begin{aligned}
\mathbf{M} \sum_{t=2}^T (\mathbf{g}(t) - \tilde{\mathbf{g}}) (\mathbf{g}(t) - \tilde{\mathbf{g}})^T &= \sum_{t=2}^T (\mathbf{f}(t) - \tilde{\mathbf{f}}) (\mathbf{g}(t) - \tilde{\mathbf{g}})^T \\
\mathbf{M} &= \left(\sum_{t=2}^T (\mathbf{f}(t) - \tilde{\mathbf{f}}) (\mathbf{g}(t) - \tilde{\mathbf{g}})^T \right) \left(\sum_{t=2}^T (\mathbf{g}(t) - \tilde{\mathbf{g}}) (\mathbf{g}(t) - \tilde{\mathbf{g}})^T \right)^{-1} \quad (\text{A.2.19})
\end{aligned}$$

Again, in all these derivations, we consider that $\langle \mathbf{f}(1) \rangle \neq \mathbf{b}$.

A.3 Optimal *ML* Parameters (*q* densities)

The *Kullback-Leibler* function between the joint posterior conditional probability J (4.2.2) and our factorised approximation Q (4.2.6) is given by

$$\begin{aligned}
KL(Q(x), J(x)) &= \int Q(x) \log \left(\frac{Q(x)}{J(x)} \right) dx \\
&= \int Q(x) \log(Q(x)) dx - \int Q(x) \log(J(x)) dx \\
&= \langle \log(Q(x)) \rangle_{Q(x)} - \langle \log(J(x)) \rangle_{Q(x)} \quad (\text{A.2.20})
\end{aligned}$$

where:

$$\begin{aligned} \langle \log(Q(x)) \rangle_{Q(x)} &= \left\langle \log \left(\prod_{t=1}^T q(\mathbf{C}(t)) q(\mathbf{f}^{(1)}(t)) q(\mathbf{f}^{(2)}(t)) \right) \right\rangle_{Q(x)} \\ &= \sum_{t=1}^T \left(\left\langle \log(q(\mathbf{C}(t))) \right\rangle_{q(\mathbf{C}(t))} + \left\langle \log(q(\mathbf{f}^{(1)}(t))) \right\rangle_{q(\mathbf{f}^{(1)}(t))} \right. \\ &\quad \left. + \left\langle \log(q(\mathbf{f}^{(2)}(t))) \right\rangle_{q(\mathbf{f}^{(2)}(t))} \right) \end{aligned} \quad (\text{A.2.21})$$

and

$$\begin{aligned} \langle \log(J(x)) \rangle_{Q(x)} &= \left\langle \log \left(\frac{P(\mathbf{Y} | \mathbf{C}, \mathbf{f}^{(1)}, \mathbf{f}^{(2)}) P(\mathbf{C}, \mathbf{f}^{(1)}, \mathbf{f}^{(2)})}{P(\mathbf{Y})} \right) \right\rangle_{Q(x)} \\ &\propto \left\langle \log(P(\mathbf{Y} | \mathbf{C}, \mathbf{f}^{(1)}, \mathbf{f}^{(2)}) P(\mathbf{C}, \mathbf{f}^{(1)}, \mathbf{f}^{(2)})) \right\rangle_{Q(x)} \\ &= \left\langle \log(P(\mathbf{Y} | \mathbf{C}, \mathbf{f}^{(1)}, \mathbf{f}^{(2)}) P(\mathbf{C}) P(\mathbf{f}^{(1)}) P(\mathbf{f}^{(2)})) \right\rangle_{Q(x)} \\ &\approx \left\langle \log \left(\prod_{t=1}^T P(\mathbf{Y}(t) | \mathbf{C}(t), \mathbf{f}^{(1)}(t), \mathbf{f}^{(2)}(t)) \cdot P(\mathbf{C}(t) | \mathbf{C}(t-1)) \right) \right\rangle_{Q(x)} \\ &= \sum_{t=1}^T \left(\left\langle \log(P(\mathbf{Y}(t) | \mathbf{C}(t), \mathbf{f}^{(1)}(t), \mathbf{f}^{(2)}(t))) \right\rangle_{q(\mathbf{C}(t))q(\mathbf{f}^{(1)}(t))q(\mathbf{f}^{(2)}(t))} \right. \\ &\quad \left. + \left\langle \log(P(\mathbf{C}(t) | \mathbf{C}(t-1))) \right\rangle_{q(\mathbf{C}(t))q(\mathbf{C}(t-1))} \right. \\ &\quad \left. + \left\langle \log(P(\mathbf{f}^{(1)}(t) | \mathbf{g}^{(1)}(t))) \right\rangle_{q(\mathbf{f}^{(1)}(t))q(\mathbf{g}^{(1)}(t))} + \left\langle \log(P(\mathbf{f}^{(2)}(t) | \mathbf{g}^{(2)}(t))) \right\rangle_{q(\mathbf{f}^{(2)}(t))q(\mathbf{g}^{(2)}(t))} \right) \end{aligned} \quad (\text{A.2.22})$$

By differentiating $KL(Q(x), J(x))$ with respect to $q(\mathbf{C}(t))$, we get

$$\frac{\partial KL}{\partial q(\mathbf{C}(t))} = \left(\begin{aligned} &\left\langle \log(q(\mathbf{C}(t))) \right\rangle - \left\langle \log(P(\mathbf{Y}(t) | \mathbf{C}(t), \mathbf{f}^{(1)}(t), \mathbf{f}^{(2)}(t))) \right\rangle_{q(\mathbf{f}^{(1)}(t))q(\mathbf{f}^{(2)}(t))} \\ &- \left\langle \log(P(\mathbf{C}(t) | \mathbf{C}(t-1))) \right\rangle_{q(\mathbf{C}(t-1))} - \left\langle \log(P(\mathbf{C}(t+1) | \mathbf{C}(t))) \right\rangle_{q(\mathbf{C}(t+1))} + \text{const} \end{aligned} \right) \quad (\text{A.2.23})$$

and by equating to zero and solving for $q(\mathbf{C}(t))$, we obtain

$$\begin{aligned} \log(q(\mathbf{C}(t))) &= \left(\begin{aligned} &\left\langle \log(P(\mathbf{Y}(t) | \mathbf{C}(t), \mathbf{f}^{(1)}(t), \mathbf{f}^{(2)}(t))) \right\rangle_{q(\mathbf{f}^{(1)}(t))q(\mathbf{f}^{(2)}(t))} \\ &+ \left\langle \log(P(\mathbf{C}(t) | \mathbf{C}(t-1))) \right\rangle_{q(\mathbf{C}(t-1))} + \left\langle \log(P(\mathbf{C}(t+1) | \mathbf{C}(t))) \right\rangle_{q(\mathbf{C}(t+1))} + \text{const} \end{aligned} \right) \\ q(\mathbf{C}(t)) \alpha \exp &\left(\begin{aligned} &\left\langle \log(P(\mathbf{Y}(t) | \mathbf{C}(t), \mathbf{f}^{(1)}(t), \mathbf{f}^{(2)}(t))) \right\rangle_{q(\mathbf{f}^{(1)}(t))q(\mathbf{f}^{(2)}(t))} \\ &+ \left\langle \log(P(\mathbf{C}(t) | \mathbf{C}(t-1))) \right\rangle_{q(\mathbf{C}(t-1))} + \left\langle \log(P(\mathbf{C}(t+1) | \mathbf{C}(t))) \right\rangle_{q(\mathbf{C}(t+1))} \end{aligned} \right) \end{aligned} \quad (\text{A.2.24})$$

Similarly, by differentiating $KL(Q(x), J(x))$ with respect to $q(\mathbf{f}^{(1)}(t))$, we obtain

$$\frac{\partial KL}{\partial q(\mathbf{f}^{(1)}(t))} = \left[\begin{array}{l} \left\langle \log(q(\mathbf{f}^{(1)}(t))) - \left\langle \log(P(\mathbf{Y}(t) | \mathbf{C}(t), \mathbf{f}^{(1)}(t), \mathbf{f}^{(2)}(t))) \right\rangle_{q(\mathbf{C}(t))q(\mathbf{f}^{(2)}(t))} \right\rangle_{q(\mathbf{f}^{(1)}(t))} \\ - \left\langle \log(P(\mathbf{f}^{(1)}(t) | \mathbf{g}^{(1)}(t))) \right\rangle_{q(\mathbf{g}^{(1)}(t))} \\ - \left\langle \log(P(\mathbf{f}^{(1)}(t+1) | \mathbf{g}^{(1)}(t+1))) \right\rangle_{q(\mathbf{f}^{(1)}(t+1), \mathbf{f}^{(1)}(t-1), \dots, \mathbf{f}^{(1)}(t-o+1))} + \text{const} \end{array} \right] \quad (\text{A.2.25})$$

and by equating to zero and solving for $q(\mathbf{f}^{(1)}(t))$ we get

$$\begin{aligned} \log(q(\mathbf{f}^{(1)}(t))) &= \left[\begin{array}{l} \left\langle \log(P(\mathbf{Y}(t) | \mathbf{C}(t), \mathbf{f}^{(1)}(t), \mathbf{f}^{(2)}(t))) \right\rangle_{q(\mathbf{C}(t))q(\mathbf{f}^{(2)}(t))} + \left\langle \log(P(\mathbf{f}^{(1)}(t) | \mathbf{g}^{(1)}(t))) \right\rangle_{q(\mathbf{g}^{(1)}(t))} \\ + \left\langle \log(P(\mathbf{f}^{(1)}(t+1) | \mathbf{g}^{(1)}(t+1))) \right\rangle_{q(\mathbf{f}^{(1)}(t+1))q(\mathbf{f}^{(1)}(t-1)) \dots q(\mathbf{f}^{(1)}(t-o+1))} + \text{const} \end{array} \right] \\ q(\mathbf{f}^{(1)}(t)) &\propto \exp \left[\begin{array}{l} \left\langle \log(P(\mathbf{Y}(t) | \mathbf{C}(t), \mathbf{f}^{(1)}(t), \mathbf{f}^{(2)}(t))) \right\rangle_{q(\mathbf{C}(t))q(\mathbf{f}^{(2)}(t))} + \left\langle \log(P(\mathbf{f}^{(1)}(t) | \mathbf{g}^{(1)}(t))) \right\rangle_{q(\mathbf{g}^{(1)}(t))} \\ + \left\langle \log(P(\mathbf{f}^{(1)}(t+1) | \mathbf{g}^{(1)}(t+1))) \right\rangle_{q(\mathbf{f}^{(1)}(t+1))q(\mathbf{f}^{(1)}(t-1)) \dots q(\mathbf{f}^{(1)}(t-o+1))} \end{array} \right] \end{aligned} \quad (\text{A.2.26})$$

Finally, an analogous procedure done for $q(\mathbf{f}^{(1)}(t))$ follows for $q(\mathbf{f}^{(2)}(t))$, so we get

$$q(\mathbf{f}^{(2)}(t)) \propto \exp \left[\begin{array}{l} \left\langle \log(P(\mathbf{Y}(t) | \mathbf{C}(t), \mathbf{f}^{(1)}(t), \mathbf{f}^{(2)}(t))) \right\rangle_{q(\mathbf{C}(t))q(\mathbf{f}^{(1)}(t))} + \left\langle \log(P(\mathbf{f}^{(2)}(t) | \mathbf{g}^{(2)}(t))) \right\rangle_{q(\mathbf{g}^{(2)}(t))} \\ + \left\langle \log(P(\mathbf{f}^{(2)}(t+1) | \mathbf{g}^{(2)}(t+1))) \right\rangle_{q(\mathbf{f}^{(2)}(t+1))q(\mathbf{f}^{(2)}(t-1)) \dots q(\mathbf{f}^{(2)}(t-o+1))} \end{array} \right] \quad (\text{A.2.27})$$

A.3.1 Optimal $q(\mathbf{f}^{(s)}(t))$

By introducing the probability density functions (4.1.3) and (4.1.11) into (A.2.26), we get

$$q(\mathbf{f}^{(1)}(t)) \propto \exp \left[\begin{array}{l} \left\langle \log \left(\exp \left(-\frac{1}{2\sigma^2} [\mathbf{Y}(t) - c^{(1)}(t)\mathbf{f}^{(1)}(t) - c^{(2)}(t)\mathbf{f}^{(2)}(t)]^2 \right) \right) \right\rangle_{q(\mathbf{C}(t))q(\mathbf{f}^{(2)}(t))} \\ + \left\langle \log \left(\exp \left(-\frac{1}{2\tau^2} [\mathbf{f}^{(1)}(t) - \mathbf{M}^{(1)}\mathbf{g}^{(1)}(t) - \mathbf{b}^{(1)}]^2 \right) \right) \right\rangle_{q(\mathbf{g}^{(1)}(t))} \\ + \left\langle \log \left(\exp \left(-\frac{1}{2\tau^2} [\mathbf{f}^{(1)}(t+1) - \mathbf{M}^{(1)}\mathbf{g}^{(1)}(t+1) - \mathbf{b}^{(1)}]^2 \right) \right) \right\rangle_{q(\mathbf{f}^{(1)}(t+1))q(\mathbf{f}^{(1)}(t-1)) \dots q(\mathbf{f}^{(1)}(t-o+1))} \end{array} \right]$$

$$\begin{aligned}
q(\mathbf{f}^{(1)}(t)) &= \exp \left\{ -\frac{1}{2\sigma^2} \left\langle \left[\mathbf{Y}(t) - c^{(1)}(t)\mathbf{f}^{(1)}(t) - c^{(2)}(t)\mathbf{f}^{(2)}(t) \right]^2 \right\rangle_{q(\mathbf{C}(t))q(\mathbf{f}^{(2)}(t))} \right. \\
&\quad \left. -\frac{1}{2\tau^2} \left\langle \left[\mathbf{f}^{(1)}(t) - \mathbf{M}^{(1)}\mathbf{g}^{(1)}(t) - \mathbf{b}^{(1)} \right]^2 \right\rangle_{q(\mathbf{g}^{(1)}(t))} \right. \\
&\quad \left. -\frac{1}{2\tau^2} \left\langle \left[\mathbf{f}^{(1)}(t+1) - \mathbf{M}^{(1)}\mathbf{g}^{(1)}(t+1) - \mathbf{b}^{(1)} \right]^2 \right\rangle_{q(\mathbf{f}^{(1)}(t+1))q(\mathbf{f}^{(1)}(t-1))\dots q(\mathbf{f}^{(1)}(t-o+1))} \right\} \\
&= \exp \left\{ -\frac{1}{2\sigma^2} \left\langle \begin{array}{l} \mathbf{Y}^T(t)\mathbf{Y}(t) - 2c^{(1)}(t)\mathbf{Y}^T(t)\mathbf{f}^{(1)}(t) \\ -2c^{(2)}(t)\mathbf{Y}^T(t)\mathbf{f}^{(2)}(t) \\ +2c^{(1)}(t)c^{(2)}(t)\mathbf{f}^{1T}(t)\mathbf{f}^{(2)}(t) \\ +c^{(1)}(t)^2\mathbf{f}^{(1)T}(t)\mathbf{f}^{(1)}(t) + c^{(2)}(t)^2\mathbf{f}^{(2)T}(t)\mathbf{f}^{(2)}(t) \end{array} \right\rangle_{q(\mathbf{C}(t))q(\mathbf{f}^{(2)}(t))} \right. \\
&\quad \left. -\frac{1}{2\tau^2} \left\langle \begin{array}{l} \mathbf{f}^{(1)T}(t)\mathbf{f}^{(1)}(t) - 2\mathbf{f}^{(1)T}(t)\mathbf{M}^{(1)}\mathbf{g}^{(1)}(t) \\ -2\mathbf{f}^{(1)T}(t)\mathbf{b}^{(1)} + 2\mathbf{b}^{(1)T}\mathbf{M}^{(1)}\mathbf{g}^{(1)}(t) \\ +(\mathbf{M}^{(1)}\mathbf{g}^{(1)}(t))^T(\mathbf{M}^{(1)}\mathbf{g}^{(1)}(t)) + \mathbf{b}^{(1)T}\mathbf{b}^{(1)} \end{array} \right\rangle_{q(\mathbf{g}^{(1)}(t))} \right. \\
&\quad \left. -\frac{1}{2\tau^2} \left\langle \begin{array}{l} \mathbf{f}^{(1)T}(t+1)\mathbf{f}^{(1)}(t+1) \\ -2\mathbf{f}^{(1)T}(t+1)\mathbf{M}^{(1)}\mathbf{g}^{(1)}(t+1) - 2\mathbf{f}^{(1)T}(t+1)\mathbf{b}^{(1)} \\ +2\mathbf{b}^{(1)T}\mathbf{M}^{(1)}\mathbf{g}^{(1)}(t+1) \\ +(\mathbf{M}^{(1)}\mathbf{g}^{(1)}(t+1))^T(\mathbf{M}^{(1)}\mathbf{g}^{(1)}(t+1)) + \mathbf{b}^{(1)T}\mathbf{b}^{(1)} \end{array} \right\rangle_{q(\mathbf{f}^{(1)}(t+1))q(\mathbf{f}^{(1)}(t-1))\dots q(\mathbf{f}^{(1)}(t-o+1))} \right\} \\
\end{aligned} \tag{A.2.28}$$

where $c^{(s)}(t)^2$ equals $c^{(s)}(t)$, since c is a Boolean variable.

After eliminating all those terms that are not dependent on $\mathbf{f}^{(1)}(t)$,¹ our equation simplifies to:

¹ The constant terms, if being included in (A.2.25), would have been dropped at that time. However, since we are inserting the elements after they were explained and are required for further progress, by including them at that point could confuse the reader (more).

$$q(\mathbf{f}^{(1)}(t)) \propto \exp \left[-\frac{1}{2\sigma^2} \begin{pmatrix} -2\langle c^{(1)}(t) \rangle_{q(\mathbf{C}(t))} \mathbf{Y}(t)^T \mathbf{f}^{(1)}(t) \\ +2\langle c^{(1)}(t)c^{(2)}(t) \rangle_{q(\mathbf{C}(t))} \mathbf{f}^{(1)T}(t) \langle \mathbf{f}^{(2)}(t) \rangle_{q(\mathbf{f}^{(2)}(t))} \\ +\langle c^{(1)}(t) \rangle_{q(\mathbf{C}(t))} \mathbf{f}^{(1)T}(t) \mathbf{f}^{(1)}(t) \end{pmatrix} \right] \\
-\frac{1}{2\tau^2} \left(\mathbf{f}^{(1)T}(t) \mathbf{f}^{(1)}(t) - 2\mathbf{f}^{(1)T}(t) \mathbf{M}^{(1)} \langle \mathbf{g}^{(1)}(t) \rangle_{q(\mathbf{g}^{(1)}(t))} - 2\mathbf{f}^{(1)T}(t) \mathbf{b}^{(1)} \right) \\
-\frac{1}{2\tau^2} \begin{pmatrix} \langle -2\mathbf{f}^{(1)T}(t+1) \rangle_{q(\mathbf{f}^{(1)}(t+1))} \mathbf{M}^{(o1)(1)} \mathbf{f}^{(1)}(t) \\ +2\mathbf{b}^{(1)T} \mathbf{M}^{(o1)(1)} \mathbf{f}^{(1)}(t) \\ +\langle (\mathbf{M}^{(1)} \mathbf{g}^{(1)}(t+1))^T (\mathbf{M}^{(1)} \mathbf{g}^{(1)}(t+1)) \rangle_{q(\mathbf{f}^{(1)}(t+1)q(\mathbf{f}^{(1)}(t-1))\dots q(\mathbf{f}^{(1)}(t-o+1))} \end{pmatrix} \quad (\text{A.2.29})$$

where $\mathbf{M}^{(o1)(1)}$, as explained in (4.1.10), is the part of the $\mathbf{M}^{(1)}$ matrix that multiplies with the autoregressive components of order one. Also, when possible, all elements of $\mathbf{g}^{(1)}(t+1)$ that are not of order 1 are dropped, therefore resulting into $\mathbf{f}^{(1)}(t)$. By factorising (A.2.29), we get the following expression for each $f_i^{(1)}(t)$:

$$q(f_i^{(1)}(t)) \propto \exp \left[-\frac{1}{2\sigma^2} \begin{pmatrix} -2\langle c^{(1)}(t) \rangle_{q(c^{(1)}(t))} y_i(t) f_i^{(1)}(t) \\ +2\langle c^{(1)}(t)c^{(2)}(t) \rangle_{q(\mathbf{C}(t))} f_i^{(1)}(t) \langle f_i^{(2)}(t) \rangle_{q(f_i^{(2)}(t))} \\ +\langle c^{(1)}(t) \rangle_{q(c^{(1)}(t))} f_i^{(1)}(t)^2 \end{pmatrix} \right] \\
-\frac{1}{2\tau^2} \left(f_i^{(1)}(t)^2 - 2f_i^{(1)}(t) \sum_j M_{ij}^{(1)} \langle g_j^{(1)}(t) \rangle_{q(g_j^{(1)}(t))} - 2f_i^{(1)}(t) b_i^{(1)} \right) \\
-\frac{1}{2\tau^2} \begin{pmatrix} -2f_i^{(1)}(t) \sum_j M_{ji}^{(o1)(1)} \langle f_j^{(1)}(t+1) \rangle_{q(\mathbf{f}^{(1)}(t+1))} \\ +2f_i^{(1)}(t) \sum_j M_{ji}^{(o1)(1)} b_j^{(1)} \\ +2\langle \mathbf{g}^{(1)T}(t+1) \rangle_{q(\mathbf{f}^{(1)}(t))\dots q(\mathbf{f}^{(1)}(t-o))} \mathbf{M}^{(1)T} \mathbf{M}^{(1)} \mathbf{Z}_i^{(ON)} f_i^{(1)}(t) \\ -f_i^{(1)}(t)^2 \mathbf{Z}_i^{(ON)T} \mathbf{M}^{(1)T} \mathbf{M}^{(1)} \mathbf{Z}_i^{(ON)} \end{pmatrix} \quad (\text{A.2.30})$$

which makes use of an operator $\mathbf{Z}_i^{(ON)}$ that creates a vector with ON Boolean values where all but i are zero.

As we can see, (A.2.30) can be parameterised by a *Gaussian* with:

$$\begin{aligned}
q(f_i^{(1)}(t)) &\propto \exp\left\{-\frac{1}{2}\left(\mathbf{A}f_i^{(1)}(t)^2 + \mathbf{B}f_i^{(1)}(t) + \mathbf{X}\right)\right\} \\
&= \exp\left\{-\frac{\mathbf{A}}{2}\left(f_i^{(1)}(t)^2 + \frac{\mathbf{B}}{\mathbf{A}}f_i^{(1)}(t)\right) + \frac{\mathbf{X}}{2}\right\} \\
&= \exp\left\{-\frac{\mathbf{A}}{2}\left(f_i^{(1)}(t)^2 + \frac{2\mathbf{B}}{2\mathbf{A}}f_i^{(1)}(t) + \frac{\mathbf{B}^2}{4\mathbf{A}^2}\right) + \frac{\mathbf{A}\mathbf{B}^2}{8\mathbf{A}^2} + \frac{\mathbf{X}}{2}\right\} \\
&= \exp\left\{-\frac{\mathbf{A}}{2}\left(f_i^{(1)}(t) + \frac{\mathbf{B}}{2\mathbf{A}}\right)^2\right\} + \text{const} \tag{A.2.31}
\end{aligned}$$

By collecting the associated terms A and B from (A.2.30), we obtain:

$$\begin{aligned}
\mathbf{A} &= \frac{\langle c^{(1)}(t) \rangle_{q(c^{(1)}(t))}}{\sigma^2} + \frac{\left(1 + [\mathbf{M}^{(1)\text{T}}\mathbf{M}^{(1)}]_{ii}\right)}{\tau^{(1)2}} \\
&= \frac{\tau^{(1)2} \langle c^{(1)}(t) \rangle_{q(c^{(1)}(t))} + \sigma^2 \left(1 + [\mathbf{M}^{(1)\text{T}}\mathbf{M}^{(1)}]_{ii}\right)}{\sigma^2 \tau^{(1)2}} \tag{A.2.32}
\end{aligned}$$

and

$$\begin{aligned}
\mathbf{B} &= \left\{ \frac{1}{\sigma^2} \begin{pmatrix} -2\langle c^{(1)}(t) \rangle_{q(c^{(1)}(t))} y_i(t) \\ +2\langle c^{(1)}(t)c^{(2)}(t) \rangle_{q(c(t))} \langle f_i^{(2)}(t) \rangle_{q(f_i^{(2)}(t))} \end{pmatrix} + \frac{1}{\tau^{(1)2}} \begin{pmatrix} -2\sum_j M_{ij}^{(1)} \langle g_j^{(1)}(t) \rangle_{q(g_j^{(1)}(t))} - 2b_i^{(1)} \\ -2\sum_j M_{ji}^{(o1)(1)} \langle f_j^{(1)}(t+1) \rangle_{q(f_j^{(1)}(t+1))} + 2\sum_j M_{ji}^{(o1)(1)} b_j^{(1)} \\ +2\langle \mathbf{g}^{(1)\text{T}}(t+1) \rangle_{q(\mathbf{r}^{(1)}(t) \dots q(\mathbf{r}^{(1)}(t-o))} \mathbf{M}^{(1)\text{T}}\mathbf{M}^{(1)}\mathbf{Z}_i^{(ON)} \\ -2\mathbf{Z}_i^{(ON)\text{T}}\mathbf{M}^{(1)\text{T}}\mathbf{M}^{(1)}\mathbf{Z}_i^{(ON)} f_i^{(1)}(t) \end{pmatrix} \right\} \\
&= \frac{\tau^{(1)2}}{\sigma^2 \tau^{(1)2}} \begin{pmatrix} -2\langle c^{(1)}(t) \rangle_{q(c^{(1)}(t))} y_i(t) \\ +2\langle c^{(1)}(t)c^{(2)}(t) \rangle_{q(c(t))} \langle f_i^{(2)}(t) \rangle_{q(f_i^{(2)}(t))} \end{pmatrix} + \frac{\sigma^2}{\sigma^2 \tau^{(1)2}} \begin{pmatrix} -2\sum_j M_{ij}^{(1)} \langle g_j^{(1)}(t) \rangle_{q(g_j^{(1)}(t))} - 2b_i^{(1)} \\ -2\sum_j M_{ji}^{(o1)(1)} \langle f_j^{(1)}(t+1) \rangle_{q(f_j^{(1)}(t+1))} + 2\sum_j M_{ji}^{(o1)(1)} b_j^{(1)} \\ +2\langle \mathbf{g}^{(1)\text{T}}(t+1) \rangle_{q(\mathbf{r}^{(1)}(t) \dots q(\mathbf{r}^{(1)}(t-o))} \mathbf{M}^{(1)\text{T}}\mathbf{M}^{(1)}\mathbf{Z}_i^{(ON)} \\ -2\mathbf{Z}_i^{(ON)\text{T}}\mathbf{M}^{(1)\text{T}}\mathbf{M}^{(1)}\mathbf{Z}_i^{(ON)} f_i^{(1)}(t) \end{pmatrix} \\
&= \frac{2}{\sigma^2 \tau^{(1)2}} \left\{ \tau^{(1)2} \langle c^{(1)}(t) \rangle_{q(c^{(1)}(t))} \begin{pmatrix} -y_i(t) \\ +\langle c^{(2)}(t) \rangle_{q(c^{(2)}(t))} \langle f_i^{(2)}(t) \rangle_{q(f_i^{(2)}(t))} \end{pmatrix} + \sigma^2 \begin{pmatrix} -\sum_j M_{ij}^{(1)} \langle g_j^{(1)}(t) \rangle_{q(g_j^{(1)}(t))} - b_i^{(1)} \\ -\sum_j M_{ji}^{(o1)(1)} \langle f_j^{(1)}(t+1) \rangle_{q(f_j^{(1)}(t+1))} + \sum_j M_{ji}^{(o1)(1)} b_j^{(1)} \\ +\langle \mathbf{g}^{(1)\text{T}}(t+1) \rangle_{q(\mathbf{r}^{(1)}(t) \dots q(\mathbf{r}^{(1)}(t-o))} \mathbf{M}^{(1)\text{T}}\mathbf{M}^{(1)}\mathbf{Z}_i^{(ON)} \\ -\mathbf{Z}_i^{(ON)\text{T}}\mathbf{M}^{(1)\text{T}}\mathbf{M}^{(1)}\mathbf{Z}_i^{(ON)} f_i^{(1)}(t) \end{pmatrix} \right\} \tag{A.2.33}
\end{aligned}$$

So $q(f_i^{(1)}(t))$ is a *Gaussian* with variance

$$\left\langle \left(f_i^{(1)}(t)\right)^2 \right\rangle - \left\langle f_i^{(1)}(t) \right\rangle^2 = \frac{\sigma^2 \tau^{(1)2}}{\tau^{(1)2} \langle c^{(1)}(t) \rangle_{q(c^{(1)}(t))} + \sigma^2 \left(1 + [\mathbf{M}^{(1)\text{T}}\mathbf{M}^{(1)}]_{ii}\right)} \tag{A.2.34}$$

and mean

$$\langle f_i^{(1)}(t) \rangle = \frac{\tau^{(1)2} \langle c^{(1)}(t) \rangle_{q(c^{(1)}(t))} \left(\begin{array}{l} +y_i(t) \\ -\langle c^{(2)}(t) \rangle_{q(c^{(2)}(t))} \langle f_i^{(2)}(t) \rangle_{q(f_i^{(2)}(t))} \end{array} \right) + \sigma^2 \left(\begin{array}{l} \sum_j M_{ij}^{(1)} \langle g_j^{(1)}(t) \rangle_{q(g_j^{(1)}(t))} + b_i^{(1)} \\ + \sum_j M_{ji}^{(o1)(1)} \langle f_j^{(1)}(t+1) \rangle_{q(f_j^{(1)}(t+1))} - \sum_j M_{ji}^{(o1)(1)} b_j^{(1)} \\ - \langle \mathbf{g}^{(1)\top}(t+1) \rangle_{q(\mathbf{r}^{(1)}(t) \dots q(\mathbf{r}^{(1)}(t-o))} \mathbf{M}^{(1)\top} \mathbf{M}^{(1)} \mathbf{Z}_i^{(ON)} \\ + \mathbf{Z}_i^{(ON)\top} \mathbf{M}^{(1)\top} \mathbf{M}^{(1)} \mathbf{Z}_i^{(ON)} f_i^{(1)}(t) \end{array} \right)}{\tau^{(1)2} \langle c^{(1)}(t) \rangle_{q(c^{(1)}(t))} + \sigma^2 \left(1 + [\mathbf{M}^{(1)\top} \mathbf{M}^{(1)}]_{ii} \right)} \quad (\text{A.2.35})$$

Hence, $q(f_i^{(2)}(t))$ is also a *Gaussian* with variance

$$\left\langle (f_i^{(2)}(t))^2 \right\rangle - \langle f_i^{(2)}(t) \rangle^2 = \frac{\sigma^2 \tau^{(2)2}}{\tau^{(2)2} \langle c^{(2)}(t) \rangle_{q(c^{(2)}(t))} + \sigma^2 \left(1 + [\mathbf{M}^{(1)\top} \mathbf{M}^{(1)}]_{ii} \right)} \quad (\text{A.2.36})$$

and mean

$$\langle f_i^{(2)}(t) \rangle = \frac{\tau^{(2)2} \langle c^{(2)}(t) \rangle_{q(c^{(2)}(t))} \left(\begin{array}{l} +y_i(t) \\ -\langle c^{(1)}(t) \rangle_{q(c^{(1)}(t))} \langle f_i^{(1)}(t) \rangle_{q(f_i^{(1)}(t))} \end{array} \right) + \sigma^2 \left(\begin{array}{l} \sum_j M_{ij}^{(2)} \langle g_j^{(2)}(t) \rangle_{q(g_j^{(2)}(t))} + b_i^{(2)} \\ + \sum_j M_{ji}^{(o1)(2)} \langle f_j^{(2)}(t+1) \rangle_{q(f_j^{(2)}(t+1))} - \sum_j M_{ji}^{(o1)(2)} b_j^{(2)} \\ - \langle \mathbf{g}^{(2)\top}(t+1) \rangle_{q(\mathbf{r}^{(2)}(t) \dots q(\mathbf{r}^{(2)}(t-o))} \mathbf{M}^{(2)\top} \mathbf{M}^{(2)} \mathbf{Z}_i^{(ON)} \\ + \mathbf{Z}_i^{(ON)\top} \mathbf{M}^{(2)\top} \mathbf{M}^{(2)} \mathbf{Z}_i^{(ON)} f_i^{(2)}(t) \end{array} \right)}{\tau^{(2)2} \langle c^{(2)}(t) \rangle_{q(c^{(2)}(t))} + \sigma^2 \left(1 + [\mathbf{M}^{(1)\top} \mathbf{M}^{(1)}]_{ii} \right)} \quad (\text{A.2.37})$$

A.3.2 Optimal $q(\mathbf{C}(t))$

By using the *mean field theory* (4.2.6), equation (A.2.24) derives

$$q(c^{(1)}(t)) \alpha \exp \left(\begin{array}{l} \left\langle \log \left(P(\mathbf{Y}(t) | c^{(1)}(t), \mathbf{f}^{(1)}(t), \mathbf{f}^{(2)}(t)) \right) \right\rangle_{q(\mathbf{r}^{(1)}(t)) q(\mathbf{r}^{(2)}(t))} \\ + \left\langle \log \left(P(c^{(1)}(t) | c^{(1)}(t-1)) \right) \right\rangle_{q(c^{(1)}(t-1))} + \left\langle \log \left(P(c^{(1)}(t+1) | c^{(1)}(t)) \right) \right\rangle_{q(c^{(1)}(t+1))} \end{array} \right) \quad (\text{A.2.38})$$

and

$$q(c^{(2)}(t)) \alpha \exp \left(\begin{array}{l} \left\langle \log \left(P(\mathbf{Y}(t) | c^{(2)}(t), \mathbf{f}^{(1)}(t), \mathbf{f}^{(2)}(t)) \right) \right\rangle_{q(\mathbf{r}^{(1)}(t)) q(\mathbf{r}^{(2)}(t))} \\ + \left\langle \log \left(P(c^{(2)}(t) | c^{(2)}(t-1)) \right) \right\rangle_{q(c^{(2)}(t-1))} + \left\langle \log \left(P(c^{(2)}(t+1) | c^{(2)}(t)) \right) \right\rangle_{q(c^{(2)}(t+1))} \end{array} \right) \quad (\text{A.2.39})$$

Before substituting into each equation and solving for $q(c^{(s)}(t))$, we still need to clarify what are the last two elements inside each exponential; this is, we need to define what $P(c^{(s)}(t) | c^{(s)}(t-1))$ and $P(c^{(s)}(t+1) | c^{(s)}(t))$ are. By using our approximation from (4.1.6), we have stipulated that each conditional probability is given by γ when $c^{(s)}(t) = c^{(s)}(t-1)$ (cases $\{0,0\}$ and $\{1,1\}$), and $1-\gamma$ when $c^{(s)}(t) \neq c^{(s)}(t-1)$ (cases $\{0,1\}$ and $\{1,0\}$). By unifying these respective cases, we get

$$\begin{aligned} P(c^{(s)}(t) | c^{(s)}(t-1)) &= \left(\begin{array}{l} (1-c^{(s)}(t))(1-c^{(s)}(t-1))\gamma + (c^{(s)}(t))(c^{(s)}(t-1))\gamma \\ + (1-c^{(s)}(t))(c^{(s)}(t-1))(1-\gamma) + (c^{(s)}(t))(1-c^{(s)}(t-1))(1-\gamma) \end{array} \right) \\ &= \left(\begin{array}{l} \gamma - 2c^{(s)}(t)\gamma - 2c^{(s)}(t-1)\gamma + 4c^{(s)}(t)c^{(s)}(t-1)\gamma \\ -2c^{(s)}(t)c^{(s)}(t-1) + c^{(s)}(t) + c^{(s)}(t-1) \end{array} \right) \\ &= \left(\begin{array}{l} (1-2c^{(s)}(t) - 2c^{(s)}(t-1) + 4c^{(s)}(t)c^{(s)}(t-1))\gamma \\ -2c^{(s)}(t)c^{(s)}(t-1) + c^{(s)}(t) + c^{(s)}(t-1) \end{array} \right) \end{aligned} \quad (\text{A.2.40})$$

and similarly,

$$P(c^{(s)}(t+1) | c^{(s)}(t)) = \left(\begin{array}{l} (1-2c^{(s)}(t+1) - 2c^{(s)}(t) + 4c^{(s)}(t+1)c^{(s)}(t))\gamma \\ -2c^{(s)}(t+1)c^{(s)}(t) + c^{(s)}(t+1) + c^{(s)}(t) \end{array} \right) \quad (\text{A.2.41})$$

Therefore, we have this relation for $c^{(s)}(t)$ given a previous time step:

$$\begin{aligned} \langle P(c^{(s)}(t) | c^{(s)}(t-1)) \rangle_{q(c^{(s)}(t-1))} &= \left(\begin{array}{l} q(c^{(s)}(t-1) = 0) \log((1-2c^{(s)}(t))\gamma + c^{(s)}(t)) \\ + q(c^{(s)}(t-1) = 1) \log((2c^{(s)}(t)-1)\gamma - c^{(s)}(t) + 1) \end{array} \right) \\ &= \left(\begin{array}{l} \left(1 - \langle c^{(s)}(t-1) \rangle_{q(c^{(s)}(t-1))} \right) \log((1-2c^{(s)}(t))\gamma + c^{(s)}(t)) \\ + \left(\langle c^{(s)}(t-1) \rangle_{q(c^{(s)}(t-1))} \right) \log((2c^{(s)}(t)-1)\gamma - c^{(s)}(t) + 1) \end{array} \right) \end{aligned} \quad (\text{A.2.42})$$

and for $c^{(s)}(t+1)$ we get a similar equation, being

$$\begin{aligned} \langle P(c^{(s)}(t+1) | c^{(s)}(t)) \rangle_{q(c^{(s)}(t+1))} &= \left(\begin{array}{l} q(c^{(s)}(t+1) = 0) \log((1-2c^{(s)}(t))\gamma + c^{(s)}(t)) \\ + q(c^{(s)}(t+1) = 1) \log((2c^{(s)}(t)-1)\gamma - c^{(s)}(t) + 1) \end{array} \right) \\ &= \left(\begin{array}{l} \left(1 - \langle c^{(s)}(t+1) \rangle_{q(c^{(s)}(t+1))} \right) \log((1-2c^{(s)}(t))\gamma + c^{(s)}(t)) \\ + \left(\langle c^{(s)}(t+1) \rangle_{q(c^{(s)}(t+1))} \right) \log((2c^{(s)}(t)-1)\gamma - c^{(s)}(t) + 1) \end{array} \right) \end{aligned} \quad (\text{A.2.43})$$

With this already defined, we can start solving for $q(c^{(1)}(t))$. By introducing the probability density function (4.1.3) and equations (A.2.42) and (A.2.43) into (A.2.38), we get:

$$\begin{aligned}
q(c^{(1)}(t)) &\propto \exp \left\{ \left\langle \log \left(\exp \left(-\frac{1}{2\sigma^2} [\mathbf{Y}(t) - c^{(1)}(t)\mathbf{f}^{(1)}(t) - c^{(2)}(t)\mathbf{f}^{(2)}(t)]^2 \right) \right) \right\rangle_{q(\mathbf{f}^{(1)}(t))q(\mathbf{f}^{(2)}(t))q(c^{(2)}(t))} \right. \\
&\quad + \left(1 - \langle c^{(1)}(t-1) \rangle_{q(c^{(1)}(t-1))} \right) \log \left((1 - 2c^{(1)}(t))\gamma + c^{(1)}(t) \right) \\
&\quad + \left(\langle c^{(1)}(t-1) \rangle_{q(c^{(1)}(t-1))} \right) \log \left((2c^{(1)}(t) - 1)\gamma - c^{(1)}(t) + 1 \right) \\
&\quad + \left(1 - \langle c^{(1)}(t+1) \rangle_{q(c^{(1)}(t+1))} \right) \log \left((1 - 2c^{(1)}(t))\gamma + c^{(1)}(t) \right) \\
&\quad + \left(\langle c^{(1)}(t+1) \rangle_{q(c^{(1)}(t+1))} \right) \log \left((2c^{(1)}(t) - 1)\gamma - c^{(1)}(t) + 1 \right) \\
&= \exp \left\{ -\frac{1}{2\sigma^2} \left\langle [\mathbf{Y}(t) - c^{(1)}(t)\mathbf{f}^{(1)}(t) - c^{(2)}(t)\mathbf{f}^{(2)}(t)]^2 \right\rangle_{q(\mathbf{f}^{(1)}(t))q(\mathbf{f}^{(2)}(t))} \right. \\
&\quad + \left(1 - \langle c^{(1)}(t-1) \rangle_{q(c^{(1)}(t-1))} \right) \log \left((1 - 2c^{(1)}(t))\gamma + c^{(1)}(t) \right) \\
&\quad + \left(\langle c^{(1)}(t-1) \rangle_{q(c^{(1)}(t-1))} \right) \log \left((2c^{(1)}(t) - 1)\gamma - c^{(1)}(t) + 1 \right) \\
&\quad + \left(1 - \langle c^{(1)}(t+1) \rangle_{q(c^{(1)}(t+1))} \right) \log \left((1 - 2c^{(1)}(t))\gamma + c^{(1)}(t) \right) \\
&\quad + \left(\langle c^{(1)}(t+1) \rangle_{q(c^{(1)}(t+1))} \right) \log \left((2c^{(1)}(t) - 1)\gamma - c^{(1)}(t) + 1 \right) \\
&= \exp \left\{ -\frac{1}{2\sigma^2} \left\langle \begin{array}{l} \mathbf{Y}^T(t)\mathbf{Y}(t) - 2c^{(1)}(t)\mathbf{Y}^T(t)\mathbf{f}^{(1)}(t) \\ -2c^{(2)}(t)\mathbf{Y}^T(t)\mathbf{f}^{(2)}(t) \\ +2c^{(1)}(t)c^{(2)}(t)\mathbf{f}^{(1)T}(t)\mathbf{f}^{(2)}(t) \\ +c^{(1)}(t)^2\mathbf{f}^{(1)T}(t)\mathbf{f}^{(1)}(t) + c^{(2)}(t)^2\mathbf{f}^{(2)T}(t)\mathbf{f}^{(2)}(t) \end{array} \right\rangle_{q(\mathbf{f}^{(1)}(t))q(\mathbf{f}^{(2)}(t))q(c^{(2)}(t))} \right. \\
&\quad + \left(1 - \langle c^{(1)}(t-1) \rangle_{q(c^{(1)}(t-1))} \right) \log \left((1 - 2c^{(1)}(t))\gamma + c^{(1)}(t) \right) \\
&\quad + \left(\langle c^{(1)}(t-1) \rangle_{q(c^{(1)}(t-1))} \right) \log \left((2c^{(1)}(t) - 1)\gamma - c^{(1)}(t) + 1 \right) \\
&\quad + \left(1 - \langle c^{(1)}(t+1) \rangle_{q(c^{(1)}(t+1))} \right) \log \left((1 - 2c^{(1)}(t))\gamma + c^{(1)}(t) \right) \\
&\quad + \left(\langle c^{(1)}(t+1) \rangle_{q(c^{(1)}(t+1))} \right) \log \left((2c^{(1)}(t) - 1)\gamma - c^{(1)}(t) + 1 \right) \\
&\left. \right\} \tag{A.2.44}
\end{aligned}$$

and as in (A.2.28), $c^{(1)}(t)^2$ translates to $c^{(1)}(t)$ because of being Boolean.

After eliminating all those terms in (A.2.44) that are not dependent on $c^{(1)}(t)$, our equation turns to be:

$$q(c^{(1)}(t)) \propto \exp \left[\begin{aligned} & -\frac{1}{2\sigma^2} \left(\begin{aligned} & -2c^{(1)}(t) \mathbf{Y}^T(t) \langle \mathbf{f}^{(1)}(t) \rangle_{q(\mathbf{f}^{(1)}(t))} \\ & +2c^{(1)}(t) \langle c^{(2)}(t) \rangle_{q(c^{(2)}(t))} \langle \mathbf{f}^{(1)T}(t) \rangle_{q(\mathbf{f}^{(1)}(t))} \langle \mathbf{f}^{(2)}(t) \rangle_{q(\mathbf{f}^{(2)}(t))} \\ & +c^{(1)}(t) \text{var}(\mathbf{f}^{(1)}(t)) + \langle \mathbf{f}^{(1)T}(t) \rangle_{q(\mathbf{f}^{(1)}(t))} \langle \mathbf{f}^{(1)}(t) \rangle_{q(\mathbf{f}^{(1)}(t))} \end{aligned} \right) \\ & + \left(1 - \langle c^{(1)}(t-1) \rangle_{q(c^{(1)}(t-1))} \right) \log \left((1 - 2c^{(1)}(t)) \gamma + c^{(1)}(t) \right) \\ & + \left(\langle c^{(1)}(t-1) \rangle_{q(c^{(1)}(t-1))} \right) \log \left((2c^{(1)}(t) - 1) \gamma - c^{(1)}(t) + 1 \right) \\ & + \left(1 - \langle c^{(1)}(t+1) \rangle_{q(c^{(1)}(t+1))} \right) \log \left((1 - 2c^{(1)}(t)) \gamma + c^{(1)}(t) \right) \\ & + \left(\langle c^{(1)}(t+1) \rangle_{q(c^{(1)}(t+1))} \right) \log \left((2c^{(1)}(t) - 1) \gamma - c^{(1)}(t) + 1 \right) \end{aligned} \right] \quad (\text{A.2.45})$$

And analogous to $q(c^{(1)}(t))$, $q(c^{(2)}(t))$ results to be:

$$q(c^{(2)}(t)) \propto \exp \left[\begin{aligned} & -\frac{1}{2\sigma^2} \left(\begin{aligned} & -2c^{(2)}(t) \mathbf{Y}^T(t) \langle \mathbf{f}^{(2)}(t) \rangle_{q(\mathbf{f}^{(2)}(t))} \\ & +2c^{(2)}(t) \langle c^{(1)}(t) \rangle_{q(c^{(1)}(t))} \langle \mathbf{f}^{(2)T}(t) \rangle_{q(\mathbf{f}^{(2)}(t))} \langle \mathbf{f}^{(1)}(t) \rangle_{q(\mathbf{f}^{(1)}(t))} \\ & +c^{(2)}(t) \text{var}(\mathbf{f}^{(2)}(t)) + \langle \mathbf{f}^{(2)T}(t) \rangle_{q(\mathbf{f}^{(2)}(t))} \langle \mathbf{f}^{(2)}(t) \rangle_{q(\mathbf{f}^{(2)}(t))} \end{aligned} \right) \\ & + \left(1 - \langle c^{(2)}(t-1) \rangle_{q(c^{(2)}(t-1))} \right) \log \left((1 - 2c^{(2)}(t)) \gamma + c^{(2)}(t) \right) \\ & + \left(\langle c^{(2)}(t-1) \rangle_{q(c^{(2)}(t-1))} \right) \log \left((2c^{(2)}(t) - 1) \gamma - c^{(2)}(t) + 1 \right) \\ & + \left(1 - \langle c^{(2)}(t+1) \rangle_{q(c^{(2)}(t+1))} \right) \log \left((1 - 2c^{(2)}(t)) \gamma + c^{(2)}(t) \right) \\ & + \left(\langle c^{(2)}(t+1) \rangle_{q(c^{(2)}(t+1))} \right) \log \left((2c^{(2)}(t) - 1) \gamma - c^{(2)}(t) + 1 \right) \end{aligned} \right] \quad (\text{A.2.46})$$

References

- Attias, H. (1999). "Independent Factor Analysis." Neural Computation **11**(4): 803-851.
- Cardoso, J.-F. (1998). Blind Signal Separation: Statistical Principles. Proceedings of the IEEE.
- Chui, C. K. and G. Chen (1987). Kalman Filtering with Real-Time Applications, Springer-Verlag.
- Cover, T. M. and J. A. Thomas (1991). Elements of Information Theory. New York, John Wiley.
- Doyle, R. S. (1995). Kalman Filter. Handbuch der Sensortechnik. E. Obermeier and H. Trankler, Springer-Verlag GmbH & Co.
- Forney, D. G. (1973). The Viterbi algorithm. Proceedings of the IEEE.
- Gerhard, D. B. (2000). Ph.D. Depth Paper: Audio Signal Classification, Simon Fraser University, School of Computing Science.
- Ghahramani, Z. and G. E. Hinton (1998). "Variational learning for switching state-space models." Neural Computation **12**(4): 963-996.
- Harma, A., M. Juntunen, et al. (2001). Frequency-warped autoregressive modeling and filtering. Finland, Helsinki University of Technology.
- Howard, D. M. and J. Angus (1996). Acoustics and Psychoacoustics. Oxford, Focal Press.

Jaakkola, S. and M. I. Jordan (1999). Improving the mean field approximation via the use of mixture distributions. Learning in Graphical Models. M. I. Jordan. Cambridge, MIT Press.

Johnson, K. (1997). Acoustic & Auditory Phonetics. Oxford, UK, Blackwell Publishers Ltd.

Klapuri, A. (1998). Automatic Transcription of Music. Department of Information Technology, Tampere University of Technology: 82.

Klapuri, A. (2001). Automatic Transcription of Music, Tampere Institute of Technology & Nokia Research Center.

Leon-Garcia, A. (1993). Probability and Random Processes for Electrical Engineering, Prentice Hall.

Maybeck, P. (1979). Stochastic models, estimation and control. New York, Academic Press.

Murphy, K. (1998). Learning Switching Kalman Filter Models, Compaq Cambridge Research Lab.

Neumaier, A. and T. Schneider (2001). "Estimation of parameters and eigenmodes of multivariate autoregressive models." ACM Transactions on Mathematical Software (TOMS) **27**(1): 27-57.

Parisi, G. (1998). Statistical Field Theory. Redwood City, Addison-Wesley.

Pierce, A. D. (1994). Acoustics. An Introduction to its Physical Principles and Applications. New York, Acoustical Society of America.

Plumbley, M. D., S. A. Abdallah, et al. (in press). "Automatic music transcription and audio source separation." Cybernetics and Systems.

Roberts, S. and R. Everson (2001). Independent Component Analysis: Principles and Practice. Cambridge, Cambridge University Press.

Roederer, J. G. (1995). The Physics and Psychophysics of Music. New York, Springer-Verlag.

Rowe, D. B. (1999). The General Blind Source Separation Model and a Bayesian Approach. California, California Institute of Technology.

Taylor, C. A. (1965). The Physics of Musical Sounds. London, The English Universities Press Ltd.