THESE
présentée pour obtenir le grade de

Docteur de TELECOM ParisTech

Spécialité: Signal et Image

Antony Schutz

Quelques Contributions au Traitement de Signal Musical et à la Séparation Aveugle Mono-Microphone de Source Audio

Thèse soutenue le 2 Décembre 2010 devant le jury composé de :

Président	Gaël Richard	TELECOM ParisTech
Rapporteurs	Olivier Michel	INPG
	Frédéric Bimbot	CNRS-INRIA
Examinateurs	Nicholas Evans	EURECOM
	Mahdi Triki	Philips
Directeur de thèse	Dirk T.M. Slock	EURECOM

DISSERTATION
In Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy
from TELECOM ParisTech

Specialization: Signal and Image

Antony Schutz

Some Contributions to Music Signal Processing and to Mono-Microphone Blind Audio Source Separation

Thesis defensed on the 2^{nd} of December 2010 before a committee composed of:

President	Gaël Richard	TELECOM ParisTech
Reporters	Olivier Michel	INPG
	Frédéric Bimbot	CNRS-INRIA
Examiners	Nicholas Evans	EURECOM
	Mahdi Triki	Philips
Thesis supervisor	Dirk T.M. Slock	EURECOM

Résumé

Pour les êtres humains, le son n'a d'importance que pour son contenu. La voie est un langage parlé, la musique une intention artistique. Le processus physiologique est hautement développé, tout comme notre capacité à comprendre les processus sous-jacent. C'est un défi de faire exécuter la même tâche à un ordinateur: ses capacités n'égalent pas celles des humains lorsqu'il s'agit de comprendre le contenu d'un son composé de paroles et/ou d'instruments de musique. Dans la présente thèse nous avons envisagé deux des aspects reliés à cette problématique: la séparation aveugle de source ainsi que le traitement musical. Dans la première partie nos recherches portent sur la séparation aveugle de source en n'utilisant qu'un seul microphone. Le problème de séparation de source audio apparaît dès que plusieurs sources audio sont présentes au même moment, mélangées puis acquises par des capteurs, un unique microphone dans notre cas. Dans ce genre de situation il est naturel pour un être humain de séparer et de reconnaître plusieurs locuteurs. Ce problème, connu sous le nom de *Cocktail Problem* à reçu beaucoup d'attention mais est toujours ouvert. Dans cette partie nous présentons deux types d'algorithmes afin de résoudre ce problème. Comme nous ne travaillons qu'avec une seule observation nous ne pouvons pas utiliser d'indice lié à la spatialisation et nous sommes dans l'obligation de modéliser les sources. Nous utilisons un modèle paramétrique dans lequel une source est représentée comme étant la résultante de deux processus autorégressifs, de longueur de corrélation différentes, en cascade et excités par un bruit blanc Gaussien, la mixture est la somme de ces sources plus un bruit blanc Gaussien. Les signaux étant non stationnaires, le premier type d'algorithme proposé suit une méthodologie adaptative. Le deuxième type d'algorithme, lui, analyse l'observation par morceaux en considérant une stationnarité locale. Dans ce contexte la séparation se passe en deux étapes, tout d'abord les paramètres sont estimés dans le mélange puis ils sont utilisés afin d'effectuer la séparation. La deuxième partie traite du traitement musical et est composée de plusieurs annexes. La tâche analysée est liée au traitement automatique de la musique, qui a pour but de comprendre un contenu musical afin d'en générer la partition. Cependant la musique ne peut pas être réduite à une succession de notes et un bon transcripteur devrait être capable de détecter les effets d'interprétations et la qualité de jeu du musicien. Les outils conçus peuvent également être utilisés dans un but pédagogique pour aider un apprenti musicien à améliorer ses compétences. Dans cette partie nous avons porté attention à la détection de certains effets et défaut de jeu. Puis nous proposons une méthode afin de détecter la présence de l'octave lorsqu'elle est jouée avec sa fondamentale. Comme l'octave d'une note a une fréquence double de sa fondamentale, elles partagent les mêmes pics fréquentiels et la détection est ardue. Nous proposons un critère basé sur le rapport d'énergie des pics fréquentiels pairs et impairs pour faire la détection. Finalement cette partie se termine avec la description d'un transcripteur Audio-Visuel de Guitare dont le but est de fournir une tablature et non une partition.

Abstract

For humans, the sound is valuable mostly for its meaning. The voice is spoken language, music, artistic intent. Its physiological functioning is highly developed, as well as our understanding of the underlying process. It is a challenge to replicate this analysis using a computer: in many aspects, its capabilities do not match those of human beings when it comes to speech or instruments music recognition from the sound, to name a few. In this thesis, two problems are investigated: the source separation and the musical processing.

The first part investigates the source separation using only one Microphone. The problem of sources separation arises when several audio sources are present at the same moment, mixed together and acquired by some sensors (one in our case). In this kind of situation it is natural for a human to separate and to recognize several speakers. This problem, known as the *Cocktail Problem*, receives a lot of attention but is still open. In this part we present two algorithms for separating the speakers. Since we work with only one observation, no spatial informations can be used and a modelization of the sources is needed. We use a parametric model for constraining the solution: a mixture is modeled as a sum of Autoregressive sources with an additive white noise. The sources are themselfs modeled by a cascade of two AR model with differents correlation lenghts. The first algorithm is adaptive, for a non stationary signal it is natural to want to follow the variation of the signal over the time. The second algorithm works with consecutive frames of short duration. The procedure is splitted in two parts: first an estimation of the sources parameters is done on a frame, then a non iterative separation algorithm is used. Finally the estimated parameters are used for the initilization of the next analysed frame.

The second part deals with Musical Processing and is composed of several annexe. The task that we investigate is connected to the Automatic Music Transcription task, which is the process of understanding the content of a song in order to generate a music score. But, music cannot be reduced to a succession of notes, and an accurate transcriptor should be able to detect other performance characteristics such as interpretations effects. The tools built for automatic transcription can also be used in a pedagogic way, so that even a student can improve his performances with the help of a software. This means that the software should be able to detect some interpret's flaws. In this part, first of all we collect several samples of interpretation effects and performing defects. Then, we have built some tools for finding the presence (or not) of the considered effect. Another problem in music transcription is called the octave problem, it appears when a note and its octave are present together. As the octave has a frequency twice the note, they share the periodicity and the partials of the spectrum are perfectly overlapped. This makes the detection laborious. We propose an energetic criterion based on the estimation of the energy of the odd and even partials of the chord. Finally, the last chapter deals with the description of an audio-video simulator specialized for writing guitar tablature instead of partition.

Acknowledgements

I would like to thank particularly Professor Dirk T.M Slock, my thesis supervisor, for so many things that it is impossible to list them all. First of all he gave me the opportunity to do a PhD with him and in EURECOM, a very special place. He was also very supportive during some hard moments. I will never forget some dinners, around a good bottle of wine and luscious food. He allowed me to work, at my own pace, on challenging topics and to assist him in several projects, students' project and courses management. I also thank him for his guidance during the years I spent at EURECOM.

I am truly grateful to Professor Olivier Michel and to Frédéric Bimbot for accepting to be the reporters of my thesis and for all the feedback they gave me for improving my work. I also thank the president Professor Gaël Richard that I met several times during project meeting. I finally thank Nicholas Evans and Mahdi Triki (my elder brother) for being examiners.

I would like to thank David, Bassem, Lorenzo, Daniel, Turgut and specially Stephanie for their help for this manuscript. I also thank people with whom I had collaborations, as Henri, Eric, Sacha, Marco, Benoit, Siouar, Simon, Valentin, Nancy, Roland and Bertrand.

My memories about EURECOM are not limited to just scientific research, but also to side activities like organizing conferences and seminars, as well as to social initiatives as being the goal keeper of the EURECOM football team (thanks Seb).

I had the chance to meet great colleagues and collaborators, but mostly, great friends. I especially think about Raul, Nadia, Fadi, Farrukh, Hicham, Jerome and Saad, who welcome me at EURECOM, as well as those I met later, as Bassem, Marco, Virginia, Randa, Daniel, Erhan, Lorenzo, Zoe, Zuleita, Umer, Turgut, Kostaks and Ikbal. And of course everyone at EURECOM.

I also appreciated to work with EURECOM students for semester projects. For this reason, I give my thanks to Lucia, Valeria, Guillaume and Pierre-Etienne. I also thank Siouar, whom I followed during her Master Project, and who became my little sister. Thanks to all the other colleagues, secretaries and to the IT service.

I will always cherish wonderful memories of these past years at EURECOM.

Last but not the least, I am grateful to my family and my friends, for their sincere love and unconditional support, before, during, and after my PhD studies.

Contents

Résumé i
Abstract i
Acknowledgements v
Contents vii
List of Figures xv
Acronyms xix
Notations xxi
Overview of the thesis and contributions xxiii
0.1 Thesis Overview xxiii
0.2 Contributions xxv
    0.2.1 Blind Audio Source Separation xxv
    0.2.2 Annexe: Musical Processing xxvii
Résumé des travaux de thèse xxix
0.3 Introduction à la séparation aveugle de source xxix
    0.3.1 Quelques solutions existante xxx
    0.3.2 Positionnement du travail exposé xxxi
    0.3.3 La séparation aveugle de source Mono-Microphone xxxi
0.4 Modèle utilisé xxxiv
    0.4.1 La parole xxxiv
    0.4.2 Modélisation à long terme xxxiv
    0.4.3 Modélisation à court terme xxxv
    0.4.4 Modèle de source xxxvi
    0.4.5 Modèle de mélange xxxvi
0.5 Traitement adaptatif xxxviii
    0.5.1 Algorithme de type EM-Kalman xxxviii
    0.5.2 Algorithme EM-Kalman adaptatif xxxviii
    0.5.3 Séparation de source Mono-Microphone avec un EM-Kalman xxxix
    0.5.4 Estimation des paramètres et discussion sur les états partiels xl
    0.5.5 Résultat de simulations sur des signaux synthetiques xli
0.6 Traitement ”par fenêtre” xlii
    0.6.1 Distance d'Itakura Saito xlii
    0.6.2 Interprétation Naïve de la distance IS xlii
        0.6.2.1 Estimation de paramètres par la méthode Naïve xliii
    0.6.3 Minimisation de la distance IS xliii
    0.6.4 Algorithme de séparation de source xliv
        0.6.4.1 Modèle conjoint xliv
0.7 Simulations xlvi
    0.7.1 Signaux de courte durée xlvii

- 0.7.2 Signaux de longue durée: Parole . . . xlvii
- 0.7.3 Signaux de longue durée: Musique . . . xlix
- 0.7.4 Discussion des performances . . . l
- 0.7.5 Extraction du bruit de fond . . . li
  - 0.7.5.1 paramètres utilisés . . . lii
- 0.8 Conclusions . . . liii
- 0.9 Perspectives . . . liv

I Mono-Microphone Blind Audio Source Separation 1

1 Introduction 3

- 1.1 Blind Source Separation . . . 4
  - 1.1.1 Introduction . . . 4
  - 1.1.2 Determination . . . 4
  - 1.1.3 Effect of the propagation . . . 5
- 1.2 Application . . . 5
  - 1.2.1 Audio Processing . . . 5
  - 1.2.2 Biomedicine . . . 6
  - 1.2.3 Diarization . . . 6
  - 1.2.4 Security . . . 6
  - 1.2.5 Telecommunications . . . 6
- 1.3 Some existing solutions . . . 7
  - 1.3.1 A brief Chronology . . . 7
  - 1.3.2 Determined and Over determined BSS . . . 7
  - 1.3.3 Under determined BSS (UBSS) . . . 8
  - 1.3.4 Stereophonic BSS . . . 8
- 1.4 Positioning of this thesis . . . 9
- 1.5 Mono-Microphone Blind Audio Source Separation . . . 9
  - 1.5.1 Problem Formulation . . . 9
  - 1.5.2 Factorial vector quantization (VQ) . . . 10
  - 1.5.3 Gaussian Mixture Models (GMM) . . . 10
  - 1.5.4 Hidden Markov Model (HMM) . . . 10
  - 1.5.5 Non-negative Matrix Factorisation (NMF) . . . 11
  - 1.5.6 Structured/Parametrized Model Based Approach . . . 11
    - 1.5.6.1 Sinusoids Plus Noise . . . 11
    - 1.5.6.2 AR/ARMA-based source separation . . . 13
- 1.6 Model Considerered in this work . . . 14
- 1.7 Relationship with another approach . . . 14
- 1.8 Proposed Approach . . . 15
- 1.9 Summary of chapter 1 . . . 16

2 Model of Speech Production 17

- 2.1 Introduction . . . 18
- 2.2 Speech Model Production . . . 18
  - 2.2.1 How to describe Human Voice . . . 18
  - 2.2.2 A speech signal . . . 18
  - 2.2.3 Modeling the Periodicity . . . 19

2.2.3.1 Comb Filter . . . 19
2.2.3.2 Formulation of the Long Term Model . . . 20
2.2.3.3 Traditional Long Term Parameters estimation . . . 21
2.2.4 Modeling the Spectral Shape . . . 22
2.2.4.1 Short Term Autoregressive Model . . . 22
2.3 Mixing Short and Long Term Autoregressive Model: A source . . . 23
2.3.1 Modeling and parameters estimation . . . 23
2.3.2 Spectra Comparison . . . 25
2.4 Multiple Source Plus Noise Model . . . 26
2.4.1 Signal Model . . . 26
2.4.2 Spectral Notation . . . 27
2.5 Summary and discussion . . . 29

3 Adaptive EM-Kalman Filter 31
3.1 Kalman Filter (KF) . . . 32
3.2 Expectation Maximization algorithm (EM) . . . 33
3.2.1 Kalman Smoothing With the EM Algorithm . . . 33
3.2.1.1 Fixed Interval Smoothing . . . 33
3.2.1.2 Fixed lag Smoothing . . . 34
3.3 Adaptive EM-KF with Fixed-Lag Smoothing . . . 34
3.4 Adaptive EM-KF for Mono-Microphone BASS . . . 35
3.4.1 Introduction . . . 35
3.5 State Space Model Formulation . . . 35
3.6 Algorithm . . . 38
3.6.1 Partial states discussion . . . 38
3.6.2 Parameters estimation . . . 39
3.7 Other Approaches using Kalman Filter . . . 41
3.8 Simulations - Synthetic Signals . . . 42
3.8.1 Used parameters to generate synthetic signals . . . 42
3.8.2 Comparison of Models . . . 42
3.8.2.1 Filtering results comparison . . . 43
3.8.2.2 Estimation results comparison . . . 44
3.8.3 Amplitudes tracking . . . 46
3.8.4 Periods Variations . . . 47
3.9 Summary of Chapter 3 and discussion . . . 49

4 Frame Based Source Separation 51
4.1 Frame Based Algorithm . . . 52
4.1.1 Introduction . . . 52
4.1.2 Windowing . . . 52
4.1.2.1 Perfect Reconstruction . . . 53
4.1.3 Related work . . . 53
4.2 Parameters and Sources estimation . . . 54
4.2.1 EM Like Algorithm . . . 54
4.2.2 VB-EM Like Algorithm . . . 54
4.2.3 Proposed Approach . . . 55
4.3 Model . . . 56
4.3.1 Spectral Model . . . 57

4.3.2 Set of Parameters . . . 58
4.4 Parameters Estimation Using The Itakura-Saito (IS) Distance . . . 58
4.4.1 Definition of the IS Distance . . . 58
4.4.2 IS Distance for the model . . . 59
4.5 Naive Interpretation of the IS distance . . . 59
4.5.1 Algorithm . . . 59
4.5.1.1 Short term parameters estimation . . . 60
4.5.1.2 Long term parameters estimation . . . 60
4.5.1.3 Noise variance . . . 61
4.6 Minimization of the IS distance . . . 61
4.6.1 Weighted Spectrum Matching . . . 61
4.6.2 Gaussian Maximum Likelihood . . . 61
4.6.3 Short-term AR Parameters Estimation . . . 61
4.6.4 Source Power Estimation . . . 62
4.6.5 Overall iterative process . . . 63
4.6.5.1 A note about the Long Term Model . . . 63
4.7 Parameters Estimation, Synthetic Spectrum . . . 63
4.8 Source Separation Algorithm . . . 64
4.8.1 Joint Source Representation . . . 64
4.8.2 Frequency Domain Window Design . . . 65
4.8.3 Joint Model . . . 67
4.8.4 Estimating the Sources . . . 67
4.8.4.1 Summary of the algorithm . . . 69
4.9 Summary and discussion . . . 69

5 Simulations **71**
5.1 Introduction and database details . . . 72
5.1.1 Used Algorithms and initialization details . . . 72
5.2 Short Duration real signals . . . 73
5.2.1 Short Duration real signals, Comparison of Models and orders . . . 76
5.3 Long Duration real signals . . . 78
5.3.1 Filtering Results . . . 79
5.3.2 Estimation Results with **Alt-EMK** and ”known” periods . . . 81
5.3.2.1 Speech Signals . . . 81
5.3.2.2 Discussion about the simulations and obtained results . . . 83
5.3.2.3 Instrumental Signals . . . 85
5.4 Performances discussion . . . 86
5.5 Vuvuzela remover . . . 88
5.5.1 Procedure and details . . . 88
5.5.2 Results . . . 88
5.6 Summary . . . 90

6 Conclusion **91**
6.1 Summary and conclusions . . . 91
6.2 Potential Improvements . . . 93

II Annexe
Music Signal Processing 95

A Instruments and Interpretation Effects: Description 97
- A.1 Introduction 97
- A.2 Violin 98
  - A.2.1 Pizzicato 98
- A.3 Electric Bass Guitar 98
  - A.3.1 *Slap* 99
- A.4 Piano 99
  - A.4.1 *Forte* Pedal 99
  - A.4.2 *Practice* Pedal 100
- A.5 Guitar 100
  - A.5.1 *Palm* mute 102
  - A.5.2 *Slide*, *Bend* and *Hammer* 102

B Tools 103
- B.1 Sinusoidal modeling for *SNR* Estimation 103
  - B.1.1 Model 103
  - B.1.2 Motivation, Analysis and synthesis method 104
  - B.1.3 Estimation - Interpolation - Amplitudes estimation 106
  - B.1.4 Synthesis, Noise extraction and SNR estimation 109
  - B.1.5 Discussion on the method 109
- B.2 Onset Detection 109
  - B.2.1 Example of onsets detection 111
- B.3 Pitch estimation 112
- B.4 Fundamental to Harmonics Energy Ratio (FHER) 113
- B.5 Slow variation fondamental frequency tracking 113
  - B.5.1 Example 114

C Interpretation Effects and Playing Defects: Detection 115
- C.1 Violin Playing defects detection 115
  - C.1.1 Orientation defects 115
- C.2 Instruments Interpretation Effects Detection 117
  - C.2.1 *Slap* Detection 117
  - C.2.2 *Forte* Pedal Detection 119
    - C.2.2.1 Database 119
    - C.2.2.2 Feature extraction 120
    - C.2.2.3 Pre-processing 120
    - C.2.2.4 Harmonics amplitudes tracking 121
    - C.2.2.5 Noise estimation 121
  - C.2.3 *Palm* mute, *Pizzicato* and *Pratice* Pedal detection 124
    - C.2.3.1 Description 124
    - C.2.3.2 *Palm* mute detection result 124
    - C.2.3.3 Violin *Pizzicato* 124
    - C.2.3.4 *Pratice* Pedal 124
  - C.2.4 Guitar: *Bend, slide* and hammer 127
    - C.2.4.1 Fundamental Frequency and Amplitude tracking 127

D Periodic Signal Modeling 131
- D.1 Problem introduction . . . 131
  - D.1.1 Illustration . . . 131
- D.2 Periodic signal Modeling . . . 133
  - D.2.1 Definition of the periodic signature . . . 135
  - D.2.2 As a pitch detector . . . 136
  - D.2.3 Simulation for the pitch estimation . . . 136
    - D.2.3.1 Diagram for the pitch determination . . . 137
    - D.2.3.2 Illustration for the octave determination . . . 137
  - D.2.4 Application to a true signal . . . 138
  - D.2.5 Application to the Octave Problem . . . 140
  - D.2.6 Note plus its Octave . . . 140
  - D.2.7 Note plus its first two octaves . . . 142
  - D.2.8 Note plus its second octave . . . 142
  - D.2.9 Parameters used . . . 143
- D.3 Conclusion and Future work . . . 143

E Audio Visual Guitar Transcription 145
- E.1 Introduction . . . 145
- E.2 Guitar Transcription . . . 146
  - E.2.1 Automatic Fretboard Detection . . . 146
  - E.2.2 Fretboard Tracking . . . 147
  - E.2.3 Hand Detection . . . 147
  - E.2.4 Audio Visual Information Fusion . . . 148
- E.3 Prototype . . . 149
- E.4 Future Work . . . 150
- E.5 Conclusions and Future work . . . 151

F Miscellaneous 153
- F.1 Short Term AR Coefficients Generation Using Levinson Algorithm . . . 153
- F.2 Iterative Algorithm for estimating Short plus Long Term AR Model . . . 154
  - F.2.1 Short Term AR Coefficients . . . 154
  - F.2.2 Long Term AR Coefficient . . . 154
  - F.2.3 Iterative estimation . . . 155
- F.3 Short Term Fourier Transform (STFT) . . . 155
- F.4 Evaluation Criteria . . . 155
  - F.4.1 Decomposition . . . 155
  - F.4.2 Global Criteria . . . 156
    - F.4.2.1 Source to Distortion Ratio . . . 156
    - F.4.2.2 Source to Interferences Ratio . . . 156
    - F.4.2.3 the Sources to Artifacts Ratio . . . 156
  - F.4.3 Local Criteria . . . 156
- F.5 Windows Properties . . . 157
  - F.5.1 Notations . . . 157
  - F.5.2 Typical Windows . . . 157
  - F.5.3 Windows and Spectra . . . 158
- F.6 Circulant Matrix . . . 159
  - F.6.1 Circulant Matrix construction . . . 159

- F.6.2 Circulant Matrix Properties . . . 159
  - F.6.2.1 Product . . . 159
  - F.6.2.2 Inverse . . . 160
- F.7 Frame based algorithms Initialization . . . 160
  - F.7.1 Per Source Weighted Itakura-Saito Distance Minimization . . . 160
  - F.7.2 Pitch Estimation . . . 161
  - F.7.3 AR coefficients estimation . . . 161
  - F.7.4 Multipitch Simulation Example . . . 162
- F.8 **Alt-EMK** Versus **Joint-EMK** . . . 163
- F.9 Spectral Roll Off . . . 163
- F.10 Parameters Used for Sources Separation Simulations . . . 164
  - F.10.1 Simulations of Chapter 2 . . . 164
    - F.10.1.1 Evaluation criteria Versus SNR . . . 164
  - F.10.2 Simulations of Chapter 3 . . . 164
    - F.10.2.1 Weighted sources . . . 164
    - F.10.2.2 Fondamentals Frequencies variations . . . 164
  - F.10.3 Simulations of Chapter 4 . . . 165
    - F.10.3.1 LDU decomposition . . . 165
  - F.10.4 Simulations of Chapter 5 . . . 165
- F.11 Noise Variance esimation . . . 165

List of Figures

1 Détermination xxx
2 Une décomposition FMN typique d'un signal audio. xxxii
3 Filtre en peigne en mode "Feedback". xxxiv
4 Réponse en magnitude pour différentes valeurs positive de b xxxiv
5 Un signal "long terme" et sa séquence de corrélation. xxxvi
6 Spectre d'un modèle AR court plus long terme et du mélange associé. . . . xxxvii
7 Signaux de courte durée, l'observation et les sources. xlvi
8 Transformée de Fourier à court terme, résultat du **Alt-EMK**. xlviii
9 L'observation, les sources et leurs estimés obtenues avec **Alt-EMK**. xlix
10 Exemple de suppression du Vuvuzela. lii

1.1 Determination of the problem 4
1.2 Typical NMF decomposition for audio signal. The observation is typically a Time Frequency matrix, such as the magnitude Short Time Fourier Transform. The decomposition finds a set of time-varying sources with constant spectrum. 8

2.1 A speech signal and its STFT (log magnitude). 19
2.2 Feedback comb filter structure. 20
2.3 Magnitude response for various positive values of b 20
2.4 Long Term signal and its Auto correlation sequence. 21
2.5 Example of a periodic signal. Sampling frequency $F_s = 8000Hz$, $b = 0.9$ and $\tau = 0.005s$. Temporal signal and its periodogram 22
2.6 Example of AR signals of order $p = 5$. 24
2.7 Example of a Short plus Long term model of order $p = 5$. Sampling frequency $F_s = 8000Hz$, $b = 0.9$ and $\tau = 0.005s$. 25
2.8 Synthesis Comparaison between iterative and non iterative methods for a real speech 26
2.9 Short plus Long Term modeling of two sources and associated mixture. . . . 28
2.10 Spectra of Short plus Long Term modeling of two sources and associated mixture. 28

3.1 State Space Model and State Vector for 2 sources. 38
3.2 Models comparison. SDR, SIR, SAR and MSE in the filtering case. 44
3.3 Models comparison. SDR, SIR, SAR in the Estimation case. 45
3.4 With parameters estimation results, weighted sources. 46
3.5 Zoom on the attenuated part. 47
3.6 Comparison example, STFT for **Joint-EMK** and **Alt-EMK**, a fixe and a varying fondamental frequencies. 48

3.7 Comparison example, STFT for **Joint-EMK** and **Alt-EMK**, the two fondamental frequencies vary. 49

4.1 Perfect reconstruction windowing, Hann and Triangulare window with an overlap of 50% and 75%. 53
4.2 Comparison example Naive Versus True Minimization of IS Distance. 64
4.3 Example of True Minimization of IS Distance. 65
4.4 Perfect reconstruction windowing. 66
4.5 LDU Decomposition. 68

5.1 The observations and the sources, Short duration speech signals. 73
5.2 The sources and Estimated sources. Kalman Filter, **Joint-EMK** and **Alt-EMK**. 74
5.3 The sources and Estimated sources. Filtering , **Naive-IS** and **Tmin-IS**. . . 75
5.4 Models comparison. SDR, SIR, SAR and MSE in the filtering case. 76
5.5 Models comparison. SDR, SIR, SAR and MSE in the Estimation case. . . . 77
5.6 Models comparison. SDR, SIR, SAR and MSE in the Estimation case for different short term order. 77
5.7 The observation and the sources, Long duration speech signals. 78
5.8 STFT of sources, and estimates extracted with a Kalman filter. 79
5.9 STFT of sources, and estimates extracted with a Wiener filter. 80
5.10 Estimated Fondamental Frequency (0-500Hz) and related strenght 81
5.11 The observation, sources and estimates extracted with **Alt-EMK**. Long duration speech signals. 82
5.12 STFT of the observation, sources, and estimates extracted with **Alt-EMK**. 84
5.13 The observation, sources and estimates extracted with **Alt-EMK**. Long duration Music signals. 85
5.14 Vuvuzela Remover example. 89

A.1 *Music Transcription* 98
A.2 *A violin* 99
A.3 *Pizzicato* 100
A.4 *Electric Bass (5 strings)* 100
A.5 *Slap example* 101
A.6 *Soft (left), pratice (middle) and sustain pedals (right)* 101
A.7 *Electro-Classical Guitar* 102
A.8 *PalmMute* 102

B.1 *Example of the Overlap and Add Method* 105
B.2 *Example for the reconstruction error* 105
B.3 *Standard deviation of the frequency error for the phase vocoder method and parabolic interpolation for a cisoid* 107
B.4 *Estimation error of the parabolic interpolation, for one and two cisoids* . . . 107
B.5 *Estimation error of the parabolic interpolation and for the phase vocoder with the cleaning of the other peaks on the periodogramme* 107
B.6 *Example of a parabolic interpolation, The circles correspond to samples of the spectrum* 108
B.7 *General Diagram for the Harmonic+Noise Decomposition* 110
B.8 *General Diagram for the Onsets Detection* 111

B.9 *Example of detection function* 111
B.10 *Exemple of Modified Pisarenko Method on a Violin Note* 114

C.1 *SNR Estimation for a Violin piece played by a student* 116
C.2 *Orientation of the Bow* 116
C.3 *Estimation of the SNR for bass sequence of two notes, the first is play with the finger and the second is play by slap.* 117
C.4 *Bass played with Finger and SNR Estimation.* 118
C.5 *Slap Bass and SNR Estimation.* 118
C.6 Examples of waveforms of *staccato*, *legato*, *staccato+ped.* and *legato+ped.*, note D_2 120
C.7 Evolution of the amplitudes of the first three harmonics 121
C.8 Harmonic+Noise Decomposition for *staccato*, *legato*, *staccato+ped.* and *legato+ped.* 122
C.9 Top : Autoregressive modeling of the Noise for 170 note recordings. A white line on top indicates notes with Pedal. Bottom : power of the AR model : For notes with Pedal (solid line) and notes without Pedal (dashed line). The dots indicate the notes that are estimated to be notes with Pedal. 123
C.10 *Short Time Fourier Transform, The song is composed of Non-Pizzicato and Pizzicato notes.* 125
C.11 *Signal and onsets (a), onsets detection function and onsets (b), result of the detection (c).* 125
C.12 *Results of the analysis for the detection of the Pizzicato for a Violin* 126
C.13 *STFT of notes played with and without the practice pedal* 126
C.14 *Instantaneous frequency and amplitude tracking of a Bend, guitar note.* . . 127
C.15 *Instantaneous frequency and amplitude tracking of a Slide, guitar note.* . . . 128
C.16 *Instantaneous frequency and amplitude tracking of a Hammer, guitar note.* . 128
C.17 *sound containing interpretation effects, onset detection function and instantaneous frequency and amplitude tracking* 129

D.1 *Illustration of the Octave Problem, Temporal point of view* 132
D.2 *Illustration of the Octave Problem, Spectral point of view* 132
D.3 *Difference between note and note+octave* 133
D.4 Even and odd parts of the spectrum. 135
D.5 Pitch detection and Octave Selection for a synthetic signal. 136
D.6 *Diagram of the pitch estimation algorithm* 137
D.7 *Octave* -2 138
D.8 *Octave* -1 138
D.9 *Good Octave* 139
D.10 Pitch detection and Octave Selection for guitar. 139
D.11 Octave problem, a note with its octave. 140
D.12 Octave problem in the prediction error, a note with its octave with the temporal method (top) and the spectral method (bottom). 141
D.13 Octave problem in the prediction error, a note with its first and second octaves. Temporal method (top) and Spectral method (bottom). 142
D.14 Octave problem in the prediction error, a note with its second octave. Temporal method (top) and Spectral method (bottom). 143

E.1 Notes on a guitar fretboard 146

E.2 Interface of the Automatic Transcription System 148
E.3 Example of video errors 149
E.4 Transcription Errors 150

F.1 Windows and Spectra. 158
F.2 MultiPitch Algorithm Versus Spectral Sum. 162
F.3 MultiPitch Bi-Dimensional research. 162
F.4 Comparison **Alt-EMK** and **Joint-EMK** with non-stationary sources. . . . 163
F.5 The weight for each sources. 164
F.6 Estimated Noise Variance for all algorithms. 165

Acronyms

Here are the main acronyms used in this document. The meaning of an acronym is usually indicated once, when it first occurs in the text. The English acronyms are also used for the French summary.

BSS	Blind Source Separation
BASS	Blind Audio Source Separation
UBSS	Under-determined Blind Source Separation
LPC	Linear Predictive Coding
PSD	Power Spectral Densities
NMF	Non-negatif Matrix Factorization
CELP	Code Excited Linear Prediction
AR	Auto Regressive
VQ	Vector Quantization
ST	Short Term
LT	Long Term
SNR	Signal to Noise Ration
SyNR	Synthesis to Estimated Noise Ratio
MSE	Mean Square Error
MMSE	Minimum Mean Square Error
LMMSE	Linear Minimum Mean Square Error
dB	decibel
ML	Maximum Likelihood
MAP	Maximum *A Posteriori*
ADSR	Attack Decay Sustain Release
KF	Kalman Filter
EKF	Extended Kalman Filter
SSM	State Space Model
EM	Expectation Maximization
RTS	Rauch Tung Striebel
SAGE	Space-Alternating Generalized Expectation-maximization
VB	Variational Bayesian
SDR	Source to Distorsion Ratio
SIR	Source to Interference Ratio
SAR	Source to Artefact Ratio
DFT	Discret Fourier Transform
IDFT	Inverse Discret Fourier Transform
FFT	Fast Fourier Transform

LDU	Lower triangular, Diagonal and Upper triangular matrix Factorization
STFT	Short Time Fourier Transform
MIDI	Musical Instrument Digital Interface
PR	Perfect Reconstruction
IS	Itakura-Saito
YW	Yule-Walker
SVD	Singular Value Decomposition
PV	Phase Vocoder
PI	Parabolic Interpolation
PEA	Pitch estimation algorithms
HPS	Harmonic Product Spectrum
SHS	Sub-Harmonic Summation
SHR	Subharmonic to Harmonic Ratio
FHER	Fundamental to Harmonics Energy Ratio
ESPRIT	Estimation of Signal Parameters via Rotational Invariance Techniques
SVM	Support Vector Machine
ICA	Independent Component Analysis
DUET	Degenerate Unmixing Estimation Technique
CASA	Computational Auditory Scene Analysis
GMM	Gaussian Mixture Model
GSMM	Gaussian Scaled Mixture Model
MIR	Music Information Retrieval
DOA	Direction-Of-Arrival
MDCT	Modified Discret Cosinus Transform

Notations

E	Expectation operator
$\|x\|$	Absolute value of x
$\lfloor x \rfloor$	Floor operation, rounds the elements of x to the nearest integers towards minus infinity
$\lceil x \rceil$	Ceil operation, rounds the elements of x to the nearest integers towards infinity
$\mathbf{h}$	vector
h	scalar
$\mathbf{H}$	matrix
$\mathbf{H}^*$	Conjugate operation
$\mathbf{H}^H$	Hermitian operation
$\mathbf{H}^\mathrm{T}$	Transpose operation
$\mathbf{O}$	Matrix containing only zeros
$\mathbf{I}$	Identity Matrix
$\mathbf{T}$	Banded Toeplitz matrix
$\mathbf{1}$	vector of one
$diag(\mathbf{H})$	Takes the diagonal elements of a matrix
$diag(\mathbf{h})$	Create a diagonal matrix with a vector
q	advance operator
$blockdiag$	block diagonal Matrix
$\oplus$	Kronecker Sum
$\otimes$	Kronecker Product
$\odot$	Hadamard Product
$*$	Convolution

Overview of the thesis and contributions

0.1 Thesis Overview

This thesis is the results of more than four years passed at **EURECOM**. The original subject was: Musical Processing, that is to apply signal processing techniques on musical signal and is reported in the annexe of the second part of the thesis. This work was supported by three projects namely the french ANR project: "**SIEPIA**", The European Network of Excellence: "**KSpace**" and in parrallel the project "**TAM-TAM**" founded by the *Institut Télécom Crédits Incitatifs GET2007*.

The project "**SIEPIA**" (*Système Interactif d'Education et de Pratique Instrumental Acoustique* was done in collaboration with the Sart-Up *SigTone* [1]. The goal was to provide some tools, included in a real time simulator, for the detection of musical performances in a pedagogic context. Typically, the applications are destined to beginners musicians who want to improve their skill. The tools developped and integrated in the simulator were able to detect some defects of the bow playing for violin player, the slap playing effect for the bass guitar and the pizzicato playing for the guitar. Other tools were developped by *SigTone* and are not presented in this thesis, they include real time chords recognition, real time automatic guitare transcription and comparison (with a predefined piece), tempo analysis etc. This simulator was presented during the "*Grand Colloque STIC 2006*" (with a high background noise) in *Lyon* (France), and was ranked in the first eight best projects of the competition (over more than 140 projects). Unfortunately the collaboration has stopped with the project.

Our contributions in the project "**KSpace**" (*Knowledge Space of semantic inference for automatic annotation and retrieval of multimedia content*) was more fuzzy. The aim of the research was to narrow the gap between low-level content descriptions and subjectivity of semantics in high-level human interpretations of audiovisual media. For this project we have continued to provide low level tools related to the detection of musical interpretation effects for the guitar and the piano. We also propose a solution for solving the octave probeme which appears when a note and its octave are played together. Then during a "**KSpace**" PhD workshop, in the last year of the project, an idea of collaboration with the **EURECOM** *Multimedia Department* born. The work of *Marco Paleari* was focused on emotional analysis of facial gesture [2] and gave us, at the begining, the idea of analysing facial gesture of musician (as a violinist performing a solo sequence) in order to evaluate the degree of emotion. As this analysis was highly specialized and, also, as a very specialized database was needed, we gave up the idea. However, we change it. The new concept was to make an audio-visual analysis of a guitar player for solving some ambiguities that are difficult to solve with only the Audio part. In parrallel we discover the work of *Olivier Gillet* [3] which a part focused on the audio-visual analysis of drum song. This work was an intra department collaboration in **EURECOM** and allowed us to supervize students projects leading to a prototype of the simulator.

The "**TAM-TAM**" Project (*Transcription automatique de la musique : Traitements avancés et mise en oeuvre*) was done with the *TSI* departement of *Télécom Paris Tech* and in parrallel with the "**KSpace**" Project. The goal of this project was the Automatic Transcription of piano songs. Our contributions was to provide an analysis of the MIDI protocol, not presented in this thesis, for the trancription (advantage and drawback). Also we propose to work on the detection of the Sustain Pedal of the piano. We propose a method which was able to detects the presence of the pedal on our database and for single note. However, normally the pedal is used for playing a succession of notes which are normally infeasible. So the single note case is not interesting but if the conditions allow the detection of the pedal, the analysis was able to find something that the ear is not abble to detect.

When these projects were finished and due to the lack of consistent database we stopped to work on Musical Processing. However all this works were considered as original at least in terms of issues addressed.

The first part of the thesis is related to one another project collaboration between the *TSI* departement of *Télécom Paris Tech*, the *Multimedia* and *Mobile Department* of **EURECOM**: The Institut Telecom "*Futur et Ruptures*" project "**SELIA**" (*Suivi et séparation de Locuteurs pour l'Indexation Audio.* The aim of the project was to focuse on audio document indexing by investigating new approaches to source separation, dereverberation and speaker diarization. Our task in this project was to propose new Mono-Microphone Source Separation Algorithms. This project give us the opportunitie to supervize a Master project in the person of Siouar Bensaid who is now a PhD Student at **EURECOM**. Together we work on an Adaptive EM-Kalman Algorithm and she continue to work on it while I begin to analyse Frame Based Algorithm.

0.2 Contributions

These contributions are divided into two parts:

- Mono-Microphone Blind Audio Source Separation
- Musical Processing.

A brief overview of the general framework of this thesis, and of each part is given in this section.

0.2.1 Blind Audio Source Separation

The first part deals with Mono-Microphone Blind Audio Source Separation (BASS). This is considered as a difficult problem because only one observation is available. In fact it is the most under-determined case, but also the more realistic one for many applications. A vast number of fast and effective methods exist for solving the determined problem [4, 5]. In the under-determined case, the number of sources n is greater than the number of mixtures m and the problem is degenerate [6] because traditional matrix inversion (demixing) cannot be applied. In this case, the separation of the under-determined mixtures requires prior [7, 8] information of the sources to allow for their reconstruction. Estimating the mixing system is not sufficient for reconstructing the sources, since for $m < n$, the mixing matrix is not invertible. Here we focus on the presumed quasi periodic nature of the sources.

Chapter 1

The first Chapter is an introductive chapter. The goal of this chapter is to motivate the BSS problem and to recall some existing solution. We recall some general definitions of the Source Separation Problem as well as a summary of a majority of the possible cases. More specially we focus on the Mono-Microphone literrature. We motivate the model and the approach that we will use.

Chapter 2

Chapter 2 is dedicated to the description of the source model. Here we model speech as a combination of a sound source: the vocal cords and a linear acoustic filter [9]. We assume that a source can be represented by the combination of auto-regressive (AR) models; the periodicity is represented by a long term AR model followed by a short term AR for its timbre. This model is largely used in CELP codding but it didn't receive a lot of interest in the BASS community, despite of its simplicity. In this chapter we explain the parametric model of a source and of the mixture. For each AR models of a source the correlation lenghts are very different and are related to different parameters.

Chapter 3

The first algorithm that we have designed is an adaptive Expectation Maximization (EM) Kalman Filtering (KF) algorithm. The KF corresponds to optimal Bayesian estimation of the state sequence if all random sources involved are Gaussian. The EM-Kalman algorithm permits to estimate parameters and sources adaptively by alternating two steps :

E-step and M-step [10]. As we focus on speakers' separation, an adaptive algorithm seems to be ideal for tracking the quick variations of the considered sources. The traditional smoothing step, needed for the M-Step, is included in the state space model. Chapter 3 presents the algorithm and associated results which have been presented in:

- S. Bensaid, A. Schutz, and D. T. M. Slock, Monomicrophone blind audio source separation using EM-Kalman filters and short+long term AR modeling, 43rd Asilomar Conference on Signals Systems and Computers, November 1-4, 2009, Asilomar, California, USA.
- S. Bensaid, A. Schutz, and D. T. M. Slock, Single Microphone Blind Audio Source Separation Using EM-Kalman Filter and Short+Long Term AR Modeling, in LVA-ICA, 9th International Conference on Latent Variable Analysis and Signal Separation, September 27-30, 2010, St. Malo, France.

Chapter 4

We consider the problem of parameters estimation as well as the source separation in a frame based context. We present, in particular, two algorithms based on the Itakura-Saito (IS) Distance for estimating the parameters of the sources individually. The estimation is done directly from the mixture and without alternating between the separation of the sources and the estimation of the parameters. The first algorithm is derived from a naive interpretation of the IS distance. It consists on alternatively estimating the short term and long term subsets of parameters. Each subset estimation needs to be iterated between all the sources (including the additive noise) until convergence. The second one is based on the true minimization of the IS distance but it is still unfinished. We remark in particular that the IS gradient is the same as for Optimally Weighted Spectrum Matching and Gaussian ML.

- Antony Schutz and Dirk T M Slock, "Blind audio source separation using short+long term AR source models and iterative itakura-saito distance minimization," in IWAENC 2010, International Workshop on Acoustic Echo and Noise Control, August 30-September 2nd, Tel Aviv, Israel.
- Antony Schutz and Dirk T M Slock, "Blind audio source separation using short+long term AR source models and Spectrum Matching", accepted in DSPE 2011, International Workshop on Digital Signal Processing and Signal Processing Education, January 4-7, Sedona Arizona, USA.

Then we focus on a frame based source separation algorithm in a Variational Bayesian context. The separation is done in the spectral domain. In the formulation of the problem, the convolution operation is done by a circulant matrix (circular convolution). We have introduced a more rigorous use of frequency domain processing via the introduction of carefully designed windows. The design of the window and the use of circulant matrices lead to simplication. The frame based algorithm extracts the windowed sources and is non-iterative. This work was presented in:

- A. Schutz and D. T. M. Slock, Single-microphone blind audio source separation via Gaussian Short+Long Term AR Models, in ISCCSP 2010, 4th International Symposium on Communications, Control and Signal Processing, March 3-5, Limassol, Cyprus.

Chapter 5 and 6

The Chapter 5 is dedicated to simulations on real signals. We have used speech (a man and a woman) and instrumental (cello and guitare) mixture on long duration. In the considered mixture the signal are non stationary, everything move. Sometimes the sources are not active, theirs periods varies. We compare the proposed algorithm, and mainly the so called **Alt-EMK** in a source separation context. We have also used this algorithm for the background extraction task involving data from the FIFA world cup 2010 in order to remove the vuvuzellas from the original signals. In Chapter 6 we give the general conclusion of our work and some possible improvement to take into account.

0.2.2 Annexe: Musical Processing

The second part of the thesis is about Musical Processing which is a very general topic. We mean by Musical Processing that all the work is related to music and more precisely to musical instruments. Several tools were developed and presented; some of which were implemented in a real time simulator and have given the expected results. Most of the work focus on the detection of interpretation effects. We present a work about the detection of the octave when it is played with a note and the last Annexe deals with an Audio-Video simulator for the analysis of a guitar player.

In this part we don't use the same model as in the first part, we use the sinusoidal plus noise model [11]. A sound is described as a sum of a sinusoid plus an additive noise. The frequencies, the amplitudes and the phases of the sinusoids are unconstrained.

Annexe A describes the musical instruments, the interpretation effects and the playing defects that will be analysed after. The instruments considered here are the Guitar, Bass, Violin, Piano.

Annexe B describes all the tools developed for the detection and Annexe C the associated results. We present an analysis/synthesis algorithm for obtaining the deterministic and the stochastic parts of the model, where the model is composed of harmonic (deterministic) and residual (stochastic part). The obtained residual component is composed of the noise, the roundoff/approximation errors and what we call the instrumental noise. The instrumental noise is a kind of signature, for a violin it is mainly due to the bow, for a piano it is more linked to the soundboards and to the hammers etc. This Harmonic plus Noise decomposition is also a transient detector when the separation is correctly performed. For example, if a violin is played by a good musician the resulting sound will be better (stronger harmonic part) and more constant than by a beginners'. The defects of the beginner are also hidden in his bowing techniques, the pressure, the orientation and the constancy of the displacement are important, and a defect on one of these points leads to noise apparition in the resulting note. Several interpretation effects are investigated for the guitar but no particular features are detected except that, in the most general case, an interpretation effect can be summarized as a frequency variation with only one attack.

- A. Schutz and D. T. M. Slock, "Modéle sinusoïdale: Estimation de la qualité de jeu dun musicien, détection de certains effets d'interprétation," in GRETSI 2007, 21eme colloque traitement du signal et des images, September 11-14, 2007, Troyes, France.

- A. Schutz and D. T. M. Slock, "Estimation of the parameters of sinusoidal signal components and of associated perceptual musical interpretation effects," in Jamboree

2007: Workshop By and For KSpace PhD Students, September, 14th 2007, Berlin, Germany.

- A. Schutz and D. T. M. Slock, ”Toward the detection of interpretation effects and playing defects,” in DSP 2009, 16th International Conference on Digital Signal Processing, July 05-07, 2009, Santorini, Greece.
- A. Schutz, N. Bertin, D. Slock, B. David, and R. Badeau, ”Piano forte pedal analysis and detection,” AES124, 2008.

In Annexe D we present an analysis of the octave problem. The octave problem appears when a note and its octave are present together. As the octave has a frequency twice as the note. If we don't take the inharmonicity into consideration, then they share the same periodicity and the partials of the spectrum are perfectly overlapped which makes the detection difficult. We propose, in this Annexe, an energetic criteria based on the energy of the odd and even partials of the chord. But we are not working in the spectral domain. We show that the odd and even harmonics of the spectrum can be represented by the odd and the even parts of a cyclic correlation. This method is also used as a pitch detector, the set of fundamental frequency has to be defined. It works in two steps; first the pitch is found in the lower octave of the instrument and then the good octave of the note is found. This approach assumes that the signal is harmonic and fails if it is not.

- A. Schutz and D. T. M. Slock, ”Periodic signal modeling for the octave problem in music transcription,” in DSP 2009, 16th International Conference on Digital Signal Processing, July 05-07, 2009, Santorini, Greece.

Annexe E gives the description of an audio-video simulator specialized for guitar. A guitar can indeed chime the same note (i.e. a note with the same pitch) at different positions of the fretboard on different strings. This is why the musical transcription of a guitar usually takes form of a tablature. A tablature is a musical notation which includes six lines (one for each guitar string) and numbers representing the position at which the string has to be pressed to perform a note with a given pitch. The proposed approach combines information from video (webcam quality) and audio analysis in order to provide the tablature. We have investigated the monophonic case, one note at a time. The audio processing is composed of an onset detector followed by a mono-pitch estimation algorithm. In case of a tablature the duration of the note is not needed. The first frame of the video is analyzed to detect the guitar and its position, then we make use of the Tomasi Lukas Kanade algorithm to follow the same points, corresponding to each string/fret intersection along the video. Filtering is done on the frame to detect the skin color and to estimate the hand position. Then knowing the pitch and the position of the hand we estimate the string and the fret which was played.

- M. Paleari, B. Huet, A. Schutz, and D. T. M. Slock, ”A multimodal approach to music transcription,” in 1st ICIP Workshop on Multimedia Information Retrieval : New Trends and Challenges, October 12-15, 2008, San Diego, USA.
- M. Paleari, B. Huet, A. Schutz, and D. T. M. Slock, ”Audio-visual guitar transcription,” in Jamboree 2008, Workshop By and For KSpace PhD Students, July, 25 2008, Paris, France.

Résumé des travaux de thèse

Ce chapitre est un résumé rédigé en Français du présent document. Il reprend les grands axes du document originellement écrit en Anglais. Dans une première partie, nous introduisons le problème de la séparation aveugle de source puis le cas monomicrophone ainsi que le positionnement du travail. Nous exposons ensuite le modèle de production de la parole utilisé dans cette thèse. Nous proposerons deux types d'algorithme, les premiers sont adaptatifs et les suivants sont basés sur une analyse par fenêtre. Les systèmes mis en oeuvre sont alors décrits. Enfn nous concluons ce chapitre par un résumé de nos contributions et développons certaines pistes à explorer à l'avenir pour améliorer les performances des systèmes présentés.

0.3 Introduction à la séparation aveugle de source

La séparation aveugle de source (SAS) est une discipline générique qui consiste à estimer K sources à partir de N observations. Elle trouve place dans de nombreuses disciplines tel que: le traitement audio, le traitement d'image, les télécommunications, le génie biomédical etc. [12]. Suivant le type d'application envisagé la nature des sources et des capteurs varient, dans le cas de l'audio les sources seront soit des instruments de musique soit de la parole et les capteurs seront des microphones. Les sources, les quantités qui nous intéressent, sont inconnues ainsi que le processus lié à leur propagation jusqu'aux microphones. Chaque microphone captera une observation, différente, qui sera un mélange déformé des sources d'origine. Les applications concernées sont par exemple dans le traitement audio: analyser les sources indépendamment et de manière automatique pour des applications de parole vers texte [13], réhaussement de la parole [14], reconnaissance de locuteur. Si l'on considère des applications liées à la musique tel que la reconnaissance automatique d'instrument [15] ou la transcription automatique de la musique [16, 17], analyser une source à la fois aidera grandement le procédé. La SAS s'applique également aux applications d'extraction de la mélodie principale [18] la restauration d'enregistrement ancien [19, 20], suppression de la voi pour le karaoke [18, 20], l'aide à l'écoute [12, 21] etc. La SAS trouve cependant de nombreuses formulations en fonction du type de problème rencontré, la Figure 1 permet de résumer ces différents cas auxquels ont peut être confronté:

- Le cas sous déterminé extrême, si seulement une observation est présente.
- Le cas sous déterminé, quand il y a moins de capteurs que de sources.
- Le cas déterminé, quand on a autant de sources que de capteurs.
- Le cas surdéterminé, le nombre de capteurs est supérieur au nombre de sources.

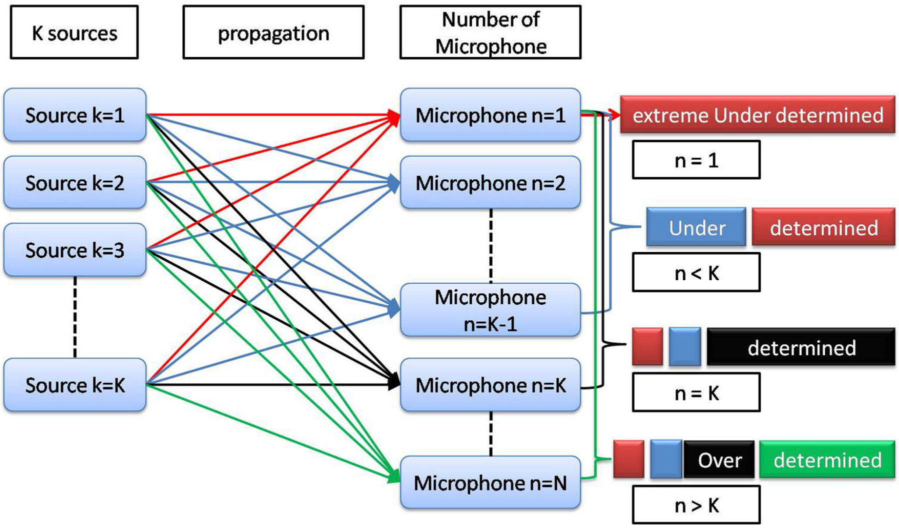

K sources
propagation
Number of Microphone
Source k=1
Source k=2
Source k=3
Source k=K
Microphone n=1
Microphone n=2
Microphone n=K-1
Microphone n=K
Microphone n=N
extreme Under determined
n = 1
Under determined
n < K
determined
n = K
Over determined
n > K

Figure 1: Détermination

La propagation des sources jusqu'aux capteurs tient aussi un rôle essentiel dans la définition du problème. On peut différencier plusieurs cas:

- Les mélanges linéaires instantanés sont les plus simples. Dans ce cas chaque observation est la somme pondérée des sources. Si on écrit les coefficients de mélanges dans une seule matrice (appelé la matrice de mélange) le problème se résume à identifier cette matrice et à l'inverser.
- Les mélanges atténués et décalés sont rencontrés lorsque les observations sont composées de versions atténuées et temporellement décalées des sources. Une source n'arrive pas au même moment sur tous les capteurs, pour chaque sources et pour chaque capteur ce décalage est différent.
- Les mélanges convolutifs sont eux les plus généraux. Dans ce cas chaque sources est filtrée avant l'acquisition par un capteur.
- Les mélanges non linéaires n'ont été étudiés que dans des cas particuliers, comme lorsque la non linéarité est introduite au moment de l'acquisition (Post non linéaire)

0.3.1 Quelques solutions existante

La SAS est apparue au milieu des années 1980 et à été tout d'abord formulée par Hérault, Ans et Jutten [22–24] pour les mélanges linéaires instantanés [23] puis, dans les années 90 pour le cas convolutif [25, 26]. Comon a introduit l'analyse en composantes indépendentes (ACI) en 1991 [27, 28], bien que de nombreux travaux aient été élaborés avant, la plupart d'entre eux peuvent être inclus dans l'ACI. L'idée original de l'ACI envisage le cas déterminé de mélange instantané et cherche à trouver l'inverse de la matrice de mélange qui rend les sources les plus indépendantes possible. Si l'on peut considérer les sources indépendantes, au sens statistique du terme, il est plus difficile de considérer que les observations le soient aussi. Une variante de l'ACI consiste à minimiser l'information mutuelle qui est aussi une mesure d'indépendance entre des variables aléatoires [29]. Suivant le même état d'esprit, si un signal se retrouve parcimonieux dans une certaine base ou transformation alors la somme des signaux le sera moins, de ceci resultera la recherche de solution amenant à maximiser la parcimonie des sources [30]. Les méthodes basées sur l'utilisation de dictionnaires cherche donc à estimer les coefficients liés aux dictionnaire et non les série temporelle elle même [30, 31]. Les séries seront reconstruites à partir du dictionnaire et des coefficients estimé, l'idée est de décomposer les observations (Y) à partir d'un dictionnaire connu(ϕ), en prenant en compte qu'il y a une matrice de mélange (A) et une matrice de coefficients (C) ce qui amène au système suivant $Y = AC\phi$ il faudra donc trouver les deux matrices A et C tel que C soit le plus parcimonieux. possible. Dans le cas sous déterminé, trouver la matrice n'est pas toujours faisable et dans ce cas son inversion n'est pas possible. Cependant ce cas, qui est plus réaliste que les cas (sur)déterminé apparaît fréquemment. Pour pouvoir séparer il faut avoir une information a priori sur les sources [7, 8] comme une hypothèse de parcimonie dans une certaine base. Par exemple l'algorithme DUET [32, 33] considère que les sources sont déjà séparées dans le plan temps fréquence, ce que les auteurs appellent la propriété d'orthogonalité W-Disjoint, et qu'un masque binaire peut être appliqué pour la séparation. Le cas sousdéterminé inclus le cas stéréophonique, le dernier cas avant le cas sous-determiné extrême. Le fait d'avoir deux observations est le cas limite pour utiliser des informations spatiales et une campage d'évaluation lui est dévolue [34, 35]. Pour des enregistrements stéréophoniques, [36, 37] prennent l'hypothèse qu'une source est dominante dans un des deux canaux, généralement les caractéristiques étudiées sont les différences de puissance et/ou de phase entre les canaux et permettent de remonter jusqu'aux angles d'arrivée des sources, avec une telle information on peut alors former un masque temps fréquence et séparer les sources [6, 33, 38–41]. Cependant si les sources sont trop proches les unes des autres ou que d'une manière générale ces caractéristiques sont trop similaires les methodes failliront.

0.3.2 Positionnement du travail exposé

Dans le travail exposé dans cette thèse nous nous placerons dans le cas Mono-Microphone, nous considérerons aussi bien le cas des mélanges linéaires instatannées que les cas atténués et décalés. Puisque nous ne pouvons, avec un seul microphone estimer les possibles retards et atténuations des signaux. Bien que les algorithmes ne soient pas restreints à ce cas précis, nous utiliserons dans les simulations des observations composées de deux sources plus un bruit blanc. Nous envisageons aussi bien des signaux de parole que de musique.

0.3.3 La séparation aveugle de source Mono-Microphone

Le cas mono-microphone est qualifié de sous-déterminé extrême et est sans doute le plus difficile puisque dans ce cas précis aucune information spatialle ne peut être utilisée. Nous sommes obligés de modèliser les sources afin de contraindre la solution. Il est néanmoins, pour beaucoup d'applications le plus réaliste, si l'on considère que $y(t)$ est une observation composée par exemple de deux sources $x_1(t)$ et $x_2(t)$ tel que $y(t) = x_1(t) + x_2(t)$ et bien n'importe quelle solution $s(t)$ satisfaisant $\hat{x}_1(t) = s(t)$ et $\hat{x}_2(t) = y(t) - s(t)$ sera une solution évidente du problème [42]. Cet exemple met en exergue la nécessité de modéliser les sources afin de contraindre les solutions. Pour cela il a été proposé d'utiliser différents types de modélisation, les premières utilisent des modèles dont les paramètres sont issues d'un certain apprentissage, les suivantes ont pour but de s'adapter (dans une certaine mesure) aux données. Dans la première catégorie, parmi les approches utilisées citons la factorisation vectorielle qui consiste à représenter un mélange comme une séquence de vecteurs.

La première étape consiste à apprendre des vecteurs sur des données d'entrainement puis à trouver la meilleure combinaison de ces vecteurs pour représenter la source. Dans cet esprit [43], Roweis utilise une représentation log magnitude d'une transformée de Fourier à court terme associé à l'approximation Log-Max $log(\ a+b\ )\ \approx \max\ \ (\ log(a),\ log(b)\ )$ ainsi chaques points temps fréquence n'appartient qu'à une source. Puis viennent les modèle de mélanges de Gaussienne (MMG), l'idée est de représenter chaque source par la réalisation d'une variable aléatoire représentée par un ensemble fini de formes spectrales caractéristiques. Dans [44], un problème considérant deux sources est analysé, chaque source est modèlisée à partir de Q états et chaque état q est représenté par une enveloppe spectrale et une distribution a priori, la séparation est effectué par l'utilisation d'un filtre de Wiener "adaptatif". Comme plusieurs trames temporelles consécutives possèdent plus ou moins le même contenu, c'est à dire un même spectre dont seul l'intensité varie, les auteurs introduisent le modèle de mélange de Gaussiennes amplifiées (MMGA), qui permet de dinstinguer l'intensité de la forme spectrale [45]. Dans le cas où les données apprises ne collent pas très bien aux données observées une adaptation du modèle est envisagée dans [42]. L'évolution temporelle des paramètres peut être pris en compte en remplacant le modèle de mélange de Gaussienne par un modèle de Markov caché (MMC). Les MMC peuvent être vue comme une généralisation des MMG. Roweiss [46] discutte l'utilisation de MMC factoriel, dans son approche les MMC/MMG sont appris pour chaque source sur des sons isolés puis appliqués à un mélange, la séparation utilise un masque binaire dans le plan temps fréquence dans lequel les sources sont supposées disjointes. Benaroya et al. [44] utilisent aussi des MMC, leur conclusion est que l'amélioration apportée par les MMC comparés au MMG n'est pas significative, au vue de certains critères d'évaluations. La séparation et l'analyse s'opère d'une manière générale dans le plan temps fréquence. Il en résulte en l'analyse d'une matrice (temps et fréquence) dont les éléments sont positifs. La factorisation en matrice non négative (FMN) devient donc un outil logique pour l'analyse. Le concept est de séparer une matrice non négative en deux matrices non négatives, la définition de ces deux matrices dépend de l'application visée et est illustré dans le cas de l'audio dans le Figure 2. La FMN est devenue très propulaire grace à l'apparition d'algorithme rapide invoquant des mises a jours multiplicatives. Originellement attribuée à Lee et Seung [47], ces mise à jour sont connues depuis longtemps dans le traitement d'image, notamment la déconvolution d'image [48] dans le cas où la matrice de base est connue [49] ou partiellement [50, 51] voir estimée. Les deux matrices sont appelées dans le cas de l'audio la matrice de base, contenant des spectres caractéristiques (spectre représenté par des peignes de fréquences fondamentales différentes) et une matrice d'activation, qui elle, est estimée. Si le spectre d'une certaine colonne est présent dans le signal alors la ligne de la matrice d'activation correspondante à cette colonne sera activée, décrivant l'évolution de l'intensité du spectre considéré.

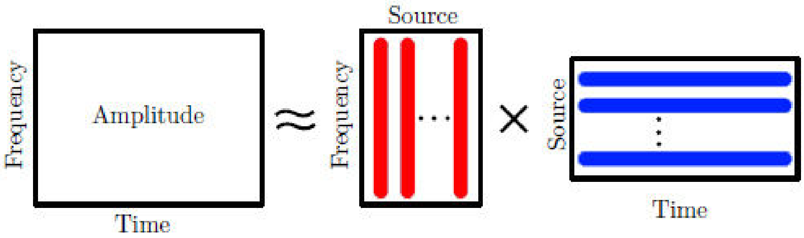


Figure 2: Une décomposition FMN typique d'un signal audio.

Parmis les méthode utilisant des modèles s'adaptant aux données le plus connu est le modèle sinusoidale bruité [11,52]. Dans ce cas là un son est une somme de sinusoides harmoniques, inharmoniques, le but consiste à détecter dans le spectre les pics correspondant à une note (d'où la contrainte d'harmonicité) puis d'estimer leurs amplitudes et phases afin de reconstruire la note seule. Plusieurs problèmes sont rencontrés comme la précision qui peut être améliorée en utilisant des interpolations sur les maximas locaux [53–55], le recouvrement fréquentiel qui peut être palié en utilisant une contrainte sur l'enveloppe spectrale résultante la forçant à être douce [56]. D'autres travaux utilisent la méthode de "matching pursuit" (MP) qui consiste à décomposer un signal en atome élémentaire [57]. Leveau dans [15] propose d'utiliser des atomes liés à chaque note de plusieurs instruments rendant l'estimation des notes et de l'instrument conjointe. Triki [58] considère l'extraction de signaux périodiques avec une faible modulation d'amplitude et de phase et rend compte que dans les zones les plus harmoniques sa méthode donne de meilleur résultat alors que le MP [57] est meilleur pour les zones transitoire (fortement non stationnaire). Un autre type de modélisation est la modélisation autorégressive (AR) ainsi certains auteurs utilisent des filtres en peigne [59], qui peuvent être implémentés en utilisant des périodes non entières [60]. Emiya [17] modèlise une note comme un processus AR pour la note et à moyenne mouvante (MM) pour le bruit (les vallées entre les pics), en utilisant un critère de maximum de vraisemblance il est capable de séparer des pièces polyphoniques. [61] Carpentier et al. utilise un filtre Kalman avec une modélisation AR de faible ordre et montre que les sources peuvent être séparées à condition que les supports soient différents.

Nous considérerons un mélange de ces derniers modèles, une source sera représentée par sa périodicité grâce à un filtre en peigne alors que les enveloppes spectrales, elles, seront modélisées par un filtre AR d'ordre faible. Ce modèle qualifié de modèle source filtre est très proche de celui utilisé par Durrieu [18]. Son modèle utilise une partie connue, un dictionnaire de fréquence fondamentale utilisé de la même manière que les FMN. Nous ne contraignons pas notre modèle.

Pour plus d'information

L'état de l'art présenté dans cette partie peut être complété par une lecteur des documents suivants

- La factorisation de quantification vectorielle, une analyse peut être trouvée dans la thèse de Ron J. Weiss [62].
- Pour le modèle de mélange de Gaussien (Amplifié ou pas), le modèle de Markov caché ainsi que le Filtre de Wiener Adaptatif, nous nous réfererons aux thèse de Laurent Benaroya [63], Alexey Ozerov [64] et Jean-Louis Durrieu [18].
- La factorisation en matrice non négative appliquée à la transcription automatique de la musique, la thèse de Nancy Bertin y est consacrée [16].
- La séparation basée sur le modèle sinusoidale bruité peut être trouvée dans la thèse de Tuomas Virtanen [65] et dans la thèse de Mahdi Triki [66] où l'on trouve aussi une revue des approches de type MP.
- La modélisation ARMM pour la transcription de la musique dans la thèse de Valentin Emiya [17].

0.4 Modèle utilisé

Comme indiqué dans la section précédente nous utiliserons un modèle paramètrique. Ce modèle est très proche du modèle CELP utilisé dans le codage de la parole. Il s'agit d'un modle source-filtre qui prend en compte les deux aspects principaux d'un son: l'aspect périodique et le timbre.

0.4.1 La parole

Une base théorique largement utilisée pour la modèlisation de la parole est le modèle source filtre. Ce modèle est basé sur la combinaison de deux aspects: les cordes vocales et un filtre acoustique linéaire [9]. Bien que ce modèle ne soit qu'une approximation, il a été largement utilisé et ceci, essentiellement, grâce à sa robustesse et sa simplicité. Ce modèle est basé sur l'hypothèse que la parole peut être modèlisée par l'utilisation de deux aspects indépendents l'un de l'autre. Ainsi les vibrations engendrées par les cordes vocales et les résonnateurs n'ont aucune interaction. Au point de vue implémentation, la technique la plus utilisée reste la modèlisation tout-pôle ou la prédiction linéaire [67]. De cette manière la, l'excitation est modélisée par un train d'impulsion qui est filtrée par le filtre tout-pôle.

0.4.2 Modélisation à long terme

Comme indiqué précédement, la modélisation à long terme s'effectuera par l'intermédiaire d'un train d'impulsion qui entraîne, dans le domaine fréquentiel, un peigne fréquentiel. Une manière de générer ce peigne est l'utilisation de filtre en peigne dont la réponse en magnitude est donnée dans la Figure 4.

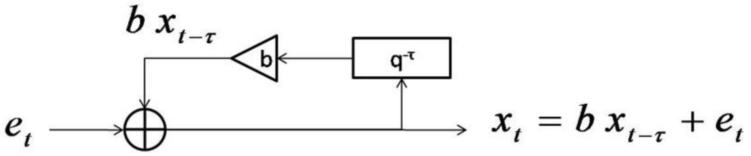


Figure 3: Filtre en peigne en mode "Feedback".

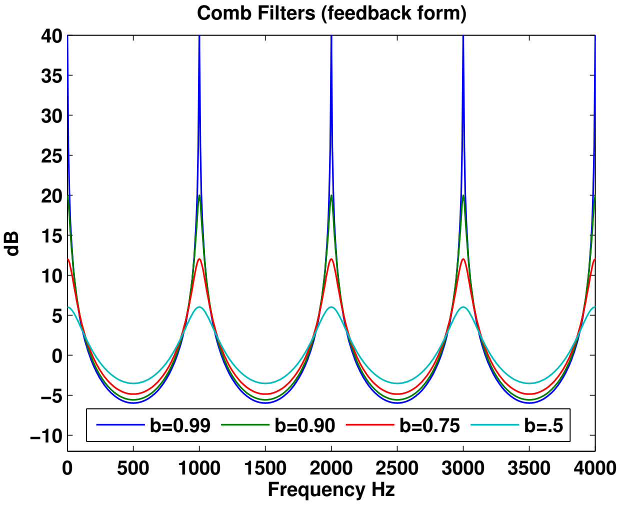


Figure 4: Réponse en magnitude pour différentes valeurs positive de b

On peut alors considérer la formulation suivante, puisque le signal obtenu en sortie du filtre en peigne dépend linéairement de son passé on peut alors le considérer comme un processus autorégréssif d'ordre élevé (égal à sa période) dont seulement deux ou trois coefficients sont non nuls. On formulera donc le signal en sortie du filtre de la manière suivante:

$$x_t = bx_{t-\tau} + e_t \tag{1}$$

$$x_t = (1-\alpha)\, bx_{t-\lfloor\tau\rfloor} + \alpha\, bx_{t-(\lfloor\tau\rfloor+1)} + e_t \tag{2}$$

x_t représente la source, e_t est un bruit blanc Gaussien, τ est la période du signal et b le coefficient de prédiction long terme Puisque la période n'est pas nécessairement entière nous introduisons un facteur d'interpolation ici linéaire α, car nous considérons que la fréquence d'échantillonnage est raisonnable [68], afin d'atteindre n'importe quelle valeur de la période. Le coefficient long terme b sera alors partagé entre deux échantillons succéssifs: $(1-\alpha)\ b$ au retard $\lfloor\tau\rfloor$ et $\alpha\ b$ au retard $\lfloor\tau\rfloor + 1$ with $\alpha = 1 - \lfloor\tau\rfloor/\tau$. Il faut noter qu'en moyenne la parole humaine s'étalone sur l'intervalle [80;250] Hz pour les hommes et [150;350] Hz pour les femmes ce qui, pour une fréquence d'échantillonage de 8KHz représente des retards de [20;100] échantillon pour les périodes. La période sera définie comme la période à un retard particulier τ qui, dans un interval pré défini, minimisera l'erreur quadratique moyenne normalisée [9]

$$J_{t,\tau} = \sum_{t=\tau-N+1}^{m} \left[|x_t - bx_{t-\tau}|^2\right] / \sum_{t=\tau-N+1}^{m} |x_t|^2 \tag{3}$$

Avec cette définition le coefficient de prédiction long terme et la période seront estimés conjointement de la manière suivante:

$$\hat{b}(\tau) = \sum_{t=\tau-N+1}^{m} [x_t\ x_{t-\tau}] / \sum_{t=\tau-N+1}^{m} x_{t-\tau}\ x_{t-\tau} = \frac{r_\tau}{r_0}$$

Où la séquence de correlation au retard k est estimé de la manière suivante

$$r_k = \frac{1}{N} \sum_{n=0}^{N-1-k} x_k x_{n-k},\ k = 1, ..., n \tag{4}$$

Le coefficient long term est alors trouvé dans la séquence de corrélation, la période est estimé comme le retard qui maximise le coefficient b. La Figure 5 illustre un signal quasi périodique et sa séquence de correlation, on y voit clairement les pics liée à la période. Le coefficient de prédiction sera la valeur du pic au retard τ normalisé par la valeur au retard 0 (ici mit à 1).

0.4.3 Modélisation à court terme

Dans le modèle source filtre les résonnateurs sont généralement modélisés par un filtre tout pôle, ici nous utiliserons un modèle auto régressif d'ordre faible qui corrélera des échantillons successifs. Puisque l'ordre est faible nous nommerons cette partie du modèle le modèle à court terme, défini comme:

$$x_t = -\sum_{n=1}^{p} a_n x_{t-n} + e_t \tag{5}$$

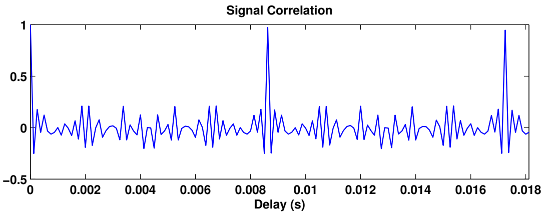


Figure 5: Un signal "long terme" et sa séquence de corrélation.

où x_t est le signal de sortie, a_n les coefficients court terme d'ordre p ($a_0 = 1$) et e_t un bruit blanc Gaussien de variance σ_e^2. Ces coefficients sont estimés par prédiction linéaire sur le signal x_t

0.4.4 Modèle de source

Une source sera alors représentée avec les deux aspects: court et long terme. Elle sera la résultante d'un bruit blanc gaussien rendu quasi-périodique par l'aspect long terme puis à laquelle on appliquera la partie court terme. Si le coefficient long terme est très faible voir nul, le bruit d'excitation sera alors l'entré de la partie court terme et pourra ainsi modéliser les sons non voisés. Une source (un signal court plus long terme) sera définie comme:

$$x_t = -\sum_{n=1}^{p} a_n x_{t-n} + \tilde{x}_t, \quad \tilde{x}_t = b\tilde{x}_{t-\tau} + e_t \tag{6}$$

Nous allons maintenant définir quelques éléments de vocabulaire utilisés dans toute la partie de la thèse. e_t est un bruit blanc Gaussien de moyenne nulle et de variance σ_e^2 nous le désignerons sous le nom d'erreur de prédiction court plus long terme. τ est la période du signal quasi périodique dont la force est réglable par l'intermédiaire du coefficient long terme b. $\tilde{x}_t$ est l'erreur de prédiction court terme, les coefficients court terme a_n ($a_0 = 1$) sont d'ordre p et finallement x_t est la source. Dans cette écriture les deux aspects sont à présent combinés et les techniques d'estimation précédemment définies doivent être opérées alternativement et itérativement. Cela consiste à estimer les coefficients long terme dans l'erreur de prédiction court terme et non dans le signal lui même, la procédure inverse doit être utilisée pour estimer les coeffients court terme, cette procédure est expliqué plus en détail dans l'annexe F.2. Il faut noter que l'estimation conjointe des paramètres peut amener à de grandes erreurs d'estimation voir à des instabilités [69].

0.4.5 Modèle de mélange

Dans notre cas, le cas mono-microphone en considérant un mélange non convolutif, l'observation ou mélange sera alors la somme de K sources autorégressives (AR) court plus long terme Gaussienne auxquelles on ajoutera un bruit blanc Gaussien v.

$$\begin{aligned} y_t &= \sum_{k=1}^{K} x_{k,t} + v_t, && (7)\\ x_{k,t} &= -\sum_{n=1}^{p_k} a_{k,n}\, x_{k,t-n} + \tilde{x}_{k,t} && (8)\\ \tilde{x}_{k,t} &= b_k\, \tilde{x}_{k,t-\tau_k} + e_{k,t} && (9) \end{aligned}$$

Chaque source est excitée avec un bruit d'excitation différent, dans la majorité des cas les périodes τ_k seront différentes. Il en va de même pour les autres paramètres, cependant il faut noter que les ordres court terme peuvent être différent et nous les supposerons connus, nous ferons de même pour le nombre de sources présentes dans le signal. Dans le domaine fréquentiel nous écrirons le modèle de le manière suivante. Tout d'abord nous introduisons les fonctions de transfert d'erreur de prédiction court et long terme comme:

$$A_k(f) = \sum_{n=0}^{p_k} a_{k,n}\; e^{-j2\pi f n}, \quad B_k(f) = 1 - b_k\; e^{-j2\pi f \tau_k} \tag{10}$$

Le spectre d'une source sera alors défini comme:

$$\begin{aligned} S_k(f) &= \frac{\sigma_k^2}{|A_k(f)\; B_k(f)|^2}\;, \quad k = 1, \ldots, K & (11)\\ &= \sigma_k^2\; S'_k(f; \theta_k) & (12) \end{aligned}$$

où nous considérons que le bruit est un modèle AR d'ordre 0 $(\sigma_0^2 = \sigma_v^2)$, le mélange sera toujours la somme des sources plus un bruit additif et est illustré sur la Figure 6:

$$Y(f) = \sum_{k=0}^{K} S_k(f) = S_0(f) + \sum_{k=1}^{K} S_k(f) = \sigma_v^2 + \sum_{k=1}^{K} S_k(f) \tag{13}$$

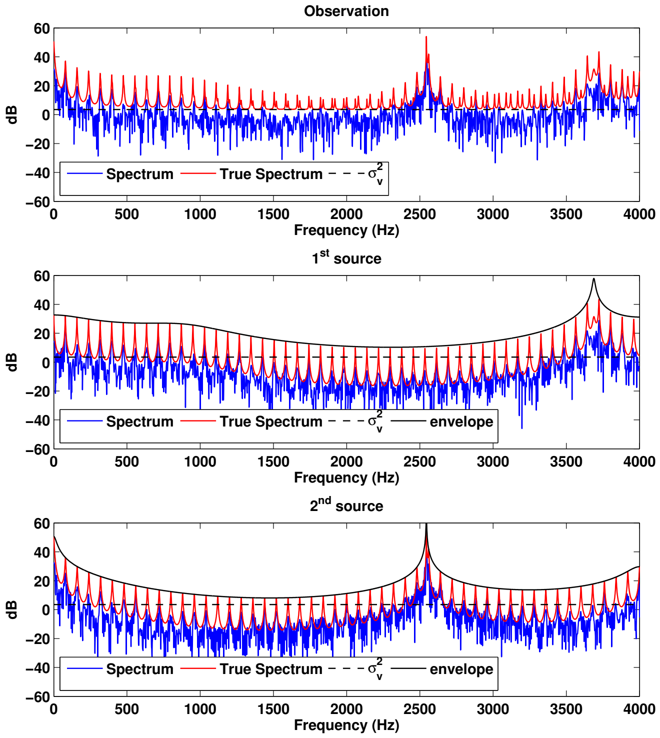


Figure 6: Spectre d'un modèle AR court plus long terme et du mélange associé.

0.5 Traitement adaptatif

Dans cette section nous expliquons le premier type d'algorithme que nous avons utilisé, il s'agit d'un algorithme de type EM-Kalman. Nous proposons et dérivons plusieurs versions de cet algorithme en considérant différentes versions du modèle.

0.5.1 Algorithme de type EM-Kalman

Un algorithme de type EM Kalman est avant tout un algorithme de type EM. On l'appelle EM pour Esperance et Maximisation, et fonctionne en deux étapes. La première étape, l'étape d'Espérance est ici réalisée par l'utilisation d'un filtre de Kalman. Cette étape a pour but d'estimer les états des sources ainsi que leurs matrices de covariance d'erreur. Le filtre de Kalman fonctionne lui aussi en deux étapes, une étape pour mettre à jour les états qui est généralement appelée mise à jour de l'observation, et une étape de prédiction dans laquelle on prédit l'observation future. L'erreur de prédiction sert à connaitre l'écart entre la nouvelle mesure et la mesure estimée afin d'effectuer une correction de la prédiction. D'étape en étape le système devient, après convergence, adaptatif. L'étape de Maximisation, elle s'occupera de maximiser les paramètres, c'est à dire de les estimer d'une manière optimale en utilisant des matrices de covariances lisées, cependant, pour effectuer le lissage le système est basé sur une connaissance de tous les états passés ce qui consomme beaucoup de mémoire [70]. Une alternative est de ne considérer un lissage qu'avec un échantillon de décalage, le filtre de Kalman étant un processus AR d'ordre 1, un décalage de 1 devrait être suffisant pour lisser les matrices [71, 72].

0.5.2 Algorithme EM-Kalman adaptatif

L'algorithme se formule de la manière suivante:

$$
\begin{array}{rcl}
\boldsymbol{K}_{k;1} & = & \boldsymbol{P}_{k-1|k-1}\,\boldsymbol{F}_{k-1}^T\,\boldsymbol{H}_{k-1}^T \\
\widehat{\boldsymbol{x}}_{k-1|k} & = & \widehat{\boldsymbol{x}}_{k-1|k-1} + \boldsymbol{K}_{k;1}\,\left(\boldsymbol{H}_{k-1}\,\boldsymbol{P}_{k|k-1}\,\boldsymbol{H}_{k-1}^T + \boldsymbol{R}_{k-1}\right)^{-1}\,\left(\boldsymbol{y}_k - \boldsymbol{H}_{k-1}\,\widehat{\boldsymbol{x}}_{k|k-1}\right) \\
\boldsymbol{P}_{k-1|k} & = & \boldsymbol{P}_{k-1|k-1} - \boldsymbol{K}_{k;1}\,\left(\boldsymbol{H}_{k-1}\,\boldsymbol{P}_{k|k-1}\,\boldsymbol{H}_{k-1}^T + \boldsymbol{R}_{k-1}\right)^{-1}\,\boldsymbol{K}_{k;1}^T \\
\boldsymbol{K}_k & = & \boldsymbol{P}_{k|k-1}\,\boldsymbol{H}_{k-1}^T\,\left(\boldsymbol{H}_{k-1}\,\boldsymbol{P}_{k|k-1}\,\boldsymbol{H}_{k-1}^T + \boldsymbol{R}_{k-1}\right)^{-1} \\
\widehat{\boldsymbol{x}}_{k|k} & = & \widehat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{K}_k\,\left(\boldsymbol{y}_k - \boldsymbol{H}_{k-1}\,\widehat{\boldsymbol{x}}_{k|k-1}\right) \\
\boldsymbol{P}_{k|k} & = & \boldsymbol{P}_{k|k-1} - \boldsymbol{K}_k\,\boldsymbol{H}_{k-1}\,\boldsymbol{P}_{k|k-1} \\
 & & \text{mise à jour des paramètres} \\
\widehat{\boldsymbol{x}}_{k+1|k} & = & \boldsymbol{F}_k\,\widehat{\boldsymbol{x}}_{k|k} \\
\boldsymbol{P}_{k+1|k} & = & \boldsymbol{F}_k\,\boldsymbol{P}_{k|k}\,\boldsymbol{F}_k^T + \boldsymbol{G}_k\,\boldsymbol{Q}_k\,\boldsymbol{G}_k^T
\end{array}
\tag{14}
$$

où $\boldsymbol{y}_k$ est l'observation à l'instant k, $\widehat{\boldsymbol{x}}_{k|k-1}$ est l'état $\widehat{\boldsymbol{x}}$ à l'instant k estimé par rapport à l'instant $k-1$, $\boldsymbol{P}$ est la matrice de covariance d'erreur, $\boldsymbol{K}$ est appelé le gain de Kalman et corrige l'estimé en fonction de l'écart de prédiction, $\boldsymbol{H}$ est appelé le modèle d'observation qui permet de transformer les états afin de les faire coller à l'observation, $\boldsymbol{G}$ est appliqué au bruit d'état $\boldsymbol{Q}$ qui est la matrice de covariance du bruit de l'observation, finallement $\boldsymbol{F}$ est la matrice de transition qui régit la dynamique du système.

Dans la suite de cette section nous exprimons le passage entre le modèle que nous avons défini et l'utilisation de l'algorithme EM-Kalman. Le modèle d'état tel que nous l'exprimerons intégrera l'opération de lissage et permettra une extraction conjointe des sources.

0.5.3 Séparation de source Mono-Microphone avec un EM-Kalman

Pour pouvoir utiliser le traitement adaptatif décrit dans la section précédente il faut pouvoir exprimer le modèle (7) sous la forme d'un modèle d'état, nous l'exprimerons de la manière suivante:

$$\begin{aligned} \mathbf{x}_t &= \mathbf{F}\,\mathbf{x}_{t-1} + \mathbf{G}\,\mathbf{e}_t \quad (15) \\ y_t &= \mathbf{h}^T\mathbf{x}_t + v_t \quad (16) \end{aligned}$$

où $\mathbf{x}$ est l'état de toutes les sources (concaténées), $\mathbf{e}$ de toutes les erreurs de prédiction court plus long terme (concaténées), $\mathbf{h}$ sera utilisé pour effectuer la sommation des sources et où $\mathbf{F}$ (diagonale par blocs) réalisera les opérations de filtrage: filtrage avec les coefficients "court terme" sur le signal et le filtrage "long terme" sur l'erreur de prédiction court terme, cette matrice est dite augmentée pour permettre le lissage. Ainsi l'état de la source k sera exprimé en concatenant des échantillons du signal et de son erreur de prédiction de la manière suivante:

$$\begin{aligned} \mathbf{x}_{k,t} = [x_k(t)\ x_k(t-1)\cdots x_k(t-p_k-1) \mid \tilde{x}_k(t)\ \tilde{x}_k(t-1)\cdots \\ \cdots\tilde{x}_k(t-\lfloor\tau_k\rfloor)\cdots\tilde{x}_k(t-N+1)]^T \end{aligned} \quad (17)$$

nous prenons p_k+2 échantillons du signal, p_k+1 pour estimer les p_k coefficients court terme plus un échantillon supplémentaire afin d'intégrer le lissage de 1 dans notre modèle directement sans devoir faire une opération séparée. Et nous prendrons N échantillons de l'erreur de prédiction court terme, nous prenons un nombre arbitraire N qui devra être supérieur à l'arrondit suppérieur de la période plus 1 afin d'également effectuer le lissage directement. Avec cette définition de l'état d'une source nous pouvons à présent écrire l'équation d'état de la source k: $\mathbf{x}_{k,t} = \mathbf{F}_k\,\mathbf{x}_{k,t-1} + \mathbf{g}_k\,e_{k,t}$, le vecteur $\mathbf{g}_k$ est de dimension $(N+p_k+2)$ et s'écrit: $\mathbf{g}_k = [\,1\ 0\cdots0 \mid 1\ 0\ \cdots\cdots\ 0]^T$ il a pour but de sélectionner le bon échantillon de l'erreur de prédiction court plus long terme afin de refleter le modèle, et est représenté par $\mathbf{G}$ (diagonale par blocs) dans (15). La matrice de transition de la source k, $\mathbf{F}_k$ s'écrira comme un assemblage de sous matrice:

$$\begin{aligned}
\mathbf{F}_k &= \begin{bmatrix} \mathbf{F}_{11,k} & \mathbf{F}_{12,k} \\ \mathbf{O} & \mathbf{F}_{22,k} \end{bmatrix} \quad (18) \\
\mathbf{F}_{11,k} &= \left[\begin{array}{cccccc} -a_{k,1} & -a_{k,2} & \cdots & -a_{k,p_k} & 0 & 0 \\ & & & & & \vdots \\ & & I_{(p_k+1)} & & & \vdots \\ & & & & & 0 \end{array}\right] \\
\mathbf{F}_{12,k} &= \begin{bmatrix} 0 & \cdots & (1-\alpha_k)\,b_k & \alpha_k\,b_k & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \end{bmatrix} \\
\mathbf{F}_{22,k} &= \left[\begin{array}{ccccccc} 0 & \cdots & (1-\alpha_k)\,b_k & \alpha_k\,b_k & 0 & \cdots & 0 \\ & & & & & & \vdots \\ & & & I_{(N-1)} & & & \vdots \\ & & & & & & 0 \end{array}\right]
\end{aligned}$$

α_k est le coefficient d'interpolation linéaire pour la source k si la période n'est pas entière.

0.5.4 Estimation des paramètres et discussion sur les états partiels

La relation qui lie dans (15) les paramètres et l'erreur de prédiction court plus long terme s'exprime en faisant intervenir ce que nous appelerons des états partiels:

$$e_{k,t-1} = \mathbf{v}_k^T \breve{\mathbf{x}}_{k,t-1} \tag{19}$$

où $\mathbf{v}_k = [1\ a_{k,1}, \cdots, a_{k,p_k},\ -(1-\alpha_k)\ b_k,\ -\alpha_k\ b_k]^T$ est un $(p_k+3)\times 1$ vecteur colonne et $\breve{\mathbf{x}}_{k,t-1} = [x_k(t-1,\theta)\cdots x_k(t-p_k-1,\theta)\ \tilde{x}_k(t-\lfloor\tau_k\rfloor-1,\theta)\ \tilde{x}_k(t-\lfloor\tau_k\rfloor-2,\theta)]^T$ est appelé état partiel de la source k. L'état partiel est obtenu en selectionnant certains échantillons dans l'état $\mathbf{x}_t$ par le biais d'une matrice de selection $\mathbf{S}_k$, cette matrice est aussi responsable du décalage d'échantillon qui mènera au lissage de 1. La relation qui lie l'etat partiel au temps $t-1$ et l'état complet à t est $\breve{\mathbf{x}}_{k,t-1} = \mathbf{S}_k\mathbf{x}_t$. Il faut à présent noter que l'on peut définir différents états partiels tout comme nous pouvons simplifier le modèle de départ, on peut aussi bien ne considérer que les parties court terme (CT) et/ou parties long terme (LT), de plus on peut faire l'approximation de découplage de ces deux parties. Ainsi on peut définir trois types d'algorithme: CT seulement, LT seulement et CT+LT, ils mèneront aux algorithmes que nous nommerons **AR-ST(ordre)**, **AR-LT(τ)** et **AR-STLT** respectivement. Pour ce dernier on peut coupler les paramètres CT et LT pour une estimation conjointe, algorithme **Joint-EMK**, ou alors les découpler pour obtenir une estimation alternée, algorithme **Alt-EMK**.
Après avoir multiplié dans (19) par $\breve{\mathbf{x}}_{k,t-1}^T$ et une fois l'opérateur d'espérance conditionelle $(E\{\ |y_{1:t}\})$ appliquée on obtient apres une inversion matricielle la relation suivante:

$$\mathbf{v}_k = \sigma_k \mathbf{R}_{k,t-1}^{-1}[1, 0\cdots 0]^T \tag{20}$$

On peut à présent remarquer qu'il s'agit bien d'une matrice de covariance lissée avec un décalage de 1 car $\mathbf{R}_{k,t-1}$ est définit comme étant $E\left\{\breve{\mathbf{x}}_{k,t-1}\breve{\mathbf{x}}_{k,t-1}^T|y_{1:t}\right\}$ ce qui correspond à $\mathbf{R}_{k,t-1} = \mathbf{S}_k E\left\{\mathbf{x}_t\mathbf{x}_t^T|y_{1:t}\right\}\mathbf{S}_k^T$.
La variance du bruit aditif sera, elle, estimée par maximum de vraisemblance (log $P\left(y_t|\mathbf{x}_t,\sigma_v^2\right)$), l'estimé sera $\hat{\sigma_v^2} = E\left[\left(y_t - \mathbf{h}^T\hat{\mathbf{x}}_{t|t}\right)^2\right] + \mathbf{h}^T\mathbf{P}_{t|t}\mathbf{h}$. En pratique toutes les opérations d'espérance, pour la variance du bruit additif et pour la matrice de covariance utilisée pour les paramètres, seront effectuées par l'intermédiaire de facteurs d'oublie λ, le facteur d'oublie est positif et généralement proche de 1. Il faut noter que pour l'algorithme **Alt-EMK** qui invoquera une estimation alternée des paramètres deux facteurs d'oulies pourront être utilisés pour les parties ST et LT, de plus pour assurer la symétrie des matrices $\boldsymbol{P}_{t|t-1}$ et $\boldsymbol{P}_{t|t}$ nous utiliserons la forme de Jospeh [73]:

$$\begin{array}{rcl} \mathbf{P}_{t|t} & = & \mathbf{P}_{t|t-1} - \mathbf{K}_t\mathbf{h}^T\mathbf{P}_{t|t-1} \\ \rightarrow & = & \left(\mathbf{I} - \mathbf{K}_t\mathbf{h}^T\right)\mathbf{P}_{t|t-1}\left(\mathbf{I} - \mathbf{K}_t\mathbf{h}^T\right) + \mathbf{K}_t\mathbf{K}_t^T\hat{\sigma_v^2} \end{array} \tag{21}$$

Dans les simulations nous considérerons différents algorithmes. Tout d'abord nous les testerons avec des signaux synthétiques dont tout est connu, nous analyserons les performances des algorithmes dans le cas du filtrage et de l'estimation adaptative. Pour le filtrage aucune estimation n'est nécessaire il n'y aura donc pas de différence entre les algorithmes ***Joint-EMK*** *et* ***Alt-EMK*** *et ils seront remplacés par le filtrage CT+LT à savoir l'algorithme* ***AR-STLT****. Les autres algorithmes seront* ***AR-LT(τ)*** *et* ***AR-ST(ordre)****, pour ce dernier nous envisagerons deux ordres: l'ordre court terme p_k des sources et l'ordre totale des sources $p_k + \tau_k$.*

0.5.5 Résultat de simulations sur des signaux synthetiques

Le première série de simulations consiste à comparer les différents algorithmes sur des signaux à la fois synthétiques et stationnaires, nous comparons ainsi les algorithmes dans le cas du "Filtrage" en supposant les paramètres connus (ceux utilisé pour générer les signaux) puis le cas de l'estimation adaptative. Dans le cas de l'estimation nous supposons toutefois les périodes connues, tous les autres paramètres sont initialisés de la même manière pour tous les algorithmes. Les coefficients CT sont mis à zéro, aucun a priori sur les enveloppes spectrales, les coefficients LT à 0.99 pour renforcer les informations liées à la périodicité, nous initialisons les sources avec la même valeur de variance $\sigma_k^2 = 1$ (aucun a priori sur la puissance des sources) et finalement la variance du bruit additif à 1 également. Bien entendu les algorithmes diminués (CT ou LT uniquement) ne sont initialisés qu'avec les paramètres liés à leur formulation, pour le CT uniquement de grand ordre nous utilisons l'algorithme de Levinson [67]. Dans le cas du filtrage il apparait que le CT uniquement d'ordre de la source donne la meilleur séparation mais il est suivi de très près par le LT uniquement ainsi que le CT+LT, le plus mauvais étant le CT de faible ordre. Dans le cas de l'estimation les choses changent, les algorithmes basés sur le CT seulement n'ont aucune information liée à la périodicité et ne sont pas capables de séparer les sources, les autres algorithmes donnent eux des resultats encore une fois équivalents. Nous conserverons le modèle nous avons proposé car en plus de suivre la périodicité il prend en compte les enveloppes spectrales ce qui fera une différence lorsque nous traiterons des signaux réels. Les algorithmes sont comparés en utilisant 4 critères d'évaluation, ces critères sont calculés dans une zone où les algorithmes ont convergé et sont l'erreur quadratique moyenne (MSE) ainsi que les rapports source à distorsion/interférence/artefact (SDR/SIR/SAR) définient dans [74]. Nous considérons deux types de non stationnarité: les variations d'amplitude à périodes constantes générés avec le modèle AR CT+LT et les variations de période avec une enveloppe spectrale fixe en utilisant le modèle sinusoidale ($b_k = 1$). Les résultats des critère d'évaluation sont montrés dans la Table 3. La variation d'amplitude est faite de la manière suivante: tout d'abord deux signaux sont générés, le premier est alors pondéré vers la fin de signal l'amenant à s'éteindre doucement. Le deuxième signal, lui, est pondéré en son milieu il décroit progressivement vers 0 puis réapparait, il faut noter que dans ce cas qu'il s'agisse du milieu ou de la fin du signal il n'y a plus qu'une source de présente à la fois. Les algorithmes, eux, n'en sont pas informés et chercheront tout de même deux sources. Les variations de périodes sont choisis de telle sorte que les fréquences fondamentales se croisent à un moment, ceci aurait pu générer une inversion des sources si les enveloppes spectrales n'etaient pas prises en compte. On y constate que pour les variations d'amplitude que les deux algorithmes d'estimation donnent des résultats similaires cependant lorsque les périodes varient le **Alt-EMK** est bien meilleur, ceci est en partie du à l'utilisation de facteurs d'oublie différents pour l'estimation des parties CT et LT, dans le cas présent la partie CT est fixe alors que la partie LT varie il faut donc être capable de s'adapter plus rapidement.

Table 3: Critère d'évaluation en dB: variation d'amplitude ou de périodicité.

Cas	Algorithme	SDR	SAR	SIR	MSE
Variation d'Amplitude	**Joint-EMK**	14.5 (17)	18.3 (16.5)	19.3 (16)	-27.8 (-13)
Variation d'Amplitude	**Alt-EMK**	15.3 (17)	18 (16)	20 (16)	-28 (-13.5)
Variation de Période	**Joint-EMK**	7.3 (17)	15.5 (16.5)	9.2 (16)	-15.6 (-13)
Variation de Période	**Alt-EMK**	10.5 (17)	16.8 (16)	13.4 (16)	-15.5 (-13.5)

0.6 Traitement "par fenêtre"

Le deuxième type d'algorithmes que nous considérons est basé sur un découpage du signal en trame en utilisant des fenêtres. Les signaux sont non-stationnaires mais nous considérons qu'ils sont stationnaires par morceaux. Le choix de la fenêtre d'analyse ainsi que de sa durée est très important, la durée doit être suffisamment grande pour inclure les informations liées à la périodicité et suffisament courte pour pouvoir considérer les signaux stationnaires. Pour renforcer la stationnarité, les fenêtres sont souvent choisies pour concentrer le maximum d'énergie au centre de la fenêtre qui par conséquence se retrouve être décroissante sur les bords. Cette décroissance impose que des fenêtres consécutives se recouvrent amenant au problème de reconstruction parfaite des signaux, tels que les signaux extraits décalés et sommés doivent être les mêmes que les signaux d'origine. Tous ces points seront pris en compte dans notre analyse et la séparation de source s'effectuera en deux étapes, la première consistera à estimer les paramètres d'une trame et de s'en servir comme initialisation pout l'étude de la trame suivante. Nous proposons deux algorithmes d'estimation des paramètres en utilisant la distance d'Itakura Saito appliquée à notre modèle. La deuxième étape sera un algorithme de séparation utilisant les paramètres estimés. Puisque l'observation est découpée en trame stationnaire les traitement se feront dans le domaine spectral et réduiront ainsi la charge de calcul, nous utiliserons donc dans cette partie le modèle spectrale défini dans (13). Bien que nous n'utilisons pas l'algorithme EM, nous conserverons dans notre analyse son état d'esprit pour la définition des paramètres, ainsi les paramètres de la source k seront pour le CT $\mathbf{a}_k = [a_{k,1} \cdots a_{k,p_k}]$ et $\varphi_{\mathbf{k}} = [\ \mathbf{b}_{\mathbf{k}}\ \tau_{\mathbf{k}}\ \sigma^{\mathbf{2}}_{\mathbf{k}}\ ]$ pour les paramètres LT. Ces paramètres, seront, définit par source dans $\theta_k = [\ \mathbf{a}_k\ \varphi_{\mathbf{k}}]^{\mathbf{T}}$ et pour toutes les sources dans $\theta = [\sigma_v^2\ \theta_1^T \cdots \theta_K^T]^T$.

0.6.1 Distance d'Itakura Saito

La distance d'Itakura Saito (IS) [75] est une distance de mesure de différence spectrale et possède la propriété de séparation qui nous interesse, à savoir que $IS(a,b) = 0$ si $a = b$. Si l'on cherche à l'appliquer à la mesure de différence entre le spectre de l'observation $Y(f) = \frac{|y(f)|^2}{N}$ et notre spectre paramètrique $S(f;\theta)$ la distance prendra la forme suivante:

$$\int df \left[\frac{Y(f)}{S(f;\theta)} - ln\left(\frac{Y(f)}{S(f;\theta)}\right) - 1 \right] \tag{22}$$

0.6.2 Interprétation Naïve de la distance IS

Si nous considérons une estimation alternée des paramètres c'est à dire estimer un type de paramètre d'une source en gardant les autres paramètres et les autres sources fixes et que l'on remarque le fait suivant:

$$\frac{Y(f)}{S(f;\theta)} = \frac{1}{S_k(f;\theta_k)} \frac{S_k(f;\theta_k)}{\sum_k S_k(f;\theta_k)} Y(f) = \frac{1}{S_k(f;\theta_k)}\ \frac{Y(f)}{1 + S_k(f;\theta_k)^{-1} \sum_{\bar{k} \neq k} S_{\bar{k}}(f;\theta_{\bar{k}})} \tag{23}$$

nous remarquons que le rapport impliqué dans IS peut être vue comme:

$$\frac{Y(f)}{S(f;\theta)} = \frac{1}{S_k(f;\theta_k)}\ \frac{1}{N} \frac{y(f)}{1 + S_k(f;\theta_k)^{-1} \sum_{\bar{k} \neq k} S_{\bar{k}}(f;\theta_{\bar{k}})} y^*(f) = \frac{\hat{S}_k(f)}{S_k(f;\theta_k)} \tag{24}$$

L'interpretation naïve consiste donc à réécrire la distance IS de cette manière la et de minimiser par rapport à $S_k(f;\theta_k)$ en ignorant la dépendance de l'estimé de Wiener $\hat{S}_k(f)$

par rapport aux paramètres. Ceci grâce au découpage des paramètres, amène à estimer alternativement les paramètres par prédiction linéaire sur des versions nettoyées de l'observation.

0.6.2.1 Estimation de paramètres par la méthode Naïve

L'estimation des paramètres CT et LT sera alternée, de plus lorsque nous estimerons les paramètres d'une source nous considérerons les autres comme constantes. L'alternance entre les sources fera que les paramètres d'une source seront estimés sur une version de l'observation qui sera nettoyée des autres sources estimées dans $\hat{S}_k(f)$. L'alternance entre les paramètres fera que la source estimée $\hat{S}_k(f)$ sera en plus nettoyée de l'influence des autres paramètres. L'algorithme itérera donc entre les paramètres des sources jusqu'à convergence des paramètres et par alternance entre les parties CT et LT. Ainsi les paramètres CT seront estimés dans $r_k = F^{-1}\left(\hat{S}_k(f)\ |B_k(f)|^2\right)$ en utilisant l'algorithme de Levinson et les paramètres LT seront eux estimés dans $r_k = F^{-1}\left(\hat{S}_k(f)\ |A_k(f)|^2\right)$. Le coefficient LT et la période seront estimés conjointement comme il est expliqué dans (4), la variance de l'erreur de prédiction CT+LT sera la moyenne de $\hat{S}_k(f)\ |A_k(f)B_k(f)|^2$ et la variance du bruit additif, considérée comme une source AR d'ordre 0, comme la moyenne de $\hat{S}_0(f)$

0.6.3 Minimisation de la distance IS

La minimisation par rapport à un jeu de paramètre, θ_i, d'une source amène à:

$$\frac{\partial}{\partial\theta_i}\int df\left[\frac{Y(f)}{S(f;\theta)} - ln\left(\frac{Y(f)}{S(f;\theta)}\right) - 1\right] = \int df\frac{1}{S(f;\theta)^2}[S(f;\theta) - Y(f)]\frac{\partial S_i(f;\theta_i)}{\partial\theta_i} \tag{25}$$

Qui possède le même gradient le maximum de vraisemblance Gaussien de $y(f)$, avec une moyenne nulle et une variance $S(f;\theta)$:

$$\int df\left[\frac{Y(f)}{S(f;\theta)} + ln\left(S(f;\theta)\right)\right] \tag{26}$$

Et qu'à la correspondance de spectre par moindres carrés pondérés pour $Y(f)$ de moyenne $S(f;\theta)$ et de variance $S(f;\theta)^2$

$$\int df\ \frac{1}{S(f;\theta)^2}\ [Y(f) - S(f;\theta)]^2 \tag{27}$$

que nous utiliserons pour l'estimation des variances. Si l'on considère la variable $\psi(f)$ pour représenter $A(f)$ ou $B(f)$ le gradient devient:

$$\begin{aligned}\frac{\partial}{\partial\psi_i}\int df\left[\frac{Y(f)}{S(f;\theta)} - ln\left(\frac{Y(f)}{S(f;\theta)}\right) - 1\right] &= \int df\frac{1}{S(f;\theta)^2}[S(f;\theta) - Y(f)]\frac{\partial S_i(f;\theta_i)}{\partial\psi_i}\\ \frac{\partial S_k(f;\theta_k)}{\partial\psi_k^*} &= -S_k(f;\theta_k)\frac{\psi_k(f)}{|\psi_k(f)|^2}\end{aligned} \tag{28}$$

qui amène à résoudre le système suivant:

$$\left(\frac{Y(f)}{S(f;\theta)}\frac{S_k(f;\theta_k)}{S(f;\theta)}\frac{1}{|\psi_k(f)|^2}\right)\psi_k(f) = \frac{S_k(f;\theta_k)}{S(f;\theta)}\frac{1}{\psi_k(f)^*} \tag{29}$$

qui correspond aux équations de Yule-Walker avec un second membre, c'est à dire que contrairement à l'interprétation naïve la minimisation n'amène pas à un problème de prédiction linéaire basique. Ce système invoque deux membres qui, une fois transposés dans le domaine temporel, correspondent à deux séquences de corrélation devant être estimées itérativement puisqu'elles dépendent toutes les deux des paramètres. Le système à résoudre devenant $T\left(r_{k,(0,\cdots,p_k-1)}\right)\ \mathbf{a}_k \ = \ g_{k,(1,\cdots,p_k)} - r_{k,(1,\cdots,p_k)}$. Où T est une matrice de Toeplitz construite avec les p_k premiers éléments de r_k, ψ_k sont les coefficients CT ou LT en fonction de ce que l'on estime g_k la séquence de corrélation résultante, r_k et g_k sont:

$$r_k = F^{-1}\left(\frac{Y(f)}{S(f;\theta)}\frac{S_k(f;\theta_k)}{S(f;\theta)}\frac{1}{|A_k(f)|^2}\right),\ g_k = F^{-1}\left(\frac{S_k(f;\theta_k)}{S(f;\theta)}\frac{1}{A_k(f)^*}\right) \tag{30}$$

Les variances seront estimées en utilisant la correspondance de spectre par moindres carrés pondérés, en pondérant directement avec le périodogramme:

$$\int df \frac{1}{Y(f)^2}\left[\sum_{k=0}^{K}\sigma_k^2 S'_k(f;\theta_k) - Y(f)\right]^2 . \tag{31}$$

La mimnimisation par rapport à $\underline{\sigma}^2 \ = \ [\ \sigma_v^2\ \sigma_1^2 ... \sigma_K^2\ ]^T$ amène à résoudre le systeme suivant $G\ \underline{\sigma}^2 \ = \ d$, où:

$$G_{ik} = \int df\ \frac{S'_i(f;\theta_i)\, S'_k(f;\theta_k)}{Y(f)^2}\ ,\ d_i = \int df\ \frac{S'_i(f;\theta_i)}{Y(f)} \tag{32}$$

0.6.4 Algorithme de séparation de source

Pour l'algorithme de séparation de source nous ferons plusieurs simplifications: les operations de filtrage seront réalisées par des matrices circulantes et dans le domaine spectral la fenêtre d'analyse sera réduite à son lobe principale. L'algorithme effectuera un extraction conjointe des sources, pour cela comme pour les algorithmes adaptatifs présentés les sources seront concaténées dans un seul vecteur. En prenant les matrices circulante pour les opérations de filtrage l'erreur de prédiction CT+LT s'écriera $\mathbf{W}\mathbf{e}_k = \mathbf{A}_k\mathbf{B}_k\,\mathbf{W}\mathbf{x}_k$. Où $\mathbf{W} = diag(w)$ et w la fenêtre d'analyse, $\mathbf{e}_k$ l'erreur de prédiction CT+LT de la source k, $\mathbf{A}_k$ et $\mathbf{B}_k$ les matrices circulantes de filtrage pour le CT et LT respectivement et $\mathbf{x}_k$ la source k.

0.6.4.1 Modèle conjoint

Afin d'écrire le modèle conjoint, nous concaténerons les sources dans un seul vecteur ce qui amène à écrire le modèle sous la forme suivante:

$$\begin{array}{rcl}
\underline{\mathbf{W}} & = & \oplus_{k=1}^{K}\mathbf{W} \ = \ I_K \otimes \mathbf{W} \\
\underline{\mathbf{I}} & = & [I_N \dots I_N] \ = \ \mathbf{1}_k^T \otimes I_N \\
\underline{\mathbf{A}} & = & \oplus_{k=1}^{K}\mathbf{A}_k \ = \ \text{blockdiag}\{\mathbf{A}_1, \dots, \mathbf{A}_K\} \\
\underline{\mathbf{B}} & = & \oplus_{k=1}^{K}\mathbf{B}_k \ = \ \text{blockdiag}\{\mathbf{B}_1, \dots, \mathbf{B}_K\} \\
\underline{\Lambda} & = & \oplus_{k=1}^{K}\lambda_k I_N \ = \ \Lambda \otimes I_N \\
\Lambda & = & diag\{\lambda_1, \dots, \lambda_k\} \\
\underline{\mathbf{x}}' & = & \underline{\mathbf{W}}\,\underline{\mathbf{x}} \\
\underline{\Lambda}' & = & \underline{\mathbf{W}}^{-1}\underline{\Lambda}\underline{\mathbf{W}}^{-1} = \Lambda \otimes \mathbf{W}^{-2} \\
\underline{\mathbf{e}} & = & [\mathbf{e}_1^T \cdots \mathbf{e}_K^T]^T.
\end{array} \tag{33}$$

Où $\oplus$ et $\otimes$ sont la somme et le produit de Kronecker respectivement. Nous avons effectué un changement de variable, ainsi les sources sont maintenant fenêtrées, ceci est nécessaire afin de prendre en compte la fenêtre d'une manière correcte durant l'analyse. Il en resultera que nous reconstruirons des sources fenetrées. Avec cette notation le modèle peut maintenant s'écrire:

$$
\begin{aligned}
\mathbf{W}\mathbf{y} &= \underline{\mathbf{I}}\ \underline{\mathbf{x}}' + \mathbf{W}\ \mathbf{v} && (34)\\
\underline{\mathbf{A}}\ \underline{\mathbf{B}}\ \underline{\mathbf{x}}' &= \underline{\mathbf{W}}\ \underline{\mathbf{e}}. && (35)
\end{aligned}
$$

Avec $\mathbf{A}_k$, $\mathbf{B}_k$ des matrices ciculantes. En prenant un a priori Gaussien pour les sources nous obtenons la matrice de covariance et la moyenne suivante pour les sources:

$$
\begin{aligned}
C_{\underline{\mathbf{x}}'} &= (\lambda_v \underline{\mathbf{I}}^T \mathbf{W}^{-2} \underline{\mathbf{I}} + \mathbf{C}_1)^{-1} \ = \ \mathbf{C}_1^{-1} - \mathbf{C}_1^{-1} \underline{\mathbf{I}}^T \mathbf{C}_2^{-1} \underline{\mathbf{I}} \mathbf{C}_1^{-1} \\
m_{\underline{\mathbf{x}}'} &= \mathbf{C}_1^{-1} \underline{\mathbf{I}}^T \mathbf{C}_2^{-1} \mathbf{W}\mathbf{y}\,.
\end{aligned} \qquad (36)
$$

Où

$$
\begin{aligned}
\mathbf{C}_1 &= \underline{\mathbf{B}}^T \underline{\mathbf{A}}^T \underline{\Lambda}' \underline{\mathbf{A}}\, \underline{\mathbf{B}} \\
&= \operatorname{blockdiag}\{\lambda_k \mathbf{B}_k^T \mathbf{A}_k^T \mathbf{W}^{-2} \mathbf{A}_k \mathbf{B}_k\}_{k=1}^K \\
\mathbf{C}_2 &= \tfrac{1}{\lambda_v} \mathbf{W}^2 + \underline{\mathbf{I}} \mathbf{C}_1^{-1} \underline{\mathbf{I}}^T \\
&= \tfrac{1}{\lambda_v} \mathbf{W}^2 + \textstyle\sum_k \tfrac{1}{\lambda_k} \mathbf{B}_k^{-1} \mathbf{A}_k^{-1} \mathbf{W}^2 \mathbf{A}_k^{-T} \mathbf{B}_k^{-T}
\end{aligned} \qquad (37)
$$

En se placant dans le domaine fréquentiel il faut noter que les matrices circulantes deviendront diagonales (matrices $\mathbf{A}_k$ et $\mathbf{B}_k$) et, inversement, la matrice diagonale $\mathbf{W}$ deviendra circulante. Cependant nous restreindrons le spectre de la fenêtre à son lobe principale ceci aura pour effet de rendre la matrice bande diagonale et non circulante. Dans le domaine fréquentiel, les calculs n'invoqueront que des matrices diagonales ou bandes diagonales, on réduira la charge de calcul par une décomposition LDU:

$$
\begin{aligned}
\mathbf{F}\mathbf{C}_2\mathbf{F}^{-1} &= \mathbf{F}\left[\frac{1}{\lambda_v}\mathbf{W}^2 + \sum_k \frac{1}{\lambda_k} \mathbf{B}_k^{-1} \mathbf{A}_k^{-1} \mathbf{W}^2 \mathbf{A}_k^{-T} \mathbf{B}_k^{-T}\right] \mathbf{F}^{-1} \\
&= \frac{1}{\lambda_v} \breve{\mathbf{W}}_2 + \sum_k \frac{1}{\lambda_k} \breve{\mathbf{B}}_k^{-1} \breve{\mathbf{A}}_k^{-1} \breve{\mathbf{W}}_2 \breve{\mathbf{A}}_k^{-H} \breve{\mathbf{B}}_k^{-H} \ = \ \mathbf{L}\mathbf{D}\mathbf{L}^H \qquad (38)
\end{aligned}
$$

Afin de séparer les sources, calculer $m_{\underline{\mathbf{x}}'}$, il faudra appliquer les étapes suivantes :

- $\breve{\mathbf{y}} \ = \ \mathbf{F}\ \mathbf{W}\ \mathbf{y}$
- résoudre $\breve{\mathbf{u}}$ depuis $\mathbf{L}\breve{\mathbf{u}} = \breve{\mathbf{y}}$ par substitution arrière
- résoudre $\breve{\mathbf{z}}$ depuis $\mathbf{L}^H \breve{\mathbf{z}} = \mathbf{D}^{-1}\breve{\mathbf{u}}$ par substitution arrière
- $m_{\mathbf{x}'_k} \ = \ \frac{1}{\lambda_k}\ \mathbf{F}^{-1}\ \breve{\mathbf{B}}_k^{-1} \breve{\mathbf{A}}_k^{-1} \breve{\mathbf{W}}_2 \breve{\mathbf{A}}_k^{-H} \breve{\mathbf{B}}_k^{-H}\ \breve{\mathbf{z}}$

De plus en prenant en compte que $\breve{\mathbf{B}}_k = \operatorname{diag}\{\breve{\mathbf{b}}_k\}$ et $\breve{\mathbf{A}}_k = \operatorname{diag}\{\breve{\mathbf{a}}_k\}$ sont diagonales on peut simplifier le calcul suivant:

$$
\breve{\mathbf{B}}_k^{-1} \breve{\mathbf{A}}_k^{-1} \breve{\mathbf{W}}_2 \breve{\mathbf{A}}_k^{-H} \breve{\mathbf{B}}_k^{-H} \ = \ \frac{1}{\breve{\mathbf{a}}_k} \frac{1}{\breve{\mathbf{a}}_k^H} \odot \frac{1}{\breve{\mathbf{b}}_k} \frac{1}{\breve{\mathbf{b}}_k^H} \odot \breve{\mathbf{W}}_2\,. \qquad (39)
$$

0.7 Simulations

Les données réelles de parole utilisées dans ce chapitre proviennent de la campagne de séparation de source stéréo -phonique SASSEC 2007 [35]. Les sources originales sont stéréophoniques, nous ne prenons cependant qu'un seul des deux canaux (droite), ils ont une durée de 10 seconde et sont échantillonnés à 16 KHz, nous les décimons à 8 KHz. Les deux fichiers utilisés sont composés d'un locuteur et d'une locutrice, les deux parlant en anglais. Les fichiers utilisés sont "$female\_inst\_sim\_1.wav$" et "$male\_inst\_sim\_1.wav$". Le mélange est artificiel et le bruit additif l'est aussi, lorsque nous travaillerons avec de court segment nous pourrons fixer le rapport signal à bruit (RSB), chose que nous ne pourrons pas faire lorsque les signaux seront non stationnaires. Nous effectuerons aussi des simulations avec des signaux musicaux, ici le mélange sera composé d'un son de violoncel provenant d'un enregistrement CD, un concerto de Paganini joué par Yo Yoma [76], et une pièce de guitare provenant d'un enregistrement personnel, encore une fois un seul canal sera conservé et les signaux sont décimés. Il faut noter qu'à présent nous ne travaillons plus avec des signaux synthétiques et donc que les paramètres estimés, sur chaque sources individuellement, ne sont ni stationnaires ni parfaitement estimés. Nous ferons un test d'extraction de bruit de fond sur un enregistrement de coupe du monde 2010 dans lequel un ensemble de vuvuzela pollue l'écoute, ces fichiers sont accessibles sur le site de l'industriel Audionamix [77] qui fournit des fichiers MP3 avant et après traitement donnant ainsi accès à leur estimé du vuvuzela. Tous les algorithmes sont programmé sous Matlab et pour les simulations le bruit sera choisi de telle sorte que le RSB global sera de 20 dB. Nous utiliserons tous les algorithmes présentés précédemment à savoir les algorithmes adaptatifs de type EM-Kalman et les algorithmes de traitement par fenêtre. Nous comparerons les résultats obtenus par filtrage à ceux obtenus par estimation. Les algorithmes utilisés sont pour le traitement adaptatif **KF** pour le filtrage de Kalman avec notre modèle ainsi que **Alt-EMK** et **Joint-EMK** pour les estimations alternatives et conjointes des paramètres. Pour les algorithmes par fenêtre le cas du filtrage sera nommé **VBSS** car il a été initialement conçu dans un cadre variationel bayésien, lorsque nous utiliserons l'estimation naïve des paramètres l'algorithme sera **Naive-IS** et pour la minimisation de la distance IS **Tmin-IS**. Dans le cas de l'estimation les algorithmes seront tous initialisés avec les mêmes valeurs et nous supposerons les périodes connues.

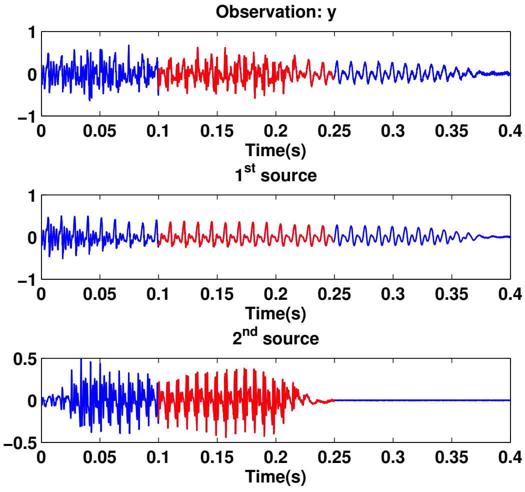


Figure 7: Signaux de courte durée, l'observation et les sources.

0.7.1 Signaux de courte durée

La première simulation utilisera des signaux de courte durée, illustrée dans la Figure 7, les critères d'évaluation seront calculés dans la zone mise en rouge c'est à dire là où les deux sources sont présentes et où les algorithmes adaptatif devraient déjà avoir convergé. Nous travaillons tout d'abord avec ces signaux car leur périodes sont stables et sont considérés constantes. Les critères sont donnés dans la Table 4.

Dans la zone dans laquelle les critères sont calculés les fenêtres n'ont aucune influence car cette zone se situe dans la phase de reconstruction parfaite. Les résultats, dans cette zone, semblent similaires pour tous les algorithmes. Ils sont néanmoins plus mauvais pour le **Tmin-IS** que pour les autres, ceci est essentiellement dù au fait que cet algorithme n'est pas encore finalisé, l'effet de la fenêtre d'analyse y est plus crucial que pour la version naïve (**Naive-IS**) et n'est pas encore prise en compte. L'algorithme **Alt-EMK** donne de résultats légerement meilleurs que le **Joint-EMK**, l'amélioration est en moyenne comprise entre 0.5 et 1 dB. Toutefois en terme de filtrage l'algorithme de séparation par fenêtre donne de meilleur résultat que le filtre de Kalman.

Table 4: MSE, SIR, SAR et SDR pour des signaux de courte durée.

Source	Algorithme	MSE	SIR	SAR	SDR
1	Filtrage **KF**	-25.14 (-26.34)	11.16 (10.60)	21.76 (21.41)	10.77 (10.23)
2	Filtrage **KF**	-25.55 (-26.67)	08.83 (16.31)	23.81 (21.30)	08.67 (15.09)
1	**Joint-EMK**	-23.42 (-24.88)	09.27 (09.03)	14.66 (21.61)	08.05 (08.76)
2	**Joint-EMK**	-23.51 (-24.90)	05.40 (12.68)	11.82 (20.26)	04.28 (11.95)
1	**Alt-EMK**	-23.71 (-25.44)	09.63 (09.59)	14.98 (21.61)	08.41 (09.29)
2	**Alt-EMK**	-23.82 (-25.50)	06.30 (13.75)	12.60 (21.90)	05.20 (13.11)
1	Filtrage **VBSS**	-31.07 (-29.49)	15.88 (15.11)	24.40 (21.52)	14.30 (14.30)
2	Filtrage **VBSS**	-31.31 (-29.51)	15.19 (15.20)	19.92 (21.80)	13.90 (14.30)
1	**Naive-IS**	-24.62 (-22.79)	12.58 (09.80)	09.40 (11.07)	07.54 (07.15)
2	**Naive-IS**	-26.03 (-28.96)	08.78 (08.02)	14.75 (15.28)	07.68 (07.22)
1	**Tmin-IS**	-16.70 (-14.76)	07.06 (07.29)	01.84 (05.52)	-00.09 (02.84)
2	**Tmin-IS**	-18.78 (-15.31)	03.70 (00.64)	05.28 (07.62)	00.71 (-00.73)

0.7.2 Signaux de longue durée: Parole

Nous faisons de même avec des signaux de longue durée, nous ne présenterons que les résultats obtenus avec l'algorithme **Alt-EMK**. Les raisons sont les suivantes, les algorithmes d'estimation basés sur le traitement par fenêtre souffre d'une grosse lacune: la fenêtre d'analyse n'est pour le moment pas prise en compte dans l'estimation des paramètres. Ceci a pour effet de rendre très longue la convergence des paramètres de l'algorithme basé sur l'interprétation naïve de la distance et de créer des instabilités pour la vraie minimisation de la distance. En ce qui concerne l'estimation conjointe et adaptative des paramètres de l'algorithme **Joint-EMK** plusieurs auteurs ont déjà discuté l'instabilité des filtres obtenus, liée à l'estimation conjointe des paramètres [69, 78]. Pour toutes ces raisons les simulations suivantes ne seront effectuées qu'avec l'algorithme **Alt-EMK**. Comme dit précédement, cet algorithme a besoin de connaître les périodes des sources, dans les simulations nous lui donnerons les périodes estimées sur les sources individuellement mais elles ne seront pas nécessairement parfaites, les autres paramètres seront initialisés à l'accoutumé et devront s'adapter.

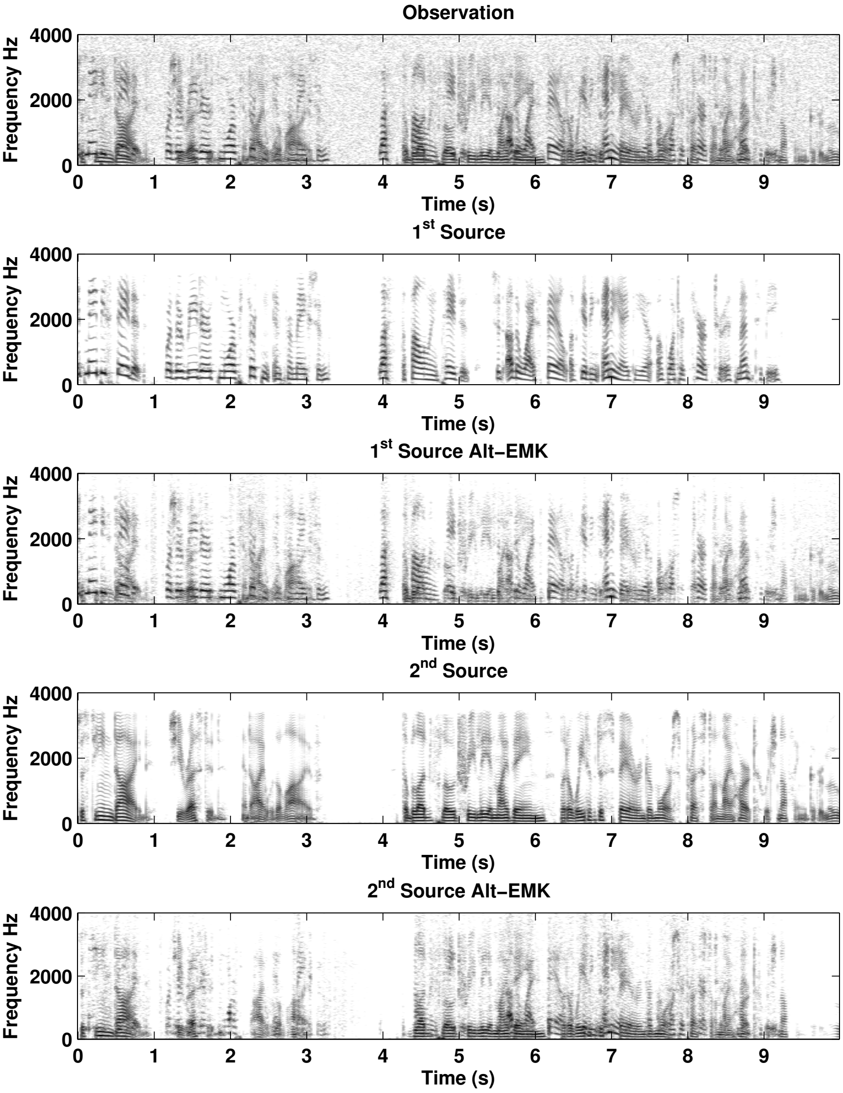


Figure 8: Transformée de Fourier à court terme, résultat du **Alt-EMK**.

Les résultats des critères d'évaluation sont donnés dans la Table 5 et on peut "voir" les résultats de la séparation dans la représentation temps fréquence dans la Figure 8. Les résultats sont plutôt intéressant car à chaques instant l'algorithme cherche deux sources, même lorsqu'une seule source est présente, on peut observer qu'il y a séparation, certes elle n'est pas parfaite, mais les sources sont suivies. Autour de 4 secondes on peut nettement voir que seule la première source est présente, et que seule la première source estimée l'est aussi. Ce suivi de l'amplitude est intimement lié à la périodicité de cette source, comme l'algorithme est renseigné sur cette période il ne l'attribue qu'à la première source estimée.

Table 5: MSE, SIR, SAR et SDR pour des signaux de longue durée.

Algorithme	MSE	SIR	SAR	SDR
Kalman	-29.1 — -29.2	16.9 — 15.1	21.3 — 20.8	14.6 — 12.9
Wiener	-32.2 — -32.3	19.2 — 17.7	22.7 — 21.7	16.9 — 15.4
Alt-EMK	-33.6 — -33.5	12.6 — 15.1	19.2 — 18.6	12.9 — 10.9

0.7.3 Signaux de longue durée: Musique

Nous faisons de même avec des instruments de musique, le mélange est cette fois ci constitué d'une guitare et d'un violoncel, le RSB global est de 20dB, la séparation est illustrée sur la Figure 9. Les critères d'évaluation moyenné sont (violoncel/guitare) 8.8 dB et 9.4 dB pour le SDR, 13.9 dB et 13.4 db pour le SAR, 12.8 dB et 12.3 dB pour le SIR et finalement -27.1 et -27.2 pour le MSE. Au point de vue critère d'évaluation ces résultats peuvent paraître plus mauvais que pour la parole, cependant à l'écoute (qui, en audio reste la seule juge) les résultats sont clairement meilleurs. Le violoncel extrait est, à l'écoute, parfaitement extrait et non pollué par la guitare mais est en réalité sous estimé. Une partie du violoncel est en réalité absorbé par la guitare, il s'agit des zones les moins stationnaires: les zones transitoires contenant les bruits de frottement de l'archet. Notre modèle ne peut pas modèliser de tel comportement, encore moins de manière adaptative, enfin il faut noter que la pièce de Paganini n'est pas non plus une pièce simple à analyser et que les fréquences fondamentales présentes dans la pièce jouée peuvent varier très rapidement. Si nous avions utilisé un traitement par fenêtre nous aurions pu adapter la taille de la fenêtre pour les zones transitoires comme il est fait dans [79].

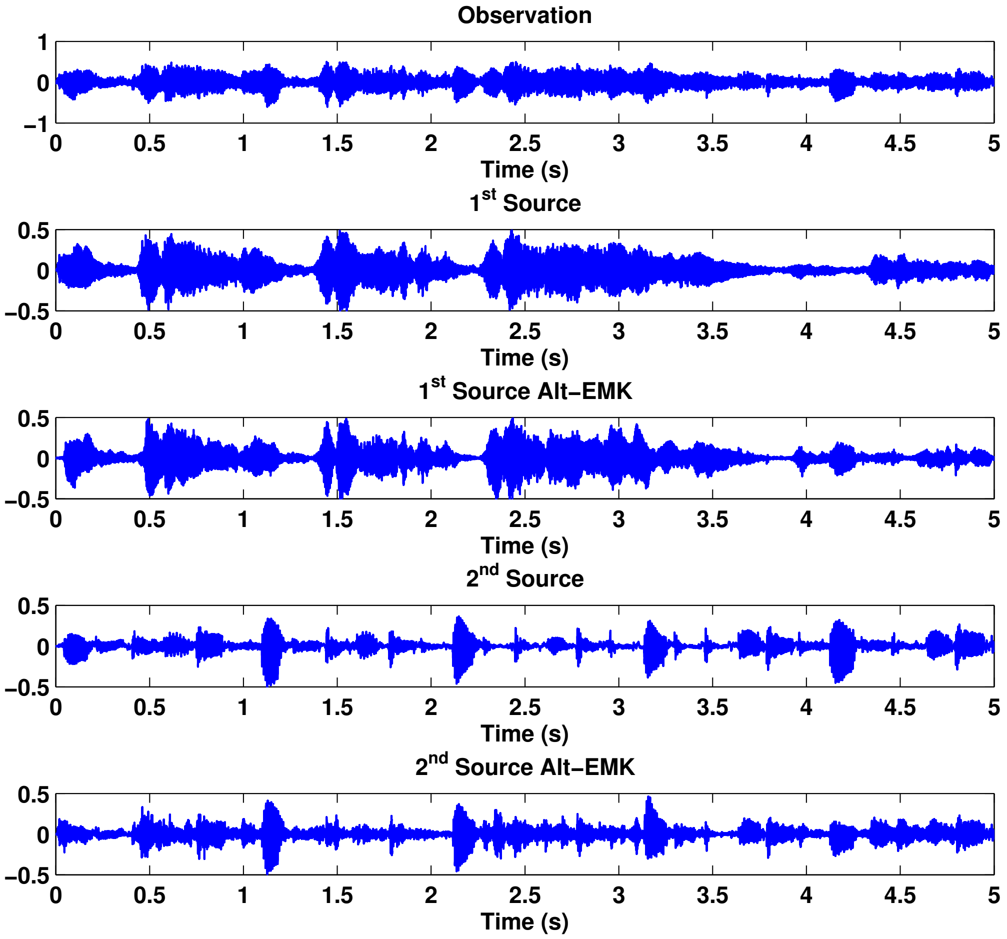


Figure 9: L'observation, les sources et leurs estimés obtenues avec **Alt-EMK**.

0.7.4 Discussion des performances

Dans le travail exposé dans cette thèse nous nous intéressons à la séparation de source mono microphone pour des signaux audio, pour évaluer les performances nous utilisons des critères d'évaluation qui sont devenus standards dans la communauté. Ces critères ont été introduits dans [80] où l'on peut également trouver le code d'origine [81] ainsi les résultats obtenus le sont de la même manière que par d'autres auteurs. Ces autres travaux se placent généralement dans des contextes différents et n'analysent pas nécessairement que des sources de même nature. Dans cette section nous voulons essayer de comparer nos résultats avec ceux existants, bien que la comparaison soit difficile puisque nous n'avons pas utilisé les mêmes signaux ni les mêmes conditions de travail. La plupart des travaux s'interessent à la séparation chant musique qui trouve une application directe pour les applications de Karaoke par exemple, dans ces méthodes les modèles sont généralement appris avec des données d'entrainement et s'attachent à séparer des sources de nature différente. Le modèle que nous avons choisi est essentiellement basé sur la différence de périodicité des sources, si aucune périodicité n'est clairement présente notre modèle ne pourra distinguer les sources, cependant si les sources sont de même nature mais avec des périodicités différentes le modèle sera capable de le faire. Dans la Table 6 nous montrons les résultats obtenus par différents travaux, nous y indiquons la nature de la tâche à accomplir et les méthodes utilisées. Ces résultats, et méthodes, proviennent des travaux suivants:
MMG/MMGA methods [18, 42, 45, 82, 83].
MMG/MMC methods [44, 84].
AR/MMG/Facteur d'amplitude [83].
Nous nommons les méthodes par le modèle utilisé et les résultats des critères d'évaluation obtenus dans la Table 6. Ces méthodes sont également appliquées de manière différente, par exemple le nombre d'état caché/modèle peut varier etc. nous donnons donc les gammes de résultats obtenus.

Table 6: Critères en dB.

Contexte	Algorithme	objet	SDR	SAR	SIR
Musique Jazz	MMG [82]	musique	3.8 → 4.2	7.7→ 8.6	6.1 → 7.4
Vs Chant		chant	-2.3 → -1.9	-1.7 → 0.1	5.1 → 10.1
Piano Vs	MMG [45]	piano	**X**	2→ 8.9	10.5 → 20
Basse et Batterie		batterie	**X**	4→ 9.6	2.8 →18.8
Chant Vs	MMG/AR [83]	voice	3.9 → 5.4	2 → 4.7	2.2 → 4.6
Musique	/Facteur d'Amp.	musique	2.1 → 3	3.2 → 12.9	5 → 11.1
Piano	MMG/MMC [44]	piano	**X**	4.2 → 8.9	10 → 35.7
Vs Batterie		Batterie	**X**	9.7 → 12.5	4.9 → 5.9
Voice Vs	Factorial [84]	voice	0.4 → 5.7	**X**	**X**
Musique	MMC	musique	9.6 → 14.9	**X**	**X**
Inst Prcpl	Modèle hybride [18]	Inst.	1.6 → 8.2	4.1 → 8.9	5.4 → 17.1
Vs Accpgmnt		Acc.	7.7 → 10.1	10.7 → 12.9	14.1 → 15.4
Chnt Vs Msqu	MMG [42]	voice	NSDR 5 → 13		

Le dernier travail exposé utilise un autre critère qui est le SDR normalisé, il s'agit du

SDR source/source estimée moins le SDR observation/source. La comparaison n'est ici qu'informative car les autres travaux ne bruitent pas les mélanges traités, nous obtenons des résidus du bruit dans nos estimés. Dans la Table 7 nous résumons les résultats que nous avons obtenus, à la différence de la Table précédente, nous n'avons pas utilisé différentes formes de notre modèle, les résultats indiqués représentent les valeurs minimales et maximales obtenues lors de l'analyse de signaux de longue durée. Sur ces signaux le RSB variant beaucoup, la qualité des résultats le sont aussi, de plus nous avons traité essentiellement le problème parole/parole ou musique/musique dans ces conditions, la gamme de fréquences fondamentales se recouvre et nous avons déjà souligné l'importance des périodes de notre modèle, si les signaux ont des périodes qui se croisent ou qui se recouvrent, notre séparation ne sera pas nécessairement bonne.

Table 7: Critère en dB.Algorithmes proposés

Contexte	Algorithme	SDR	SAR	SIR	NSDR
Courte Durée					
Parole Vs Parole	**Joint-EMK**	4.3 → 11.9	11.8→ 21.6	5.4 → 12.6	3.3 → 9.4
Parole Vs Parole	**Alt-EMK**	4.3 → 11.9	12.6→ 21.9	6.3 → 13.7	3.7 → 10.5
Parole Vs Parole	**Naive-IS**	7.1 → 7.7	9.4 → 15.2	8 → 12.5	4 → 7.6
Longue Durée					
Parole Vs Parole	**Alt-EMK**	4.9 → 20.7	11→ 29	8.6 → 30	3.7 → 27
Guitare Vs Violoncel	**Alt-EMK**	3.1 → 18	5→ 21.2	3.6 → 21.4	2.2 → 17.7

0.7.5 Extraction du bruit de fond

Le Vuvuzela est très utilisé en Afrique du Sud pendant les matches de football, la pression acoustique de ces instruments est très forte et le son peut être très inconfortable pour les personnes qui ne sont pas habituées, pouvant amener à des pertes de capacité auditive. Pendant la coupe du monde 2010 ils ont été présents lors de tous les matches. De nombreuse approches ont été proposées pour le supprimer des enregistrements, le vuvuzela possède un son fixe dont la fréquence fondamentale est d'environ 230 Hz [85], des méthodes consistant à filtrer cette fréquence et ses harmoniques ont donc été proposées cependant elles réduisent également les commentaires ce qui n'est pas désiré. Pour traiter ce problème nous utiliserons notre Algorithme basé sur la méthodologie EM-Kalman, toute fois nous y apportons certaines modifications. Comme précédemment indiqué nous avons accès à la solution proposée par Audionamix, nous pouvons par soustraction avoir un estimé du vuvuzela. Nous avons donc apporté les modifications suivantes à l'algorithmes, nous considérons le vuvuzela comme un signal CT+LT dont nous connaissons les paramètres pour cette source connue nous n'estimerons pas les paramètres et nous effectuerons un filtrage. Le fond, nous le considèrerons comme étant un bruit coloré, c'est à dire un AR d'ordre élevé, nous avons pris un ordre 50, que nous devrons estimer et adapter, cette seconde source sera alors estimée alors que la première sera filtrée. Il en résulte que seul un modèle AR, sans partie LT, doit être estimé donc que les algorithmes **Joint-EMK** et **Alt-EMK** sont dans ce cas les même. Au son analysé nous avons ajouté un bruit additif menant à un RSB Global de 30 dB, mais non constant. Le résultat dans le plan temps fréquence est montré dans la Figure 10, les critères d'évaluation estimés sont 14.15 dB pour le SDR, un SIR de 17.75 dB, un SAR de 16.7 dB et un MSE de -12.41 dB en prenant comme solution celle proposée par Audionamix.

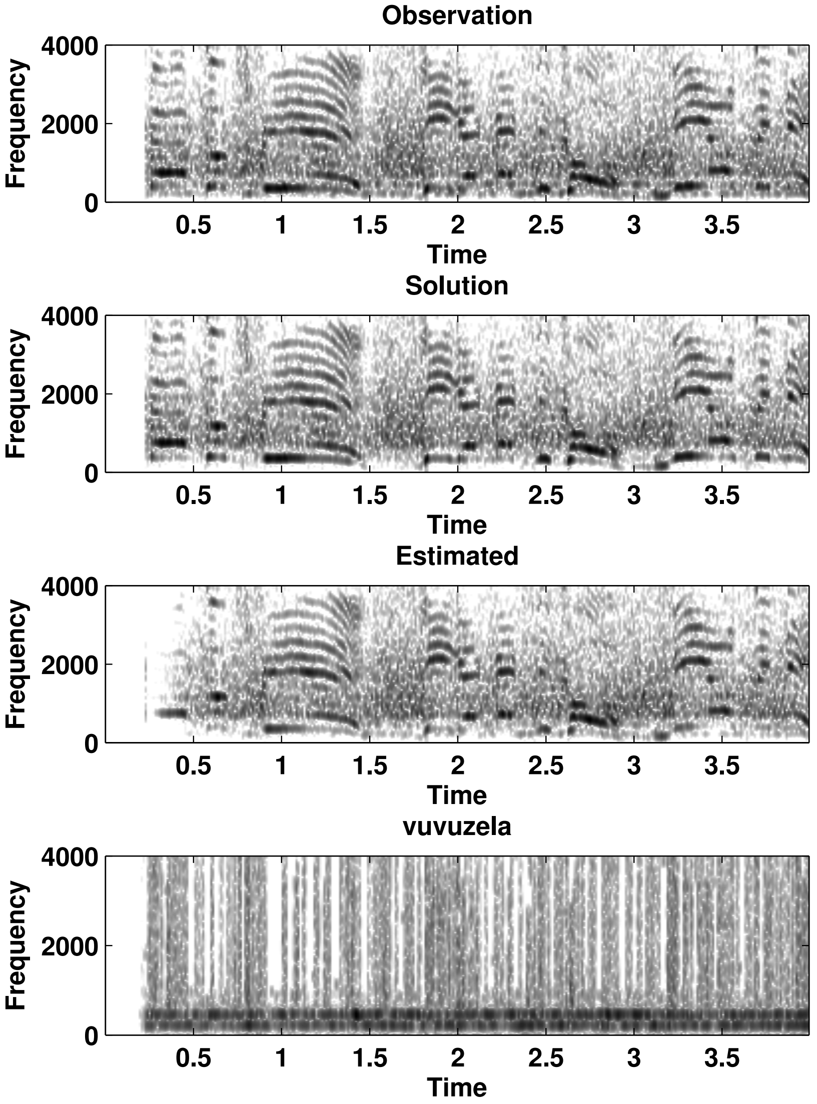


Figure 10: Exemple de suppression du Vuvuzela.

0.7.5.1 paramètres utilisés

Dans les simulations précédentes nous montrons des magnitudes de transformée de Fourier à court terme (TFCT voir l'Annexe F.3) en représentation logarithmique, les paramètres utilisés pour calculer ces représentations sont les suivants: tout d'abord les signaux sont découpés en trame de 1024 échantillons, chaque trame est pondérée par une fenêtre de Hann (voir l'Annexe F.5) avec un recouvrement de 75%. Les tranformées de Fourier sont calculées avec 4096 points. Les fenêtres utilisées pour les pondérations en amplitude de signaux synthétique sont montrées dans l'annexe F.10.2.1. Les critères d'évalation SDR/SIR/SAR sont décrits dans [74] et sont donnés en Annexe F.4. Une comparaison entre les algorithmes **Alt-EMK** et **Joint-EMK** pour l'analyse de signaux réels de longue durée est donnée en Annexe F.8 et montre que le **Joint-EMK** ne s'adapte pas bien, les valeurs de facteurs d'oublie utilisés sont pour le **Alt-EMK** 0.8 pour le long terme et 0.98 pour le court terme. Les signaux de courtes durées ont été extraits dans les signaux de longue durée, il s'agit des premiers 4096 échantillons, la première fois que les sources sont actives, de ces deux fichiers.

0.8 Conclusions

Ce résumé en français résume la première partie de la thèse, la deuxième partie de la thèse n'étant que des annexes elle n'est pas présentée dans ce résumé. Cependant dans cette partie nous avons traité le problème de séparation aveugle de source audio mono microphone. Nous avons proposé deux algorithmes dans le domaine temporel et deux dans le domaine fréquentiel. Nous avons toujours utilisé le même modèle, à savoir le modèle source filtre non contraint. Dans ce modèle, une source est définie comme étant la cascade de deux processus AR, de longueur de corrélation très diffèrentes, un de grand ordre l'autre de petit ordre. Le modèle de faible ordre appelé modèle court terme (CT) correspond à l'enveloppe spectral du signal. L'autre d'ordre plus élevé est en réalité un filtre en peigne, qui peut être vu comme un AR très parcimonieux et est traité comme tel. Ce filtre d'ordre élevé représente les corrélations à long terme du signal et modélise la périodicité, sa force peut être ajustée en modifiant le coefficient long terme (LT) pouvant le rendre nul si nécessaire. Dans sa formulation le modèle n'est pas restreint à des périodes entières grâce à l'utilisation d'une interpolation linéaire. Le mélange est une somme de ces sources AR Gaussiennes plus un bruit blanc Gaussien.

Dans le domaine temporel, les algorithmes proposés suivent une méthodologie de type EM-Kalman et deux versions, en suivant le modèle utilisé, sont proposés. L'un d'entre eux effectue une estimation alternative des paramètres alors que le deuxième les estime conjointement, dans les deux cas les sources sont extraites conjointement. La différence principale entre ces deux algorithmes est l'utilisation de facteur d'oubli différent pour les parties CT et LT, puisque ces deux aspects ne changent pas nécessairement à la même vitesse, ce point est très important et se reflette dans les simulations sur des signaux réels, seule l'estimation alternative est pour le moment capable de séparer les sources. Nous avons néanmoins comparé ces deux algorithmes à ceux obtenus en ne prenant en compte qu'une seule partie du modèle et sommes restés sur ceux que nous avons proposés. Le défaut principal de ces algorithmes est aussi ce qui en fait leur force: la prise en compte de la périodicité, car elle nécessite d'estimer les périodes. Ceci n'étant pas pour le moment inclus dans nos algorithmes, un estimateur de périodes multiples doit tourner en parralll̀e, et bien que les périodes soient le paramètre de notre modèle le plus facile à estimer cette tâche n'en reste pas moins hardue. Les autres points qui ne sont pour le moment pas traités sont l'estimation de l'ordre CT des sources ainsi que le nombre de sources présentes.
Dans le domaine fréquentiel nous avons proposé un algorithme de séparation de source qui est un filtre de Wiener où l'on exprime la prise en compte de la fenêtre correctement, cette fenêtre d'analyse est, dans le domaine spectrale, limité à son lobe principale. Nous avons également décrit les opérations de filtrage par des matrices circulantes, ces deux approximations nous amènent à de grandes simplifications de calcul tout en prenant en compte la fenêtre d'analyse. Nous avons proposé deux algorithmes d'estimation des paramètres basés sur la distance d'Itakura Saito IS, le premier algorithme est un interprétation naïve de cette distance et amène à un algorithme itératif basé sur de la prédiction linéaire directement sur l'observation, plus précisement sur un estimé de la source en question, cet algorithme est néanmoins capable d'estimer tous les paramètres. Le second algorithme traite la véritable minimisation de la distance vis à vis des paramètres et n'amène pas à de la prédiction linéaire classique, cet algorithme amène à des résultats frolant la perfection lorsque l'on travaille avec des données synthétique et à des connections avec le maximum de vraissemblance Gaussien et la correspondance de spectre par moindres carrés pondérés.

La dernière partie de la thèse est consacrée aux simulations, tous les algorithmes y sont comparés sur des données réelles dans le cadre de la séparation de source mono microphone. Une comparaison impossible avec l'état de l'art est aussi faite, impossible dans le sens que les algorithmes n'ont pas été comparés avec les mêmes signaux et les même conditions, cependant en restant prudent sur la comparaison des résultats nos algorithmes semblent se situer dans l'état de l'art. Nous avons aussi testé l'utilisation d'un de nos algorithmes, modifié pour l'experience, dans une application consistant à extraire le fond d'un enregistrement provenant de la coupe du monde 2010 pollué par un ensemble de vuvuzela, le but étant de tout extraire sauf le vuvuzela.

0.9 Perspectives

Lorsque l'on traite un sujet aussi spécifique que la séparation de la parole et/ou de la musique il pourrait paraître naturel d'envisager d'inclure des modèles linguistiques ou musicologiques. Comme une gamme de fréquence fondamental fixe (comme pour certains instruments de musique), de suivre des lois d'évolution temporelle des amplitudes des signaux etc. Cependant ces connaissances supplémentaires impliqueraient d'être ajustées amenant à estimer encore plus de paramètres.

Pour les algorithmes temporels basés sur la théorie du filtre de Kalman beaucoup de contributions peuvent être envisagées. Une relaxation robuste des facteurs d'oubli quand les sources changent trop rapidement, l'utilisation de facteurs d'oubli dépendant des sources pourrait aider lorsqu'une source est moins stationnaire qu'une autre. La conjugaison de plusieurs algorithmes pourrait être envisagé comme créer un système, toujours de type EM-Kalman, spécialisé dans le suivi des périodes et l'optimisation des facteurs d'oubli en entrée de notre système pourrait être très intéressant.

Les algorithmes développpés pour un traitement par fenêtre pourraient eux aussi être améliorés. Premièrement dans la communauté les tranformées de Fourier sont l'outil dominant cependant il existe de nombreuses représentations temps fréquence comme la transformée de Wigner Ville. Il a été mis en exergue que la taille des fenêtres d'analyse est cruciale, cependant rien n'impose d'utiliser une taille fixe pour l'analyse, il faut néanmoins noter qu'une longueur optimale serait dure à trouver car le mélange étudié est composé de plusieurs sources et que les longeurs optimales ne seraient pas les même pour les différentes sources. L'algorithme basé sur la minimisation de la distance IS doit être finalisé, dans son état actuel lorsque la partie long terme est mal estimée, nous pouvons estimer des variances négatives, ceci pourrait être amélioré en imposant une contrainte de non négativité dans leur estimation, de même l'estimation de la partie long terme pourrait prendre en compte le fait que cette partie du modèle est très parcimonieuse dans le domaine temporel.

Pour les deux types d'algorithmes (spectral et temporel) nous n'avons utilisé aucun pré traitement et les deux algorithmes souffrent d'un manque d'information lié aux sources. L'utilisation de modèle psychoacoustique pour prédire quelle information est cachée par les composantes les plus intenses pourrait amenér à des résultats plus agréables pour l'oreille. Pour éviter les problèmes de sur-estimation, estimer le nombre de sources présente serait un grand avantage car lorsque le nombre de source est sous estimé la/les source(s) restantes sont répartis dans les souces éstimées.

Part I

Mono-Microphone Blind Audio Source Separation

Chapter 1

Introduction

Imagine that you call a friend on a Saturday evening. You dial the number, the phone rings and your friend answers. He is out in a nightclub with his friends, the music is loud and you can hear that people are talking and dancing all around him. Despite the fact that he speaks at the top of his voice for you to hear him, you cannot get a word of what he is saying. The reception gets saturated from the noise and the loud sound of his voice. You explain it to him in order to have him repeat what he said more quietly. Even though there is no saturation this time, you still cannot quite understand everything because of the music, the people around and the interferences of the signal.

This kind of situation illustrates the context of the work done in this part of the thesis. We focus on the problem of blind audio source separation with a single microphone.

This introductive chapter is organized as follow: section 1.1 is a reminder of some general definitions of the Source Separation Problem as well as a summary of a majority of the possible cases. Section 1.2 gives an idea of the possible application in which source separation is needed/used to solve problems. Section 1.3 is a reminder of the principal solutions applied to each case. Section 1.5 is dedicated to the Mono-Microphone case.

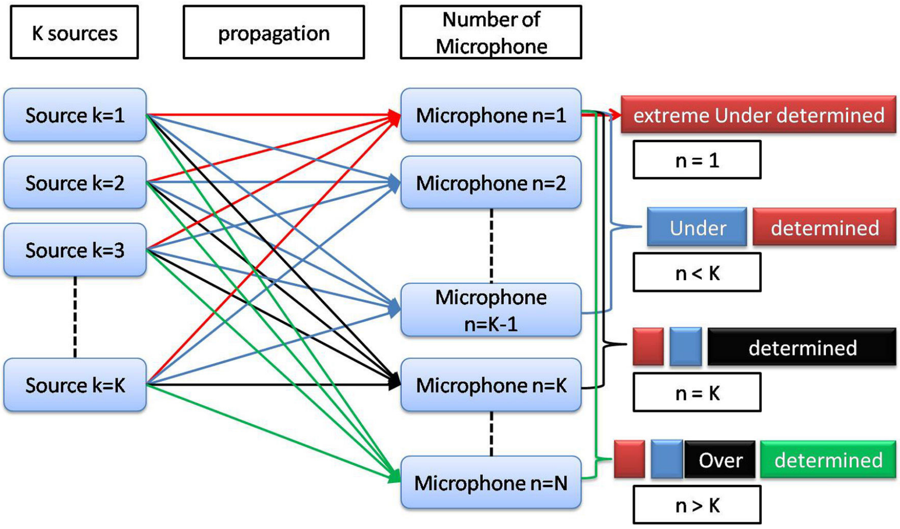

K sources
propagation
Number of Microphone
Source k=1
Source k=2
Source k=3
Source k=K
Microphone n=1
Microphone n=2
Microphone n=K-1
Microphone n=K
Microphone n=N
extreme Under determined
n = 1
Under
determined
n < K
determined
n = K
Over
determined
n > K

Figure 1.1: Determination of the problem

1.1 Blind Source Separation

1.1.1 Introduction

Blind Source Separation (BSS) aims at estimating a set of source signals from a mixture of these same signals. The separation is called blind because we don't know the mixture parameters and our source estimation is based on observations only. Presently, BSS is very up-to-date in signal processing because of the great number of applications in speech processing, image processing, telecommunications, biomedical engineering or astrophysics [12].

1.1.2 Determination

The number of observations depends on the number of sensors (Microphone). The difference between the number of sources and microphones defines the problem's determination. Figure 1.1 shows the different cases, referenced as:

- Extreme Under determined case, if only one observation is available
- Under determined, when more sources than microphones are present
- Determined case, when there is an equal number of observations and microphones
- Over determined case, when we have more observations than microphones

The determination of the problem depends also on the approach considered (2^{nd} order, High Order Statistic (HOS)) [86, 87] and will be discussed later. In this thesis we are considering the Mono-Microphone (Extreme Under determined) Blind Audio Source Separation (BASS) problem with a parametric approach.

1.1.3 Effect of the propagation

At the exception of the classification between (over) determined and underdetermined mixtures, all sources are affected by their process of propagation. Different mixture models exist depending on different scenarios:

- Linear instantaneous mixtures are the simplest. They consider that each observation is a sum of scaled versions of the source signals. If we enter all the mixture coefficients in a matrix called mixture matrix, we deal with a problem of identification concerning the inverse of this matrix up to a diagonal matrix and a permutation matrix.
- Scaled and delayed mixtures are met when the source contributions are some scaled and delayed versions of the sources. The sources do not arrive at the same moment on the microphones and in each mixture the delay of the source is different.
- Convolutive mixtures are the most general linear mixtures but also the most complicated ones to separate. In this case the source contributions are some filtered versions of the original sources. Generally speaking, this is the case when a source reaches a microphone by passing through a multiple path.
- Non-linear mixtures have been studied only on a few occasions. This case is difficult to solve and presently only some cases have been investigated as the Post-Non-Linear case. For instance, investigations include cases in which sensors are used with hard nonlinearities or signal levels lead to a saturation of the conditioning electronic circuits.

1.2 Application

According to the aforementioned definitions, the Blind Source Separation Problem can be defined as a very general problem. This problem happens very often and has a very general definition depending either on the situation. It should not be forgotten that sensors also lead to different definitions of the entities. Amongst the possible applications using this concept, we can find:

1.2.1 Audio Processing

In audio processing, when several audio sources are used, BSS is needed to analyze each source individually. For a meeting room as well as for an audio conference situation, the separation of every single speaker allows the use of some automatic speech-to-text application [13], speech enhancement [14], speaker verification etc. As far as music processing is concerned, BSS is needed for each instrument's automatic transcription [16,17], melody extraction [18] and automatic instrument recognition [15]. Other music applications include old recording restoration [19, 20], voice removal for karaoke [18, 20], re-mixing, hearing aids [12, 21] etc.

1.2.2 Biomedicine

In the context of medical signal and image processing, the BSS problem arises at least in the analysis of Electro-Encephalogram, Magneto-Encephalogram, Electro-Cardiogram signals. Since a various number of electrodes are used in each of these situations to analyze a specific or several impulses, a BSS signal can be applied [88]. For example, if the Electro-Cardiogram of a foetus has to be analyzed, the signal attributed to the foetus will be corrupted by the signal of the mother's heart. The problem turns out to be more complicated in the case of twins if the signal of only one of the two foetus is wanted [89].

1.2.3 Diarization

Speaker diarization is a process that detects the characteristics of the speaker's turns in order to regroup them and recognize the vocal identity of every single speaker . In general, the very first step is an unsupervized segmentation (often preceded by a speech detection phase) that consists in partitioning the regions of speech into segments (each segment must be as Long as possible and ideally, has to contain the speech of one speaker only). It is followed by a clustering step that consists in labeling the various segments uttered by the same speaker. As far as diarization algorithms are concerned, mono-speaker algorithms generally fail when several speakers are present at the time [90]. This problem comes from the fact that the learning is done on the mixture. If several speakers are present a the same time they can be understood as being a new person by the clustering algorithm. However, BASS would improve the result of the clustering. Because, if the separation is well achived, only one speaker is present at the same time. However the original multi-speaker signal several is now several mono-speaker signal, more data have to be treated.

1.2.4 Security

Concerning security applications, BASS can be applied in various scenarios. The separation of the speakers in a highly noisy/ interference environment can help an automatic analyzer to detect some keyword. For suspect tracking, as mentioned previously, BASS would improve the result of the recognition of the speakers and more generally enhance the contents of an audio surveillance separating the main speaker from the noise. Moreover, due to privacy policy (telephone conversations, interrogations,..), anonym recording and analyzing can be done without external intervention.

1.2.5 Telecommunications

In multi-user communication systems, when several mobiles share the same time-frequency code slot, BSS is needed. The signal of interest is corrupted by signals coming from different spatial origins. BSS is also needed in the case of intersymbol interference, that is to say when time-delayed versions of the signal corrupt the signal itself [12]. Generally speaking, the suppression of the intersymbol interference is defined as a deconvolution or a channel equalization problem. Be it assumed that the symbols are statistically independent, the channel equalization can be identified as a BSS problem of independent sources in instantaneous linear mixtures [91].

1.3 Some existing solutions

1.3.1 A brief Chronology

The BSS problem has been first formulated in the mid eighties by Hérault, Ans and Jutten [22–24] in the framework of neural modeling, even though theoretical principles have been understood later. Initially, source separation has been investigated for instantaneous linear mixtures [23]. The generalization to convolutive mixtures has been studied in the early $90s$ [92]. Eventually, nonlinear mixtures have been addressed during the $90s$ [25, 26].

1.3.2 Determined and Over determined BSS

Comon [27,28] introduced the Independent Component Analysis (ICA) in 1991 and numerous theoretical and practical works followed. Numerous algorithms have been developed before ICA but most of them can be included in the ICA method. As mentioned in [12], the idea of the statistical independence of the unknown sources has been introduced by Jutten in 1987 [93]. A basic ICA algorithm assumes a determined BSS case of instantaneous mixing model. Such as ICA, this over determined case has also been analyzed for non linear mixtures [26]. Under the independence assumptions, a demixing matrix has to be found. In order to find such a matrix, the ICA algorithm minimizes statistical dependency between unmixed channels. While the independence of sources can be accepted, their signal mixtures cannot be. The explanation is that each source signal contributes to every mixture, and therefore, the mixtures cannot be independent. Inspired by the information theory, another approach for ICA estimation is the minimization of the mutual information [29]. Mutual information is a natural measure of the dependence between random variables. It is actually equivalent to the Kullback-Leibler divergence between a joint density and the product of its marginal densities. It is always non-negative and zero if the variables are statistically independent. Along the same lines, if a signal at hand is known to be sparse in some bases, then a sum of two sparse signals will be less sparse than its components. Subsequently, the observations are less sparse than the sources in a specific domain. The BSS algorithms using this property look for a matrix that will produce the sparsest signals after demixing [30].

The sparsity is not exclusive to the frequency domain; a signal can be sparse in a given dictionary (possibly overcomplete). The aim of these methods then becomes the estimation of the source coefficients in the dictionary and not the time series themselves [57]. The time series are then reconstructed from the estimated coefficients [31]. The idea is to find, for a given set of observations (Y) and a given dictionary (ϕ), the mixing matrix (A) and the coefficient matrix (C) which lead to $Y = AC\phi$ with the sparsest coefficient matrix [30]. Non-negative Matrix Factorization (NMF) decomposes one matrix into two matrices: one matrix corresponds to the bases and the other corresponds to the weights so that each column in the matrix is a linear combination of the bases. NMF is applied to the magnitude of the Short-time Fourier transform (STFT). In some particular cases where the basis correspond to differents sources the separation is achieved by multiplying the bases and weights that correspond to each source [94]. Figure 1.2 shows a typical decomposition scheme of an audio signal. The basis is composed of independent spectrum. If one of this spectrum is present in the mixture then its power evolution (corresponding also to its activation) over the time is reported in the activation matrix.

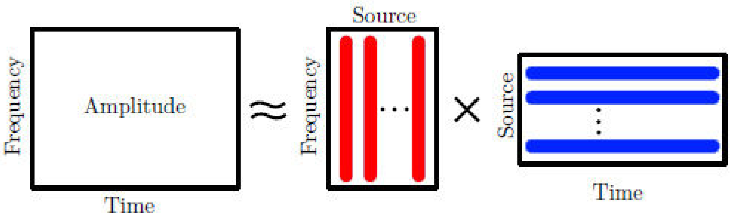


Figure 1.2: Typical NMF decomposition for audio signal. The observation is typically a Time Frequency matrix, such as the magnitude Short Time Fourier Transform. The decomposition finds a set of time-varying sources with constant spectrum.

Pursuing a totaly different idea, when dealing with source separation of audio signals, some algorithms are based on the results acquired in psychoacoustic studies. The perceptual organization of sounds contains various psychoacoustic studies and provides a basis for the computational implementation of algorithms that mimic the behavior of human auditory apparatus. Computational implementations of psychoacoustic rules are known as Computational Auditory Scene Analysis (CASA) [95].

1.3.3 Under determined BSS (UBSS)

The Underdetermined case is very important and also very realistic. It appears that when more sources than sensors are present, the demixing operation becomes difficult. The separation of the underdetermined mixtures requires prior information on the sources to allow their reconstruction [7,8]. The difficulty of the underdetermined setup can be somehow alleviated if there is a representation wherein all the sources are rarely simultaneously active, which entails finding a representation where the sources are sparse.

For example the DUET (Degenerate Unmixing Estimation Technique) algorithm is based on the basic assumption that all the sources have a sparse frequency spectrum for any given time [32, 33]. The way that DUET separates degenerate mixtures is by partitioning the Time Frequency representation. This implies that each time-frequency point in the spectrogram is associated with only one source. DUET assumes that the sources are already separated and then creates a binary mask . This property is called the W-disjoint orthogonality property.

1.3.4 Stereophonic BSS

Stereophonic BSS is included in the under determined case. It comes just before the Extreme Under determined case and an international evaluation campaign is dedicated to this problem [34, 35]. As it is included in the UBSS case, the algorithms mentioned previously are still valid. The stereo case is the last case in which spatial information can be used for the extraction of the source. For stereophonic record, [36, 37] make the assumptions that a source is dominant on one of the two channels. By equalizing the power of the two channels the method can separate the sources. Generally speaking, sources can be separated by applying binary TF masks whose features (such as the Inter-channel Level/Phase Difference (ILD/IPD)) are used to estimate the index of dominant source. Time-frequency masking aims at extracting the time frequency components dominated by target signals. Time-delay and the amplitude ratio between multiple sensory inputs correspond to the direction-of-arrival (DOA) of the signal. By analyzing the directivity patterns formed by a separating matrix, source directions can be estimated and, therefore, permutations can be aligned [38, 39]. When the sources are non-stationary signals, the inter-frequency correlations of the signals' envelopes can be used to align permutations [33, 40]. The TF mask can be built by performing a clustering on these quantities [6, 41] or by fitting some distribution using the Expectation-Maximization (EM) Algorithm [10] in the feature domain [96]. But in the case of an ill-posed problem, the method fails. For instance, it happens in the case in which the mixing filters is very similar to each other, when sources are located close to each other, or even when the DOA of the sources are similar.

1.4 Positioning of this thesis

In this thesis, the Mono-Microphone case of linear instantaneous mixtures or equivalently the scaled and delayed mixtures are considered. As we work with only one observation, we don't take into account the possible delay and/or attenuation. Most of the simulation will be done with two sources even if the algorithms are not specialized. Unlike the approaches mentioned previously, we will work with a parametric model which will force the source estimation process to follow this model. Thus resulting algorithms which are less general than the methods described before.
The next section is devoted to Mono-Microphone approaches.

1.5 Mono-Microphone Blind Audio Source Separation

The extreme case, with only one microphone, is undoubtedly the most challenging one as absolutly no spatial informations about the acoustic field is available. Nevertheless, considering its wide variety of applications, it remains a very realistic case in all days applications.

1.5.1 Problem Formulation

The Mono-Microphone source separation problem can be defined as the estimation of K original source signals: $x_1(t), \cdots, x_K(t)$, given only one observed mixture: $y(t)$. In a general formulation we may write:

$$y(t) = f(x_1(t), ..., x_K(t)) + v(t) \tag{1.1}$$

where $v(t)$ is an additive noise and f refers to the mixing operation and can be non-linear. In our situation of linear instantaneous mixtures, the problem simply becomes:

$$y(t) = \sum_{k}^{K} x_k(t) + v(t) \tag{1.2}$$

Mono-Microphone source separation is an underdetermined problem and its solution requires additional information about the sources. For example, in the case of linear noise free mixing with two sources, $\hat{x}_1(t) = s(t)$ and $\hat{x}_2(t) = y(t) - s(t)$ are the obvious solutions of the problem [42]. This being true for any candidate $s(t)$. With this example, it becomes essential to use additional information about the sources to constrain the problem. Several source models have been suggested, and here is a brief description of the most common ones:

1.5.2 Factorial vector quantization (VQ)

In factorial Vector Quantization, the mixture signal is represented as a sequence of vectors. Separate training data are required for each source in the mixture. The first step in the factorial VQ procedure helps to learn a codebook that consists of code vectors for each isolated source. Several techniques exist to learn the codebook in order to represent each source vector by a single code vector in an optimal way. The inference in a factorial VQ framework consists in finding the best combination of codebook vectors for each source. In [43], Roweis introduces a factorial VQ method to separate audio sources in a log-magnitude spectral representation using the Log-Max approximation: $log(\ a + b\ ) \approx \max\ (\ log(a),\ log(b)\ )$. In [83] three codebook based approaches are compared: Gaussian Scaled Mixture Models (GSMM, as in [45]), amplitude factor models (as in [97]), and Auto-Regressive (AR) models. The authors conclude that the autoregressive models effectively capture speech features, whereas the amplitude factor model is a better solution to separate music signals.

1.5.3 Gaussian Mixture Models (GMM)

The Gaussian Mixture Model (GMM) is a very useful source model. Indeed, each source is modeled as a combination of Multivariate Gaussian densities. The source separation technique, as presented in [44], suggests the use of GMM to model the sources' statistical behavior. They consider a two-source problem based on the learning. The idea is to represent each source as the realization of a random variable represented by a finite set of characteristic spectral shapes. In this approach, each source model is composed of Q states. The state q of the k^{th} source is represented by a spectral shape $(\sigma^2_{q,k}(f))$ and a priori distribution. In this case, the codebook is made according to GMM parameters trained from sample data representative of the sources. The separation process is done by using an "adaptive Wiener" filter because the parameters depend on the observations. When dealing with non-stationary sources, which is generally the case in Audio Processing, consecutive time frames will approximately contain the same Power Spectral Densities (PSD) shape but different amplitudes. As this formulation implies that there will be as many Gaussians as different amplitudes, in [45] Gaussian Scaled Mixture Models (GSMM) are used instead of GMM so that the amplitude can be separated from the PSD. The global system can be found in [42] in which they also adapt the models in case that the training data do not match very well the analyzed data.

1.5.4 Hidden Markov Model (HMM)

A Hidden Markov Model (HMM) can be considered as a generalization of a mixture model where the hidden variables are related through a Markov process rather independent from each other. Roweis [46] discusses the use of a factorial HMM with a GMM observation model. In this approach, a HMM/GMM is learned for each source on isolated sound. He presents a refiltering technique that estimates the weight of each source as a time-varying mask that localizes sound streams in a time-frequency representation. In his work, the sources are supposed to be disjoint in the frequency domain and he also shows that a binary mask, which can separate the sources, exists. Benaroya et al. [44] also investigate the use of HMM. In this work, they present a more advanced technique to estimate the sources based on an adaptive Wiener filter. They conclude that the improvement, based on the used evaluation criteria and compared to the GMM, is not significant.

1.5.5 Non-negative Matrix Factorisation (NMF)

As aforementioned, the principle of Non-negative Matrix Factorisation (NMF) is to approximate a non-negative matrix (e.g.: a Time-Frequency representation of the data as the magnitude of a STFT) with the product of two non-negative matrices (the principle is explained in section 1.3.2). Non-negativity is the only restriction of NMF, actually the independence of the source is not recquired in this framework. The use of NMF to solve (1.2) exploit the sparsity assumption of the source in the spectral domain. If the sources are monophonic (i.e. they have only one fondamental frequency) then the NMF will find which fondamental frequency is active [16] that is, which column of the basis is present on this frame of the STFT. Then the weight (power evolution) of this individual spectrum will be represented on a row (associated to the column of the bases matrix) of the activation matrix. The NMF has become very popular since the apparition of fast algorithms involving multiplicative update. This multiplicative update rules are, in the NMF literature, attributed to Lee and Seung [47] but have been known and used for a Long time in the image processing community [48]. Indeed, they have been used in Astrophysics for example when the base is fully [49] or partialy known [50, 51] (or estimated). NMF has a multitude of applications in audio processing, including feature extraction, music transcription, sound classification, and source separation. NMF involves the minimization of a cost function. The most common are the Euclidean Distance, the Kullback-Leibler (KL) divergence [47, 98] and the Itakura-Saito (IS) Divergence [16, 48, 99]. This distance can be weighted for example by using a psychoacoustical model [100]. In the Mono-Microphone source separation context, NMF has been applied to polyphonic music transcription [16,99] as well as for speech separation [101].

The methods based on VQ, GMM and HMM need learning on training data and are, as NMF, general methods. The definition of the differents learnt entities as well as the definition of the matrices involved in the factorization of NMF depends on the application. For more specialized applications, as it is the case in Audio, parametric models of the sources are also used.

1.5.6 Structured/Parametrized Model Based Approach

1.5.6.1 Sinusoids Plus Noise

An alternate decomposition for the sounds produced by musical instruments and/or speech is the sinusoids plus noise model [52]. It represents the signal as a sum of deterministic (sinusoids) [52] and stochastic parts (noise/residual) [11]. Sinusoidal components are usually harmonic ($f_n = n\ f_0$ with f_0 the fundamental frequency). In the Automatic Polyphonic Music Transcription Task, this model is widely used and is defined in a multipitch context as:

$$y(t) = \sum_{k}^{K} x_k(t) + v(t) \tag{1.3}$$

$$x_k(t) = \sum_{n}^{N_H(k)} a_n\ \cos(2\pi f_n t / F_s + \phi_n) \tag{1.4}$$

where $y(t)$ is the observation at a given time t, $x_k(t)$ the k^{th} source (K is the number of sources). $v(t)$ stands for both the contribution of the residual and additive noise. Each source is a sum of sinusoids with amplitudes (a_n), phases (ϕ_n) and frequencies (f_n in Hz. It is important to emphasize that all parameters may vary with respect to time. F_s is the sampling frequency, $N_H(k)$ the number of sinusoids of the source k and, as aforementioned, a source can be harmonic ($f_n = n\ f_0$). However as this kind of model is generally analyzed over a Short duration frame, the parameters are assumed to be constant and the time dependency refers to a frame. In order to perform a source separation with such a kind of model, all the parameters have to be estimated, and, then assigned to the right source. To achieve it, several methods have been developed:

Serra's original work [11] was done in the spectral domain and the frequencies were found by peak picking in the magnitude spectrum. The complex amplitude, taken from the Fourier transform, gives other parameters. Other works apply interpolation around a local maximum of the Spectrum in order to refine the estimation of the frequency [53–55]. However, these methods are usually confronted to an issue when harmonics of concurrent sounds overlap. This is essentially due to Time-Frequency incertitude (High Resolution method [102, 103] can be used instead of Fourier Transform).

In order to solve the overlapping partials, Klapuri [56] popularized the use of the spectral smoothness principles highlighting the hope that the spectral envelopes of a real sound note tend to be continuous. It consists in weighting the amplitude of a harmonic by using the information related to the other ones with the constraint that the resulting envelope is smooth.

In the Polyphonic case, the choice of the frequency peaks (be it a harmonic or not) is also crucial in order to avoid a peak due to the noise/residual. Yeh [104] puts forward a classification method based on a Harmonic plus Noise Decomposition. Meurisse et al. [105] suggest looking at the time direction of successive STFT frames. They decide that a frequency index is a harmonic (or not) by analyzing the distribution of the amplitude values in successive Time-Frequency frames. Other time frequency representation (e.g. Wigner-Ville) than STFT have been also used in the over determined case [106, 107] and are not explained here.

The source separation algorithms based on the sinusoidal model can be split into three categories at least [65]. Methods first find the sinusoids independently and then group them by source. Strictly speaking, the grouping is a hard task in itself. For example in order to deal with the grouping, Virtanen [108] uses psychoacoustic cues defined by Bregman in [109], while other works use the harmonicity constraint like in [56]. The two other methods consist in estimating the number of sources, their F0s and parameters of sinusoids jointly and iteratively respectively. The joint approach is essentially done in a Bayesian context which gives more flexibility due to the fact that the harmonicity constraint can be somewhat alleviated [110]. Goto [111] separates the bass line and the melody line from a polyphonic recording. First a Time Frequency representation is computed, each columns (time index) are then normalized leading to a probability density's interpretation. An Expectation-Maximization algorithm is used to adjust the weight of each Gaussian which are centered on the fundamental frequency multiple. The set of Gaussian defines a harmonic spectrum.

On the one hand, iterative approaches, compared to a joint estimation of the parameters, lose some flexibility because of the harmonic constraint. But on the other hand, they are usually easier to implement and are typically faster and more robust. Based on this approach, other methods have been introduced by Virtanen or Klapuri [112, 113] They estimated the parameters by minimizing the energy of the residual.

Another popular approach is based on the atomic decomposition of a signal. It is called Matching Pursuit (MP) [57]. Leveau in [15] uses instruments specific atoms. An atome refers to the note of an instrument, it takes into account the spectral enveloppe for a certain frequency, this allow to jointly find the instruments present in the sound and its fondamental frequency evolution. Triki [58] suggests a Quasi-Periodic Signal Extraction (QPSE) in order to perform sources separation. This method sets up a signal as a periodic signal with a (slow) global variation of amplitude (reflecting attack, sustain, decay) and frequency (limited time warping). The author observes that the QPSE and MP approaches have comparable enhancement performances. However, the QPSE approach outperforms the MP [57] and the Harmonic MP [114] in the steady-state region (i.e. where the quasi-periodic model allows a better fit of the audio signal). However, the MP is better in the transition region, where the structure of QPSE is too constrained.

1.5.6.2 AR/ARMA-based source separation

Several authors adress the same kind of modelization by using AR model, ARMA Model and Comb Filter. The frequency response of a comb filter has peaks at integer multiples of the frequency corresponding to the period of delay. When the delay is tuned according to the fundamental frequency, subtracting delayed versions of the input signal result in the cancellation of the fundamental frequency and its harmonics. In [59] this fact is used for separating concurrent harmonic sound. A normal discrete-time implementation of the delay restricts the fundamental frequencies to quantized values of the period, but arbitrary fundamental frequencies can be modeled by using a fractional delay filter [60].

Emiya [17] models a note with an ARMA model. The AR model is used to modeling the spectral envelope of the harmonics (he also uses an inharmonicity law) while the Moving Average (MA) represents the colored noise. The multipitch estimation is performed through a Maximum Likelihood approach and the resulting criterion is based on the spectral flatness (or whiteness). The method has been applied on a polyphonic piano sound (inharmonic sound) and turned out to give good results.

A simple model that can capture temporal correlations in the sources is an autoregressive (AR) model. In [61] Carpentier and al. use a Kalman-Filter in order to reconstruct the source while the AR sources parameters are estimated via a Maximum Likelihood estimation. The states are the source and the observations are the incoming signal mixtures. The authors came to the conclusion that it is actually possible to separate Gaussian AR sources when the spectral contents are disjoint, which correspond to an oversimplified problem. Likewise,this is applicable in an underdetermined context. Couvreur [70] also advises a Kalman approach but in the context of AR coefficients identification from a codebook. In both cases, the analysis is done with an AR model (AR(2)) of Short order.

All the approach mentioned previously are used in a frame based context or with a Short duration signal, which is related to stationary signal. We will propose in chapter 3 an adaptive method which normally allows to deal with infinite length segment duration and non stationary signals.

Additionally, Balan et al. [115] show, also, that for a single-channel mixture of stationary AR sources, the parameters of the AR processes can be uniquely identified and the sources separated. When dealing with non-stationary sources the identification problem is more difficult. For separating slowly changing non-stationary AR sources, the authors propose to first identify the constituent AR processes for the initial N samples in the signal, and use an adaptive sliding-window method to update the AR processes for each new sample.

1.6 Model Considerered in this work

Regarding the approach studied in this section, the main source of identifiability comes from exploiting the presumed quasi-periodic nature of sources. As highlighted by several studies, the spectrum of speech or musical signal can be efficiently modeled by a harmonic sum of peaks and a spectral shape. The comb filter, for example in [59], will generate peaks but with constant amplitude. Yet, it will not match the spectral envelope of a real sound. Contrarily, a Short order auto-regressive model will not match the peaks, or if it does, it will be due to the fact that the analyzed sound contains a little number of peaks. In both cases, the matching will not be good. In our approach, we want to distinguish these two aspects. We will model a sound with a sum of two imbricated autoregressive processes, with very different correlation lengths, and an additive white noise. We consider that speakers talk on different frequencies. Each source is assumed to have a quasi-periodic nature that makes its presence identifiable. Quasiperiodic means that a source is not exactly periodic but that the signal is almost the same in consecutive periods. The easiest way to model such small variations is with a stochastic signal model, the simplest one being (zero mean) Gaussian signal. A Short plus Long-term autoregressive (AR) signal model has been used in this case as it is frequently used in speech encoding algorithms like CELP and LPC [9]. This model has proven its robustness in speech coding and is simple to formulate. Finally, a source is a white Gaussian Noise filtered by two AR filters and the mixture will be a sum of Gaussian sources. The Long term AR part allows the modeling of the source's quasi-periodic nature, and is in fact a AR model of high order with only one (or two for fractional delay) non zero-coefficient(s), equivalent to a comb filter in its feedback form. The Short term AR allows the modeling of the spectral envelope, and it also refers to the notion of formant. Formant contains the information that humans require to distinguish vowels. As humans are able to pronounce the same vowels at different frequencies, the harmonic peaks are weighted by the formant. Bearing this consideration in mind, it becomes reasonable to firstly, distinguish the two aspects (Long ans Short term) and secondly, use the peaks' information in order to estimate the spectral shape and conversely.

Sumarizing we propose to derive a CELP like approach, although for CELP model, some learning dictionnaries are introduced. This is not the case in the proposed method.

1.7 Relationship with another approach

The model used in this thesis is not fully original. As aforementioned, this model is a major focus in speech coding and it is also related to all sinusoidal plus noise modeling. Representing a song with a sum of harmonic peaks modulated by an envelope is also the goal of most of the approaches mentioned previously. The hybrid model used by Durrieu [18] has the same kind of consideration as the one we use. The principal objective of his work concerns the extraction of the main melody in a polyphonic recording. According to this, he defines a model in which the model of the source of interest (essentially a monophonic melody line) is different to the one used to model the accompaniment, and, are assumed to be independent. His work is related to the work in section 1.5.3, by [42,45], the accompaniment model is the same. Concerning the melody, the representation is done via a source/filter model, aiming at matching the two aspects (pulse train frequency and the spectral shape) of music/speech. The main difference with our approach comes from the use of predefined dictionaries for one of the two aspects (namely the Long term).

1.8 Proposed Approach

For the Mono-Microphone audio source separation problem we will mainly investigate two algorithms:

The first one is based on an adaptive EM Kalman filter. At first sight, this approach can appear to be similar to some of the aforementioned works [61, 70] but actually, it turns out to be truly different in many aspects. In our approach, the model is more complete, the sources are jointly extracted and the parameters are adaptively estimated without the use of a predefined codebook. Our main contribution relies on the use of the State Space Model (SSM). In order to sum up the algorithm, the Expectation-step (E-Step) uses a Kalman Filter to extract jointly the source states and their error covariances matrices. The Maximization-step (M-Step) aims at estimating the parameters. The form of the SSM we used is called an extended SSM because it leads to Kalman Fixed lag Smoothing instead of purely filtering. Present simulations reveal some problems because of the periods' variations over the time, some relaxation have to be done but we presently analyze Short segment by Short segment with fixed periods.

The second algorithm is based on successive frames processing. We are currently considering two steps, a parameter estimation step and a separation step. In the veins of the Expectation-Maximization (EM) based algorithm, our research differentiates itself on two points. EM based algorithms iterate between the source estimation and the parameters estimation, and the parameters estimation step explicitly needs the estimated source. In our approach, the parameters are estimated in the mixture and only the covariance of the sources will be extracted but not the sources themselves. Then the separation procedure, with these parameters, is not iterative and looks like a Wiener Filter. For both algorithms an initialization step is needed, at least for the periods. *Throughout our work, AR model order estimation together the number of sources estimation are not studied here. These quantities are assumed to be known.* Temporal correlations between frames are indirectly taken into account. Indeed, from one frame to the next, the parameters are exported as initialization values for the next frame thus leading to a kind of sources tracking. Our contribution is to provide two algorithms for parameter estimation based on the Itakura-Saito (IS) Distance and a source separation algorithm. The first algorithm is a naive interpretation of IS distance. It leads to an iterative algorithm in which all the parameters are estimated using basic Linear Prediction (LP) on sources covariances extracted from the mixture. The second algorithm, is based on the true minimization of the IS distance and leads to an iterative algorithm in which the parameters are estimated by actually solving the Yule-Walker normal equation with a non zero Right Hand Side. This leads to better results. Using a weighted form of IS, we come out with a solution to initialize the different parameters. The source separation algorithm uses some approximations and was originally proposed in a Variational Bayesian (VB) context which is now reduced to a Maximum a Posteriori (MAP) context. One of the simplifications we use is due to the approximation of filtering operation by circulant matrices. We an adapted analysis window for the spectral implementation. Thus, the sources we extract are naturally windowed. The methods are illustrated with some separation results.

1.9 Summary of chapter 1

In this chapter we have introduced the Blind Source Separation problem (BSS) and its different representations in section 1.1. The determination of the problem is related to the difference between number of sources and the number of microphones, while the mixing process is related to the propagation of the sources. In section 1.3, a brief review of the literature concerning the (Over) Determined and Under Determined cases has been presented. Some of the techniques used for this multi-sensors source separation can be used for the mono-microphone BSS. However, for most of the techniques, it is impossible to adapt these latter approaches methods. In the work presented in this part of the thesis, we focus on the Blind Mono-Microphone source separation problem for audio signals. This extreme case needs assumption on the sources; the term Semi-Blind will be more appropriate as we constrain the sources to follow a prior model. Several techniques are reviewed in section 1.5 which investigates the same problem. In section 1.6 we encourage the use of the model we have chosen. Finally, section 1.8, gives an overview of this first part of the document.

The following Chapter is dedicated to present our model and its mathematical formulation.

For more information

The state of the art presented in this section follows almost the same lines as several other works. The readers are invited to read the work referenced in this section for more informations. For details on Mono-Microphone source separation using:

- The Factorial Vector Quantization, a review can be found in the Thesis of Ron J. Weiss [62].
- The Gaussian (Scaled) Mixture Model approach, Hidden Markov Model and for the adaptive Wiener Filter we refer to the thesis of Laurent Benaroya [63], Alexey Ozerov [64] and Jean-Louis Durrieu [18].
- The Non-negative Matrix Factorization applied to music transcription, the thesis of Nancy Bertin [16] is dedicated to this method.
- Sinusoids Plus Noise Model, an overview of sound separation methods based on sinusoidal modeling can be found in the thesis of Tuomas Virtanen [65] as well as for the Matching Pursuit Approach in the work of Mahdi Triki [66].
- ARMA modeling for music transcription is considered in the thesis of Valentin Emiya [17].

Chapter 2

Model of Speech Production

In this chapter we will describe and motivate the source model we use. As mentioned in the previous chapter, we can represent a speech signal by the combinaison of two aspects: one is related to the fondamental frequency while the second one aims to represent the spectral shape. This is the speech production model we use and it can be efficiently modeled by using a cascade of two Auto-Regressive model.

The chapter is organized as follow, after a brief introduction we will analyse a speech spectrum and how to model it. The Long term modeling, for its periodicity (related to the fondamental frequency), is presented in section 2.2.3 and the Short term modeling, for its timbre (spectral shape), is presented in section 2.2.4. The mixture or observation is then defined, in section 2.4, as a sum of Short plus Long Term Auto-Regressive source with an additive white noise. Finally we conclude and discuss the model.

2.1 Introduction

Speech production begin with the lungs which generate air pressure. The air flows through the trachea, vocal folds, pharynx, oral and nasal cavities. Everything that is after the vocal folds is called the vocal tract. The speech sound can be done by two differents way. Firstly, for voiced speech, the vocal cords come in an oscillating regime because of the airflow (e.g. vowels */a/*, */o/* and */i/*, and nasals */m/* and */n/*). Consequently, voiced speech sounds consist of a strong periodic component rich in harmonics. Secondly, for unvoiced speech, airflow is constricted (e.g. fricatives */f/*, */s/* and */h/*) or completely stopped for a Short interval (e.g. stops */t/*, */p/* and */k/*). Therefore and unlike to voiced speech, unvoiced speech is of either noise-like without harmonic structure.

2.2 Speech Model Production

2.2.1 How to describe Human Voice

The theoretical basis widely used for speech modelling is the source-filter model, it models speech as a combination of a sound source: the vocal cords and a linear acoustic filter [9]. While only an approximation, the model is widely used in a number of applications because of its relative simplicity and robustness.

This model is based on the assumption that the speech can be modelled in two independent parts: the source and the filter. The above assumption assumes that vocal tract resonances and vocal fold oscillations have no interaction. In practise, because the error introduced by these assumptions is small, source-filter modelling yields good results.

For the implementations of source-filter models, the prevalent technique is all-pole modelling or linear predictive modelling [67]. With this method, we aim to model the filtering effects of the speech production mechanisms, with a parametric model, obtained by linear prediction that takes the source signal as input. The sound source, or excitation signal, is often modelled as a periodic impulse train. The vocal tract filter is approximated by an all-pole filter. Convolution of the excitation signal with the filter response then produces the synthesised speech.

2.2.2 A speech signal

Figure 2.1 shows a speech signal, in the temporal domain and its time frequency representation (Short Time Fourier Transform, see appendix F.3). The length of the speech is about 10 s, the length of a segment, weighted by a Hann window, is 128 ms for a sampling frequency of 8 KHz and an overlapp of 75 %. The Fourier Transforms are done using the Fast Fourier Transform (FFT) algorithm with a zero padding factor of 4.

We can easily observe the non-stationarity of the signal, the voice is not always active. The fondamental frequency changes over time and, also, when the same fondamental frequency appears on two differents frames the frequency plane is not the same (e.g looking through the frequency direction at a given time). So for a given Fondamental Frequency, the spectral shape changes. According to this observation it appears natural to model the periodicity and the spectral shape individually.

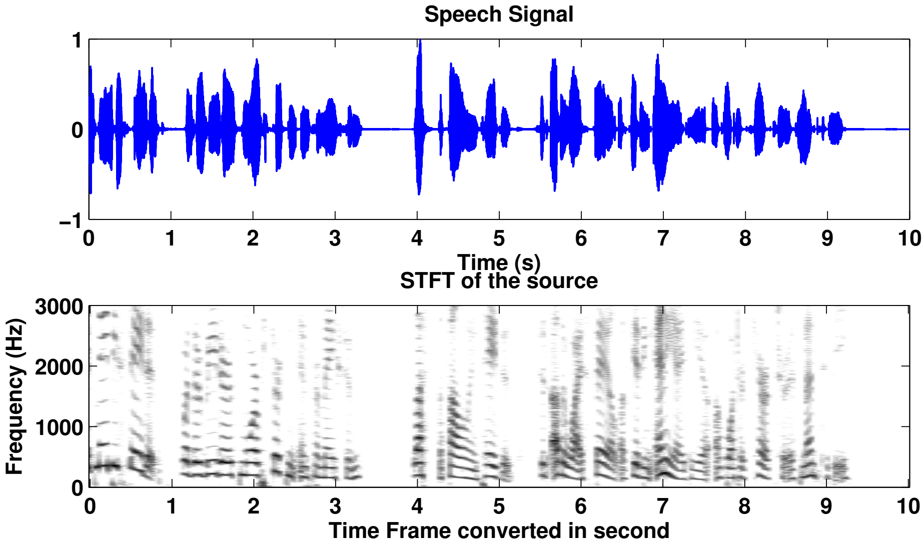


Figure 2.1: A speech signal and its STFT (log magnitude).

2.2.3 Modeling the Periodicity

As previously mentioned, instrumental sound and voiced part of speech are well modeled by the periodicity of the signal in the temporal domain. In the spectral domain it is represented by a comb like structure. If we ignore in a first time the spectral shape, the sound can be represented by a sum of sinusoids of the same amplitudes which is a frequency comb.

2.2.3.1 Comb Filter

A frequency comb can be achieve with a comb filter, in its feedback form, for which an output signal sample is the input signal sample plus a past version, with a delay equal to the desired period, of the output signal. This creates a more or less strong correlation of distante samples, as multiple past versions are added, and leads to a periodic signal. Figure 2.3 shows the feedback comb filter structure and the magnitude response of the filtering process. In the comb filter structure $e(t)$ is the input, $x(t)$ is the output and q is the advance operator so that q^{-1} is the unit delay operator and $q^{-1}x(t) = x(t-1)$, b is the attenuation factor related to the strenght of the periodicity. The filter is only stable if $|b|$ is strictly less than 1 and positive.

In the literrature, in order to achieve any arbitrary fondamental frequency, fractional delay are used. In our case we consider audio signals and we assume that the sampling frequency is relatively high (8 Khz is a typical value in speech processing, it is the sampling rate used by nearly all telephony systems). So, one possibility is to fix the period to the closest integer, or to use several samples in order to perform an interpolation between consecutif samples. The number of samples used for the fractional delay are generally called tapes [68] and, also, are generally weighted by arbitrary value. We will use a linear interpolation, it is an empirical choice, the value of b will be linearly splitted between the two closest samples. If the period τ is not an integer the value of b will be reparted into $(1-\alpha)\ b$ at the delay $\lfloor\tau\rfloor$ and $\alpha\ b$ at the delay $\lfloor\tau\rfloor + 1$ with $\alpha = 1 - \lfloor\tau\rfloor/\tau$, so that the sum is equal to b.

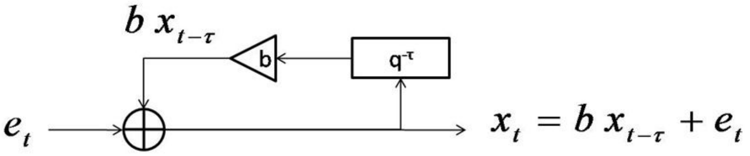


Figure 2.2: Feedback comb filter structure.

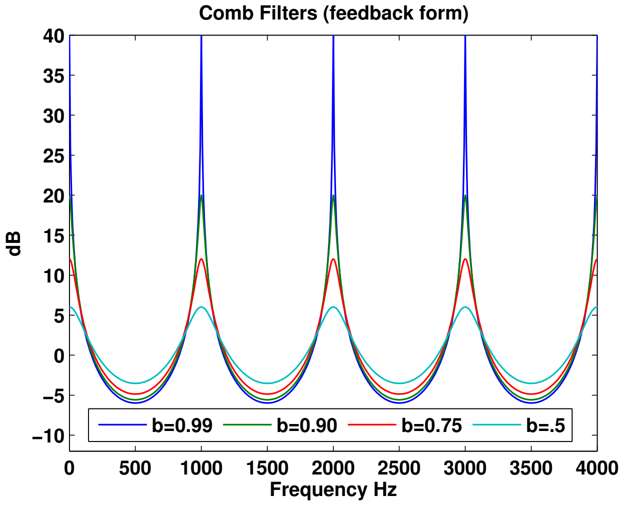


Figure 2.3: Magnitude response for various positive values of b

In order to give a range of values for the period consider some known fact about male and female human speakers. The mean fondamental frequency is between 80 and 250 Hz for a male and between 150 and 350 Hz for a woman, leading to typical period value between 20 and 100 samples at 8 KHz.

The comb filter can be seen as an autoregressif (AR) process of high order with only one (or two in case of a two taps fractional delay) non zero coefficient(s). We will denote it a Long Term (LT) AR Model, as it reflects the LT correlations of the signal.

2.2.3.2 Formulation of the Long Term Model

A Long term AR model is defined by three parameters, the period τ of the signal (related to the fundamental frequency of the harmonic sum), the Long term coefficient b which is related to the amplitudes of the harmonics and the interpolation factor α. Mathematically, the process is expressed as follow:

$$x_t = b x_{t-\tau} + e_t \tag{2.1}$$

$$x_t = (1-\alpha)\, b x_{t-\lfloor\tau\rfloor} + \alpha\, b x_{t-(\lfloor\tau\rfloor+1)} + e_t \tag{2.2}$$

for integer period (2.1) and fractional period (2.2), $\lfloor\rfloor$ is the floor operator.

The Long term error e_t is defined as a white Gaussian noise. It is independent and identically-distributed (i.i.d.) and drawn from a zero-mean normal distribution with variance σ_e^2. x_t is the source of interest at time t. Figure 2.5 shows a typical signal which follows this model, the temporal signal, its spectrum and the magnitude response of the filter are shown. The spectral reprensentation is a periodogram done with zero padding and without using an analysis window.

2.2.3.3 Traditional Long Term Parameters estimation

Here we present the traditional method to estimate the LT parameters for integer Period.

The optimal period at time instant $t = \tau$ can be defined as the particular value of τ, in a defined interval, that minimizes the normalized sum of squared error [9]

$$J_{t,\tau} = \frac{\sum_{t=\tau-N+1}^{m} \left[|x_t - b x_{t-\tau}|^2\right]}{\sum_{t=\tau-N+1}^{m} |x_t|^2} \tag{2.3}$$

The normalization term is required to compensate for the variable size of the speech segment involved and the uneven energy distribution over the pitch interval. For a given τ, the optimal value of b can be found by differentiating J with respect to b and setting the results to zero.

$$\hat{b}(\tau) = \frac{\sum_{t=\tau-N+1}^{m} [x_t \; x_{t-\tau}]}{\sum_{t=\tau-N+1}^{m} x_{t-\tau} \; x_{t-\tau}} = \frac{r_\tau}{r_0}$$

where the correlation sequence at delay k, for a signal x is estimated as:

$$r_k = \frac{1}{N} \sum_{n=0}^{N-1-k} x_k x_{n-k}, \;\; k = 1, ..., n \tag{2.4}$$

The Long term coefficient is found in the correlation sequence. The period τ ($\tau >> 1$) is estimated as the delay which maximizes b (for a realistic range of periods Ω_τ): $\hat{\tau} = \arg\max_{\tau \in \Omega_\tau} \hat{b}(\tau)$. Figure 2.4 shows the correlation sequence of a LT signal of period τ, the first peak (except the zero delay peak) is the LT coefficient b (the maximum of the sequence is one in the figure).

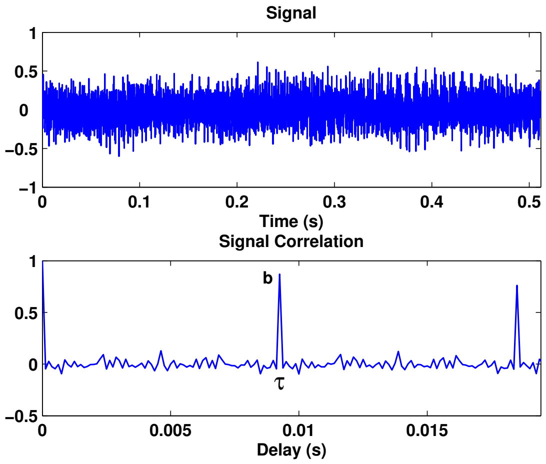


Figure 2.4: Long Term signal and its Auto correlation sequence.

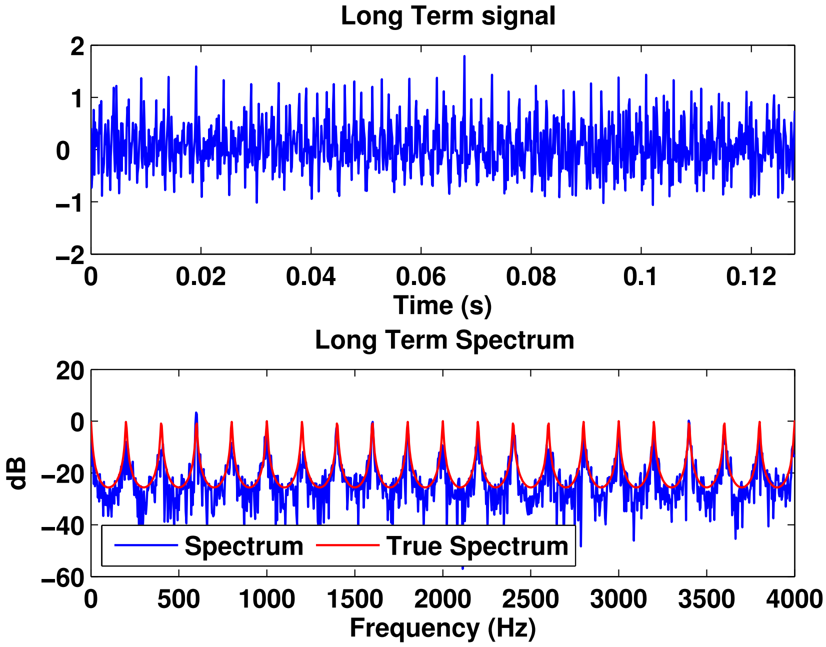


Figure 2.5: Example of a periodic signal. Sampling frequency $F_s = 8000Hz$, $b = 0.9$ and $\tau = 0.005s$. Temporal signal and its periodogram

Using (2.1): $x_t - bx_{t-\tau} = x_t(1 - b\, q^{-\tau}) = e_t$ (with q the advance operator) we get the prediction error e_t. Reexpressing the above expression in a compact form, in order to define the LT prediction error coefficient $\mathbf{b}$, we get $\mathbf{b} = [1,\ 0,\ \cdots\ 0,\ -b]^T$ where b $(0 \leq b < 1)$ is at the position τ. For fractional delay the vector becomes $\mathbf{b} = [1,\ 0,\ \cdots\ 0,\ -(1-\alpha)\times b,\ -\alpha \times b]^T$ the negative coefficients are at the position $\lfloor \tau \rfloor$ and $\lfloor \tau + 1 \rfloor$.

2.2.4 Modeling the Spectral Shape

Short order AR model allows to model the spectral envelope of a signal, for an instrument (as for a speaker) it defines the so called *Timbre* which allows to recognize an instrument to one other.

2.2.4.1 Short Term Autoregressive Model

A Short Term AR model of order p is defined by a set of coefficients. It reflects the fact that at a given moment the output signal sample is the input sample plus a linear combination of the past output samples. It generates correlation between consecutives sample refered as Short term correlation.

$$x_t = -\sum_{n=1}^{p} a_n x_{t-n} + e_t \tag{2.5}$$

x is the desired Gaussian source, a_n are the Short term AR coefficients and e is i.i.d. zero mean Gaussian with variance σ_e^2. Using (2.5) $x_t + \sum_{n=1}^{p} a_n x_{t-n} = x_t(1 + \sum_{n=1}^{p} a_n q^{-n})$ with

$a_0 = 1$ we have $x_t \sum_{n=0}^{p} a_n q^{-n} = e_t$. Also note that $a_0 = 1$ is needed in order to have a stable filter.

The Short term coefficients are also found in the correlation sequence. For an AR model of order n, the prediction coefficients $\mathbf{a} = [1 \ a_1 \cdots \ a_n]^T$ and the prediction error variance are obtained from the Yule-Walker (or normal) equations:

$$\mathbf{R}_{n+1}\mathbf{a} \ = \ \begin{bmatrix} r_0 & r_1 & \cdots & r_n \\ r_1 & \ddots & \ddots & \\ \vdots & \ddots & \ddots & r_1 \\ r_n & & r_1 & r_0 \end{bmatrix} \begin{bmatrix} 1 \\ a_1 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} \sigma_e^2 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{2.6}$$

Figure 2.6 shows two AR signals, their spectrum (periodogram) and their true spectrum, the Short term coefficients are given in the table 2.2.4.1. The first signal has three strong peaks in the spectrum (around 250, 2220 and 4000 Hz), as a result the signal seems sinusoidal compare to the other one with a very soft shape. Hopefully, the signals we are considering (e.g. speech signal) don't have this kind of structure otherwise another one periodicity will be added to the source.

Table 2.1: Short term coefficients.

$\mathbf{a}_1$	1	-0.5864	-0.2958	-0.3005	-0.6136	0.9772
$\mathbf{a}_2$	1	0.1698	-0.0974	-0.0007	-0.3489	-0.1342

2.3 Mixing Short and Long Term Autoregressive Model: A source

2.3.1 Modeling and parameters estimation

The complete source/filter model is so defines as a combination of AR process. In the procedure, a white gaussian noise is, first, passing through the so called Long Term filter. If the Long term coefficient is zero nothing happens and the output is still a white Gaussian noise. If not, the white noise becomes colored, periodic, and then its spectrum is composed by a frequency comb. The periodic Gaussian signal passes through the Short Term filter, which correlates closed samples, an gives the spectral shape to the signal.

The source (e.g. the Short plus Long term signal) is defined as:

$$x_t \ = \ -\sum_{n=1}^{p} a_n x_{t-n} + \tilde{x}_t \tag{2.7}$$

$$\tilde{x}_t \ = \ b\tilde{x}_{t-\tau} + e_t \tag{2.8}$$

Here we define some vocabulary used in this part of the thesis. e is a i.i.d. zero mean Gaussian with variance σ_e^2, we also refer to e as the Short plus Long term prediction error. τ is the desired period, b the Long term coefficient which affect the amplitudes of the all harmonics. $\tilde{x}_t$ which is the output of the first process is still Gaussian but no Longer white, after the first process it becomes a periodic signal and we also refer to it as the

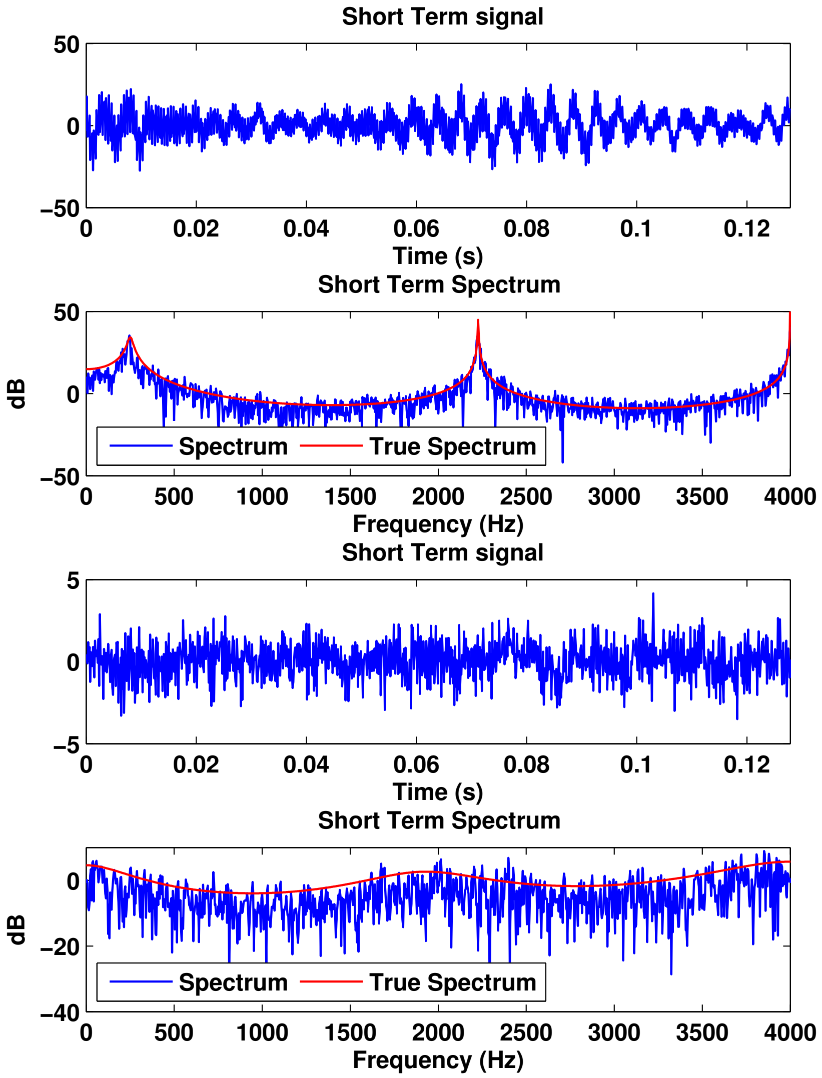


Figure 2.6: Example of AR signals of order $p = 5$.

Short term prediction error because it is the input of the second process. p is the order of the Short term AR filter with coefficients a_n $(a_0 = 1)$ finally x_t is still Gaussian and it is the signal of interest, the source. *Note that the order p of the Short term predictor is generally less than* 12 *and that the minimum period for human speech is greater than* 20.

Now the impulse train $(\tilde{x}_t\ )$ is convolved with the Short term coefficients, in the spectral domain this means that the harmonic structure is multiplied by the spectral shape, this modifies the constant amplitude of the harmonics and leads to a similar spectrum as the one obtained with a sum of sinusoids with differents amplitudes (see Figure 2.7).

Estimating the parameters is now different, we cannot directly use the same principle as before because the estimation of one of the AR model parameters will be corrupted by the other one. Joint estimation can lead to an unstable filter [69] and has to be excluded. A way is to estimates the parameters iteratively, estimating the Short term coefficient in the Long term error and conversely until convergence.

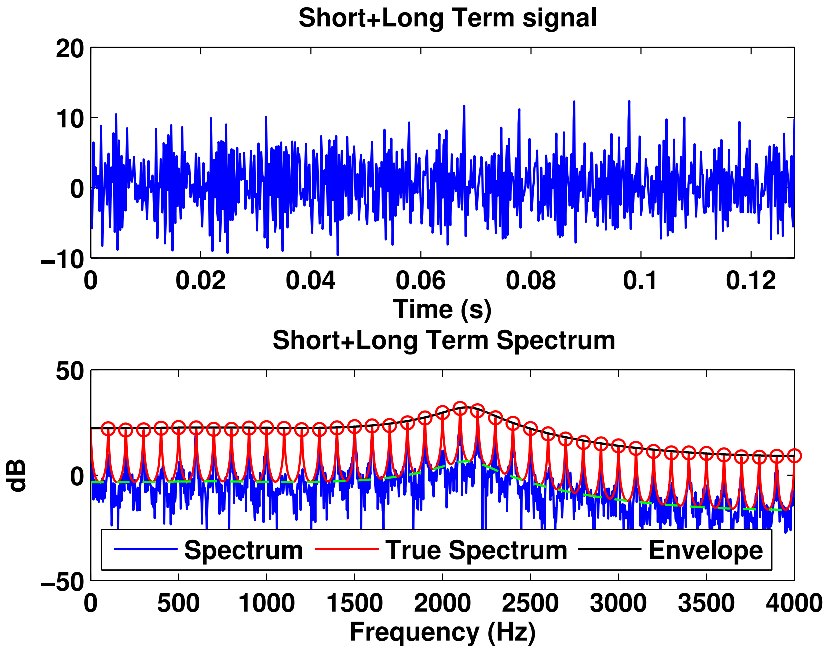


Figure 2.7: Example of a Short plus Long term model of order $p = 5$. Sampling frequency $F_s = 8000Hz$, $b = 0.9$ and $\tau = 0.005s$.

2.3.2 Spectra Comparison

In section 2.2.2 we have shown a real speech signal and then we have proposed to model it by the cascade of two filters. In previous sections we have explained how to model the speech, by separatly studying Long and Short term components, and how to estimate the parameters of each components. In order to illustrate we now show the synthesis of the real signal. Two estimation methods are used, the first one use traditional method to estimate the parameters: the Short term AR coefficients are estimated on the signal (using Levinson-Durbin Recursion [67,116]), then we use the estimated coefficients to filter the signal and to obtain the Short term prediction error. The Long term coefficients is computed on this prediction error. The second estimation method is an iterative method (described in appendix F.2) which did the same operation but iterating between the Short term and Long term prediction errors to estimate the coefficients (the average number of iterations needed for convergence is 10). The comparison is shown on Figure 2.8.

The results are not comparable, the iterative method, which estimates an aspect of the model after the cleaning of the other one, gives a better fitting of the spectrum, specially for the spectral shape. We can see on Figure 2.8 that the frequency component above $1KHz$ are better modelized for the iterative method. This is essentially due to the estimation of the Long term coefficient. The value obtained with the non iterative method is lower than for the iterative method, as a result some frequencies structure (formants) are not modeled. Note that the three figures have the same scale. This motivate us to keep this fact as much as possible into consideration for our futur estimation algorithms. Both methods finds the same estimated periods, note that it is a monophonic speech.

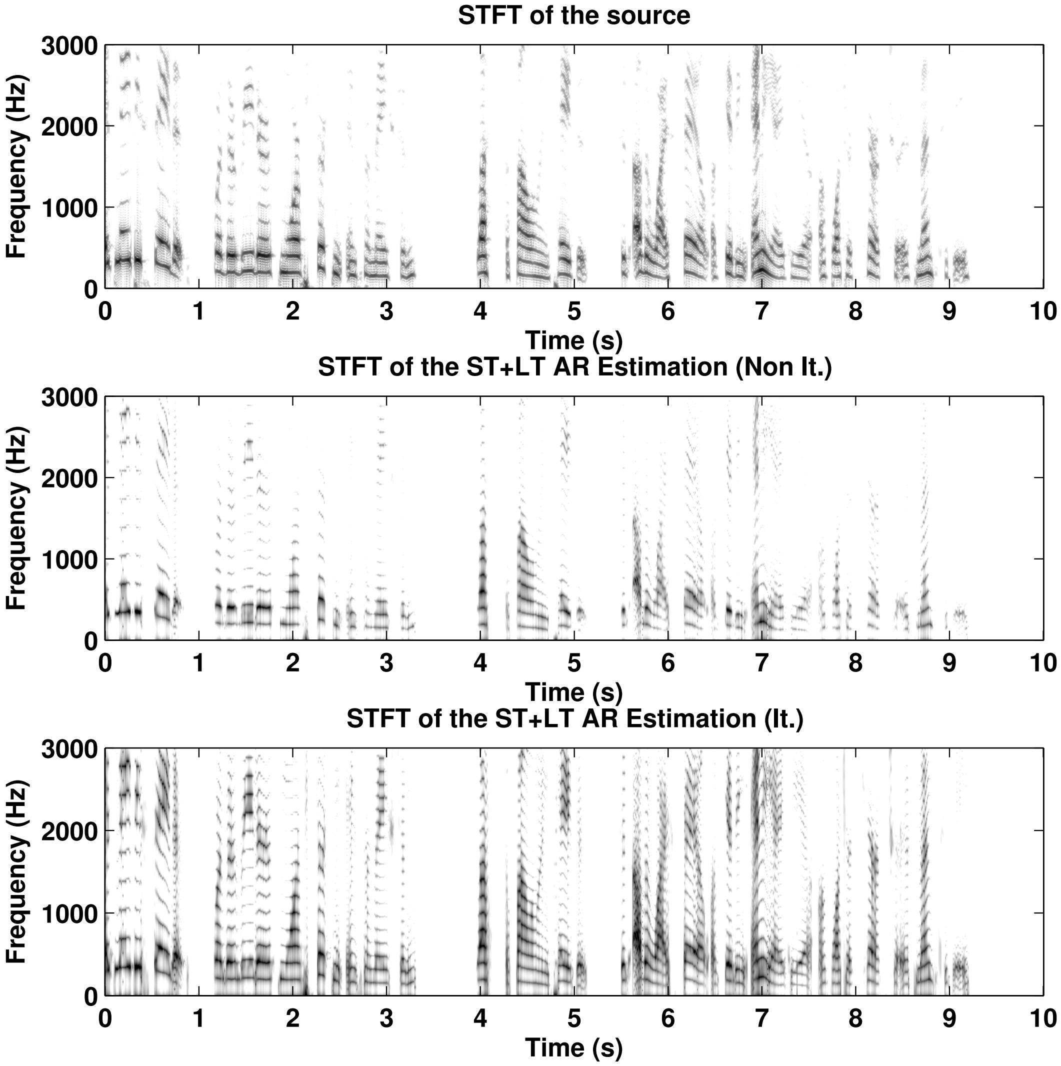


Figure 2.8: Synthesis Comparaison between iterative and non iterative methods for a real speech

2.4 Multiple Source Plus Noise Model

In the case of sources separation, several sources are involved and generally a noise is present, a source is assumed to be independent with the noise but also with the other sources. Each source is described by the model presented before.

2.4.1 Signal Model

The model for the sum of Short plus Long-term autoregressive (AR) Gaussian sources $x_{k,t}$ plus Gaussian white noise v_t (all independent) is :

$$\begin{aligned} y_t &= \sum_{k=1}^{K} x_{k,t} \ + \ v_t, && (2.9) \\ x_{k,t} &= -\sum_{n=1}^{p_k} a_{k,n} \ x_{k,t-n} \ + \ \tilde{x}_{k,t} && (2.10) \\ \tilde{x}_{k,t} &= b_k \ \tilde{x}_{k,t-\tau_k} \ + \ e_{k,t} && (2.11) \end{aligned}$$

where t is the discrete time, K is the number of sources $x_{k,t}$, v_t has variance σ_v^2, $e_{k,t}$ is the excitation signal of source k and is also assumed to be white Gaussian (i.i.d.) with variance σ_k^2. For each source x_k, τ_k is the period (its fractional part is implemented by linear interpolation or even be rounded if the sampling frequency is high enough), b_k its Long-term prediction coefficient and the Short-term prediction coefficients, of order p_k, are $a_{k,n}$. Sources, mixture and their spectra are shown in Figure 2.9 and 2.10 respectively.

2.4.2 Spectral Notation

If we introduce the Short-term and Long-term prediction error transfer functions

$$\begin{aligned} A_k(f) &= \sum_{n=0}^{p_k} a_{k,n} \ e^{-j2\pi f n} && (2.12) \\ B_k(f) &= 1 - b_k \ e^{-j2\pi f \tau_k} && (2.13) \end{aligned}$$

For the source k: $a_{k,0} = 1$ and $a_{k,n}$ are the ST coefficients, p_k the order of the ST modeling, b_k $(0 < b_k < 1)$ the LT coefficient and τ_k the period. The spectra of the sources can be written as:

$$\begin{aligned} S_k(f) &= \frac{\sigma_k^2}{|A_k(f) \ B_k(f)|^2} \ , \quad k = 1, \ldots, K && (2.14) \\ S_0(f) &= \sigma_v^2 = \sigma_0^2 && (2.15) \end{aligned}$$

where σ_k^2 is the variance of the Short plus Long term prediction error e_k and was previously called σ_e^2. The additive noise is considered as a (Short-term) AR model of order 0 (AR(0)) and is included in the signal set. The mixture spectrum is:

$$\begin{aligned} Y(f) &= \sum_{k=0}^{K} S_k(f) && (2.16) \\ &= \sigma_v^2 + \sum_{k=1}^{K} S_k(f) && (2.17) \end{aligned}$$

Why defining a model ? The problem we investigate is a source separation problem. The goal is to estimate/extract the source x_k from the mixture y. For this, we have defined a parametric model which allows to define a source by its model parameters. So to extract a source we have to first estimate its parameters, in our case the parameters to find are: the Short term parameters $a_{k,n}$ (we assume that the order p_k is known) and the Long term parameters b_k, τ_k and σ_k^2. In order to find noise free sources we also have to estimate the variance of the additive noise σ_v^2, or equivalently we have to find the source $k = 0$.

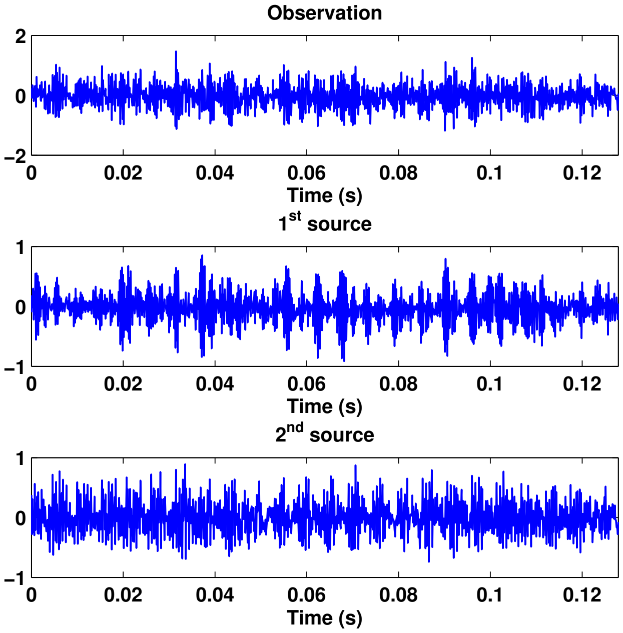


Figure 2.9: Short plus Long Term modeling of two sources and associated mixture.

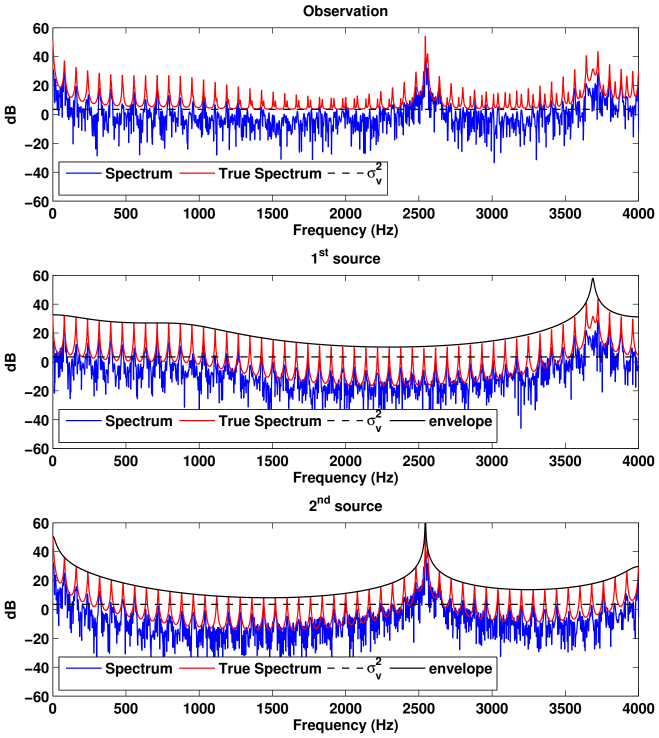


Figure 2.10: Spectra of Short plus Long Term modeling of two sources and associated mixture.

2.5 Summary and discussion

In this chapter we have presented the model that will be used in all the first part of this thesis. To summarize it, we will consider that the observation is a sum of sources with an additive white noise, each source follows the explained speech production model. We have decided to separate the Long term and the Short term components of the signal. Instead of using a single filter we describe the source by a cascade of two filters. We have made this choice for several reasons: the first one is due to the fact that the two aspect (Short and Long term) don't necessary change at the same pace. The second one, because an error on the estimation of a set of parameters will propagate errors on the second one leading to instability if the resulting AR is not stable [69]. This model is good enough to model human voice, it is able to model the most energetic part (voiced speech) which is periodic. the strenght periodicity can be varied strong by tunning the Long term coefficients. For unvoiced speech (e.g. non periodic) the Long term part will be negligeable and the obtained source will be dominated by the Short term model, as it is the case for speech. In order to model musical instruments whith such a model some problem have to be posed. First of all the periodicity, musical instruments like Piano are known to be inharmonic (e.g. a frequency peak is not exactly a multiple integer of the fundamental frequency) and this is not achieved with this model. Then, several instruments like Piano, Guitar have a strong transient part (a note needs a non negligeable time to be stable) which is not modelized. Also it is usual to consider the temporal enveloppe of a musical note, called the Attack Decay Sustain Release (ADSR) [117], used for example in the work of [16, 66]. Enveloppe that this model didn't modelizes. And finally we make the hypothesis of an additive white noise, which is less general than a colored one. However it can be good enough to model a wide range of signal.

The next chapters deal with the design of algorithms which use this model for the mono-microphone source separation problem.

Chapter 3

Adaptive EM-Kalman Filter

In this chapter we explain the first Blind Audio Source Separation (BASS) algorithm that we have designed. It is an adaptive EM-Kalman algorithm. The contribution in this chapter is not related to Kalman or EM theory but to the application. We develop an Extended State Space Model (ESSM) to represent the equations models for audio signals introduced in chapter 2. We will see that the Extended form of the SSM transforms the filtering equation into smoothing equation (of delay 1*).*

The chapter is organized as follows: we recall some notions of the Basic Kalman Filter (KF) in section 3.1. Then the idea of the Expectation Maximization (EM) algorithm is sketched in section 3.2. Finally adaptive EM-Kalman algorithm is depicted. Then section 3.4 describes the state space model and the algorithmic details. We finally present some simulation results in section 3.8 and we discuss the proposed algorithm ine section 3.9.

3.1 Kalman Filter (KF)

The Kalman filter (KF) considers the estimation of a first-order vector autoregressive (Markov) process from linear measurements in white noise. The KF performs this estimation recursively by alternating between filtering (measurement update) and prediction (time update). It can be applied to any time series model which can be written in State Space Form. Almost all the standard time series models can be written in this form. The KF corresponds to optimal (Minimum Mean Squared Error (MMSE) or Maximum A Posteriori (MAP)) Bayesian estimation of the state sequence if all random sources involved (measurement noise, state noise and state initial conditions) are Gaussian. The signal model can be written as

$$\begin{array}{rcl} & \text{state update equation:} & \\ \boldsymbol{x}_{k+1} & = & \boldsymbol{F}_k\,\boldsymbol{x}_k + \boldsymbol{G}_k\,\boldsymbol{w}_k \\ & \text{measurement equation:} & \\ \boldsymbol{y}_k & = & \boldsymbol{H}_k\,\boldsymbol{x}_k + \boldsymbol{v}_k \end{array} \tag{3.1}$$

for $k = 1, 2, \ldots$, where the initial state $\boldsymbol{x}_0 \sim \mathcal{N}(\widehat{\boldsymbol{x}}_0, \boldsymbol{P}_0)$, the measurement noise $\boldsymbol{v}_k \sim \mathcal{N}(0, \boldsymbol{R}_k)$, the state noise $\boldsymbol{w}_k \sim \mathcal{N}(0, \boldsymbol{Q}_k)$ and all these random quantities are mutually uncorrelated. $\boldsymbol{y}_k$ is the observation, $\boldsymbol{F}_k$ is the state transition model which is applied to the previous state $\boldsymbol{x}_k$, and it refers to the dynamic of the system. $\boldsymbol{G}_k$ is the control-input model which is applied to the state noise, $\boldsymbol{H}_k$ is the observation model which maps the true state space into the observed space and $\boldsymbol{v}_k$ is the observation noise. In the case of time-varying system matrices $\boldsymbol{F}_k$ etc., the form of the equations as they appear in (3.1) is the most logical one, with the state update corresponding to a prediction of the state on the basis of the quantities available at time k.

The Kalman filter is therefore a recursive estimator. This means that only estimated state from the previous time step and the current measurement are needed to compute the estimate for the current state. The prediction phase uses the estimate from the previous timestep to produce an estimate of the current state. In the update phase, measurement information from the current timestep is used to refine this prediction to get to a new, more accurate estimate. In the following, we introduce the notation $\boldsymbol{y}_{1:k} = \{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_k\}$.

The Kalman Filter is a two-step recursive procedure, going from $|k{-}1$ to $|k$:

Measurement Update

$$\begin{array}{rcl} \boldsymbol{K}_k & = & \boldsymbol{P}_{k|k-1}\,\boldsymbol{H}_k^T\,\left(\boldsymbol{H}_k\,\boldsymbol{P}_{k|k-1}\,\boldsymbol{H}_k^T + \boldsymbol{R}_k\right)^{-1} \\ \widehat{\boldsymbol{x}}_{k|k} & = & \widehat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{K}_k\,\left(\boldsymbol{y}_k - \boldsymbol{H}_k\,\widehat{\boldsymbol{x}}_{k|k-1}\right) \\ \boldsymbol{P}_{k|k} & = & \boldsymbol{P}_{k|k-1} - \boldsymbol{K}_k\,\boldsymbol{H}_k\,\boldsymbol{P}_{k|k-1} \end{array} \tag{3.2}$$

Time Update (prediction)

$$\begin{array}{rcl} \widehat{\boldsymbol{x}}_{k+1|k} & = & \boldsymbol{F}_k\,\widehat{\boldsymbol{x}}_{k|k} \\ \boldsymbol{P}_{k+1|k} & = & \boldsymbol{F}_k\,\boldsymbol{P}_{k|k}\,\boldsymbol{F}_k^T + \boldsymbol{G}_k\,\boldsymbol{Q}_k\,\boldsymbol{G}_k^T \end{array} \tag{3.3}$$

The subscript $k|k$ means that we estimate a quantity at time k given observations up to time k included. The choice of the initial conditions crucially affects the initial convergence (transient behavior). In the usual case of total absence of prior information on the initial state, one can choose $\widehat{\boldsymbol{x}}_0 = 0$, $\boldsymbol{P}_0 = p_0\,I$. This leads to $\boldsymbol{P}_{1|0} = \boldsymbol{F}_0\,\boldsymbol{P}_0\,\boldsymbol{F}_0^T + \boldsymbol{G}_0\,\boldsymbol{Q}_0\,\boldsymbol{G}_0^T$, $\widehat{\boldsymbol{x}}_{1|0} = \boldsymbol{F}_0\,\widehat{\boldsymbol{x}}_0$.

For numerical stability (in the presence of roundoff errors), it is crucial that the symmetry of the covariance matrices $\boldsymbol{P}_{k|k-1}$, $\boldsymbol{P}_{k|k}$ is maintained throughout the updates (which is not going to be the case with the updating of $\boldsymbol{P}_{k|k}$ the way it appears in (3.2)). This point will be discussed later in this chapter.

3.2 Expectation Maximization algorithm (EM)

Expectation Maximization (EM) algorithm is a method to find maximum likelihood or maximum a posteriori (MAP) estimates of parameters in statistical models, where the model depends on unobserved hidden variables x (i.e. variables that are not directly observed but are rather inferred). The EM algorithm was first introduced by Dempster et al. in [10], and has been used extensively for model parameter estimation [71, 118, 119].

The objective is to compute an estimate of the model parameters (θ) given a measurement sequence. For Gaussian models, Maximum Likelihood (ML) estimate is an obvious choice, which is given as follows: $\theta_{ML} = \arg\max_{\theta} \log p(\boldsymbol{y}_{1:k}|\theta)$ where $p(\boldsymbol{y}_{1:k}|\theta)$. Because of the dependence on the states, which are not available, direct maximization is not possible. The idea is to maximize the likelihood with respect to two unknowns: states and model parameters. The EM algorithm uses an iterative approach by first maximizing the likelihood with respect to the states in the E-step, and then maximizing with respect to the parameters in the M-step.

The E-step is given by the expected value of the complete log-likelihood function as follows:

$$\mathcal{Q} = E_{\boldsymbol{x}|\boldsymbol{y}} p(\boldsymbol{y}_{1:k}\boldsymbol{x}_{1:k}|\theta) \tag{3.4}$$

The M-step involves the direct differentiation of $\mathcal{Q}$ to find the values of the parameters.

3.2.1 Kalman Smoothing With the EM Algorithm

Now the expectation is taken, in principle with the conditional distribution given all data in (3.4). This leads to an iterative algorithm with, at each iteration, a whole fixed-interval smoothing.

3.2.1.1 Fixed Interval Smoothing

The filtering problem is to find the best estimate of the state at stage k conditioned on the measurements up to and including stage k. The fixed-interval smoothing problem is to find the best estimate of all state trajectories in time history for stages 0 to k conditioned on the measurements for the entire interval. As a result, the optimal fixed-interval smoother provides the optimal estimate of $\hat{\mathbf{x}}_{k|n}$ $(k < n)$ using the measurements from a fixed interval. One of the most popular is the Rauch Tung Striebel (RTS) Smoother [70] which is an efficient two-pass algorithm. In pratice the first pass is a basic Kalman Filter (forward pass) followed by a backward pass involving all the samples in the signal. All states and estimated covariance need to be stored during the forward pass in order to compute the backward one, as a result, the fixed interval smoothing consumes a lot of memory and will not be used.

3.2.1.2 Fixed lag Smoothing

An adaptive version of the Kalman Smoother can be obtained by replacing fixed-interval smoothing by fixed-lag smoothing and performing one iteration per time sample [71, 72]. Since the state update equation corresponds to a vector AR(1) model, one may expect (as in [72]) that a unit lag should be enough to guarantee convergence. Using the innovations approach, we have

$$\widehat{\boldsymbol{x}}_{k-1|k} = \widehat{\boldsymbol{x}}_{k-1|k-1} + R_{\boldsymbol{x}_{k-1}\widetilde{\boldsymbol{y}}_k}\, \boldsymbol{S}_k^{-1}\, \left(\boldsymbol{y}_k - \boldsymbol{H}_{k-1}\, \widehat{\boldsymbol{x}}_{k|k-1}\right) \; . \tag{3.5}$$

After a few steps, we get the following lag-1 smoothing equations that need to be added to the basic Kalman Filter equations (to be inserted between the Measurement Update and the Time Update)

$$\begin{array}{rcl} \boldsymbol{K}_{k;1} & = & \boldsymbol{P}_{k-1|k-1}\, \boldsymbol{F}_{k-1}^T\, \boldsymbol{H}_k^T \\ \widehat{\boldsymbol{x}}_{k-1|k} & = & \widehat{\boldsymbol{x}}_{k-1|k-1} + \boldsymbol{K}_{k;1}\, \boldsymbol{S}_k^{-1}\, \left(\boldsymbol{y}_k - \boldsymbol{H}_{k-1}\, \widehat{\boldsymbol{x}}_{k|k-1}\right) \\ \boldsymbol{P}_{k-1|k} & = & \boldsymbol{P}_{k-1|k-1} - \boldsymbol{K}_{k;1}\, \boldsymbol{S}_k^{-1}\, \boldsymbol{K}_{k;1}^T \; . \end{array} \tag{3.6}$$

This step can be skipped by using an appropriate State Space Model (SSM), called Extended SSM (ESSM) in this thesis. In fact (3.6) is obtained from (3.2) by extending the State Space vector, i.e. adding one older past sample to the vector.

3.3 Adaptive EM-KF with Fixed-Lag Smoothing

Consider now the case in which the state-space model is essentially time-invariant (or slowly time-varying). In that case the time index of the system matrices $\boldsymbol{F}_k$ etc. just reflects at which time the unknown system matrices have been adapted. The resulting KF equations with lag-1 smoothing become

$$\begin{array}{rcl} \boldsymbol{K}_{k;1} & = & \boldsymbol{P}_{k-1|k-1}\, \boldsymbol{F}_{k-1}^T\, \boldsymbol{H}_{k-1}^T \\ \widehat{\boldsymbol{x}}_{k-1|k} & = & \widehat{\boldsymbol{x}}_{k-1|k-1} + \boldsymbol{K}_{k;1}\, \left(\boldsymbol{H}_{k-1}\, \boldsymbol{P}_{k|k-1}\, \boldsymbol{H}_{k-1}^T + \boldsymbol{R}_{k-1}\right)^{-1}\, \left(\boldsymbol{y}_k - \boldsymbol{H}_{k-1}\, \widehat{\boldsymbol{x}}_{k|k-1}\right) \\ \boldsymbol{P}_{k-1|k} & = & \boldsymbol{P}_{k-1|k-1} - \boldsymbol{K}_{k;1}\, \left(\boldsymbol{H}_{k-1}\, \boldsymbol{P}_{k|k-1}\, \boldsymbol{H}_{k-1}^T + \boldsymbol{R}_{k-1}\right)^{-1}\, \boldsymbol{K}_{k;1}^T \\ \boldsymbol{K}_k & = & \boldsymbol{P}_{k|k-1}\, \boldsymbol{H}_{k-1}^T\, \left(\boldsymbol{H}_{k-1}\, \boldsymbol{P}_{k|k-1}\, \boldsymbol{H}_{k-1}^T + \boldsymbol{R}_{k-1}\right)^{-1} \\ \widehat{\boldsymbol{x}}_{k|k} & = & \widehat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{K}_k\, \left(\boldsymbol{y}_k - \boldsymbol{H}_{k-1}\, \widehat{\boldsymbol{x}}_{k|k-1}\right) \\ \boldsymbol{P}_{k|k} & = & \boldsymbol{P}_{k|k-1} - \boldsymbol{K}_k\, \boldsymbol{H}_{k-1}\, \boldsymbol{P}_{k|k-1} \\ & & \qquad\qquad\qquad \text{parameter update} \\ \widehat{\boldsymbol{x}}_{k+1|k} & = & \boldsymbol{F}_k\, \widehat{\boldsymbol{x}}_{k|k} \\ \boldsymbol{P}_{k+1|k} & = & \boldsymbol{F}_k\, \boldsymbol{P}_{k|k}\, \boldsymbol{F}_k^T + \boldsymbol{G}_k\, \boldsymbol{Q}_k\, \boldsymbol{G}_k^T \end{array} \tag{3.7}$$

So, the system matrices ($\boldsymbol{F}$, $\boldsymbol{G}$, $\boldsymbol{Q}$) should be adapted after the smoothing step and before the filtering and prediction steps.

3.4 Adaptive EM-KF for Mono-Microphone BASS

3.4.1 Introduction

In this section we explain the different steps of the algorithm and the different quantities used. Our main contributions in this chapter are the derivation of a State Space Model for the signal model we use and the derivation of the traditional M-Step for the parameters estimation. We will first recall the model and how to transpose it in a State Space form to estimate jointly the sources, then we will derive the M-Step and we will give the complete algorithm.

3.5 State Space Model Formulation

We still consider the problem of estimating K mixed Gaussian sources using the voice production model presented in chapter 2, that can be described by filtering an excitation signal with Long term prediction filter followed by a Short term filter, which is formulated as:

$$\begin{aligned} y_t &= \sum_{k=1}^{K} x_{k,t} \ + \ v_t, \\ x_{k,t} &= -\sum_{n=1}^{p_k} a_{k,n}\ x_{k,t-n} \ + \ \tilde{x}_{k,t} \\ \tilde{x}_{k,t} &= b_k\ \tilde{x}_{k,t-\tau_k} \ + \ e_{k,t} \end{aligned} \tag{3.8}$$

where, for the mixture:

- y_t is the scalar observation.
- $x_{k,t}$ is the k^{th} source at time t, an AR process of order p_k
- v_t is a white Gaussian noise with variance σ_v^2, independent of the innovations $\{e_{k,t}\}_{k=1..K}$

and for each source k:

- p_k is the Short term order
- $a_{k,n}$ is the n^{th} Short term coefficient
- $\tilde{x}_{k,t}$ is the Short term prediction error
- b_k is the Long term prediction coefficient
- τ_k is the period, not necessary an integer
- $e_{k,t}$ are i.i.d. zero mean Gaussian. It is the Short plus Long term prediction error, also called innovation sequences, with variance σ_k^2

We assume that we know the number of sources and that they have the same AR modeling order (fixed). We want to estimate the sources x_k individually; in order to perform the separation, the other quantities have to be estimated.

The model as described in (3.8) is divided in two part for each source. The Short plus Long term prediction error appears in $\tilde{x}_{k,t} - b_k\ \tilde{x}_{k,t-\tau_k}\ =\ e_{k,t}$. Applying KF on (3.8) as the state equations (state is defined in (3.10)) requires that all b_k and τ_k are known, or separately estimated. e_k also appears indirectly in $x_{k,t} - \sum_{n=1}^{p_k} a_{k,n}\ x_{k,t-n} - b_k\ \tilde{x}_{k,t-\tau_k}\ =\ e_{k,t}$: it needs all the parameter of the source and it is applied on the signal and on its ST prediction error. Let $\mathbf{x}_{k,t}$ be the vector of length $(N + p_k + 2)$, defined by concantenating x_k and $\tilde{x_k}$:

$$\begin{aligned}\mathbf{x}_{k,t} = [x_k(t)\ x_k(t-1) \cdots x_k(t-p_k-1) \mid \tilde{x}_k(t)\ \tilde{x}_k(t-1) \cdots \\ \cdots \tilde{x}_k(t - \lfloor\tau_k\rfloor) \cdots \tilde{x}_k(t-N+1)]^T\end{aligned} \tag{3.9}$$

where N will be discussed later in this section. The different operations related to the parameters will be reflected by the product of this concatenated vector with the transition matrix $\mathbf{F}_k$. As the signal is non stationary, $\mathbf{F}_k$ is time dependent, and we omit the time index for clarity. In the adaptive scheme the iteration refers to the time evolution. Hence the process at time t can be written as:

$$\mathbf{x}_{k,t} \quad = \quad \mathbf{F}_k\ \mathbf{x}_{k,t-1}\ +\ \mathbf{g}_k\ e_{k,t} \tag{3.10}$$

where $\mathbf{g}_k$ is the $(N + p_k + 2)$ length vector defined as $\mathbf{g}_k = [\ 1\ 0 \cdots 0 \mid 1\ 0\ \cdots\cdots\ 0]^T$. The second non null component is at position $(p_k + 3)$. The $(N + p_k + 2) \times (N + p_k + 2)$ matrix $\mathbf{F}_k$ has the following structure

$$\mathbf{F}_k \quad = \quad \begin{bmatrix} \mathbf{F}_{11,k} & \mathbf{F}_{12,k} \\ \mathbf{O} & \mathbf{F}_{22,k} \end{bmatrix} \tag{3.11}$$

The transition matrix is composed of three sub matrices. With the defined $\mathbf{x}_{k,t}$ it is clear that the sub matrices $\mathbf{F}_{12}$ and $\mathbf{F}_{22}$ only affect the second part of the vector (the ST prediction error) while the sub matrix $\mathbf{F}_{11}$ affects the signal part.

The sub matrices $(p_k + 2) \times (p_k + 2)$ matrix $\mathbf{F}_{11,k}$, the $(p_k + 2) \times N$ matrix $\mathbf{F}_{12,k}$ and the $N \times N$ matrix $\mathbf{F}_{22,k}$ are given by:

$$\mathbf{F}_{11,k} \quad = \quad \begin{bmatrix} -a_{k,1} & -a_{k,2} & \cdots & -a_{k,p_k} & 0 & 0 \\ & & & & & \vdots \\ & & I_{(p_k+1)} & & & \vdots \\ & & & & & \vdots \\ & & & & & 0 \end{bmatrix}$$

$$\mathbf{F}_{12,k} \quad = \quad \begin{bmatrix} 0 & \cdots & (1-\alpha_k)\ b_k & \alpha_k\ b_k & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \end{bmatrix}$$

$$\mathbf{F}_{22,k} \quad = \quad \begin{bmatrix} 0 & \cdots & (1-\alpha_k)\ b_k & \alpha_k\ b_k & 0 & \cdots & 0 \\ & & & & & & \vdots \\ & & & & I_{(N-1)} & & \vdots \\ & & & & & & \vdots \\ & & & & & & 0 \end{bmatrix}$$

It is noteworthy that the choice of the $\mathbf{F}_{22,k}$ matrix size N should be done carefully. In fact, the value of N should be superior to the maximum value of the periods for a possible period tracking. It can be noticed that the coefficients $(1-\alpha_k)\ b_k$ and $\alpha_k\ b_k$ are situated respectively in the $\lfloor \tau_k \rfloor^{th}$ and $\lfloor \tau_k + 1 \rfloor^{th}$ columns of $\mathbf{F}_{22,k}$ and $\mathbf{F}_{12,k}$.
In order to perform the separation jointly for the K sources, we introduce the vector $\mathbf{x}_t$ that consists of the concatenation of the $\{\mathbf{x}_{k,t}\}_{k=1:K}$ vectors $\mathbf{x}_t = \left[\mathbf{x}_{1,t}^T\ \mathbf{x}_{2,t}^T\ \cdots\ \mathbf{x}_{K,t}^T\right]^T$ which results in the time update equation, see (3.12). Moreover, by reformulating the expression of $\{y_t\}$, we introduce the observation equation (see 3.13).

We obtain the following state space model:

$$\mathbf{x}_t = \mathbf{F}\ \mathbf{x}_{t-1} + \mathbf{G}\ \mathbf{e}_t \tag{3.12}$$

$$y_t = \mathbf{h}^T \mathbf{x}_t + v_t \tag{3.13}$$

where

- $\mathbf{e}_t = [e_{1,t}\ e_{2,t}\ \cdots\ e_{K,t}]^T$ is the $K \times 1$ column vector resulting from the concatenation of the K innovations at time t. Its covariance matrix is the $K \times K$ diagonal matrix $\mathbf{Q}$ =diag$(\sigma_1, \cdots, \sigma_K)$.
- $\mathbf{F}$ is the $\sum_{k=1}^{K}(p_k+N+2) \times \sum_{k=1}^{K}(p_k+N+2)$ block diagonal matrix given by $\mathbf{F} =$ blockdiag $(\mathbf{F}_1, \cdots, \mathbf{F}_K)$. See Figure 3.1 for a two-sources example.
- $\mathbf{G}$ is the $\sum_{k=1}^{K}(p_k+N+2) \times K$ matrix given by $\mathbf{G}$ = block diag$(\mathbf{g}_1, \cdots, \mathbf{g}_K)$
- $\mathbf{h}$ is the $\sum_{k=1}^{K}(p_k+N+2) \times 1$ column vector given by $\mathbf{h} = [\mathbf{h}_1^T \cdots \mathbf{h}_K^T]^T$ where $\mathbf{h}_i = [1\ 0 \cdots\ 0]^T$ of length $(N+p_k+2)$.

Figure 3.1 shows a two-sources example. On the right the state vector is represented. It consists of the concatenation of two state vectors (themself being the concatenation of the source signal and of its Short Term prediction error signal), one by source k. The transition matrix F is shown on the right of the Figure. It is a block diagonal matrix (one block by source k). Each source block F_k of F is composed by three sub matrices defined in (3.11).
It is obvious that the linear dynamic system derived previously depends on unknown parameters recapitulated in the variable

$$\theta = \left\{ \{\mathbf{a_{k,n}}\}_{\substack{k\in\{1,\ldots,K\} \\ n\in\{1,\ldots,p_k\}}}, \{\mathbf{b_k}\}_{\mathbf{k}\in\{\mathbf{1},\ldots,\mathbf{K}\}}, \{\sigma_\mathbf{k}\}_{\mathbf{k}\in\{\mathbf{1},\ldots,\mathbf{K}\}}, \sigma_\mathbf{v}^\mathbf{2} \right\} \tag{3.14}$$

Hence, a joint estimation of sources (the state) and θ is required. In literature (see e.g. [70, 72, 118]) the EM-Kalman algorithm presents an efficient approach to estimate iteratively parameters and its convergence to the Maximum Likelihood solution is proven [10]. In the next section, the application of this algorithm to our case is developed.

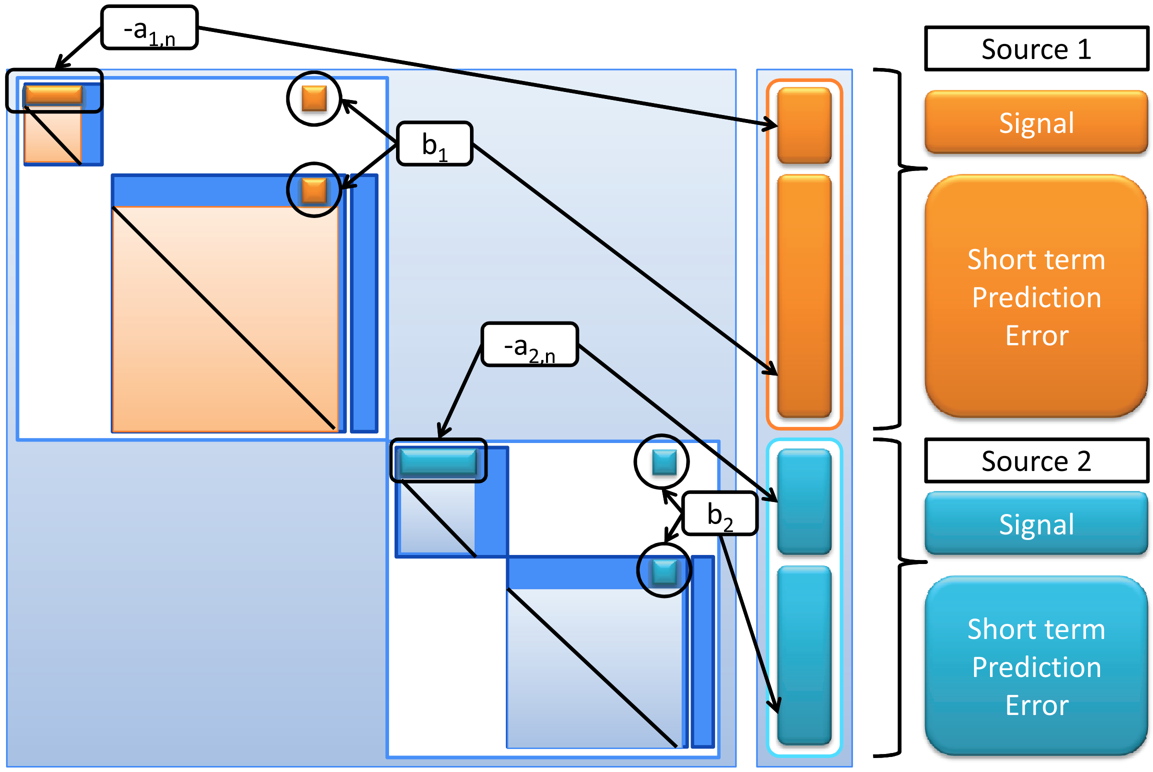


Figure 3.1: State Space Model and State Vector for 2 sources.

3.6 Algorithm

The EM-Kalman algorithm allows to estimate iteratively parameters and sources by alternating two steps, E-step and M-step [10]. In the M-step, an estimate of the parameters $\hat{\theta}$ is computed. In our problem, there are two types of parameters. The first one includes the parameters of the time update equation, in (3.12), which consist on the Short term and Long term coefficients and the innovation variance of all the K sources. The second one is the parameter of the observation in (3.13), the additive noise variance. From the state space model presented in the first part, and for each source k, the relation between the innovation process at time $t-1$ and the LT plus ST coefficients could be written as

$$e_{k,t-1} = \mathbf{v}_k^T \check{\mathbf{x}}_{k,t-1} \tag{3.15}$$

where $\mathbf{v}_k = [1\ a_{k,1}, \cdots, a_{k,p_k},\ -(1-\alpha_k)\, b_k,\ -\alpha_k\, b_k]^T$ is a $(p_k+3)\times 1$ column vector and $\check{\mathbf{x}}_{k,t-1} = [x_k(t-1,\theta)\cdots x_k(t-p_k-1,\theta)\ \tilde{x}_k(t-\lfloor\tau_k\rfloor-1,\theta)\ \tilde{x}_k(t-\lfloor\tau_k\rfloor-2,\theta)]^T$ is called the partial state deduced from the full state $\mathbf{x}_t$ with the help of a selection matrix $\mathbf{S}_k$. The relation between the partial state at time $t-1$ and the full state at time t is $\check{\mathbf{x}}_{k,t-1} = \mathbf{S}_k \mathbf{x}_t$. The lag of one time sample between the full and partial state will be justified later and will lead to the fixed-lag smoothing.

3.6.1 Partial states discussion

As we have just mentioned, the selection matrix and the extended form of the transition matrix allow to transform the filtering in fixed lag smoothing. Also, this selection matrix extracts in the source state the quantities needed to estimate the Short plus Long term variances. But note that this formulation can lead to extract the parameters (ST and LT) jointly or separately:

- The joint relation between the partial state and the innovation is the one defined before:

$$\begin{aligned}
\breve{\mathbf{x}}_{k,t-1} &= [x_k(t-1,\theta)\cdots x_k(t-p_k-1,\theta)\ \tilde{x}_k(t-\lfloor\tau_k\rfloor-1,\theta)\ \tilde{x}_k(t-\lfloor\tau_k\rfloor-2,\theta)]^T \\
\mathbf{v}_k &= [1,\ a_{k,1},\cdots,a_{k,p_k},\ -(1-\alpha_k)\ b_k,\ -\alpha_k\ b_k]^T \\
e_{k,t-1} &= \mathbf{v}_k^T\breve{\mathbf{x}}_{k,t-1}
\end{aligned}$$

- The alternative relation consists on decoupling between ST and LT parameters. The consequence is the algorithm design possibilities that it offers:

$$\begin{aligned}
\breve{\mathbf{x}}_{k,t-1}^{ST} &= [x_k(t-1,\theta)\cdots x_k(t-p_k-1,\theta)]^T \\
\mathbf{v}_k^{ST} &= [1\ a_{k,1}\cdots a_{k,p_k}]^T \\
\tilde{x}_{k,t-1} &= \left(\mathbf{v}_k^{ST}\right)^T\breve{\mathbf{x}}_{k,t-1}^{ST}
\end{aligned}$$

and

$$\begin{aligned}
\breve{\mathbf{x}}_{k,t-1}^{LT} &= [\tilde{x}_k(t-1,\theta)\ \tilde{x}_k(t-\lfloor\tau_k\rfloor-1,\theta)\ \tilde{x}_k(t-\lfloor\tau_k\rfloor-2,\theta)]^T \\
\mathbf{v}_k^{LT} &= [1,\ -(1-\alpha_k)\ b_k,\ -\alpha_k\ b_k]^T \\
e_{k,t-1} &= \left(\mathbf{v}_k^{LT}\right)^T\breve{\mathbf{x}}_{k,t-1}^{LT}
\end{aligned}$$

This lead us to design two algorithms. The first one is called **Joint-EMK** and it estimates jointly the parameters. The second one performs alternated estimation and is called **Alt-EMK**. Naturally designing algorithms with only one aspect of the speech model are also investigated in simulations.

3.6.2 Parameters estimation

After multiplying (3.15) by $\breve{\mathbf{x}}_{k,t-1}^T$ on both sides, applying the conditional expectation operator $E\{\ |y_{1:t}\}$ and doing a matrix inversion, the following relation between the vector of coefficients and the innovation variance is deduced:

$$\mathbf{v}_k = \sigma_k\mathbf{R}_{k,t-1}^{-1}[1,0\cdots 0]^T \tag{3.16}$$

The vector $\mathbf{v}_k$ contains all the parameters we want to estimate for the source k. Note that although this was performed for the joint estimation, we get a similar procedure for the separate estimation except that two covariances matrices are involved and two vectors are estimated. In (3.16) the covariance matrix $\mathbf{R}_{k,t-1}$ is defined as $E\Big\{\breve{\mathbf{x}}_{k,t-1}\breve{\mathbf{x}}_{k,t-1}^T|y_{1:t}\Big\}$. It is important to notice that the estimation of $\mathbf{R}_{k,t-1}$ is done using observations till time t, which is a fixed-lag smoothing treatment with $lag = 1$. As mentioned before, the relation between the partial state at time $t-1$ and the full state at time t is $\breve{\mathbf{x}}_{k,t-1} = \mathbf{S}_k\mathbf{x}_t$. The following key relation is used in the partial state covariance matrix computation:

$$\mathbf{R}_{k,t-1}^{-1} = \mathbf{S}_kE\Big\{\mathbf{x}_t\mathbf{x}_t^T|y_{1:t}\Big\}\mathbf{S}_k^T \tag{3.17}$$

Notice here the transition from the fixed lag smoothing with the partial state to the simple filtering with the full state. This fact justifies the selection of the partial state at time

$t-1$ from the full state at time t. This selection is possible due to the augmented form matrix $\mathbf{F}_k$. *In pratice, the expectation is done using a forgetting factor* ($\lambda < 1$)*. If we use the alternative reduced state to estimate separatly the ST and LT parameters, then it follows that we can use differents forgetting factor.* This point is not investigated, but we claim that by relaxing the covariances matrix of, e.g., LT parameters when the period is changing can be useful for a quicker adaptation of the system.

The innovation variance is simply deduced as the first component of the matrix $\mathbf{R}_{k,t-1}^{-1}$. The estimation of the observation noise power σ_v^2 is achieved by maximizing the log likelihood function $\log P\left(y_t|\mathbf{x}_t,\sigma_v^2\right)$ relative to σ_v^2. The optimal value can be easily proved to be equal to

$$\hat{\sigma_v^2}^{(t)} = E\left[\left(y_t - \mathbf{h}^T\hat{\mathbf{x}}_{t|t}\right)^2\right] + \mathbf{h}^T\mathbf{P}_{t|t}\mathbf{h} \tag{3.18}$$

The time index (t) in $\hat{\sigma_v^2}^{(t)}$ denotes the iteration number. The computation of the partial covariance matrix $\mathbf{R}_{k,t-1}$ is achieved in the $E-step$. This matrix depends on the quantity $E\Big\{\mathbf{x}_{k,t}\mathbf{x}_{k,t}^T|y_{1:t}\Big\}$ the definition of which is

$$E\Big\{\mathbf{x}_t\mathbf{x}_t^T|y_{1:t}\Big\} = \hat{\mathbf{x}}_{t|t}\hat{\mathbf{x}}_{t|t}^T + \mathbf{P}_{t|t} \tag{3.19}$$

where the quantities $\hat{\mathbf{x}}_{t|t}$ and $\hat{\mathbf{P}}_{t|t}$ are respectively the full estimated state and the full estimation error covariance computed using Kalman filtering equations. The algorithm needs an accurate initialization, which will be discussed afterward. Let us call with $\hat{x}_{k,t}$ the estimation of the source k at time t.

Adaptive EM Kalman Algorithm

- E-Step. Estimation of the sources covariance

$$\begin{array}{rcl}
\mathbf{K}_t & = & \mathbf{P}_{t|t-1}\mathbf{h}(\mathbf{h}^T\mathbf{P}_{t|t-1}\mathbf{h} + \hat{\sigma_v^2})^{-1} \\
\hat{\mathbf{x}}_{t|t} & = & \hat{\mathbf{x}}_{t|t-1} + \mathbf{K}_t(\mathbf{y}_t - \mathbf{h}^T\hat{\mathbf{x}}_{t|t-1}) \\
\mathbf{P}_{t|t} & = & \mathbf{P}_{t|t-1} - \mathbf{K}_t\mathbf{h}^T\mathbf{P}_{t|t-1} \\
\rightarrow & = & \left(\mathbf{I} - \mathbf{K}_t\mathbf{h}^T\right)\mathbf{P}_{t|t-1}\left(\mathbf{I} - \mathbf{K}_t\mathbf{h}^T\right) + \mathbf{K}_t\mathbf{K}_t^T\hat{\sigma_v^2} \\
\hat{\mathbf{x}}_{t+1|t} & = & \hat{\mathbf{F}}\hat{\mathbf{x}}_{t|t} \\
\mathbf{P}_{t+1|t} & = & \hat{\mathbf{F}}\mathbf{P}_{t|t}\hat{\mathbf{F}}^T + \mathbf{G}\hat{\mathbf{Q}}\mathbf{G}^T
\end{array}$$

- M-Step. Estimation of the AR parameters using linear prediction. $k = 1, ..., K$

$$\begin{array}{rcl}
\hat{x}_{k,t} & = & (\hat{\mathbf{x}}_{k,t|t})_{[1,1]} \\
\mathbf{R}_{k,t-1} & = & \lambda\mathbf{R}_{k,t-2} + (1-\lambda)\mathbf{S}_k(\hat{\mathbf{x}}_{t|t}\hat{\mathbf{x}}_{t|t}^T + \mathbf{P}_{t|t})\mathbf{S}_k^T \\
\sigma_k^{(t)} & = & (\mathbf{R}_{k,t-1}^{-1})_{(1,1)}^{-1} \\
\mathbf{v}_k^{(t)} & = & \sigma_k^2\mathbf{R}_{k,t-1}^{-1}[1,0\cdots0]^T \\
\hat{\sigma_v^2}^{(t)} & = & \lambda\hat{\sigma_v^2}^{(t-1)} + (1-\lambda)\left(y_t^2 - 2y_t\mathbf{h}^T\hat{\mathbf{x}}_{t|t} + \mathbf{h}^T\Big(\hat{\mathbf{x}}_{t|t}\hat{\mathbf{x}}_{t|t}^T + \mathbf{P}_{t|t}\Big)\mathbf{h}\right)
\end{array}$$

As previoulsy mentioned, it is essential that the symmetry of the covariance matrices

$\boldsymbol{P}_{t|t-1}$, $\boldsymbol{P}_{t|t}$ is maintained throughout the updates. The way as $\boldsymbol{P}_{t|t}$ appears in (3.2) involves subtraction and this can cause loss of symmetry and positive definiteness due to rounding errors. Josephs form [73] covariance update avoids this at the expense of computation burden:

$$\begin{aligned} \mathbf{P}_{t|t} &= \mathbf{P}_{t|t-1} - \mathbf{K}_t \mathbf{h}^T \mathbf{P}_{t|t-1} \\ \rightarrow &= \left(\mathbf{I} - \mathbf{K}_t \mathbf{h}^T\right) \mathbf{P}_{t|t-1} \left(\mathbf{I} - \mathbf{K}_t \mathbf{h}^T\right) + \mathbf{K}_t \mathbf{K}_t^T \hat{\sigma_v^2} \end{aligned} \tag{3.20}$$

Only subtraction is squared and preserves symmetry.

3.7 Other Approaches using Kalman Filter

There are several works which use the Kalman Filter for similar problems. In fact Kalman Filters receive interests for the problem of speech enhancement. In [14] the problem is to separate an AR(p) source corrupted by an AR(q) noise. A EM method is used to estimate the spectral parameters of the speech signal and noise process. The E-Step uses a fixed lag Kalman Smoother while the M-Step uses a nonstandard Yule-Walker equation set, in which correlations are replaced by their a posteriori values. Wan and Nelson, in [120], consider also the problem of colored noise. They transform the speech enhancement problem into a source separation problem by considering the noise as an AR source. The problem, defined as a Non-Linear mixture problem, is to find the sources and the weigths of the non-linear fonction using a KF approach. As the problem is non linear, it leads to the use an Extended Kalman Filter (EKF). The idea of the EKF is to apply the KF to a linearized version of the state-space model, via a first-order Taylor series expansion. They split the estimation problem by using a Dual-EKF which is composed of two EKF running in parrallel and interacting, one for the source and one for the weight. In [61] Carpentier and Févotte use a Kalman Filter to reconstruct the source while the AR sources parameters are estimated via a Maximum Likelihood estimation. They consider the case of two Gaussian AR sources (of order 1 or 2) linearly mixed, and one or two observations. For the mono-sensor separation problem the system works well when the two sources are disjoint. In their simulation they obtain good results for a low pass and a high pass source. Couvreur [70] proposes an EM approach in the context of AR coefficients identification from a codebook. They consider the case of several Gaussian AR sources (2 in the simulations) with only one observation. The E-step leads to a Smoothing approach to estimate the variances of the innovations and of the additive noise, and for which they use the RTS Smoother. The analysis is done with Short order AR model (AR(2)) and the goal is to find the good subset in the codebook.

All this approachs use only Short Term AR Model. The Sinusoidal Model of McAuley and Quatieri [52] is also used. In the context of source separation (under determined not Mono-Microphone), in [121] they propose to use a Sinusoidal plus AR Model. The State Space Model derived is close to the one we use, with the difference that they have two transition matrices: one for the AR model and one for the sinusoidal model. Note that for the sinusoidal model the amplitudes of each harmonic and the fondamental frequency have to be found. The sources are reconstructed by using a Kalman Smoother and they investigate different learning approches for the parameters. The simulations show an improvement for the separation when the sources are truly harmonic, but the model is vulnerable to overfitting when the energy of one or more sources are locally near-zero.

3.8 Simulations - Synthetic Signals

In this section we show some simulations results. The used signals are synthetic and follow the model described in Chapter 2. For the first simulation we compare different versions of the algorithm to emphasize the choice of the model. The model we have defined can be simplified, since one can use only the Short Term part with differents order, or can only use the Long Term part. This allows to design differents algorithm defined as:

- **AR-STLT**: Short Term + Long Term Auto-Regressive Model
- **AR-ST(p_k)**: Short Term Auto-Regressive Model of order p_k, p_k being the order used to generate the sources (here the same for all the sources)
- **AR-ST($p_k + \tau_k$)**: Short Term Auto-Regressive Model of order p_k, τ_k being the periods of the sources (different)
- **AR-LT**: Long Term Auto-Regressive Model

AR-STLT is the previously defined model. The State Space Model (SSM) used with this algorithm is defined, for each source, in (3.11). **AR-ST(p_k)** is a traditional AR model of order p_k. The SSM by source uses only the first sub-matrix F_{11} in (3.11). In the simulations p_k is the same for all the sources. **AR-ST($p_k + \tau_k$)** is an AR model with the same order as the source. Finally **AR-LT** is the Long Term model. Its SSM is composed by the last sub-matrix F_{22} for each source.

3.8.1 Used parameters to generate synthetic signals

The synthetic signals are randomly generated. For each source, the Short Term coefficients are generated using the Levinson Algorithm (Appendix F.1), the Long Term coefficient and the variance are uniformly generated between $[.75; 1[$ (1 is excluded in order to avoid singularity) and $[.75; 1]$ respectively. The set of frequencies is $[80; 400]\,Hz$, which includes the most part of natural fondamental frequencies of humans voice. This leads also to non-integer periods. We work with 2 sources and the sampling frequency is $F_s = 8000Hz$. A zero mean white Gaussian noise is added to the mixture, and its variance is such to give the desired Signal to Noise Ratio (SNR).

For all simulations when the spectrum is shown, the transient part is discarded. The adaptive algorithm needs first to converge before tracking the sources. The spectrum is computed with a Hann window, which makes the spectrum smooth. Moreover, we use a zero padding of factor 4 for the computation of the Fourier Transform, using the Fast Fourier Transform (FFT) Algorithm (see also Appendix F.10).

3.8.2 Comparison of Models

In case of synthetic data, we know perfectly the parameters used to generate the signals and the noise. Two kind of simulations can be used to compare the models, i.e. the separation of the sources with known (Filtering) or unknown parameters (Estimation).

In order to compare the results we use four criteria. Common criteria used in evaluation of BSS [34, 35] are:

- Source to Distortion Ratio (SDR)
- Source to Interference Ratio (SIR)
- Source to Artefact Ratio (SAR)

These criteria are described in Appendix F.4 [74, 80, 81]. The fourth criterion is the Mean Square Error (MSE). The SIR measures the level of distortion due to the others sources. The SAR corresponds to the distortion added to the extracted source by artefacts components, which are not explained by the interference and which, generally, are coming from the algorithm. Finally, the SDR corresponds to the distortion induced by all the noises (artefact, interference). Note that the higher the value is, the better the result.

3.8.2.1 Filtering results comparison

For the **AR-STLT**, **AR-ST**(p_k) and **AR-LT** we use the parameters used to generate the signals. We expect that the combining the Short and Long Term gives better results than using them individually. We also expect that at low SNR the peaks informations are more reliable than the spectral shape. Concerning **AR-ST**$(p_k + \tau_k)$, which is a high order AR model, the parameters are computed on the sources individually, by using the algorithm described in section F.2.

We denote by "filtering" that only the filtering part of the algorithm is used. So, parameters are not estimated and the SSM is not updated during the process. In this case we cannot compare the algorithm denoted as **Joint-EMK** and **Alt-EMK** in section 3.6.1, because the difference comes from the estimation part. In the considered simulations the filtering results are the best results we can obtain.

Simulations consist on filtering the observation with the four algorithms. We study the evolution of the four criteria with respect to the input SNR. The parameters of the used signals are given in Appendix F.10. The results of the analysis are shown in Figure 3.2.

Several points have to be emphasized in this filtering case. First of all the **AR-ST**$(p_k + \tau_k)$ gives better results. This is not surprising because, in the filtering case, with the good parameters the model which captures the entire sources is the more accurate. However, in the case of parameter estimation, it requires to estimate many coefficients. The **AR-ST**(p_k) gives the worst results except at low SNR for the MSE and at high SNR for the SAR. Finally, the proposed model seems to be a good compromize: at high SNR it converges to the **AR-ST**$(p_k + \tau_k)$ and gives better result at low SNR than the **AR-ST**(p_k). By looking at the results, we observe that the long term aspect is essential in order to separate the source. When the SNR is low, the spectral shape is degraded. At high SNR, using both short and long term correlation the result are improved and tend to the complete model performance with fewer parameters to estimate.

Note that, in the filtering case, this results are just informative about the upper performance bound and don't reflect the accuracy of the separation in a real context. The generated synthetic signals are almost stationary, the non stationary signals are also investigated in this chapter.

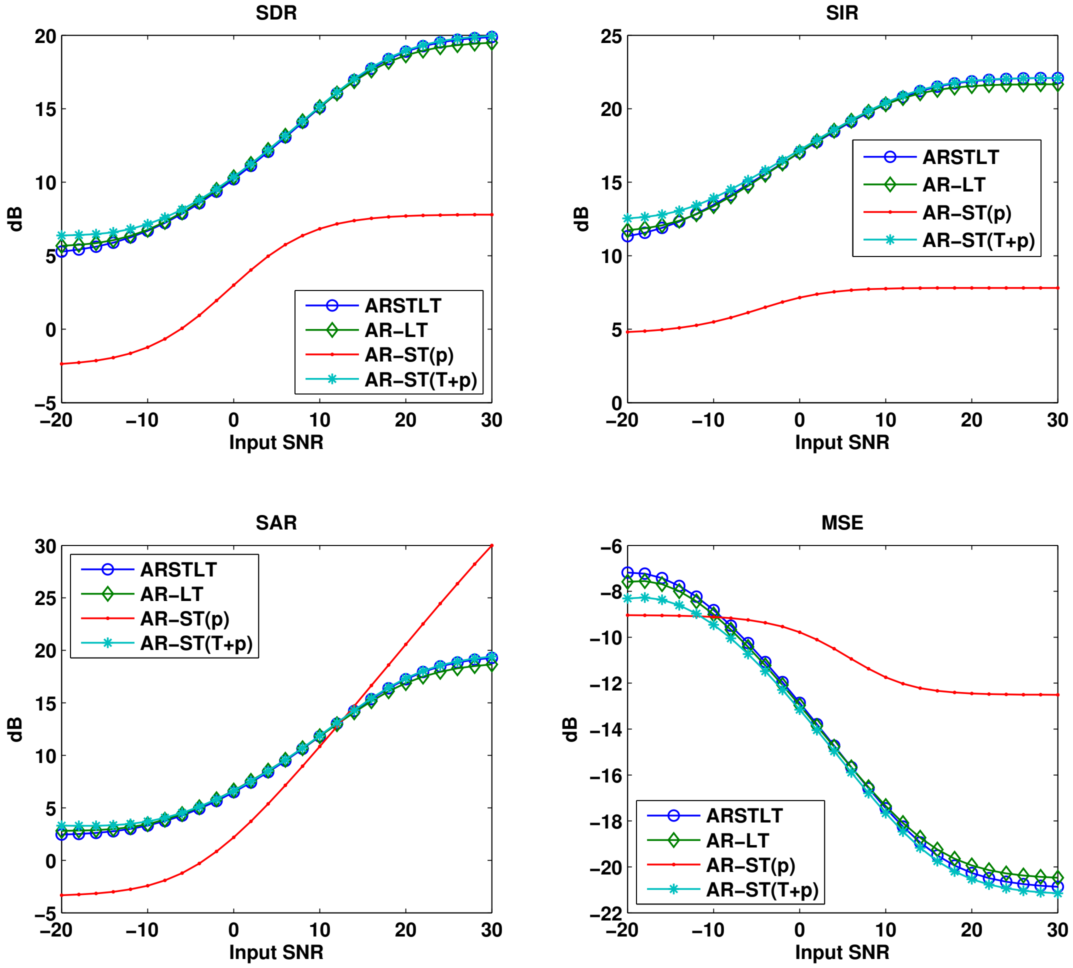


Figure 3.2: Models comparison. SDR, SIR, SAR and MSE in the filtering case.

3.8.2.2 Estimation results comparison

Here the signals are the same as in the previous simulation. The initialization of the sources is done as follows. We assume that we know the number of sources, the short term order and the periods. The short term coefficients are initially set to 0 (with $a_{k,0} = 1$) and the long term coefficients close to 1 ($b_k = .99$). We prefer to over-estimate the long term coefficient in order to emphasize the periodicity assumption of the sources. The variances of the Short plus Long Term prediction error are equal and set to 1. We give the correct variance of the additive noise which evaluates. *We call this initialization as "fixed initialization".* We compare the four previously defined algorithms in the case of parameter estimation. The results are reported in Figure 3.3 and indicate that:
A general ascertainment is that the AR based algorithms (**AR-ST**(p_k) and **AR-ST**$(\tau_k + p_k)$) give now the worst results. Since no periodicity constraint is imposed for the estimation of the coefficients, i.e. the estimated spectral shape of a source is not related to its spectral peaks, then it models a combination of sources, but not the sources separately. The algorithms using the long term correlation are the best and in the simulations they lead to comparable results with very small differences.

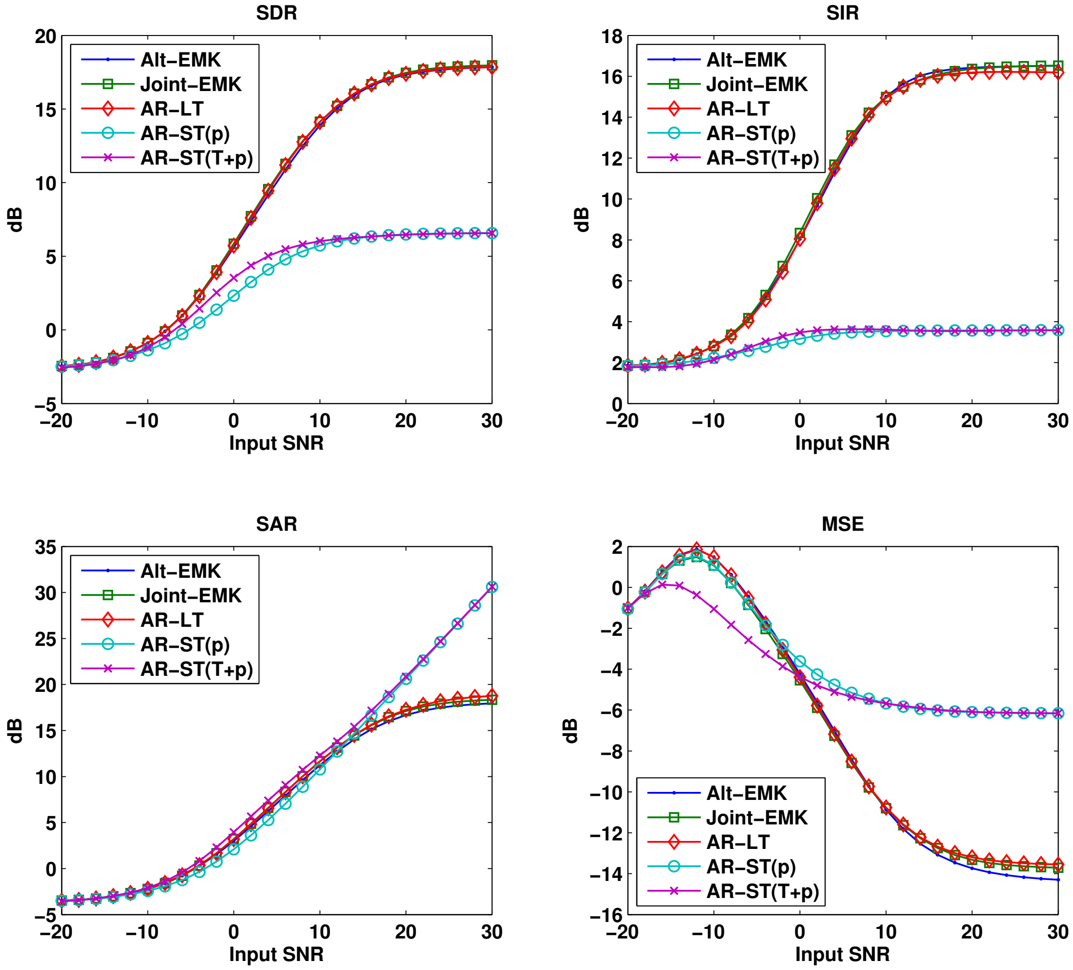


Figure 3.3: Models comparison. SDR, SIR, SAR in the Estimation case.

In this section we have presented some simulation results about the filtering and estimation of differents versions of the model. In the filtering case, where we perfectly know all parameters, a high order AR model gives the best results and it is followed by the proposed model. However, this high order AR is composed of a high number of parameters and in the estimation case it fails. The Long Term model is more robust than the short term one, in terms of evaluation criteria it is important to take the periodicity into consideration. For such stationary and synthetic signals it is hard to measure the improvement due to the short term correlation. However, in chapter 5 we will investigate the same simulation on real and not completly stationary signals. In view of the results obtained in this section we will continue to work with the proposed model.

3.8.3 Amplitudes tracking

In this section we use sources with amplitudes variations. In this case, the two sources are weighted with windows (see Appendix F.10.2.1]. The goal is to show the behavior of the algorithm in a more realistic tracking task. The first source is highly attenuated at the end and the second is attenuated around its middle. We use the same initialization as in the previous simulation for the two algorithms **Joint-EMK** and **Alt-EMK** and the SNR is 20dB. As we can see in Figures 3.4 and 3.5, **Joint-EMK** does not track the sources very well. The end of the first estimated source does not follow the attenuation and for the second estimated source the tracking of the attenuation is better for the **Alt-EMK**. The evaluation criteria, averaged over 100 realizations, are given on the top of Table 3.1 and they also indicate that the **Joint-EMK** gives slightly worse results. The results, shown in Figure 3.4, highlight two points: the tracking is reasonable, the estimate follows the original, and when only one source is present the method tries to find a second one estimate. This second remark explains what happens on the attenuated part (see Figure 3.5). The algorithm takes, for the missing source, the information related to its period on the present source. It is the same kind of overfitting problem that Olsson et al. describe when the energy of one source (or more) is locally near-zero [121].

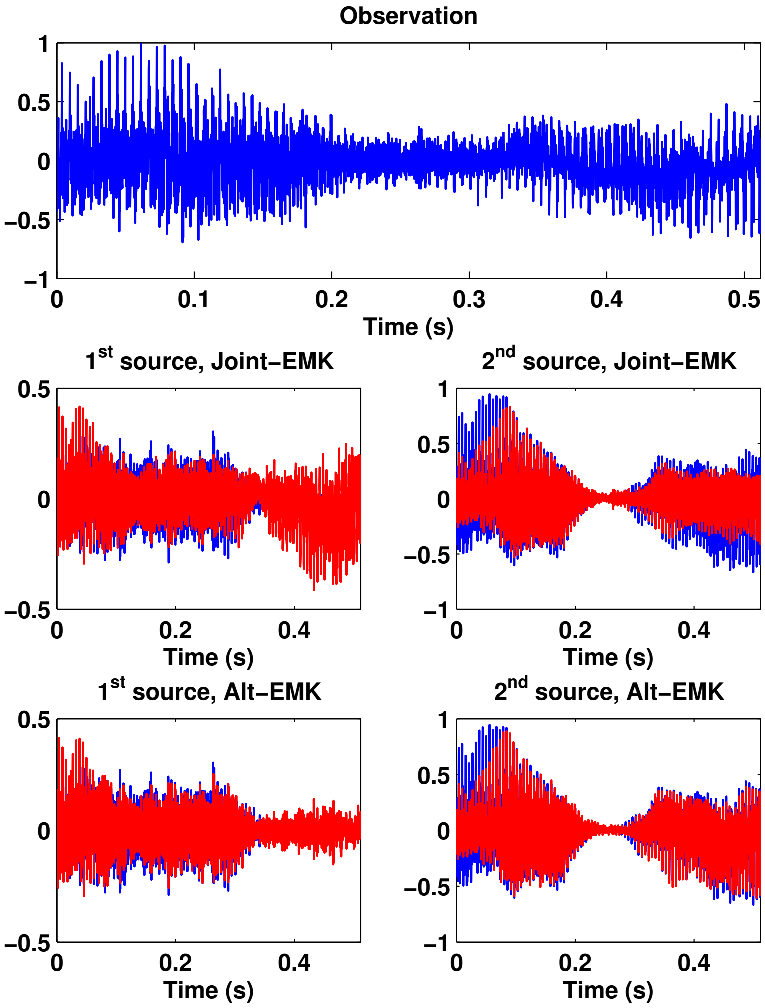


Figure 3.4: With parameters estimation results, weighted sources.

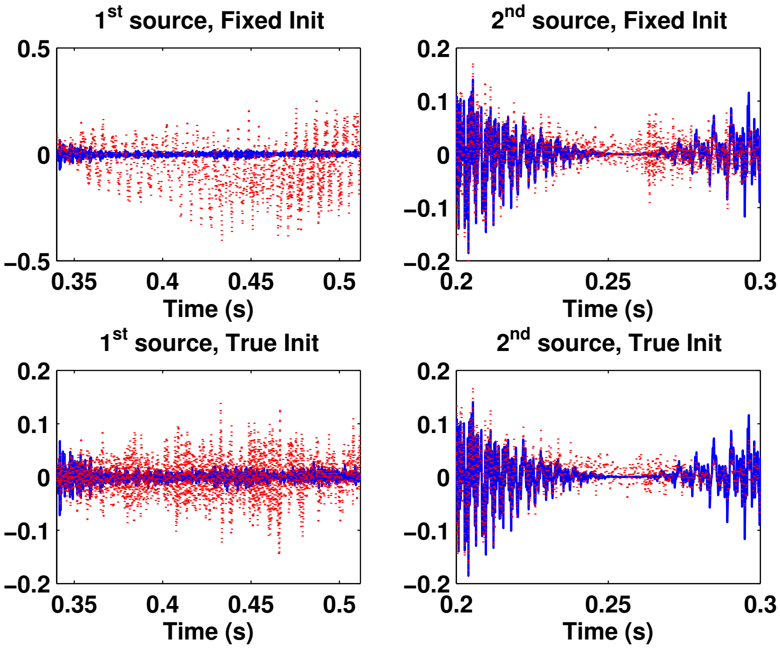


Figure 3.5: Zoom on the attenuated part.

Table 3.1: Criteria in dB for periods and amplitudes variations.

Case	Algorithm	SDR	SAR	SIR	MSE
Amplitudes Variation	**Joint-EMK**	14.5 (17)	18.3 (16.5)	19.3 (16)	-27.8 (-13)
Amplitudes Variation	**Alt-EMK**	15.3 (17)	18 (16)	20 (16)	-28 (-13.5)
Periods Variation	**Joint-EMK**	7.3 (17)	15.5 (16.5)	9.2 (16)	-15.6 (-13)
Periods Variation	**Alt-EMK**	10.5 (17)	16.8 (16)	13.4 (16)	-15.5 (-13.5)

3.8.4 Periods Variations

Now we consider that periods of sources vary with respect to time in a parameters estimation context. The observation is composed of two sources. We consider two scenarios: the first source has a constant period whereas the period of the second source varies. The frequency of the two sources may vary. In both case the variation happens after the convergence to the first frequency.

In the examples shown in Figure 3.6 and Figure 3.7, at a certain moment the frequencies intersect. We show the Short Time Fourier Transform (STFT) of the observations, of the sources and of the estimated sources. The STFT is computed with 4096 fft points for a segment of 256 samples, an overlap of 87.5% and using a Hann window. In both simulations the sources are generated with the ”sinusoids plus noise” model [11]. The duration of the signals is 1.28 s, the sampling frequency is $8KHz$. The white Gaussian noise is added and the SNR is approximatively 20 dB. The observation is composed of two sources and noise. The algorithms are both initialized, as before, with zeros for the Short Term coefficients and close to one for the Long Term. We assume that we know the periods and we compare the algorithms, defined in section 3.6.1, namely **Joint-EMK** and **Alt-EMK**. Table 3.1 contains the evaluation criteria (average of 100 realizations) for a two sources mixture with noise ($SNR = 20dB$). The value between brackets corresponds to the filtering case for fixed periods from the previous section.

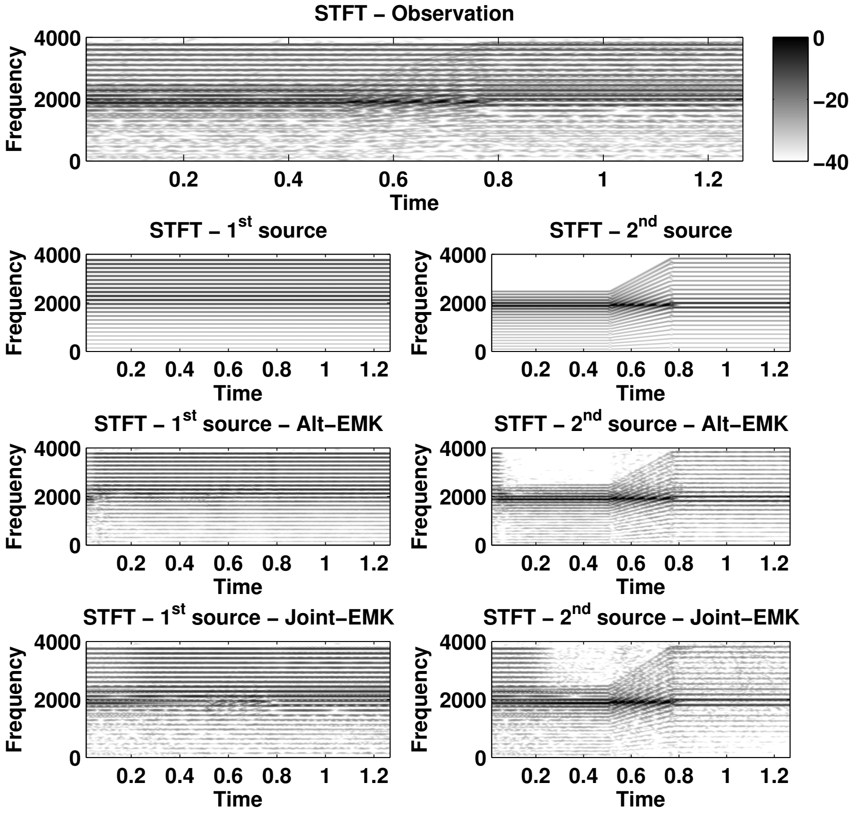


Figure 3.6: Comparison example, STFT for **Joint-EMK** and **Alt-EMK**, a fixe and a varying fondamental frequencies.

In Figure 3.6 the first source has a fondamental frequency of 160 Hz. The second source starts with a fondamental frequency of 117 from 0 to 512 ms, then the frequency increases linearly to 181 Hz in 310 ms and remains at this frequency until the end. We can observe on the estimated second source that the **Alt-EMK** converges faster that **Joint-EMK**. The residuals of the frequency comb of the first source stay longer in the second source's estimate of the **Joint-EMK**. Also the spectral shape of the first source is better modeled by the **Alt-EMK**. For the **Joint-EMK**, the estimate the first source gives a residual of the formant of the second source.

In the second example, in Figure 3.7, the two sources have time varying fundamental frequencies. For the two sources the frequency varies linearly but at different moments. For the first source the frequency vary from 360 to 260 Hz from 380 to 768 ms. The frequency of the second source is going from 250 to 80 Hz between 512 and 768 ms. In this example it is clear that the **Alt-EMK** possesses a better adaptation than the **Joint-EMK**. During the frequencies variation part both algorithms encounter separation problem, but in this particular example the **Alt-EMK** separates the sources while **Joint-EMK** fails. The evaluation criteria, averaged over 100 realizations, are given in Table 3.1 and indicates that the **Joint-EMK** gives slightly worse results.

*The simulations done on "non stationary" synthetic signals indicate that the **Alt-EMK** gives better separation results than the **Joint-EMK**. However in simulations with real signals, we will still consider the two algorithms as the evaluation criteria are not so different.*

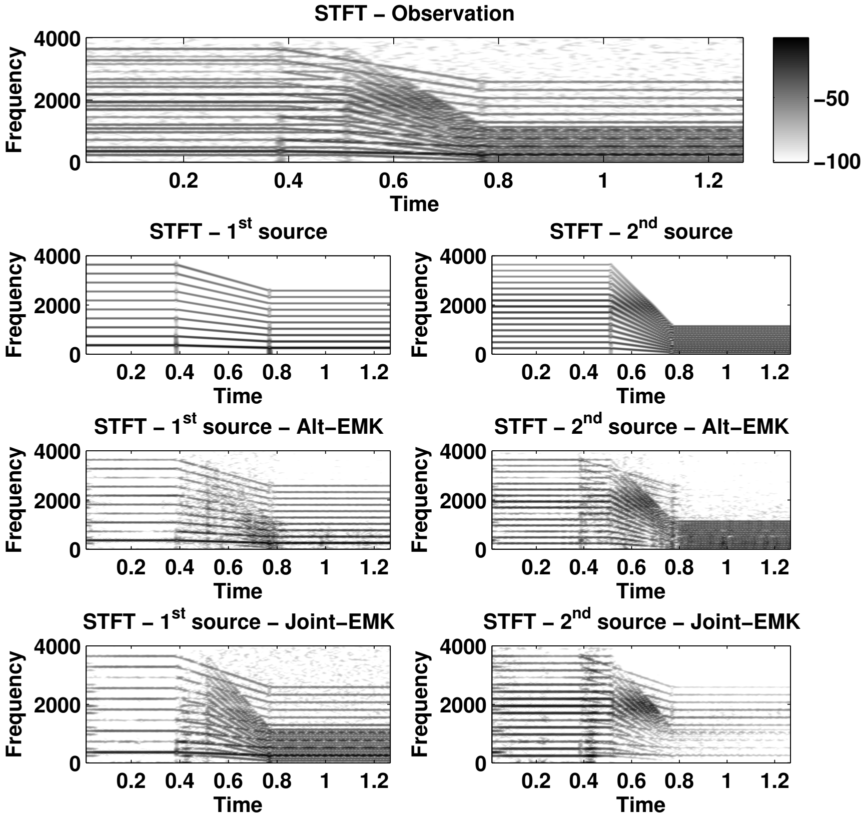


Figure 3.7: Comparison example, STFT for **Joint-EMK** and **Alt-EMK**, the two fondamental frequencies vary.

3.9 Summary of Chapter 3 and discussion

In this chapter we have used the adaptive EM-Kalman algorithm for the blind audio source separation problem. The model, presented in Chapter 2, takes into account the different aspects of speech signals production (Short and Long term correlation) and is compared to other simplified versions. We have presented the basis of Kalman Filtering and its use in an adaptive way in sections 3.1-3.4. In the algorithm the sources are jointly estimated. The traditional smoothing step is included into the algorithm due to the extended form of the State Space Model. The details of the algorithm are given in section 3.6. In this chapter the simulations, in section 3.8, are done on synthetic data. The results with real data will be presented in the last chapter of this part and compared to an other algorithm presented in the next chapter. We have considered two models for the study of non stationary synthetic signal. In the simulations done on signals with amplitude variations the used model is presented in 2 while for non-constant frequency signal we have used the sinusoids plus noise model. Note that with this second model, when the frequencies vary the amplitudes vary as well. With non stationary signals the Alternate estimation gives better results than the Joint Estimation, while it was not the case for stationary signals. For this reason, we will still consider the two algorithms for analysing Real Speech Signals. This algorithm would be more complete if an other process aimed at estimating the number of active sources could work in parallel. The order of the Short Term part and the periods of sources should also be considered.

Chapter 4

Frame Based Source Separation

In this chapter we investigate another BSS Algorithm, it is a Frame Based one. It means that the observation signal is cut into frames and that the source separation process is also applied on frames. Analyzing the signal by segments is the opposite view of the adaptive system presented in chapter 3. It assumes that we can consider the signal stationary and the parameters constant during a short duration, obviously the length of the analysis frame is crucial. If the length of the window is too short then the Long Term correlation of the signal cannot be estimated. If it is too long the stationarity cannot be considered.

The vision of the frame based algorithm we have is slightly different from the general one. We propose to separate the source, in a non iterative way, with the knowledge of the parameters. Generally frame based algorithms iterate between the separation and the estimation process of a frame. We propose to, firstly, estimate the parameters in a frame without separating the sources, just their correlation sequences and then to use the estimated parameters to separate the sources.

The chapter is organized as follows: First we introduce the frame decomposition of a signal in section 4.1 in which we briefly explain some related approaches. Then in section 4.2 we explain and motivate our approach. In section 4.3 we recall the model of chapter 2 and add some notations used in this chapter. Section 4.4 deals with the parameters estimation while section 4.8 explain the separation process. We present two algorithms based on the Itakura-Saito distance to estimate the parameters, the first one is a naive interpretation of the distance while the second one deals with its true minimization. The Source Separation algorithm was originally developed in a Variational Bayesian Context and was simplified to its current version. We finally show some simulation results in section 4.7 on synthetic spectra and we discuss the approach.

4.1 Frame Based Algorithm

4.1.1 Introduction

In chapter 2 we have analyzed a speech signal and we have observed that the speech signal (and also music signal) is not stationary. When dealing with non stationary signals two visions exist:

- The first one is to track and process the signal in an online fashion, this leads to use adaptive algorithm as in chapter 3.
- The second one, consists in buffering a certain amount of data before processing the signal.

In this Chapter we consider the second proposition. It considers that the signal is stationary during this laps of time and uses all the data for the processing. Frame based algorithms are very popular for this reason. For example, if a spectral analysis is done, it needs a lot of data to reduce the Time-Frequency uncertainty. Cutting the signal into frames implies to use an analysis window. If another window than the rectangular one is used, the problem of Perfect Reconstruction (PR), if needed, appears. The Perfect Reconstruction (PR) means that if we cut a signal into frames then we can perfectly reconstruct it with the weighted frames. As the time domain multiplication leads to a frequency domain convolution the spectral properties of the window are also very important to be able to distinguish closed peaks.

4.1.2 Windowing

When dealing with periodic signals (e.g. composed of sinusoids) some considerations must be taken into account. If we want to find the period for example, we need, at least, one period of the signal. So the length of the window cannot be shorter than the period. If the signal is composed of several sinusoids, it results that its spectrum is composed of frequency peaks. As just mentioned, in the spectral domain a convolution between the Fourier transform of the window and of the signal occurs. As the window is temporally finished its Fourier transform is not, and then interferences between peaks appears. Generally, the window functions used are non-negative smooth bell-shaped curves (except for the rectangular and triangular ones) and theirs spectra are composed of a main lobe (with a non negligible width) and adjacent lobes (side-lobes). The windows definitions and properties are recalled in Appendix F.5. The width of the main lobe and the power of the side-lobes can pose separation problem, as a result the Spectral properties of an analysis window are crucial when a spectrum is analyzed. The bell-shaped windows have the property to decrease the side-lobe power. Any window which are zero-valued outside of the analyzed interval are multiplied by a rectangular window. The bell-shaped windows use this fact to reduce the side-lobes, they are composed of raised cosines function and put a negative contribution at the first secondary lobe positions. This reduces the secondary lobe but increase the width of the main lobe. If the signal has to be reconstructed after the processing, the perfect reconstruction constraint is needed. This means that the grouping of windowed signal segments have to be equal to the original signal and, obviously, without adding discontinuity between two consecutive frames. Since the window needs to decay towards its edges, consecutive frames need to overlap.

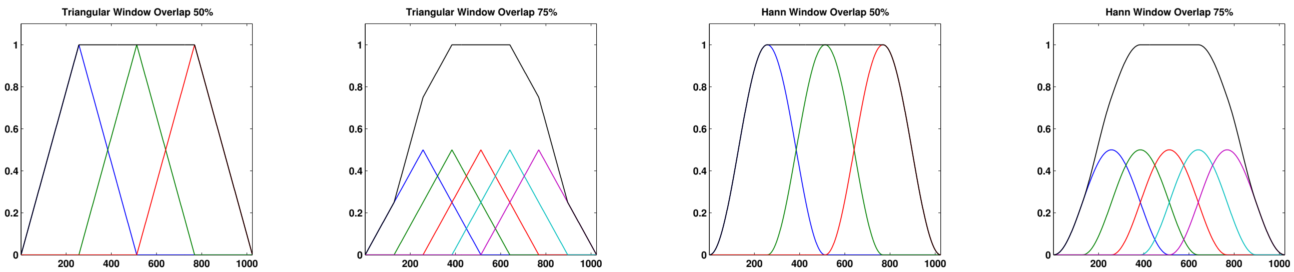


Figure 4.1: Perfect reconstruction windowing, Hann and Triangulare window with an overlap of 50% and 75%.

4.1.2.1 Perfect Reconstruction

Just like the original data signal will be cut into a series of windowed frames, a processed signal will be reconstructed by superposing its reconstructed windowed frame segments, it is called the overlap and add method. Let M be the hop size (time jump) from one frame to the next, then a perfect reconstruction (PR) window w_t requires

$$\sum_{i=-\infty}^{\infty} w_{t-iM} = 1 \; , \quad \forall n \tag{4.1}$$

Figure 4.1 shows the cases of relative overlap of $(N-M)/N = 50\%, 75\%$ for the Hann and the triangular windows, both the individual windows and their sums are shown for a finite set of windows.

Note that one could consider extensions to non-PR windows, in which the superposition of windowed signal frames could be followed by a zero-forcing rescaling with $1/(\sum_{i=-\infty}^{\infty} w_{t-iM})$. The PR window that will be used in the simulations in this chapter is a Hann window [122]

$$w_t = \frac{1}{2}\left[1 - cos\left(2\pi\frac{t}{N}\right)\right] \; , \quad t = 0, 1, \ldots, N-1 \, . \tag{4.2}$$

The use of a bell-shaped window also has another one effect. *As the window is maximum at its center and decay to a small value, it results that the stationarity is enforced because the central samples have more weight than the other.* So the introduction of a window allows to reduce the approximation error.

4.1.3 Related work

The frame based algorithms are very popular. As working on a short time duration segment allows to consider a signal segment as stationary, a wide range of algorithms have been developed in this spirit. The analysis of a signal in the spectral domain needs an accurate estimation of the spectrum. While the periodogram uses the whole frame, the Welch periodogram uses several consecutive frames (or sub frames) for the estimation. Cutting the signal into frames give, also, an another interpretation of the time. Instead of referring to a sample index it refers to a frame index, as this, a non stationary signal is assumed to be composed of consecutive stationary sub-signals. Generally the method used to analyze a frame depends on the application, if the application is speech/music related then the frames have to be long enough to catch the inherent periodic components.

The NMF related methods work generally with a time-frequency representation of the signal (generally the Short Time Fourier Transform (STFT)) [18]. Working with a STFT is motivated by the fact that audio sources are generally weakly overlapping in the Time-Frequency domain. For each frames NMF tries to find a weighted combination of a pre-defined basis matrix (which is generally composed of individual harmonic spectrum), the matrix containing the weight evolution is then an activation matrix. This matrix expresses what basis is active at what time which generally results on: what note is played at what moment in an automatic transcription problem [16].

Other methods are based on the decomposition of the signal into a dictionary which can be redundant. Searching over an extremely large dictionary for the best matches is computationally unacceptable for practical applications. Mallat and Zhang [57] proposed a greedy solution which is known from that time as Matching Pursuit (MP). The algorithm iteratively generates, for any signal and any dictionary, a sorted list of indexes and scalars which are sub-optimal solutions to the problem of sparse signal representation. The signal into a frame is then defined by its sparse representation.

Leveau [15] uses instrument and frequency specific atomes that allow to conjointly find the note and the instrument, using the time dimension the atoms are then grouped into molecule to refine the analysis. When dealing with time varying signal, also if we assume that they are stationary, the question of the length of the window is important. In [79] the authors use two different Modified Discrete Cosine Transform (MDCT) basis, one to represent the tonal part and one for the transient part (e.g. the attack of a note).

The source separation technique, as presented in [44], suggests the use of GMM to model the sources statistical behavior. In this approach, each source model is composed of Q states. The state q of the k^{th} source is represented by a spectral shapes ($\sigma^2_{q,k}(f)$) and a a priori distribution. In this case, the codebook is formed by GMM parameters trained from sample data representative of the sources.

We will consider a different approach adapted to the parametric model used in this chapter.

4.2 Parameters and Sources estimation

4.2.1 EM Like Algorithm

In the EM approach, the idea is to decompose the observed data into its signal components and then estimate the parameters of each signal component separately. The algorithm iterates using the current parameter estimates to decompose the observed data better and thus improve the next parameter estimates [118]. For AR Sources, the AR source parameters are estimated by linear prediction on the reconstructed source correlations, which are the sum of the sample correlations of the estimated source plus the correlation of the source estimation error (orthogonality property of LMMSE estimation) [71].

4.2.2 VB-EM Like Algorithm

Variational Bayesian methods, also called ensemble learning, are a family of techniques to approximate intractable integrals arising in Bayesian inference and machine learning. They can be used to lower bound the marginal likelihood (i.e. *evidence*) of several models with a view to performing model selection, and often provide an analytical approximation to the parameter posterior probability which is useful for prediction.

A recent tutorial on Variational Bayesian (VB) estimation techniques can be found in [123], see also [119]. It provides an approximate technique to determine the posterior probability density function (pdf) of the quantities to be estimated. Let θ denote the vector of all quantities to be estimated, including parameters and possibly signals (e.g. the "hidden variables" in EM terminology) and Y denotes the measurements. In many problems, the joint posterior pdf $p(\theta|Y)$ can be complicated to determine. Consider now a partition of θ into K subgroups of quantities that will get estimated per subgroup $\theta = \{\theta_k,\ k = 1, \ldots, K\}$. The idea of VB is to approximate $p(\theta|Y)$ by a product form $q(\theta|Y) = \prod_{k=1}^{K} q(\theta_k|Y)$ where the $q(\theta_k|Y)$ in general will differ from the true marginal pdfs $p(\theta_k|Y)$. The $q(\theta_k|Y)$ are determined by minimizing the Kullback-Leibler distance between $\prod_{k=1}^{K} q(\theta_k|Y)$ and $p(\theta|Y)$. This leads to the following implicit relations

$$\ln q(\theta_k|Y) = \mathrm{E}_{q(\theta_{\bar{\mathbf{k}}}|Y)} \ln p(Y, \theta_k, \theta_{\bar{\mathbf{k}}})\ , k = 1, \ldots, K \tag{4.3}$$

where $\theta_{\bar{\mathbf{k}}}$ is θ minus θ_k, hence $\theta = \{\theta_k, \theta_{\bar{\mathbf{k}}}\}$. In practice, (4.3) needs to be solved iteratively by consecutively sweeping through $k = 1, \ldots, K$, at all times using for $q(\theta_{\bar{\mathbf{k}}}|Y)$ the latest version available. This iterative process can be shown to converge monotonically. Typically, when $p(Y|\theta)$ and the prior $p(\theta)$ are exponential pdfs (typically Gaussian), then one can see from (4.3) that $q(\theta_k|Y)$ will also be an exponential pdf. Note that Variational Bayesian techniques can also be applied in the presence of deterministic unknowns θ_D. There are two ways to think about deterministic unknowns:

(i) as truly deterministic, with prior $p(\theta_D) = \delta(\theta_D - \theta_D^o)$ where θ_D^o is the unknown true value of θ_D; in other words, $\theta_D \sim \mathcal{N}(\theta_D^o, R_{\theta_D})$ where $R_{\theta_D} = 0\, I$.

(ii) as random with no prior information, hence $\theta_D \sim \mathcal{N}(\theta_D^o, R_{\theta_D})$ where $R_{\theta_D} = \infty\, I$.

In case (i), VB becomes EM [123], in which case during the iterations the deterministic parameters are simply substituted by their current estimate.
Case (ii) is closer to the VB spirit. If $\theta = \{\theta_D, \theta_S\}$ where θ_S are the stochastic parameters, then it suffices to replace $p(Y, \theta)$ in (4.3) by $p(Y, \theta_S|\theta_D) = p(Y|\theta)\, p(\theta_S)$. In this case also for the deterministic parameters not only their current estimates (posterior means) are accounted for but also their estimation error.

To summarize, EM is a special case of VB, with 2 subsets of parameters (stochastic and deterministic). Note that in the VB context the difference between EM and SAGE [124] algorithm is the splitting of the subsets [125].

4.2.3 Proposed Approach

The approach we propose in this chapter is the following, we will consider the parameters constant during a frame of the analysis. Instead of iteratively alternating the estimation of sources correlations and of parameters estimation we will design two kind of algorithms. The first one is dedicated to the estimation of the parameters directly from the mixture and without extracting the source directly. The second one uses the estimated parameters to separate the sources. Whereas the parameters estimation is an iterative process the source separation algorithm is non iterative. We will use a constant length window, despite of the fact that an optimal length can be found for a special signal, it should not be the case when several signals are present.

4.3 Model

Here we briefly recall the model presented in chapter 2 and add some notations used in this chapter. The sum of short plus long-term autoregressive (AR) Gaussian sources $x_{k,t}$ plus Gaussian white noise v_t (all independent) is :

$$\begin{aligned} y_t &= \sum_{k=1}^{K} x_{k,t} \;+\; v_t, && (4.4)\\ x_{k,t} &= -\sum_{n=1}^{p_k} a_{k,n}\, x_{k,t-n} \;+\; \tilde{x}_{k,t} && (4.5)\\ \tilde{x}_{k,t} &= b_k\, \tilde{x}_{k,t-\tau_k} \;+\; e_{k,t} && (4.6) \end{aligned}$$

where $\tilde{x}_{k,t}$ is the short term prediction error of the k^{th} source at time t. The prediction error filter can be seen from $x_{k,t} + \sum_{n=1}^{p_k} a_{k,n}\, x_{k,t-n} = \tilde{x}_{k,t}$. Consider that we regroup the N samples of a source in $\mathbf{x}_k$.

$$\tilde{\mathbf{x}}_k = \mathbf{T}_{A_k}\, \mathbf{x}_k \tag{4.7}$$

where $\mathbf{T}_{A_k}$ is the $N \times (N+p_k-1)$ banded Toeplitz matrix corresponding to the prediction error filter $A_k(q)$ (to ease the notation we shall suppress the time index of the frame). With q the advance operator, $A_k(q)$ and $B_k(q)$ are the short-term and long-term prediction error transfer functions defined as:

$$A_k(q) = \sum_{n=0}^{p_k} a_{k,n} q^{-n},\ B_k(q) = 1 - b_k q^{-\tau_k} \tag{4.8}$$

To transform a filtering matrix easily, it should be circulant, in which case the DFT diagonalizes the matrix. The direct approximation of a Toeplitz matrix by a circulant matrix is only acceptable when the matrix dimension is much larger than the filter length. Then the filtering operation can be approximated by the use of a Circulant Matrix (Appendix F.6 deals with Circulant Matrix properties). With this asumption we define the following temporal notations:

- vector of observation: $\mathbf{y}$ of size N
- vector of source k: $\mathbf{x}_k$ of size N
- vector of short term prediction error: $\tilde{\mathbf{x}}_k$ of size N
- vector of short term coefficients for the source: k, $\mathbf{a}_k$ of order p_k
- Short Term prediction error Circulant Matrix: $\mathbf{A}_k$ of size $N \times N$ created with $\mathbf{a}_k$
- vector of long term prediction error: $\tilde{\mathbf{g}}_k$ of size N
- vector of long term coefficients for the source: k, $\mathbf{b}_k$ of size τ_k
- Long Term prediction error Circulant Matrix: $\mathbf{B}_k$ of size $N \times N$ created with $\mathbf{b}_k$
- vector of short+long term prediction error: $\mathbf{e}_k$ of size N
- vector of noise: $\mathbf{v}$ of size N

With this notation we can write the model as follow:

$$\begin{aligned}
\tilde{\mathbf{x}}_k &= \mathbf{A}_k \ \mathbf{x}_k && (4.9)\\
\tilde{\mathbf{g}}_k &= \mathbf{B}_k \ \mathbf{x}_k && (4.10)\\
\mathbf{e}_k &= \mathbf{A}_k \ \mathbf{B}_k \ \mathbf{x}_k && (4.11)\\
&= \mathbf{B}_k \ \mathbf{A}_k \ \mathbf{x}_k && (4.12)
\end{aligned}$$

The properties of circulant matrix allow to invert the Short and Long Term prediction error matrices. This will be useful to alternate the estimation of the Short and Long Term aspect of signals. This allow to define the Long Term error in which the main periodicity is removed, its spectrum only contains the spectral shape.

4.3.1 Spectral Model

The model as defined in chapter 2 leads to the following Spectral model for the sources and the observation.

$$\begin{aligned}
S_k(f) &= \frac{\sigma_k^2}{|A_k(f) \ B_k(f)|^2} \ , \quad k = 1, \dots, K && (4.13)\\
S_0(f) &= \sigma_v^2 = \sigma_0^2 && (4.14)\\
S'_k(f) &= \frac{1}{|A_k(f) \ B_k(f)|^2} && (4.15)\\
S(f) &= \sum_{k=0}^{K} \sigma_k^2 S'_k(f) && (4.16)\\
&= \sum_{k=0}^{K} S_k(f) && (4.17)
\end{aligned}$$

$S_k(f)$ is the spectrum of a single source k. The source $k = 0$ is the additive white noise and it is just represented by its variance (as its spectral shape is flat). For convenience, variances are decoupled from spectral informations. $A_k(f)$ and $B_k(f)$ are defined as the Fourier Transform of the vectors $\mathbf{a}_k$ and $\mathbf{b}_k$:

$$\begin{aligned}
\mathbf{a}_k &= [1 \ a_{1,k} \ \cdots \ a_{p_k,k}]^T && (4.18)\\
\mathbf{b}_k &= [1 \ 0 \cdots -(1-\alpha_k)b_k \ \ -\alpha_k b_k]^T && (4.19)\\
A_k &= F \ [\mathbf{a}_k \ 0 \cdots 0]^T && (4.20)\\
B_k &= F \ [\mathbf{b}_k \ 0 \cdots 0]^T && (4.21)
\end{aligned}$$

Where F is the DFT Matrix and the two vectors are zero padded, A_k and B_k have the same size (N_{fft}) as the DFT of the observation. α_k is the interpolation coefficient of the source k, due to non integer period. The two terms in b_k are at the position $\lfloor \tau_k \rfloor$ and $\lfloor \tau_k + 1 \rfloor$.

To clarify the notation consider that A_k (or B_k) is the Fourier Transform of $\mathbf{a}_k$ while the circulant matrix is written with a bold font $\mathbf{A}_k$ and its Fourier Transform is written with a breve $\breve{\mathbf{A}}_k = \mathbf{F}\mathbf{A}_k\mathbf{F}^{-1}$.

4.3.2 Set of Parameters

The short and long-term aspects of the signals are very different by nature, it may seem natural to separate their analysis. Keeping the EM terminology of Hidden Variables we define a vector θ of parameters (or hyper-parameters). Except the additive noise, the parameters are sources related, we group them by source; this impose to alternate the estimation of a group between sources. The overall set of parameters contains the following subsets (short term and long term parameters):

$$\begin{aligned}\theta &= [\sigma_v^2\ \theta_1^T \cdots \theta_K^T]^T &(4.22)\\ \theta_k &= [\ \mathbf{a}_k\ \varphi_\mathbf{k}]^\mathbf{T} &(4.23)\\ \mathbf{a}_k &= [a_{k,1} \cdots a_{k,p_k}] &(4.24)\\ \varphi_\mathbf{k} &= [\ b_k\ \tau_k\ \sigma_k^2\ ] &(4.25)\end{aligned}$$

For the estimation of a given subset of parameters of a given source we consider that the other sources are constant and also the other subset of the current source.

4.4 Parameters Estimation Using The Itakura-Saito (IS) Distance

The Itakura Saito Distance/Divergence is a common distance in audio processing. This divergence was obtained by Itakura and Saito (1968) from the maximum likelihood (ML) estimation of short-time speech spectra under autoregressive modeling, it was presented as a measure of the goodness of fit between two spectra. The Itakura Saito distance can be derived from the Bregman divergence [126]. It was in particular praised for the good perceptual properties of the reconstructed signals [127].

In this section we present two iterative methods based on the interpretation of the Itakura-Saito distance. The first one is a naive interpretation of this distance and the other one is based on its true minimization. The two algorithms are designed to estimate the sources parameters directly from the mixture using a frame based analysis. We will first recall the definition of the Itakura Saito distance then we will describe the two iterative algorithms.

4.4.1 Definition of the IS Distance

The Itakura Saito distance (IS) [75] is a measure of the difference between an original spectrum $Y(f)$ and an approximation $\hat{Y}(f)$ of that spectrum. The IS distance is defined as :

$$D(Y(f)|\hat{Y}(f)) = \int df \left[\frac{Y(f)}{\hat{Y}(f)} - ln\left(\frac{Y(f)}{\hat{Y}(f)}\right) - 1\right] \tag{4.26}$$

It respects the separation constraint: $D(Y(f)|\hat{Y}(f)) = 0$ if $\hat{Y}(f) = Y(f)$ but does not the symmetry $D(Y(f)|\hat{Y}(f)) \neq D(\hat{Y}(f)|Y(f))$ nor the triangle inequality $D(Y(f)|\hat{Y}_2(f)) \leq D(Y(f)|\hat{Y}_1(f)) + D(\hat{Y}_1(f)|\hat{Y}_2(f))$. However the separation constraint is the one which is of interest in our case.

4.4.2 IS Distance for the model

Consider the IS distance between the periodogram $Y(f) = \frac{|y(f)|^2}{N}$ and the parametric spectrum $S(f;\theta)$ as defined in (4.17) with the parameters set defined in (4.22):

$$\int df \left[\frac{Y(f)}{S(f;\theta)} - ln\left(\frac{Y(f)}{S(f;\theta)}\right) - 1 \right] \tag{4.27}$$

where $S(f;\theta) = \sum_{k=0}^{K} S_k(f;\theta_k) = \sum_k \sigma_k^2 S'_k(f;\theta_k)$ with $S_k(f;\theta_k)$ the parametric spectrum of the source k, defined in (4.13) ($S'_0(f;\theta_0) = 1$). This definition means that we try to match the parametric AR Spectrum to the periodogram. As the approximate Spectrum is completely defined by its parameters, the minimization of this distance leads to find the set of parameters.

4.5 Naive Interpretation of the IS distance

If we consider an alternating maximization of the parameters, as updating the parameters of the source k while keeping the other sources parameters constant. If we rewrite $\frac{Y(f)}{S(f;\theta)}$ as:

$$\frac{Y(f)}{S(f;\theta)} = \frac{1}{S_k(f;\theta_k)} \frac{S_k(f;\theta_k)}{\sum_k S_k(f;\theta_k)} Y(f) \tag{4.28}$$

$$\frac{Y(f)}{S(f;\theta)} = \frac{1}{S_k(f;\theta_k)} \frac{Y(f)}{1 + S_k(f;\theta_k)^{-1} \sum_{\bar{k}\neq k} S_{\bar{k}}(f;\theta_{\bar{k}})} \tag{4.29}$$

we can observe that:

$$\frac{Y(f)}{S(f;\theta)} = \frac{1}{S_k(f;\theta_k)} \frac{1}{N} \frac{y(f)}{1 + S_k(f;\theta_k)^{-1} \sum_{\bar{k}\neq k} S_{\bar{k}}(f;\theta_{\bar{k}})} y^*(f) \tag{4.30}$$

is the crossed spectrum between the Wiener estimates of the source k and the observation y. Now consider that we split the expression in two part

$$\frac{Y(f)}{S(f;\theta)} = \frac{\hat{S}_k(f)}{S_k(f;\theta_k)} \tag{4.31}$$

$\hat{S}_k(f)$ can be seen as an estimate of $S_k(f;\theta_k)$. The naive interpretation consists on ignoring the dependence of $\hat{S}_k(f)$ over θ (as it is the case in (4.31), the minimization of the IS distance with respect to $S_k(f;\theta_k)$ leads to a linear prediction problem on this spectrum.

4.5.1 Algorithm

The algorithm consists by alternatively estimating the short term and long term subsets of parameters. Each subsets estimation needs to be iterated between all the sources (including the additive noise) until convergence. Also, we iterate between all the sources and the algorithm is stopped when all the subsets of all sources have converged. For convenience, we define

$$S_{\bar{k}}(f;\theta_{\bar{k}}) = \sum_{\bar{k}} \frac{\sigma_{\bar{k}}^2}{|A_{\bar{k}}(f) B_{\bar{k}}(f)|^2} \tag{4.32}$$

The index $\bar{k}$ includes all the sources (and noise) except the one of interest, the source k.

4.5.1.1 Short term parameters estimation

To estimate the short term (st) parameters of order $p_k + 1$, also if we work with a single source as said in section 2.3.2, we have to remove the effect of the long term otherwise the estimation is biased by the harmonic structure. Until convergence, and for all the sources:

$$\begin{aligned}
\hat{S}_k(f) &= \frac{Y(f)}{1 + S_k^{-1}(f;\theta_k) S_{\bar{k}}(f;\theta_{\bar{k}})} && (4.33)\\
S_k^{st}(f) &= \hat{S}_k(f)\ |B_k(f)|^2 && (4.34)\\
r_k &= F^{-1}\ S_k^{st}(f) && (4.35)
\end{aligned}$$

The correlation sequence as defined in (4.35) is, in the ideal case, the correlation sequence of the long term prediction error of the source k, in which the spectral shape is maintained but the influence of the frequency comb is removed. The short term coefficients are computed on this correlation sequence, using the Levinson-Durbin recursion, and the new estimates of $\mathbf{a}_k$ is used for the next source.

4.5.1.2 Long term parameters estimation

The long term (lt) parameters consists on three parameters: the period, the coefficient and the short+long term variance. We also need to clean the short term influence to estimate them. Until convergence, and for all the sources:

$$\begin{aligned}
\hat{S}_k(f) &= \frac{Yf)}{1 + S_k^{-1}(f;\theta_k) S_{\bar{k}}(f;\theta_{\bar{k}})} && (4.36)\\
S_k^{lt}(f) &= \hat{S}_k(f)\ |A_k(f)|^2 && (4.37)\\
r_k &= F^{-1}\ S_k^{lt}(f) && (4.38)
\end{aligned}$$

The correlation sequence as defined in (4.38) is the correlation sequence of the short term prediction error of the source k. The short term prediction error is, if we use the good short term coefficients, composed of a pulse train corresponding to the periodic excitation. If the period is known $b_k = \frac{r_k(\tau_k)}{r_k(0)}$ otherwise τ_k is estimated as the delay which maximize b_k (for a realistic range of delay), as in section 2.2.3.3.
The short+plus long term prediction error is:

$$\begin{aligned}
S^e(f) &= S_k(f)\ |A_k(f)|^2\ |B_k(f)|^2 && (4.39)\\
&= S_k^{lt}(f)\ |B_k(f)|^2 && (4.40)\\
\sigma_k^2 &= \frac{1}{N} \sum_f S^e(f) && (4.41)
\end{aligned}$$

where $S^e(f)$ is the spectrum of the short+long term prediction error of the source k, if the algorithm has converged toward the good parameters, $S^e(f)$ is white.

4.5.1.3 Noise variance

In the above formulation the noise is treated as a source. It is the only one global parameter (not related to a particular source) and needs the knowledge of all the sources parameters.

$$\hat{S}_0(f) = \frac{Y(f)}{1 + S_0^{-1}(f)\sum_{k=1} S_k(f;\theta_k)} \tag{4.42}$$

$$\sigma_v^2 = \frac{1}{N}\sum_f S_0(f) \tag{4.43}$$

4.6 Minimization of the IS distance

The minimization of the IS distance consists on the true minimization over one of the subset of parameters set. If we consider the gradient of IS with respect to parameter θ_i, we obtain:

$$\frac{\partial}{\partial\theta_i}\int df \left[\frac{Y(f)}{S(f;\theta)} - ln\left(\frac{Y(f)}{S(f;\theta)}\right) - 1\right] = \int df \frac{1}{S(f;\theta)^2}[S(f;\theta) - Y(f)]\frac{\partial S_i(f;\theta_i)}{\partial\theta_i} \tag{4.44}$$

4.6.1 Weighted Spectrum Matching

It turns out that the IS gradient is the same as for Optimally Weighted Spectrum Matching. Indeed, at high window length N, the periodogram $Y(f)$ has as mean the spectrum $S(f;\theta)$ and as variance $S(f;\theta)^2$. Hence the optimally weighted spectrum matching criterion becomes

$$\int df \; \frac{1}{S(f;\theta)^2} \; [Y(f) - S(f;\theta)]^2 \tag{4.45}$$

Taking the gradient w.r.t. a parameter θ_i in the parametric spectrum $S(f;\theta)$ (and ignoring the dependence of the weighting $\frac{1}{S(f;\theta)^2}$ on θ_i) leads to the same gradient as for the IS distance. The weighting involves the true spectrum $S(f;\theta)$, but can asymptotically be replaced by a consistent spectrum estimator such as appropriate versions of the averaged or smoothed periodogram. In our simulations we just use the periodogram itself.

4.6.2 Gaussian Maximum Likelihood

For sufficiently long window length, Maximum Likelihood (ML) can be expressed in the frequency domain and the negative Gaussian log likelihood of $y(f)$, which has zero mean and variance $S(f;\theta)$, becomes

$$\int df \left[\frac{Y(f)}{S(f;\theta)} + ln\left(S(f;\theta)\right)\right] \tag{4.46}$$

which obviously will again give the same gradient as the IS distance.

4.6.3 Short-term AR Parameters Estimation

For the short term filter of the k^{th} source ($\theta_k = A_k$) we obtain:

$$\frac{\partial S_k(f;\theta_k)}{\partial A_k^*} = -S_k(f;\theta_k)\frac{A_k(f)}{|A_k(f)|^2} \tag{4.47}$$

using (4.47) in (4.44):

$$\int df \frac{1}{S(f;\theta)^2}[Y(f) - S(f;\theta)]S_k(f;\theta_k)\frac{A_k(f)}{|A_k(f)|^2} \tag{4.48}$$

which leads to:

$$\begin{aligned} \frac{Y(f)}{S(f;\theta)}\frac{S_k(f;\theta_k)}{S(f;\theta)}\frac{A_k(f)}{|A_k(f)|^2} &= \frac{S_k(f;\theta_k)}{S(f;\theta)}\frac{A_k(f)}{|A_k(f)|^2} && (4.49) \\ \left(\frac{Y(f)}{S(f;\theta)}\frac{S_k(f;\theta_k)}{S(f;\theta)}\frac{1}{|A_k(f)|^2}\right)A_k(f) &= \frac{S_k(f;\theta_k)}{S(f;\theta)}\frac{1}{A_k(f)^*} && (4.50) \end{aligned}$$

This leads to the Yule-Walker like equation with a non zero Right Hand Side (RHS), this needs to be solved iteratively:

$$T\left(r_{k,(0,\cdots,p_k-1)}\right)\ \mathbf{a}_k \ = \ g_{k,(1,\cdots,p_k)} - r_{k,(1,\cdots,p_k)} \tag{4.51}$$

where T is a symmetric Toeplitz matrix filled with the first p_k elements of r_k, $\mathbf{a}_k$ are the short term AR coefficients (4.24) and g_k the resulting correlation, r_k and g_k are defined by:

$$\begin{aligned} r_k &= \left[F^{-1}\left(\frac{Y(f)}{S(f;\theta)}\frac{S_k(f;\theta_k)}{S(f;\theta)}\frac{1}{|A_k(f)|^2}\right)\right] && (4.52) \\ g_k &= \left[F^{-1}\left(\frac{S_k(f;\theta_k)}{S(f;\theta)}\frac{1}{A_k(f)^*}\right)\right] && (4.53) \end{aligned}$$

F is the Discrete Fourier Transform matrix and $A_k(f)$ can be replaced by $B_k(f)$.

4.6.4 Source Power Estimation

As we have shown the IS distance has the same gradient as the weighted least squares spectrum matching. The estimation of the variance by minimizing the IS is not attractive but the two approaches have the same solution. So, consider equivalently the weighted least squares spectrum matching, weighted by the inverse squared periodogram. We obtain:

$$\int df \frac{1}{Y(f)^2}\left[\sum_{k=0}^{K}\sigma_k^2 S_k'(f;\theta_k) - Y(f)\right]^2 . \tag{4.54}$$

where $S_k'(f;\theta_k)$ is defined in (4.15). The minimization with respect to $\underline{\sigma}^2 \ = \ [\ \sigma_v^2\, \sigma_1^2 ... \sigma_K^2\ ]^T$ leads to solve the system $G\ \underline{\sigma}^2 \ = \ d$, with:

$$G_{ik} = \int df\ \frac{S_i'(f;\theta_i)\, S_k'(f;\theta_k)}{Y(f)^2}\ ,\ d_i = \int df\ \frac{S_i'(f;\theta_i)}{Y(f)} \tag{4.55}$$

4.6.5 Overall iterative process

r_k and g_k are both A_k dependent, the algorithm can be summarised as:

- For all the sources k
- Until convergence of r_k: construct r_k and g_k using (4.52) and (4.53).
- Until convergence of g_k: estimate $\mathbf{a}_k$ by solving (4.51), construct $S_k(f;\theta_k)$ and g_k using (4.13) and (4.53), update $S(f;\theta)$ and $\underline{\sigma}^2$.
- Stop condition on g_k
- Stop condition on r_k

The procedure is stopped if the variation between two consecutive estimated correlations is lower than a threshold or if the number of iteration is greater than a maximum number.

4.6.5.1 A note about the Long Term Model

The estimation of the long term (LT) coefficient and of the period is under investigation. Applying the same strategy as for the short term does not give good result for the moment, also an error on the long term deteriorates the result. If no positivity constraint is used, the variance estimation can, in the case of a very bad estimation of the LT, leads to negative value. For the simulation involving the Minimization Algorithm we assumes that we know the long term coefficients and the periods of each sources.

4.7 Parameters Estimation, Synthetic Spectrum

In this section we apply the two parameters estimation algorithm presented in section 4.5, for the Naive Interpretation of the IS Distance, and in section 4.6, for the true minimization, on a synthetic spectrum as defined in (4.17).

As mentioned previously the long term estimation of the Minimization algorithm is not yet finished, so for the two algorithms we assume that we know the long term coefficients and the periods of each source. As a result in the algorithm based on the naive interpretation of the IS distance the step to estimate this two known quantities is removed.

The observed spectrum is composed of two sources. For each sources the parameters are randomly generated. The true and estimated parameters are the following

Table 4.1: Parameters used

Parameter:	$\mathbf{a}_{k,1}$	$\mathbf{a}_{k,2}$	$\mathbf{a}_{k,3}$	$\mathbf{a}_{k,4}$	$\mathbf{a}_{k,5}$	σ_k^2
1^{st} source	-0.4047	-1.2747	+0.9761	+0.6927	-0.7380	0.9255
2^{nd} source	-0.1699	-0.3636	-0.4466	-0.5440	+0.7022	0.8298
1^{st} source IS Minimization	-0.4047	-1.2747	+0.9761	+0.6927	-0.7380	0.9255
2^{nd} source IS Minimization	-0.1699	-0.3636	-0.4466	-0.5440	+0.7022	0.8298
1^{st} source IS Naive	-0.4086	-1.2762	+0.9804	+0.6921	-0.7419	0.9253
2^{nd} source IS Naive	-0.1684	-0.3617	-0.4461	-0.5437	+0.7005	0.8297

In this particular example, in Figure 4.2, the results are very good, this is essentially due to the fact that the formants frequency is not overlapped. Our observations show that the Naive interpretation gives always worse results than the minimization of IS, however they can be very close. And the IS minimization leads alway to almost perfect results on a true spectrum (if the long term is known). As an example of the performance of the IS Minimization another example is shown in Figure 4.3 in which the sources have the same formant frequency.

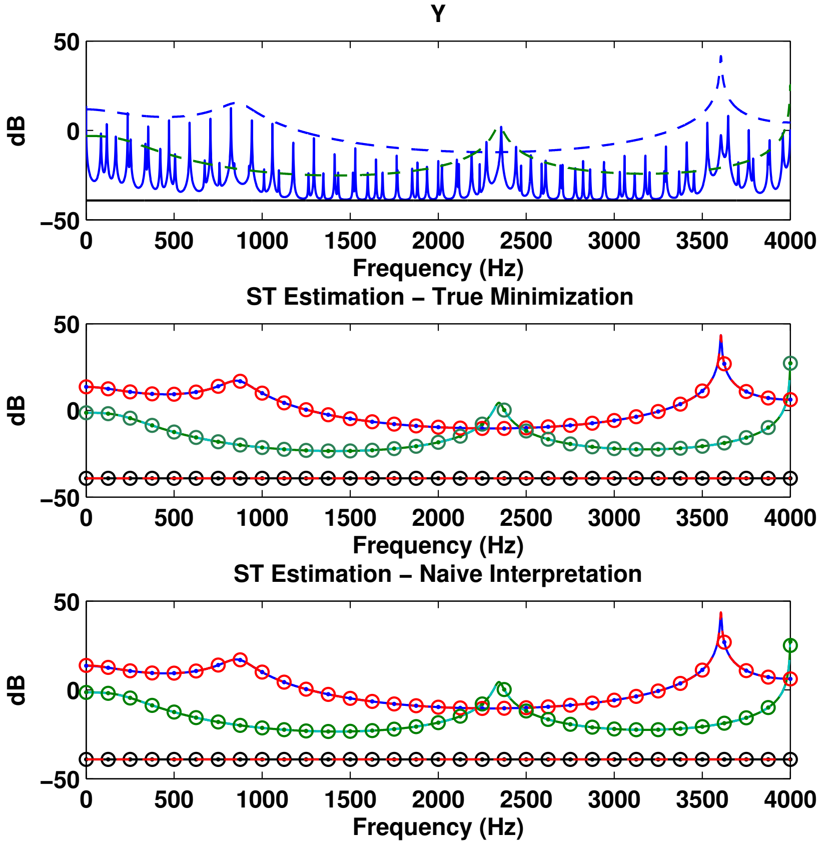


Figure 4.2: Comparison example Naive Versus True Minimization of IS Distance.

4.8 Source Separation Algorithm

In section 4.2.2 we have presented the idea of the Variational Bayesian approach. We shall simplify the VB approach by splitting the overall parameters θ into two groups: the sources $\{\mathbf{x}_k,\ k = 1,\ldots,K\}$ on the one hand, and all AR and noise parameters on the other hand. Whereas the first group shall be treated as random, the second group shall be treated as deterministic (negligible variability, delta function posterior distribution).

4.8.1 Joint Source Representation

In order to estimate the sources $\mathbf{x}_k$ jointly, we concatenate the sources of a segment into a single vector, consider the vector $\underline{\mathbf{x}} = [\mathbf{x}_1^T \cdots \mathbf{x}_k^T]^T$. One of most important point we introduce is the substitution of sources by windowed sources, this substitution leads to take into consideration the influence of the window during all the derivation. The windows is used to reduce approximation and has to be well designed (as mentioned in section 4.1.2), as a result, the separation algorithm we propose will extract windowed estimate.

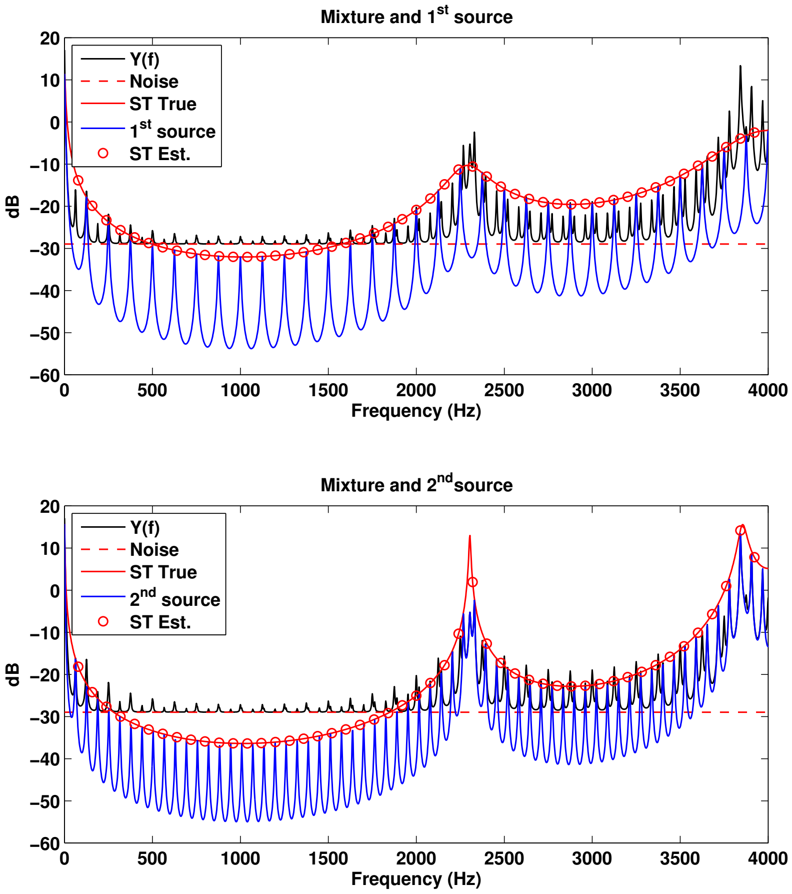


Figure 4.3: Example of True Minimization of IS Distance.

4.8.2 Frequency Domain Window Design

As mentioned before, the signals considered are by nature non-stationary. If we can consider the parameters constant during a short time, we can process the signal in frames (time segments), over which the signal can be considered stationary, which corresponds to time-invariant filtering. Many of the signal processing operations (e.g. linear time-invariant filtering and filter computation) could be largely simplified by passing to the frequency domain. However, transforming a frame of signal to the frequency domain directly via the DFT (FFT) leads to approximations due to the periodic extension of the frame assumption inherent in the DFT. In section 4.3 we have introduced the Circulant Matrix to denote the prediction error filter instead of using $\mathbf{T}_{A_k}$ a banded Toeplitz matrix corresponding to the prediction error filter. Whereas it is an approximation the introduction of a window can reinforce it. Consider the analysis window $\mathbf{w} = [w_0\, w_1 \ldots w_{N-1}]^T$, with associated diagonal weighting matrix $\mathbf{W} = \operatorname{diag}\{\mathbf{w}\}$. The windowed prediction error $\mathbf{W}\tilde{\mathbf{x}}_k$ requires $\mathbf{W}\mathbf{T}_{A_k}$. Now, assume the window decays to zero at its edges and varies sufficiently slowly, then the following approximations become valid:

$$\mathbf{W}\,\mathbf{T}_{A_k} \approx \mathbf{W}\,\mathbf{A}_k \approx \mathbf{A}_k\,\mathbf{W} \tag{4.56}$$

where $\mathbf{A}_k$ is a $N \times N$ square circulant matrix (as defined in section 4.3), corresponding to circulant convolution with $A_k(q)$ as in (4.8). We shall similarly introduce the circulant

$\mathbf{B}_k$, though the approximations considered above will be rougher for the filter $B_k(q)$ (or $B_k^{-1}(q)$) since long-term prediction has larger delay spread than short-term prediction. Note that just like $A_k(q)B_k(q) = B_k(q)A_k(q)$, also $\mathbf{A}_k\mathbf{B}_k = \mathbf{B}_k\mathbf{A}_k$. Then we get the following signal relations

$$\begin{aligned} \mathbf{W}\mathbf{e}_k &= \mathbf{A}_k\mathbf{B}_k\,\mathbf{W}\mathbf{x}_k = \mathbf{A}_k\,\mathbf{W}\tilde{\mathbf{g}}_k = \mathbf{B}_k\,\mathbf{W}\tilde{\mathbf{x}}_k \\ \mathbf{W}\tilde{\mathbf{g}}_k &= \mathbf{B}_k\,\mathbf{W}\mathbf{x}_k \\ \mathbf{W}\tilde{\mathbf{x}}_k &= \mathbf{A}_k\,\mathbf{W}\mathbf{x}_k\ . \end{aligned} \tag{4.57}$$

When applying the $N \times N$ DFT matrix $\mathbf{F}$ to the windowed signals in (4.57), we get

$$\mathbf{F}\mathbf{W}\mathbf{e}_k = (\mathbf{F}\mathbf{A}_k\mathbf{F}^{-1})(\mathbf{F}\mathbf{B}_k\mathbf{F}^{-1})\,(\mathbf{F}\mathbf{W}\mathbf{F}^{-1})\,\mathbf{F}\mathbf{x}_k$$

where we get diagonal frequency domain filtering matrices $\breve{\mathbf{A}}_k = \mathbf{F}\mathbf{A}_k\mathbf{F}^{-1}$ etc. The main non-diagonal matrix will be the covariance matrix of $\mathbf{F}\mathbf{W}\mathbf{e}_k$, which is proportional to $\breve{\mathbf{W}}_2 = \mathbf{F}\mathbf{W}^2\mathbf{F}^{-1}$ ($\mathbf{e}_k$ being white). For the case of the Hann window, both the window and a zoom on the main antidiagonal of the circulant $\breve{\mathbf{W}}_2$ appear in the bottom half of Figure 4.4. The time domain window design criteria of decaying edges and smooth behavior translate in the frequency domain to decaying spectral smear and high sidelobe attenuation. Indeed, in order to keep a low computational complexity approach, the window spectrum will be approximated by only its main lobe. This leads to an approximation error that derives from the sidelobe attenuation level. The resulting processing will no longer involve pure diagonal matrices, but banded matrices. As the FFT points in the bottom right figure indicate, for the case of a Hann window, $\breve{\mathbf{W}}_2$ can be approximated by a symmetric banded circulant matrix with 5 diagonals, with (elementwise) approximation error attenuated by at least 30dB.

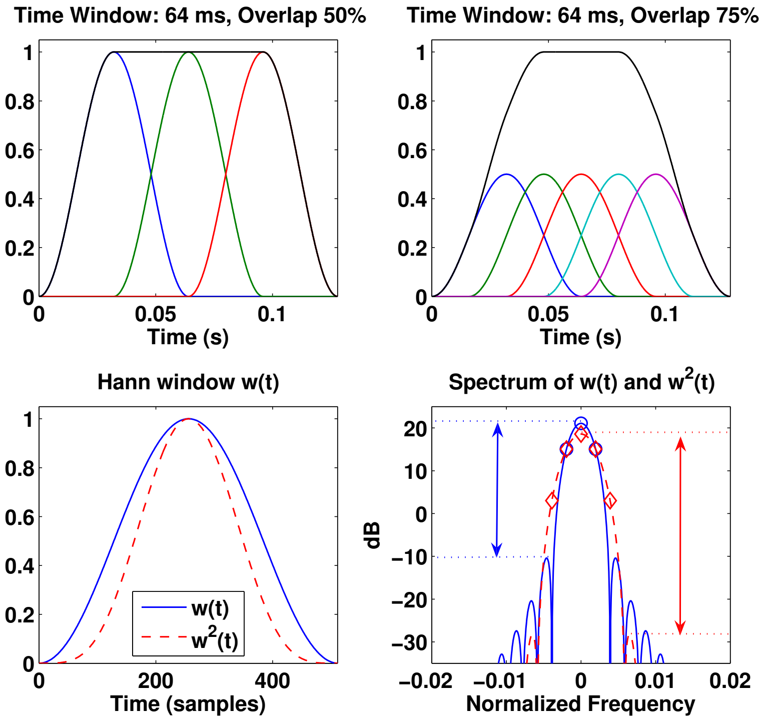


Figure 4.4: Perfect reconstruction windowing.

4.8.3 Joint Model

As mentioned previously we consider the substutition of the sources by their windowed version. Also to write the model consider now also the following notation:

$$
\begin{aligned}
\underline{\mathbf{W}} &= \oplus_{k=1}^{K}\mathbf{W} \;=\; I_K \otimes \mathbf{W} && (4.58)\\
\underline{\mathbf{I}} &= [I_N \ldots I_N] \;=\; \mathbf{1}_k^T \otimes I_N && (4.59)\\
\underline{\mathbf{A}} &= \oplus_{k=1}^{K}\mathbf{A}_k \;=\; \text{blockdiag}\{\mathbf{A}_1, \ldots, \mathbf{A}_K\} && (4.60)\\
\underline{\mathbf{B}} &= \oplus_{k=1}^{K}\mathbf{B}_k \;=\; \text{blockdiag}\{\mathbf{B}_1, \ldots, \mathbf{B}_K\} && (4.61)\\
\underline{\Lambda} &= \oplus_{k=1}^{K}\lambda_k I_N \;=\; \Lambda \otimes I_N && (4.62)\\
\Lambda &= diag\{\lambda_1, \ldots, \lambda_k\} && (4.63)\\
\underline{\mathbf{x}}' &= \underline{\mathbf{W}}\,\underline{\mathbf{x}} && (4.64)\\
\underline{\Lambda}' &= \underline{\mathbf{W}}^{-1}\underline{\Lambda\mathbf{W}}^{-1} = \Lambda \otimes \mathbf{W}^{-2} && (4.65)\\
\underline{\mathbf{e}} &= [\mathbf{e}_1^T \cdots \mathbf{e}_K^T]^T. && (4.66)
\end{aligned}
$$

where $\oplus$ and $\otimes$ are the Kronecker sum and product respectively. With this notation, the signal model can be written as

$$
\begin{aligned}
\mathbf{W}\mathbf{y} &= \underline{\mathbf{I}}\;\underline{\mathbf{x}}' \;+\; \mathbf{W}\;\mathbf{v} && (4.67)\\
\underline{\mathbf{A}}\;\underline{\mathbf{B}}\;\underline{\mathbf{x}}' &= \underline{\mathbf{W}}\;\underline{\mathbf{e}}. && (4.68)
\end{aligned}
$$

with circulant $\mathbf{A}_k$, $\mathbf{B}_k$.

4.8.4 Estimating the Sources

Following the VB approach defined in section 4.2.2, the prior probability distributions for the various parameters are chosen as follows: Let ψ be any of the groups $\{\mathbf{x}_k,\, k = 1, \ldots, K\}$, $\mathbf{a}_k$, $\varphi_k \setminus \lambda_k$. Then for any such subset of parameters ψ and for the λ_k, λ_v, the priors are chosen as

$$
\begin{aligned}
p(\psi) &= \mathcal{N}(m_\psi, C_\psi) && (4.69)\\
p(\lambda_v) &= Exponential(m_{\lambda_v}) && (4.70)\\
p(\lambda_k) &= Exponential(m_{\lambda_k})\;. && (4.71)
\end{aligned}
$$

Where the λ are the precision (inverse variance used in VB terminology).
With this choice of prior distributions, the posterior distributions obtained by VB will be of the same nature (Gaussian or Exponential). However, in this chapter we shall consider a further simplification. We consider that the parameters are deterministic. We get the Gaussian

$$
\begin{aligned}
-2\ln\; p(\mathbf{y}, \underline{\mathbf{x}}|\theta) &= \lambda_v(\mathbf{W}\mathbf{y} - \underline{\mathbf{I}}\;\underline{\mathbf{x}}')^T\mathbf{W}^{-2}(\mathbf{W}\mathbf{y} - \underline{\mathbf{I}}\;\underline{\mathbf{x}}') + \underline{\mathbf{x}}'^T\left[\underline{\mathbf{B}}^T\underline{\mathbf{A}}^T\underline{\Lambda}'\underline{\mathbf{A}}\;\underline{\mathbf{B}}\right]\underline{\mathbf{x}}'\\
&= c + (\underline{\mathbf{x}}' - m_{\underline{\mathbf{x}}'})^T C^{-1}_{\underline{\mathbf{x}}'\underline{\mathbf{x}}'}(\underline{\mathbf{x}}' - m_{\underline{\mathbf{x}}'})\;. && (4.72)
\end{aligned}
$$

Averaging this over the parameters θ now simply means evaluating at the latest estimates of these parameters, since they are considered deterministic in the simplification. We get from (4.72), after introducing the auxiliary quantities

$$\begin{aligned} \mathbf{C}_1 &= \underline{\mathbf{B}}^T \underline{\mathbf{A}}^T \underline{\Lambda}' \underline{\mathbf{A}}\, \underline{\mathbf{B}} \\ &= \text{blockdiag}\{\lambda_k \mathbf{B}_k^T \mathbf{A}_k^T \mathbf{W}^{-2} \mathbf{A}_k \mathbf{B}_k\}_{k=1}^K \\ \mathbf{C}_2 &= \tfrac{1}{\lambda_v}\mathbf{W}^2 + \underline{\mathbf{I}}\mathbf{C}_1^{-1}\underline{\mathbf{I}}^T \\ &= \tfrac{1}{\lambda_v}\mathbf{W}^2 + \textstyle\sum_k \tfrac{1}{\lambda_k}\mathbf{B}_k^{-1}\mathbf{A}_k^{-1}\mathbf{W}^2\mathbf{A}_k^{-T}\mathbf{B}_k^{-T} \end{aligned} \tag{4.73}$$

we get

$$\begin{aligned} C_{\underline{\mathbf{x}}'} &= (\lambda_v \underline{\mathbf{I}}^T \mathbf{W}^{-2} \underline{\mathbf{I}} + \mathbf{C}_1)^{-1} \;=\; \mathbf{C}_1^{-1} - \mathbf{C}_1^{-1}\underline{\mathbf{I}}^T\mathbf{C}_2^{-1}\underline{\mathbf{I}}\mathbf{C}_1^{-1} \\ m_{\underline{\mathbf{x}}'} &= \mathbf{C}_1^{-1}\underline{\mathbf{I}}^T\mathbf{C}_2^{-1}\mathbf{W}\mathbf{y}\,. \end{aligned} \tag{4.74}$$

To implement this in the frequency domain, consider the diagonal $\breve{\mathbf{A}}_k = \mathbf{F}\mathbf{A}_k\mathbf{F}^{-1}$ etc. The only non-diagonal matrix is $\breve{\mathbf{W}}_2 = \mathbf{F}\mathbf{W}^2\mathbf{F}^{-1}$ which, due to careful window design, can be approximated by a banded matrix as discussed earlier. As a result, $\breve{\mathbf{C}}_1^{-1}$ and $\breve{\mathbf{C}}_2$ are equally banded matrices. Now consider the (Lower triangular, Diagonal, Upper Triangular) LDU factorization (as shown in Figure 4.5):

$$\begin{aligned} &\mathbf{F}\mathbf{D}\mathbf{F}^{-1} = \mathbf{F}\left[\frac{1}{\lambda_v}\mathbf{W}^2 + \sum_k \frac{1}{\lambda_k}\mathbf{B}_k^{-1}\mathbf{A}_k^{-1}\mathbf{W}^2\mathbf{A}_k^{-T}\mathbf{B}_k^{-T}\right]\mathbf{F}^{-1} \\ &= \frac{1}{\lambda_v}\breve{\mathbf{W}}_2 + \sum_k \frac{1}{\lambda_k}\breve{\mathbf{B}}_k^{-1}\breve{\mathbf{A}}_k^{-1}\breve{\mathbf{W}}_2\breve{\mathbf{A}}_k^{-H}\breve{\mathbf{B}}_k^{-H} \;=\; \mathbf{L}\mathbf{D}\mathbf{L}^H \end{aligned} \tag{4.75}$$

where the unit diagonal lower triangular $\mathbf{L}$ is banded.

The steps to compute $m_{\underline{\mathbf{x}}'}$ in the frequency domain are now:

- $\breve{\mathbf{y}} = \mathbf{F}\;\mathbf{W}\;\mathbf{y}$
- solve $\breve{\mathbf{u}}$ from $\mathbf{L}\breve{\mathbf{u}} = \breve{\mathbf{y}}$ by backsubstitution
- solve $\breve{\mathbf{z}}$ from $\mathbf{L}^H\breve{\mathbf{z}} = \mathbf{D}^{-1}\breve{\mathbf{u}}$ by backsubstitution
- $m_{\mathbf{x}'_k} = \frac{1}{\lambda_k}\,\mathbf{F}^{-1}\;\breve{\mathbf{B}}_k^{-1}\breve{\mathbf{A}}_k^{-1}\breve{\mathbf{W}}_2\breve{\mathbf{A}}_k^{-H}\breve{\mathbf{B}}_k^{-H}\;\breve{\mathbf{z}}$, each time multiplying a vector with a matrix and ending with IDFT and scaling.

In practice all operations with the Discrete Fourier Transform (DFT) matrix $\mathbf{F}$ are done by using the Fast Fourier Transform algorithm (FFT). As $\breve{\mathbf{B}}_k = \text{diag}\{\breve{\mathbf{b}}_k\}$ and $\breve{\mathbf{A}}_k = \text{diag}\{\breve{\mathbf{a}}_k\}$, we can write

$$\breve{\mathbf{B}}_k^{-1}\breve{\mathbf{A}}_k^{-1}\breve{\mathbf{W}}_2\breve{\mathbf{A}}_k^{-H}\breve{\mathbf{B}}_k^{-H} \;=\; \frac{1}{\breve{\mathbf{a}}_k}\frac{1}{\breve{\mathbf{a}}_k^H} \odot \frac{1}{\breve{\mathbf{b}}_k}\frac{1}{\breve{\mathbf{b}}_k^H} \odot \breve{\mathbf{W}}_2\,. \tag{4.76}$$

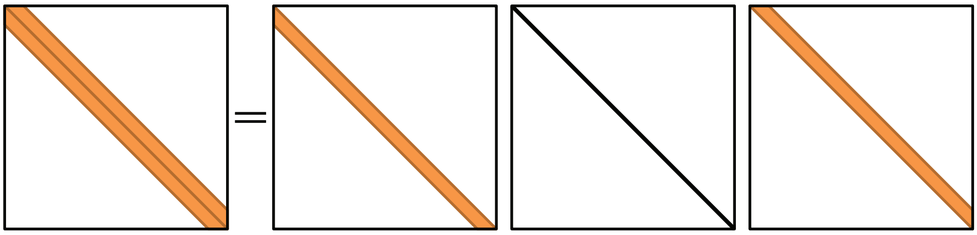


Figure 4.5: LDU Decomposition.

4.8.4.1 Summary of the algorithm

The algorithm can be summarized as follows. First of all, the window w is designed and its weights are used to construct $\mathbf{W}$ and the Discrete Fourier Transform (DFT) of its square $\breve{\mathbf{W}}_2$. A segment of the observation $\mathbf{y}$ is extracted, windowed and its DFT is computed. The diagonal elements of the DFT of the circulant matrices are computed and used in (4.76) for all the sources. A LDU decomposition is done on the sum defined in (4.75) then the procedure described before is applied. The LDU Decomposition is done by using the Cholesky Decomposition. The algorithm extracts windowed source so for the overlap and add method they don't need to be windowed.

4.9 Summary and discussion

This Chapter deals with three algorithms. Two of them focus on parameter estimation while the third one is a purely source separation algorithm. The estimation of the parameters use the Itakura Saito (IS) Distance, which is known to give good perceptual results. All the algorithms of this Chapter are frame based and most calculations are done in spectral domain. This implies to use a well designed analysis window and leads, also, to analyze overlapped segment. Section 4.8.2 deals with an approximation of the window in the frequency domain to simplify the source separation algorithm. The approximation consists on keeping only the main lobe in the Spectrum of the window. Also the original model presented in Chapter 2 is slightly modified, we approximate the filter operation (convolution) with a circular convolution using circulant matrices. These two approximations lead to simplification in the use of the Wiener filter. The two parameter estimation algorithms are both based on the Itakura Saito Distance. The first one, presented in section 4.5, is a naive interpretation of this distance and leads to an iterative algorithm able to estimate all the parameters. The second one, in section 4.6, is based on the minimization of the IS distance and gives better result for synthetic Spectra but the Long Term part is not yet finished. The overall algorithm will consists on using one of the two algorithms to estimate the parameters, on a frame of the observation, and to use the estimated parameters in the source separation algorithm. The algorithm which use the Naive interpretation of the IS distance for the parameter estimation is called **Naive-IS** while the one which minimizes the IS distance is called **TMin-IS**.

The next chapter is dedicated to real speech separation, the adaptive algorithm of section 3 will be compared to the two possible algorithms presented in the present section. An initialization procedure will also be presented as well as a background sound extraction. The results, even if the simulations are not performed in the same context, are compared to some existing solutions.

Chapter 5

Simulations

The purpose of this chapter is to show some simulation results on real signals. The considered algorithms used here were presented in previous chapters. Two versions of the adaptive algorithm, introduced in section 3, are compared to the frame-based algorithms presented in section 4. The EM-Kalman based algorithms are called ***Joint-EMK*** *and* ***Alt-EMK*** *for joint and alternate parameter estimation respectively. The Frame based algorithms are called* ***Naive-IS*** *and* ***TMin-IS****. They are used for the Naive interpretation of the Itakura Saito (IS) distance and its Minimization.*

This Chapter is organized as follows: First of all, some details about the real signals used in this chapter will be given in section 5.1. In section 5.2, the focus will be on short duration speech segments in which the periods are constant. Then in section 5.3, long time duration speech will be at the centre of the simulation. Throughout section 5.4, the performance of the presented algorithm compared to some existing algorithms will be discussed. In section 5.5, the adaptive algorithm is applied on a particular application related to the 2010 Football World Cup. Finally, section 5.6 will be the conclusion of this chapter.

5.1 Introduction and database details

The real speech sound used in this chapter comes from the 2007 Stereo Audio Source Separation Evaluation Campaign, SASSEC2007 [35]. The original signals are stereophonic and composed of source signals of 10 s' duration sampled at 16 KHz (a male and a female speakers). The files used here are for the English female and male speakers: "$female\_inst\_sim\_1.wav$" and "$male\_inst\_sim\_1.wav$" respectively. We work with one channel (right) and we resample the source signals to 8 Khz. The mixing is done artificially and the white Gaussian Noise is added artificially as well. When working with short duration segments, we can get the desired SNR. For long speech segments, as the variance of the input noise is fixed, the SNR is not constant.

For the simulation concerning the Vuvuzela we use the files provided by Audionamix [77]. It consists of two MP3 files. One is the observation composed of: the vuvuzela, comments and the crowd's ambient sound. The second file is the solution provided by Audionamix. We did not pretend to offer a solution to this problem; our algorithm is not specialized to remove a particular source. We just use these sounds as an example for future improvements of the Kalman Algorithm for our model. Because we've got access to the information related to the vuvuzela with these two files, the mean spectral shape and the Long Term coefficient, we can modify our algorithm for this task. However, the analysis we are doing is actually not on a single vuvuzela remover but on multiple vuvuzelas remover. The modification is to consider that the "vuvuzela" is known (the parameters are learned on the "clean" vuvuzela sound) whereas the rest is unknown, less structured and has to be adapted.

5.1.1 Used Algorithms and initialization details

We use all the algorithms presented previously to analyse the observation. Two versions of the adaptive algorithm of chapter 3, namely the **Joint-EMK** that estimates the parameters jointly, and the **Alt-EMK** that alternates the estimation of the Short Term parameters and of the Long Term parameters, are compared to the filtering case **KF**. The parameters used for the filtering case are estimated on individual sources with the iterative method described in chapter 2 (and detailed in Appendix F.2).

The estimated parameters reflect the mean behaviour of the sources during the analysed frame. However, as the sources are not perfectly stationary during this time, they are not necessary the optimal parameters. On the other hand, the parameters estimated adaptively can become more optimal if the tracking operates well after the convergence.

The other algorithms, detailed in chapter 4, are called the **Naive-IS** and **Tmin-IS**. They are also compared to the filtering case that is presently called **VBSS**. As the **Tmin-IS** algorithm is not yet finished we use the Long Term parameters (Long term coefficient and periods) estimated by the **Naive-IS**.

In Appendix F.7, we discuss a Per Source Initialization. It uses the Itakura Saito Distance as well but in a weighted form in order to enhance one of the sources. This method, not used in the present simulations, aims at initializing the Short Term parameters and provides a (multi) pitch estimation.

All simulations are done with Matlab.

5.2 Short Duration real signals

In this part of the simulation, we still consider an observation composed of 2 sources with a noise. The signals are true speech segments of duration 400 ms', the first 400 ms being extracted from speech files from [35]. The SNR is set to 20 dB and the segments have been chosen to have an approximately constant period.
In this short duration segment, the periods of the two sources are assumed to be known while the others parameters are unknown (except for the two filtering cases). All the algorithms are initialized with zeros for the ST coefficients, 0.99 for the long term coefficient and variances are initialized to one. The noise variance is also given.

Figure 5.1 shows the waveform of the noisy observation and of individual sources. The blue signals correspond to the total signals, the red signals correspond to a reduced part in which we can make the following assumptions: the adaptive algorithms have converged, the frame-based algorithms are not influenced by the windowing (which is not removed in this example), the two sources are present. Because the signals do not have the same temporal envelope and the second source died before the first one, we compute the criteria in all the signals and in this reduced part.

The evaluation criteria are presented in Table 5.1; the value in brackets corresponds to the criteria evaluated in the stable part of the signals (red part of Figure 5.1)

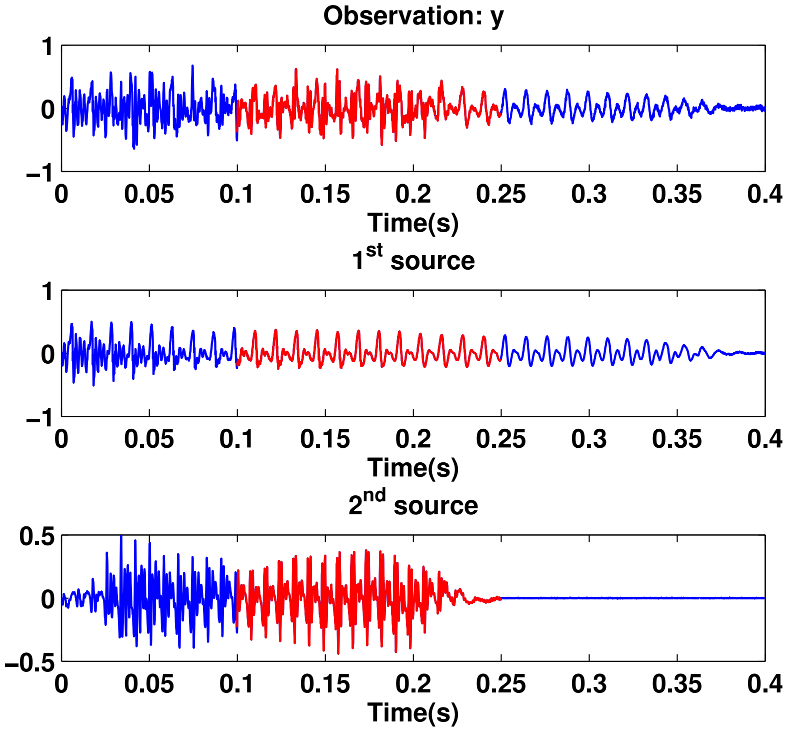


Figure 5.1: The observations and the sources, Short duration speech signals.

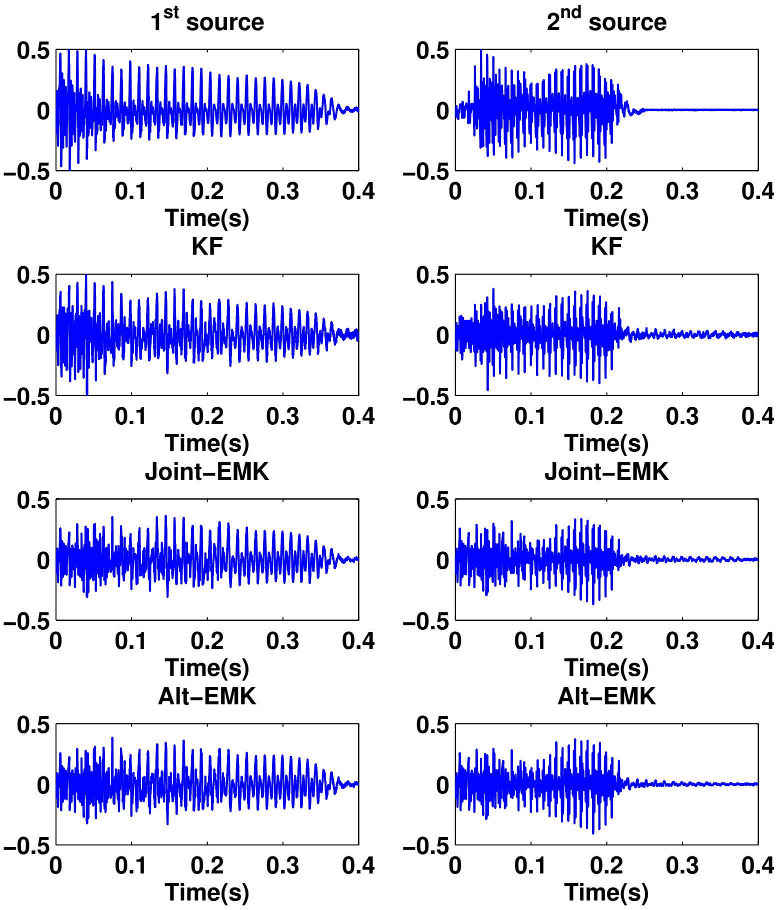


Figure 5.2: The sources and Estimated sources. Kalman Filter, **Joint-EMK** and **Alt-EMK**.

Table 5.1: MSE, SIR, SAR and SDR for real speech of short time duration.

Source	Algorithm	MSE	SIR	SAR	SDR
1	filtering **KF**	-25.14 (-26.34)	11.16 (10.60)	21.76 (21.41)	10.77 (10.23)
2	filtering **KF**	-25.55 (-26.67)	08.83 (16.31)	23.81 (21.30)	08.67 (15.09)
1	**Joint-EMK**	-23.42 (-24.88)	09.27 (09.03)	14.66 (21.61)	08.05 (08.76)
2	**Joint-EMK**	-23.51 (-24.90)	05.40 (12.68)	11.82 (20.26)	04.28 (11.95)
1	**Alt-EMK**	-23.71 (-25.44)	09.63 (09.59)	14.98 (21.61)	08.41 (09.29)
2	**Alt-EMK**	-23.82 (-25.50)	06.30 (13.75)	12.60 (21.90)	05.20 (13.11)
1	filtering **VBSS**	-31.07 (-29.49)	15.88 (15.11)	24.40 (21.52)	14.30 (14.30)
2	filtering **VBSS**	-31.31 (-29.51)	15.19 (15.20)	19.92 (21.80)	13.90 (14.30)
1	**Naive-IS**	-24.62 (-22.79)	12.58 (09.80)	09.40 (11.07)	07.54 (07.15)
2	**Naive-IS**	-26.03 (-28.96)	08.78 (08.02)	14.75 (15.28)	07.68 (07.22)
1	**Tmin-IS**	-16.70 (-14.76)	07.06 (07.29)	01.84 (05.52)	-00.09 (02.84)
2	**Tmin-IS**	-18.78 (-15.31)	03.70 (00.64)	05.28 (07.62)	00.71 (-00.73)

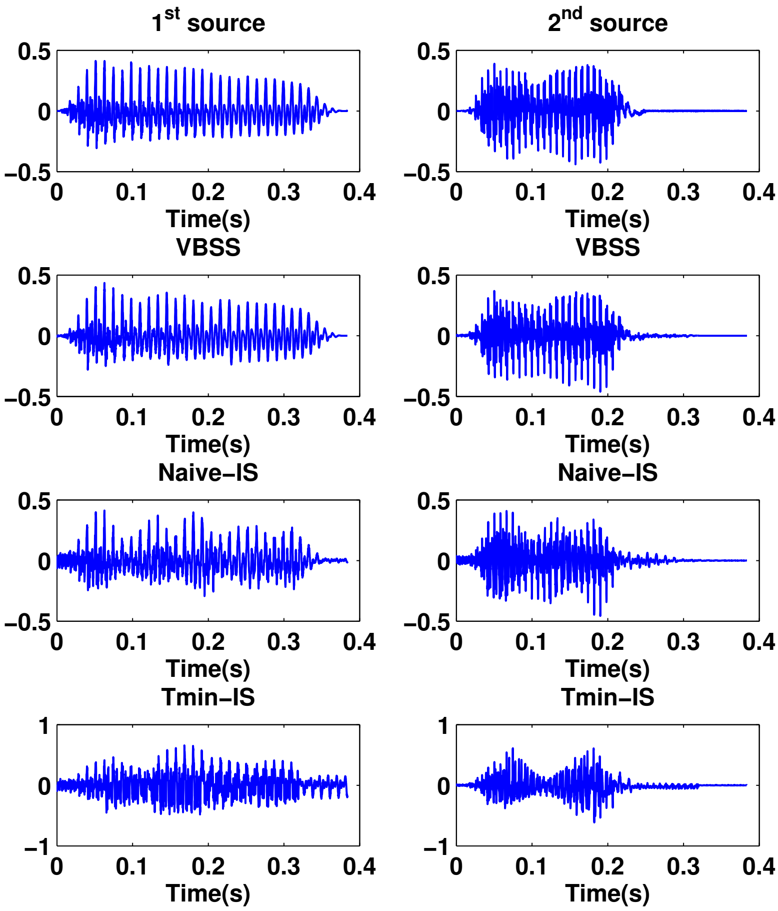


Figure 5.3: The sources and Estimated sources. Filtering , **Naive-IS** and **Tmin-IS**.

The separation results of the Kalman-based algorithms are shown in Figure 5.2 and those of the frame-based algorithms in Figure 5.3. They indicate that the algorithms are more or less similar except from the **Tmin-IS** that gives bad results. Note that the weighting, introduced by the analysis window for the frame-based algorithm, has been removed neither for the original signals nor for their estimations. However, we can notice that, for the whole signal, the **Alt-EMK** gives slightly better results than the **Joint-EMK** but neither of them reach the filtering performances. We have observed in section 3.8.4 of chapter 3 that the **Alt-EMK** converges to the sources faster than the **Joint-EMK**. Besides, compared to the **Joint-EMK**, the **Alt-EMK** always shows an improvement ranging from 0.5 to 1 dB. The **Naive-IS** has a better SIR than the Kalman-based estimation on the whole signal at the cost of a reduced SAR for the first source. The windowing affects the results; that is the reason why we show the criteria in the stable part. In this stable part, it appears that the Kalman-based methods are better. In the filtering case, the frame-based algorithm gives the best results and a source is not favoured as it is the case for Kalman-based methods.

For the next simulation we do not use the **Tmin-IS** algorithm.

5.2.1 Short Duration real signals, Comparison of Models and orders

Here, we reproduce the experience done in section 3.8.2 with real speech signals of short duration. The speech signals are the same as in the previous simulations and the evaluation criteria are computed in the same reduced interval (the red part of Figure 5.1). Concerning the filtering case, the results on real speech signals are different from those on synthetic signals. In Figure 5.4, we can now observe differences between the algorithms. The **AR-ST(T+p)** having the order of the source gives the best separation results and is followed by the proposed algorithm. Both **AR-ST(p)** and **AR-LT** give bad results.

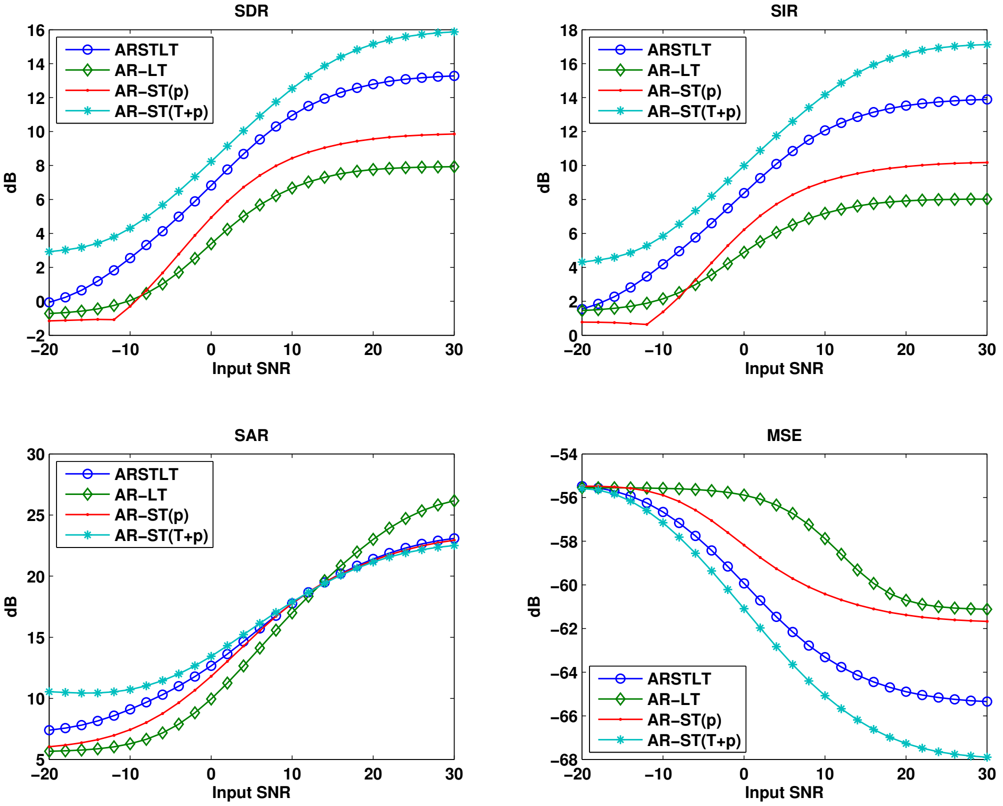


Figure 5.4: Models comparison. SDR, SIR, SAR and MSE in the filtering case.

In Figure 5.5, the results of the estimation case are shown. The models taking into account the long-term aspect of the signal give the best results now. The alternated estimation is better than the joint estimation, and AR models (low and high order) are not able to adapt the parameters sufficiently in order to separate the sources properly. Note that the **AR-LT** is the most robust, the estimation and filtering results are similar and that the peaks' information is crucial in order to separate quasi-periodic sources.

In Figure 5.6, we give the evaluation criteria for different values of the short-term order. We use the **Alt-EMK** which gave the best separation results in the previous simulation. The orders used are varying from 1 to 25 for both sources. Apart from orders of less than 3, we can observe that the results are equivalent and that the best sources' order depends on the source of interest. The best value of an evaluation criterion for a source defines the best order for this criterion of the two sources. However, this order doesn't give the best evaluation criterion result on the other source. High short-term order, ranging from 20 to 25, leads to too many coefficients to estimate, and gives bad results.

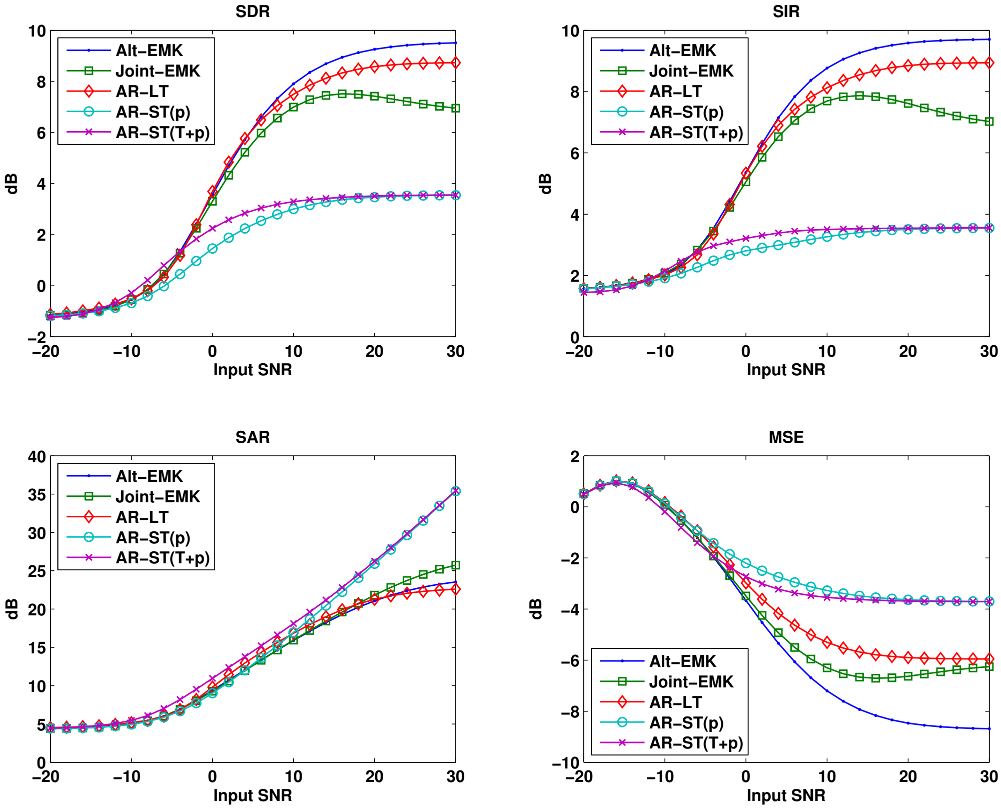


Figure 5.5: Models comparison. SDR, SIR, SAR and MSE in the Estimation case.

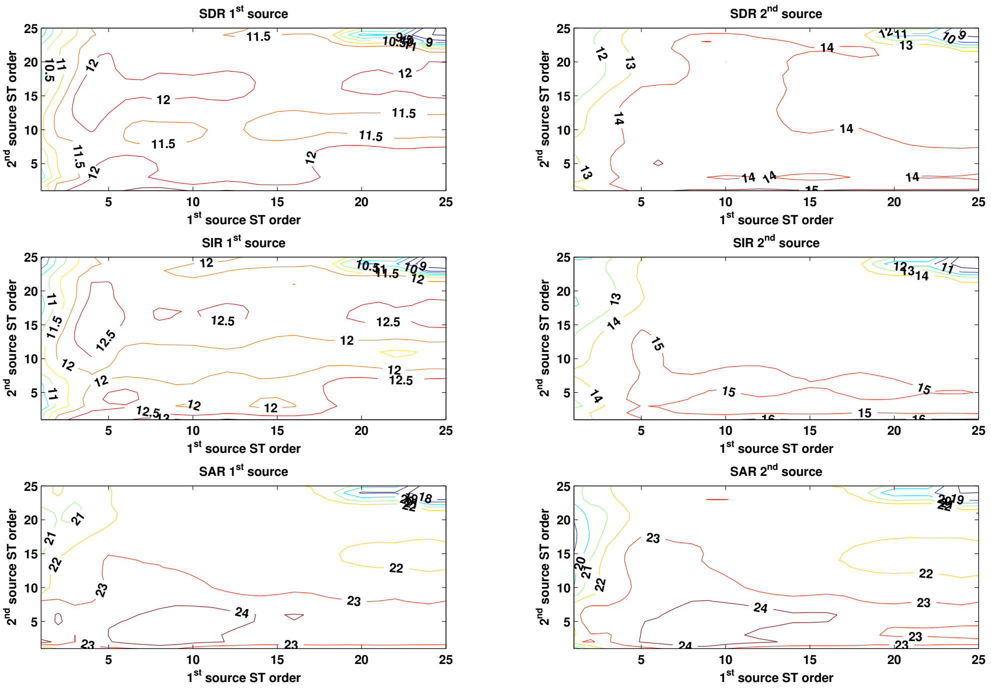


Figure 5.6: Models comparison. SDR, SIR, SAR and MSE in the Estimation case for different short term order.

5.3 Long Duration real signals

In this section, we consider long duration signals. We consider real speech signals as well as real signals of musical instruments. The speech signals have a ten seconds' duration. One of the speakers is a male and the other one is a female; both of them are English speakers. As mentioned in section 5.1, these signals come from the SASSEC 2007 Stereo Audio Source Separation Evaluation Campaign. We did not participate to this campaign but the signals are distributed under a "Creative Commons" licence [35]. The real signals of the musical instruments have a 5 s' duration, the instruments are a guitar and a cello. The guitar sound is a personal recording. Three notes are played successively and continuously. The notes played are "A2", "D3" and "E3" (midi code/frequency: "45/110Hz", "50/146.8Hz" and "52/164.8Hz") and are performed relatively quickly (six notes per second). The recording was done with a classical acoustic guitar using a "guitar pick" (or "plectrum"). The notes are more or less "palm-muted" (see section A.5.1 for more details about the interpretation of this effect). For the cello song, it is the first five seconds of the "Paganini caprice N24" played by Yo-Yoma [76] extracted from a commercial recording. The term "long duration signals" raises several issues that have not been mentioned so far. First of all, the sources are not always active so much so that we can find two, one or zero sources at a given moment. When less sources than assumed are present (this is also mentioned in section 3.8.3), an over fitting problem can appear. Such that the algorithms will try to find more sources than present. This leads to a partial share of the observation between the current estimates. The adaptive and frame-based algorithms use the parameters estimated previously for the current estimation so that after a noise/silence segment for example, they turn out to be sensitive to the apparition of a speaker and will need time to converge. We add a noise to the original signal . However, the signals have also an original noise (not necessary white) that we assimilate to the additive white Gaussian noise of our model.

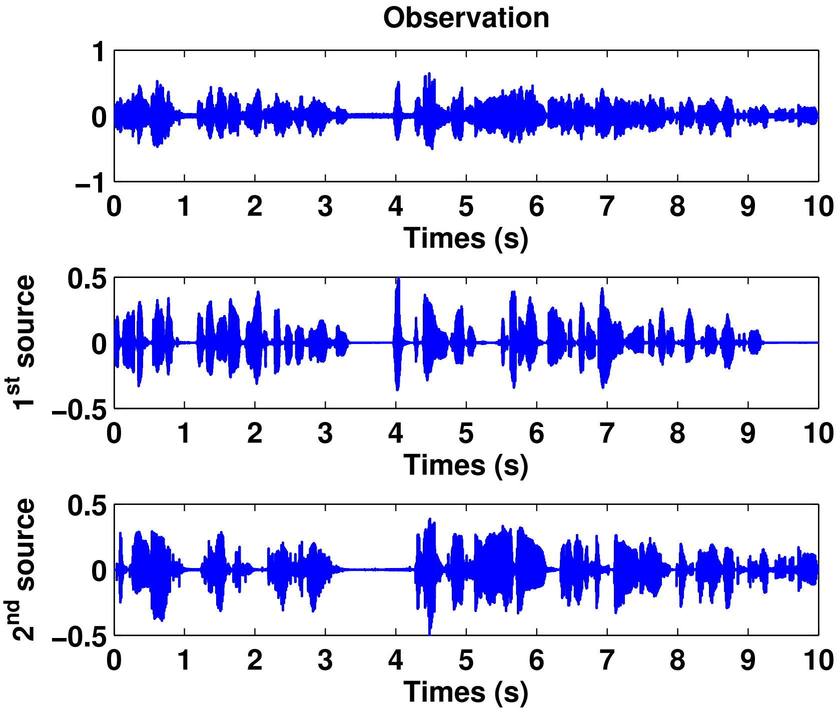


Figure 5.7: The observation and the sources, Long duration speech signals.

Regarding the speech signals, the speakers have been recorded with two microphones from different positions; we only take one channel (right). As mentioned in section 1.4 we don't take into account the possible delay and/or attenuation of the signals during their propagation process. The original signals are sampled at 16 KHz, we resample them to 8 KHz and we don't take into account the cutting frequency of the microphone. Finally, the signals are obviously not stationary and concerning the adaptive algorithm, a Multi-Pitch estimator has to run in parallel to estimate the periods.

5.3.1 Filtering Results

We first compare the two main filtering algorithms, the Kalman Filter and the Wiener Filter with our model. In both cases the parameters are estimated on individual sources, in a frame duration of 100 ms. For the Kalman filter the parameters are brutally updated every 100 ms. In the filtering case, it is not so important because we only update the State Space Model and the short plus long term error variances.

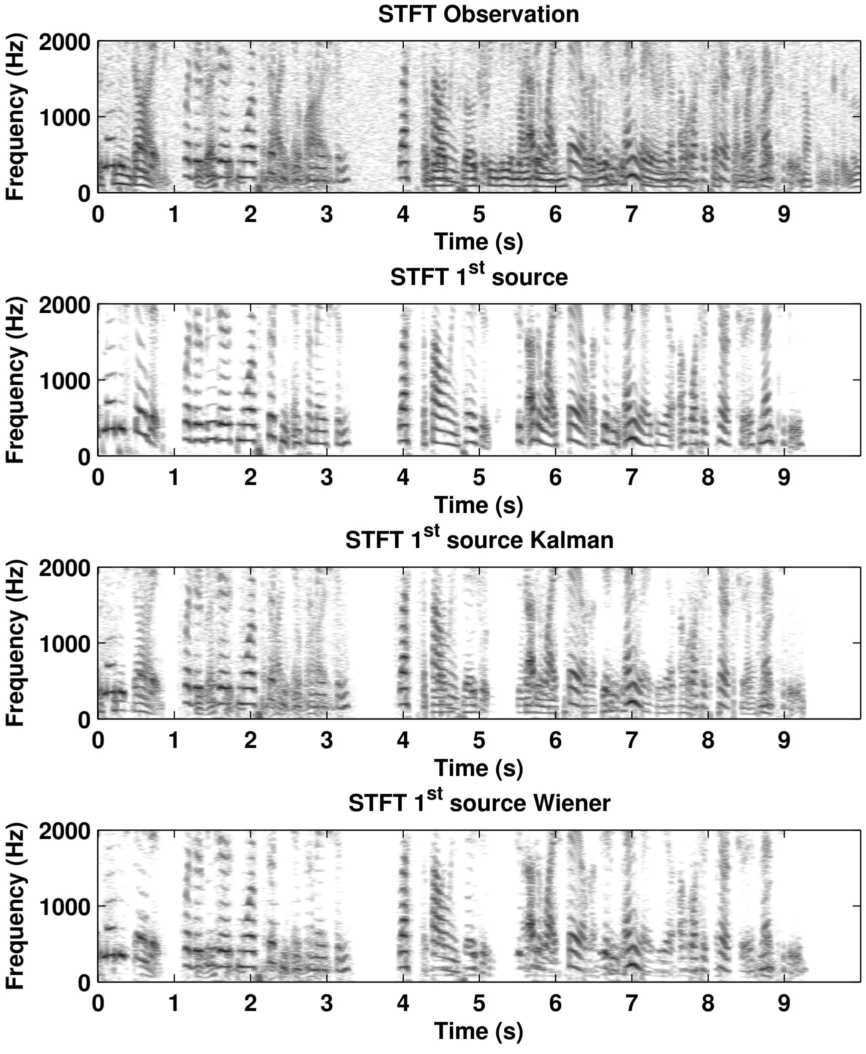


Figure 5.8: STFT of sources, and estimates extracted with a Kalman filter.

When listening to the extracted sources, we notice that the separation is better for the frame-based algorithm but the voice seems to be less natural than for the Kalman-based algorithm. Some artefacts are surely present for the frame-based algorithm, however the second source is clearly audible in the Kalman's results. This is also visible on the Short Time Fourier Transform of the extracted sources, in Figure 5.8 for the first source and in Figure 5.9 for the second source. In the Kalman's estimates, we can observe (around 1.5 s for the second source and almost everywhere for the first source) that the extracted source contains frequency components from the other source and also some noise residuals. Both algorithms use the same parameters. The estimation is done with the same frame size. In this example, the separation result (in this example) is better for the Wiener Filter than for the Kalman Filter. By averaging the (local) evaluation criteria (see Appendix F.4.3) in non-silent parts of the signal, we obtain the mean criteria in Table 5.2, which leads to the same conclusion.

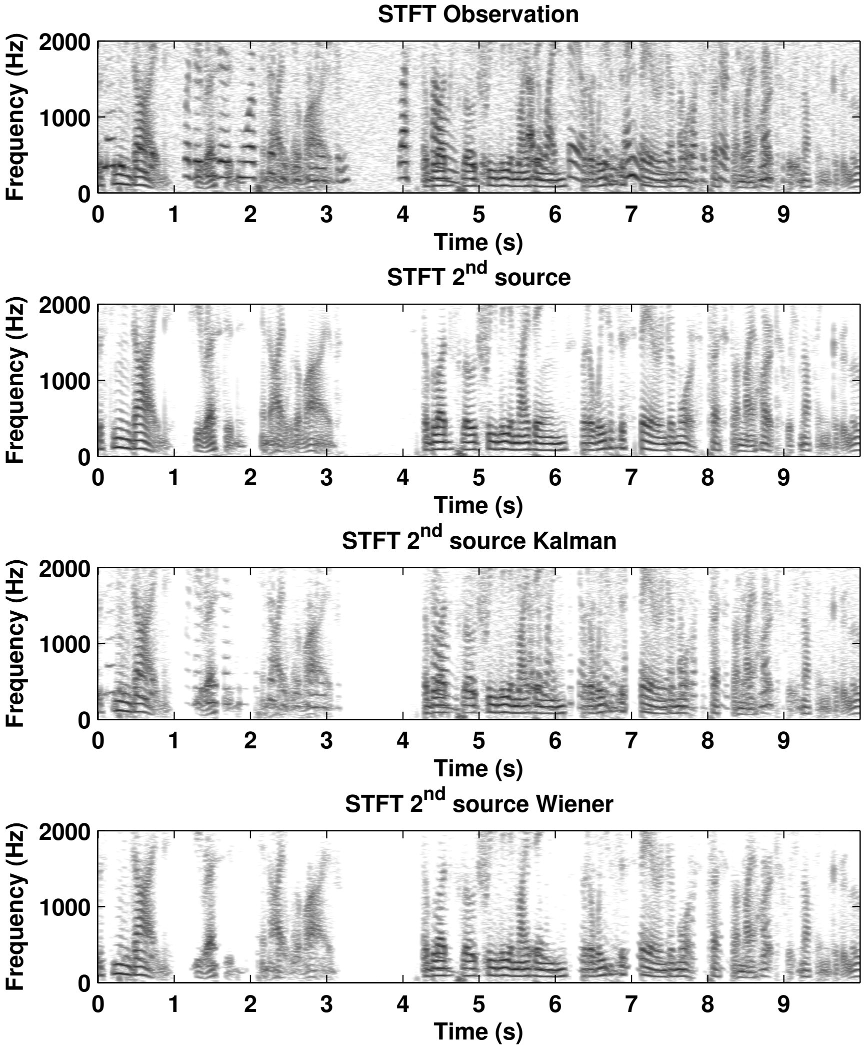


Figure 5.9: STFT of sources, and estimates extracted with a Wiener filter.

5.3.2 Estimation Results with Alt-EMK and ”known” periods

In this section, we present the separation results of the **Alt-EMK**. In section 3.8.3 and 3.8.4, we have presented the separation results on synthetic data for the two algorithms: **Joint-EMK** and **Alt-EMK**. In both cases the **Alt-EMK** shows better results compared to the speed of convergence. It appears that, on real data, the **Joint-EMK** is not able to separate the sources properly. This point will be discussed later in this section.

We assume that we know the periods and the variance of the additive noise. The periods are estimated automatically every $64ms$ (512 samples at 8Khz) on the original sources and are not necessarily the exact ones. We use the procedure explained in Appendix F.7.2 to estimate the periods. This is a spectral method; it gives similar results to the Spectral Sum [128]. Spectral methods tend to make octave errors and, in fact, it is not the case in the simulations (only some frames for the second source). We did not correct the estimate as a manually annotated ground truth. However, as the estimation is updated every 64ms, the errors are quickly forgotten.

If the pitch strength (i.e. the result of the detection function of the pitch estimator) is lower than a defined threshold, the segment is assumed to be unvoiced. In order to limit the overfitting problem when a source is declared to be unvoiced, we force the Long Term coefficient to be equal to zero and the Short plus Long Term prediction error variance is replaced by the Short Term prediction error variance.

5.3.2.1 Speech Signals

The two speech signals considered here are the same as the ones used for the filtering. Figure 5.10 shows the estimated Fundamental Frequencies of the sources, the strength and the threshold which is very low. The results of the estimation are shown in Figure 5.11 in the temporal domain and in the time-frequency domain (magnitude of the STFT in dB) in Figure 5.12.

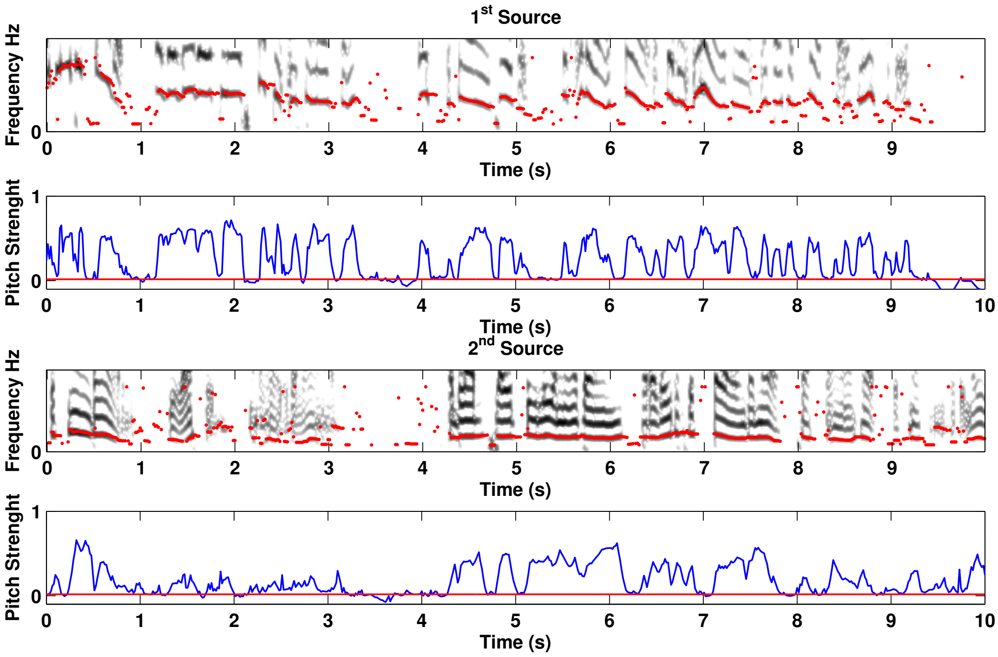


Figure 5.10: Estimated Fondamental Frequency (0-500Hz) and related strenght

The temporal amplitude of the estimated sources is not too badly tracked for the first five seconds. In the next five seconds, the original sources are more overlapped and the results are deteriorated. For example, we can observe that, at the end (9-10s), the first source estimate takes the second source. Also, we can observe that the estimated sources are able to get close to zero and so to follow the appearance and the disappearance of the sources. At four seconds, after a "long" silence, the first source is well modelled and the second estimate does not try to model it. This is surprising compared to previous observations.

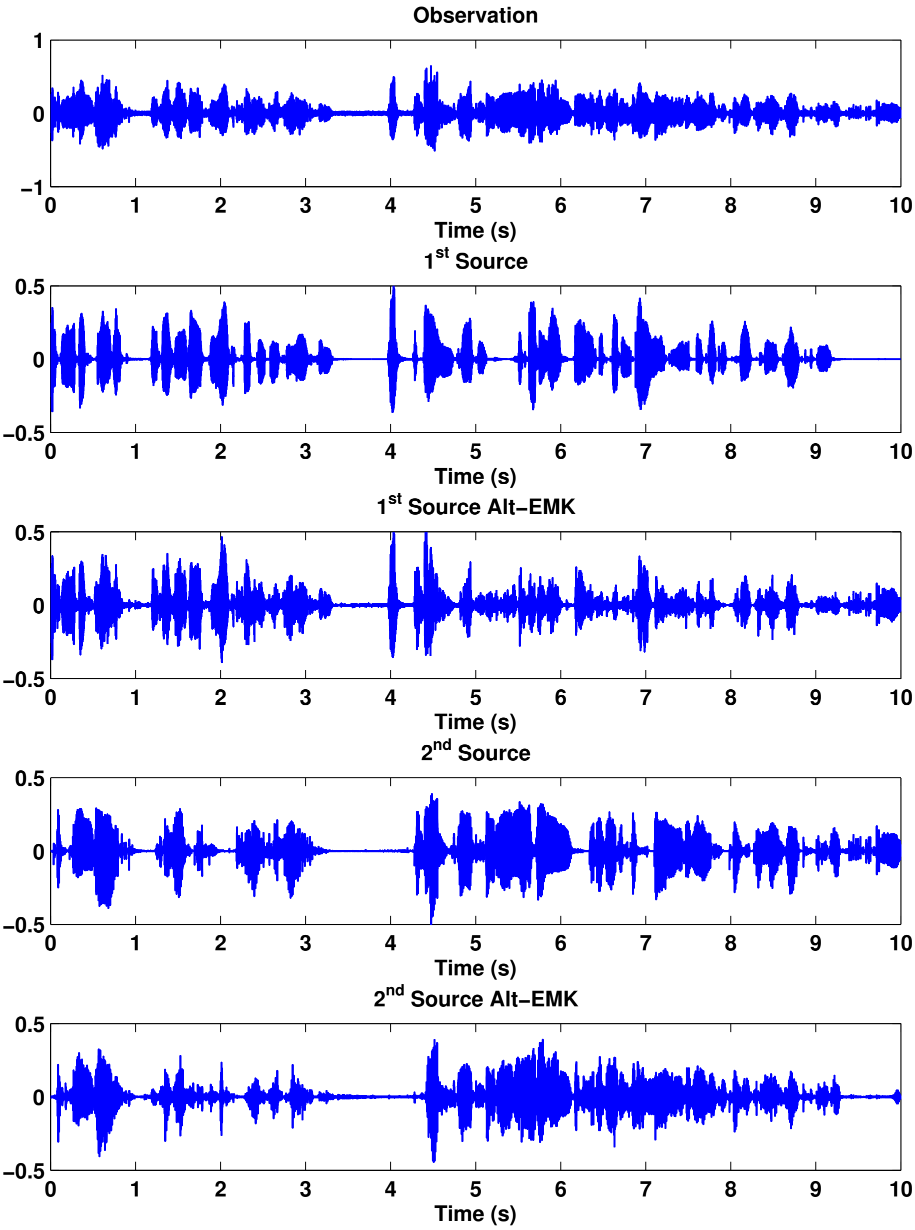


Figure 5.11: The observation, sources and estimates extracted with **Alt-EMK**. Long duration speech signals.

In Figure 5.12, we show the magnitude STFT (in dB) of the observation, the sources and the estimated sources. Except for some problems like the source inversion at the end and some interference residuals, the result is quite satisfying. The results are not perfect but most of the spectral contents are well modelled. The frequency comb's residual seems to be equivalent in the estimation case to the one in the filtering case. The frequency comb's components are sometimes shared between the two estimated sources just after 3s or around 7s.

As we can observe on Table 5.2, the performance of the **Alt-EMK** is very close to the filtering one. In terms of MSE, the results are better. This can indicate that the parameters used for the filtering case were not the perfect ones. Also, regarding the filtering case, the parameters are supposed to be constant during two updates. For the estimation case, even if the algorithm has first to converge to the sources, it updates the parameters at each iteration/sample leading sometimes to a better modelling of the sources.

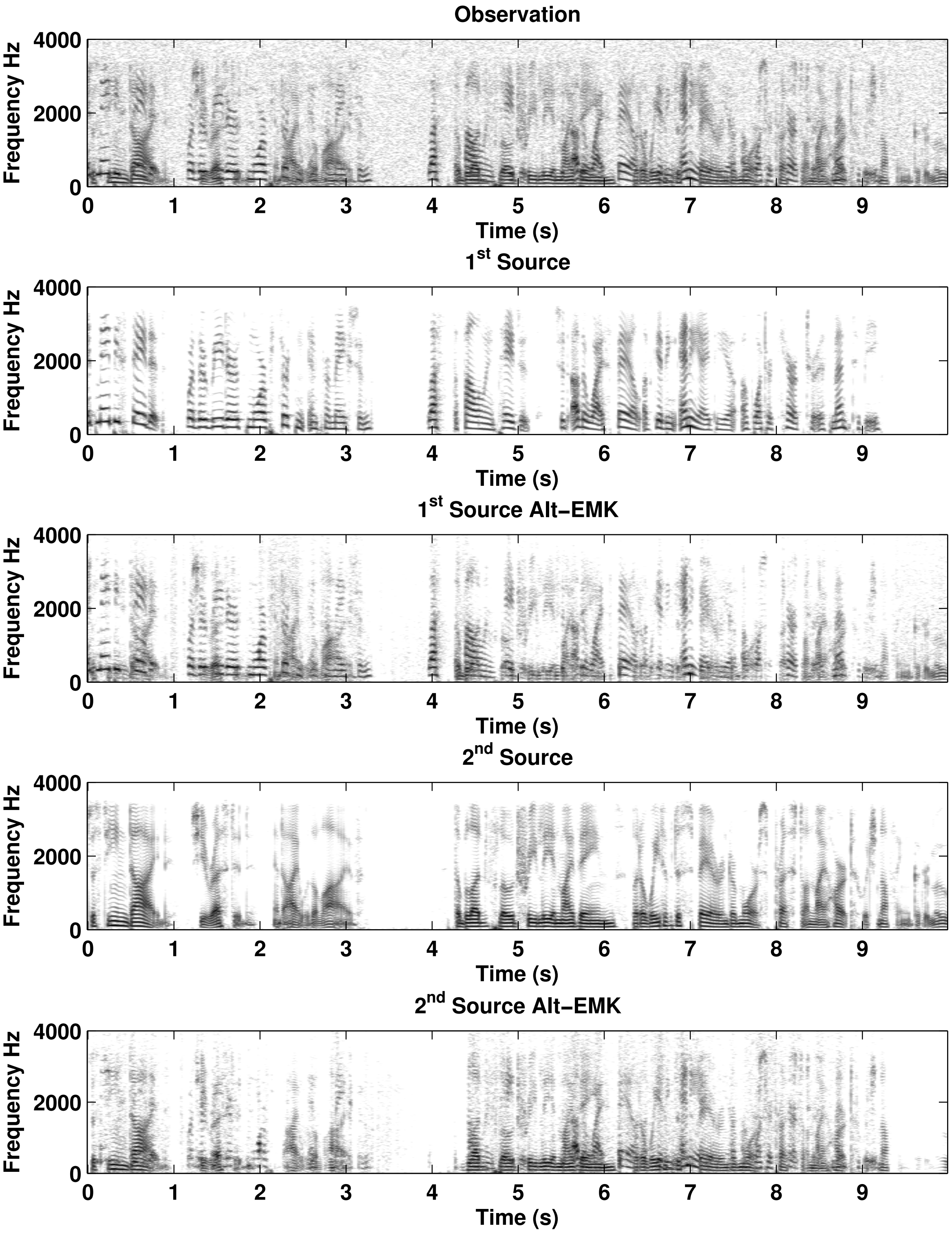
Table 5.2: Evaluation criteria, filtering of a real speech of long time duration

Algorithm	MSE	SIR	SAR	SDR
Kalman	-29.1 — -29.2	16.9 — 15.1	21.3 — 20.8	14.6 — 12.9
Wiener	-32.2 — -32.3	19.2 — 17.7	22.7 — 21.7	16.9 — 15.4
Alt-EMK	-33.6 — -33.5	12.6 — 15.1	19.2 — 18.6	12.9 — 10.9

5.3.2.2 Discussion about the simulations and obtained results

In the simulations, the global SNR is set to 20 dB, the order of the Short term AR model is fixed and equal to 5 for the two sources. The forgetting factor values are the following: 0.99 for the Short Term estimation and 0.8 for the Long Term estimation. This difference between the forgetting factor values is responsible for the performance differences between the **Joint-EMK** and **Alt-EMK**. The results of the **Joint-EMK** are not presented because it does not separate the source when the periods vary. An example is shown in Appendix F.8. This is due to a problem in the Joint estimation. Sometimes, the estimated variance explodes (this stability problem for the joint estimation of short and long term predictor is also mentioned in [69]).

The lack of convergence can come from different parameters. If the estimated Long Term coefficient of all the sources is too low (or equal to zero), we fall in a purely Short Term aspect of the model. Simulations of section 3.8.2.1 show that, in this case, the separation power is not so good. If, due to the inactivity/apparition of a source, the estimated variances are too different, such as a source having a variance very low compared to the other one, the source that has the greatest variance absorbs the other one. As a result, one of the sources contains most of the observation and the other one contains a very low residual. This fact is visible on the waveform (Figure 5.11), just after 6s, the first source estimate gets close to zero (tracking its source) while the second estimate doesn't track its source. From this moment, the second estimate stops to follow the temporal envelope and the first estimate is always underestimated (except at the end). This observation leads to an "obvious" conclusion, the algorithm sometimes needs to be re-initialized, at least when the observation is too low.


Observation

1st Source

1st Source Alt–EMK

2nd Source

2nd Source Alt–EMK

Figure 5.12: STFT of the observation, sources, and estimates extracted with **Alt-EMK**.

5.3.2.3 Instrumental Signals

We also consider the separation of instrumental sounds. Speech and music can be modelled with the same model. However, they are different: the sound of an instrument is richer than the sound of a voice. As mentioned in section 5.3, the observation is composed of a cello song and of a guitar song with an additive white Gaussian noise leading to a global SNR of 20 dB. The mean evaluation criteria for this example are (cello/guitar) 8.8 dB and 9.4 dB for the SDR, 13.9 dB and 13.4 db for the SAR, 12.8 dB and 12.3 dB for the SIR and finally -27.1 and -27.2 for the MSE.

The results seem to be worse than for speech signals. However, when listening to the results it is not the case. The used model is then responsible of the degradation. The extracted cello signal is almost the same, but the estimated signal is slightly under-estimated; its transient part due to the bowing is not modelled (see section C.1 for more details about this). As a result, in the estimation of the second source i.e. the guitar, the fricative song of the bow is clearly audible. Note that the Paganini song is not really simple. This example is satisfactory in the sense that the model can also be used for music application. The structure is more complex than for the speech (the periods change more quickly). However, none of the sources disappears, they are always present. In an adaptive algorithm, we cannot use different window lengths to model the transient part like in [79]; it is a limitation of the model. Also, it is not really possible to model the period evolution during the transient part, and if it were, the speed of convergence of the algorithm would have to be highly improved.

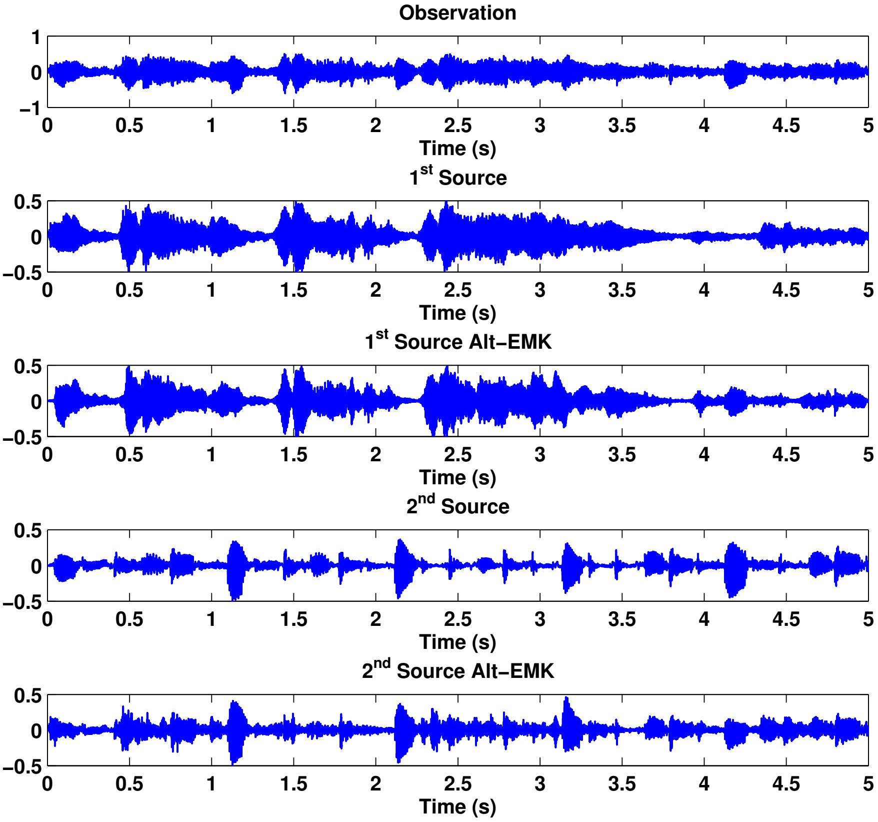


Figure 5.13: The observation, sources and estimates extracted with **Alt-EMK**. Long duration Music signals.

5.4 Performances discussion

In this part of the thesis, we consider the Mono-Microphone source separation problem for audio signals. We focused more precisely on speech signals during the simulations. In order to evaluate the performances of proposed algorithms, we used some evaluation criteria introduced in [80] with the associated matlab code [81]. Other works use the same evaluation criteria in a different context but not necessarily with speech signals. In this section, we aim at comparing the results with other methods as much as possible.

Most of the works we compared focused on Music/Speech separation or Music/Music separation and are essentially based on training. In the case of speech/speech separation, if no constraint is used as for example the Male/Female separation, the use of a model based on training will not necessarily give good results. On the other hand, the model we used is essentially based on the fact that we can represent a source by its periodicity. If the periodicity is not clearly present and/or if the song is inharmonic, our model is not necessarily better. The model we used needs also more precision, in the sense that the periods have to be estimated and tracked. However, this constraint allows separating the signals that have the same nature (if the periods are different).

In Table 5.4 we show the results we obtained on short time duration speech signals (≈ stationary periods) and on long time duration signals (with all the inconveniences this may have). We try to compare the obtained results with methods based on:

- GMM/GSMM methods [18,42,45,82,83]
- GMM/HMM methods [44,84]
- AR/GMM/Amplitude factor methods [83]

We will name the methods with the used method, give the separation context and the results of the criteria, namely the SDR, SIR and SAR. Concerning the criteria, in Table 5.3, we give the range of values obtained by the authors because they generally use different versions of their algorithms, for example they use different number of states and/or models.

Table 5.3: Criteria in dB.

Context	Algorithm	object	SDR	SAR	SIR
Jazz Music	GMM [82]	music	3.8 → 4.2	7.7→ 8.6	6.1 → 7.4
Vs Voice		voice	-2.3 → -1.9	-1.7 → 0.1	5.1 → 10.1
Piano Vs	GMM [45]	piano	**X**	2→ 8.9	10.5 → 20
Bass and Drum		drum	**X**	4→ 9.6	2.8 →18.8
Voice Vs	GMM/AR [83]	voice	3.9 → 5.4	2 → 4.7	2.2 → 4.6
Music	/Amp Factor	music	2.1 → 3	3.2 → 12.9	5 → 11.1
Piano	GMM/HMM [44]	piano	**X**	4.2 → 8.9	10 → 35.7
Vs Drum		drum	**X**	9.7 → 12.5	4.9 → 5.9
Voice Vs	Factorial [84]	voice	0.4 → 5.7	**X**	**X**
Music	Scaled HMM	music	9.6 → 14.9	**X**	**X**
Main Instrument	Hybrid Model [18]	Main	1.6 → 8.2	4.1 → 8.9	5.4 → 17.1
Vs Accompaniment		Acc.	7.7 → 10.1	10.7 → 12.9	14.1 → 15.4
Voice Vs Music	GMM [42]	voice	NSDR 5 → 13		

The last work uses a criteria named Normalised SDR (NSDR). It is the SDR of Estimation/source minus the SDR of Observation/source. Here, the comparison is only informative as we are working with an additive noise in the observation, leading to noise residuals in the estimated source, and not them.
In Table 5.4, we give the range of values for the evaluation criteria for the proposed algorithms. In the previous table (Table 5.3), the range of values concerned different methods/models/parameters. The present table represents the range of values (min to max for each source): some segments are modelled better than others.
As we can observe when comparing the results, we are in the state of the art. However, the comparison is not so easy to do. Separating Voice Vs Voice can look easier than separating Voice Vs Music. However, note that in the Voice/Voice problem, the two sources have the same nature but that also the range of fundamental frequencies for a male and a female overlap. We can still say that the Kalman-based algorithm introduces fewer artefacts than the other algorithms.

Table 5.4: Criteria in dB. Proposed algorithm

Context	Algorithm	SDR	SAR	SIR	NSDR
Short Duration					
Voice Vs Voice	**Joint-EMK**	4.3 → 11.9	11.8→ 21.6	5.4 → 12.6	3.3 → 9.4
Voice Vs Voice	**Alt-EMK**	4.3 → 11.9	12.6→ 21.9	6.3 → 13.7	3.7 → 10.5
Voice Vs Voice	**Naive-IS**	7.1 → 7.7	9.4 → 15.2	8 → 12.5	4 → 7.6
Long Duration					
Voice Vs Voice	**Alt-EMK**	4.9 → 20.7	11→ 29	8.6 → 30	3.7 → 27
Guitar Vs Cello	**Alt-EMK**	3.1 → 18	5→ 21.2	3.6 → 21.4	2.2 → 17.7

The proposed algorithms also have a strong limitation due to the Long Term Aspect. However, note that in the result of the state of the art, the voice separation gives worse results than for the associated music. In our model, a source is monophonic with a defined period. If the speech/instrument is not voiced/harmonic, as it is the case for unvoiced speech or for an instrument like the piano, the bass or the drum etc., we are not sure to obtain the same quality of separation. As far as the evaluation criteria are concerned, the simulation with the instrument's sounds gives slightly worse results than for the speech. This can be explained by the fact that an instrument possesses more harmonics to model and that the evolution of the spectral can be quicker, compared to a voice.
As aforementioned, for the long duration signal, the adaptive estimation can lead to a better result on some segments than for the filtering.

5.5 Vuvuzela remover

The vuvuzela is used during ”Football” matches in South Africa where the stadiums are filled with its loud sound. The acoustical sound pressure is about 120 dB at one meter and that is the threshold of pain. These high sound pressure levels, at short distance, can lead to permanent hearing loss for unprotected ears. The intensity of the sound caught the attention of the football ”community” during the 2010 FIFA World Cup and has been a subject of controversy.

For the audience, it can become annoying to hear the vuvuzela’s sound permanently during the broadcasting of the games. A first approach to remove this annoyance would be to use a notch filter in order to suppress the frequency bands corresponding to the pitch (frequency of $\approx$ 230 Hz) of the vuvuzela [85]. Besides, this method greatly affects the comments. Numerous solutions based on adaptive filtering can be found but are not reviewed here.

5.5.1 Procedure and details

In the simulation considered in this section, we are not considering a single vuvuzela but a multitude of vuvuzelas playing together. As mentioned in section 5.1, we consider that we know the source parameters of the vuvuzelas. These parameters are estimated on a segment of an extracted vuvuzela sound from a real recording, and reflect its mean parameters. The extracted segments we used come from the Audionamix solution [77]. They provide a real football match recording before and after the vuvuzela removing. The difference between the two sounds gives the vuvuzela with a very low residual. At this moment, we can filter the vuvuzela but the background sounds (comments etc.) have also to be modelled. We will consider that this background sound can be modelled by an Auto-Regressive model of high order (50 in the simulation) that has to be adapted. We modify the algorithm presented in chapter 3 assuming that a source is fixed/known and that the other one is a high order AR model. In this situation, no Long Term AR parameters (according to the definition given in this thesis) have to be estimated so that the algorithms, namely **Joint-EMK** and **Alt-EMK**, are the same. We also add a white Gaussian noise to the original sound leading to a global but not constant SNR of 30dB.

5.5.2 Results

The Short Time Fourier Transform (STFT) of the first 4 seconds of the original song is shown in Figure 5.14. On the same Figure, the STFT of the observation (Background plus vuvuzela plus additive white noise), the estimate and the vuvuzela (obtained as aforementioned) are also shown. Regarding the vuvuzela’s sound, we can see that it is essentially composed of two tones ($\approx$ 230 and 460 Hz). These two tones are visible in the observation (between voiced segment, for example between 1.5 s and 2 s or around 3 s) and seem to be insignificant but when we listen to the sound, we hear them easily. On the estimate, we notice that the time of convergence is not negligible (before 0.5 s). We can also observe the reduction of the two vuvuzela’s tones (same range of time than before). This reduces the tones of the comments but they are still present and understandable. For the record, the evaluation criteria are: 14.15 dB for the SDR, a SIR of 17.75 dB, a SAR of 16.7 dB and a MSE of -12.41 dB evaluated with the Audionamix solution. The STFT is computed with 4096 FFT points. It uses a hanning window of 128 samples’ length with a sampling frequency of 8Khz and an overlap of 25%.

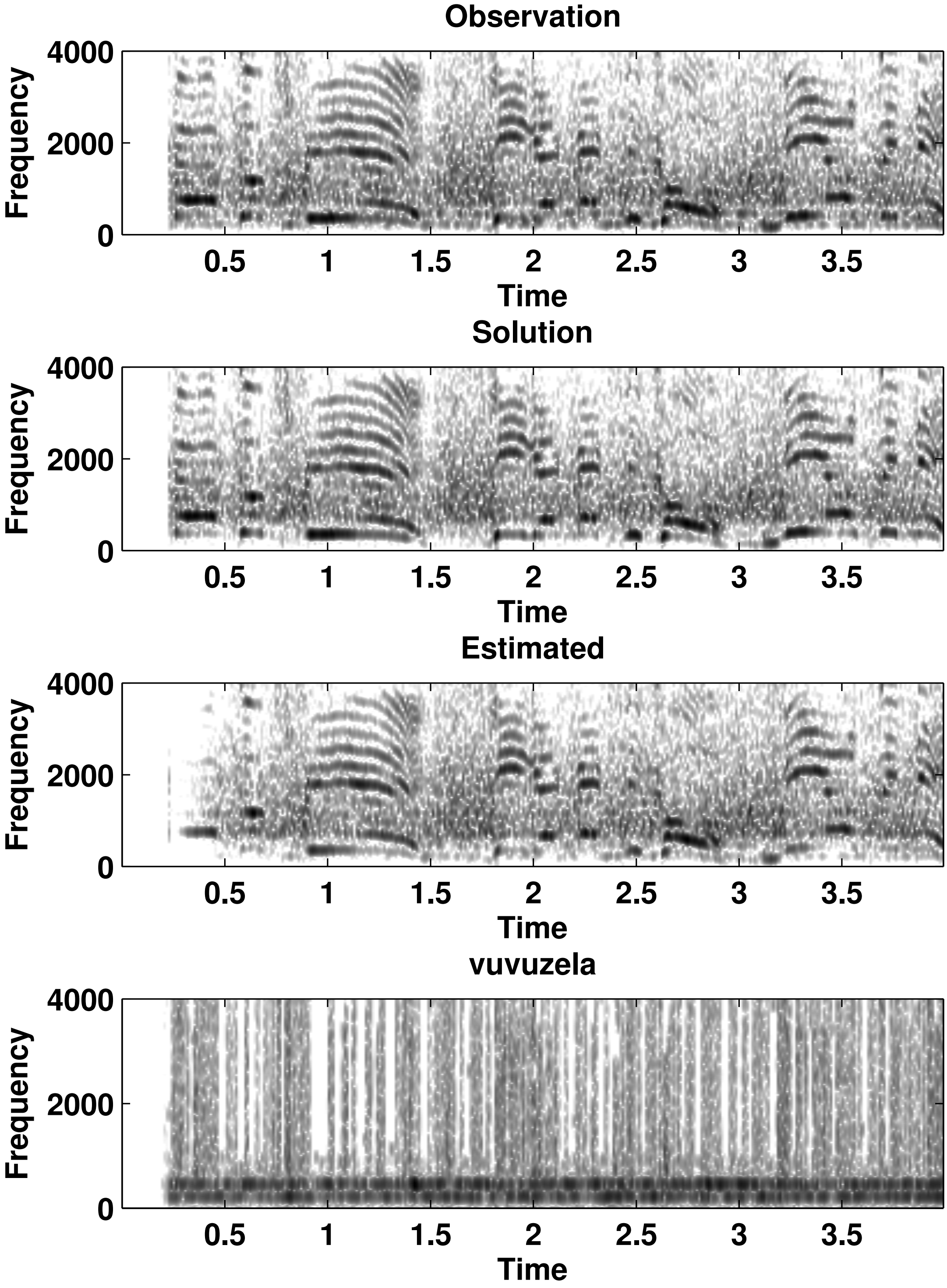


Figure 5.14: Vuvuzela Remover example.

This is just an example of a different use of the algorithm.

5.6 Summary

In the present chapter, we considered simulations on real speech signals as well as on musical instrument signals. We also compared the results with other works, as much as possible. The obtained results are comparable to the state of the art in this field. But the comparison is difficult. The context is different for each work and we did not use the existing methods on our signals. However, the simulations have been mostly done with the **Alt-EMK** algorithm because the **Joint-EMK** shows poor results when the periods vary (as shown in Appendix F.8) but also because the **Tmin-IS** is not finalized and the **Naive-IS** takes too much time for the moment. We also used this algorithm for the background extraction task involving data from the FIFA world cup 2010 in order to remove the vuvuzelas from the original signals.

In the next chapter, we will give the general conclusion of our work and some potential improvements to take into account.

Chapter 6

Conclusion

6.1 Summary and conclusions

In this part of the thesis, we have tackled the problem of Blind Mono-Microphone Audio Source Separation. We proposed two algorithms in the temporal domain and two in the frequency domain. Throughout this part, we have worked with a source filter model defined in Chapter 2. With this model, a source is modeled as the cascade of two Auto-Regressive processes of a very different order: a short and a high order. The short order one models the short-term correlations of the signal that corresponds to the spectral shape. The high order model is very sparse with only two or three non-zero elements. We call it a Long Term AR. It is a comb-like filter and it aims at representing the long-term correlations of the signal. It defines the temporal periodicity and the resulting frequency comb. The strength of the periodicity can be tuned by the so-called Long Term coefficient. When the long-term parameter is high, the model corresponds to a voiced speech. However, when it is low, it corresponds to an unvoiced speech. Also, arbitrary fundamental frequency can be achieved thanks to an interpolation factor so that the model is not restricted to the integer period. The observations, a linear mixture of the sources, are then composed of a sum of AR Gaussian sources to which we add a white Gaussian noise. The noise can be observed as an AR of order zero. This parametric model is entirely defined but its parameters as well as the designed algorithms estimate the sources' parameters in order to extract the sources.

In the temporal domain, the algorithms follow the Adaptive Expectation Maximization Kalman Filter methodology (EMKF). According to the model, two versions of the algorithms are proposed. In both cases, the sources are jointly estimated. The first one, namely the **Joint-EMK**, estimates the parameters jointly, while the second algorithm, **Alt-EMK**, alternates the estimation of the Short Term and Long Term parameters. This second version of the algorithm allows the use of different forgetting factors: one for the Short Term and one for the Long Term. As these two aspects are not necessary evolving at the same speed, the use of two forgetting factors helps the tracking of the sources. In synthetic data simulations, we have also considered taking only one of the two AR source models (long term and/or short term). It appears that a traditional AR model of high order leads to better filtering results but lower estimation results. The number of coefficient that has to be estimated is too high, whereas the proposed model is comparatively represented by a low number of coefficients. The comparison leads us to continue with the proposed model.

The algorithms are defined in Chapter 3 with some results on synthetic data. The two algorithms give almost the same kind of results. On synthetic data, the difference is too low to conclude if one is better than the other. However, the alternate estimation seems to converge faster to the source, which is not negligible when dealing with non-stationary signals. In any case, the EMKF algorithms need to know the periods of the sources and a multipitch estimator has to run in parallel. This is actually a significant drawback because the multipitch estimation is not an easy task, but compared to the other parameters that need to be estimated, it is the easiest estimation to do. Another problem related to the periods appears when the periods of different sources are closed or cross each other, as shown in Chapter 3. In these simulations, the periods of two synthetic sources vary linearly and intersect at some point in time, depending on the situation, which is related to the relaxation time of the Long Term part of the model, a source inversion is possible. Another problem comes from the fact that the sources are not always active, the number of sources varies with respect to time. This issue can also be solved by a robust multipitch estimation, except if the sources have the same fundamental frequency. Finally, during the simulations, we considered the sources with a constant Short Term order equal for the different sources. A single simulation on real signals with different Short Term order (for the two sources) shows that the best model order is included between the orders 5 and 20. In order to improve the results, it would be beneficial to estimate the order of the sources. In some cases, if the "good" order is used, and when the support is really different, it could allow to separate the sources with a low periodicity (when the sources are dominated by the short term AR). We also applied the algorithm to musical instruments signals. In this case, the algorithm had a better performance and this is essentially due to the fact that musical instruments possess more frequency peaks, and, in the considered simulations the sources are active more often. However, treating the single sensor source separation problem in the temporal domain is not so used. Besides, we used the EMKF in an enjoyable application, in order to remove the vuvuzela from a football match recording. The algorithm is modified and does not look like the one used for the speech separations. We considered a half filtering scheme by assuming a known source, in this instance the vuvuzela, while the other signals were adapted and considered as a colored noise.

Regarding the Frequency domain algorithms, as the analyzed signals are not stationary, the use of frames based on algorithms is unavoidable. In Chapter 4, two algo- rithms based on the Itakura Saito (IS) distance have been described in order to estimate the parameters; one of them is not yet finalized. We also proposed a Wiener like filter in order to separate the sources. This algorithm uses some simplification thanks to the design of the window in the frequency domain and to the use of circulant matrices for the filtering process. The extracted sources are windowed by default and this leads to adding only the sources in the overlap add procedure. We tested the parameters estimation algorithms on a syn- thetic spectrum and the results were quite good. The algorithm leads to almost perfect results when the Long Term part of the model is known. The first algorithm, named **Naive-IS**, is able to estimate all the parameters, including the periods of the sources, and gives reasonable results on real data. It is based on a Naive interpretation of the IS distance and uses a basic linear prediction to estimate the source parameters directly from the mixture, or more exactly a cleaned version of the mixture. The second algorithm, namely **Tmin-IS**, is based on the minimization of the IS distance and its gradient has a connection with the Weighted Spectrum Matching and the Gaussian Maximum Likelihood. Nevertheless, we used the Weighted Spectrum Matching to estimate the sources' variances. This algorithm is really promising but the Long Term part estimation is not yet valid and leads to problems. The Short Term parameters are estimated iteratively by solving a set of (Yule-Walker like) equations with a non-zero right hand side.

In the previous chapter, we considered simulations on real speech signals as well as musical instrument signals. We also compared the results with other works as much as possible. The obtained results are comparable to the state of the art in this field. But the comparison is difficult. The context is different for each work and we did not use the existing methods on our signals. However, the simulations have been mostly done with the **Alt-EMK** algorithm because first the **Joint-EMK** showed poor results when the periods were varying (as shown in Appendix F.8), then the **Tmin-IS** was not finalized and finally because the **Naive-IS** took too much time for the moment.

6.2 Potential Improvements

When dealing with a task as specialized as the separation of speech and/or Music, it may seem natural to use linguistic/Musicologic facts for the separation. For example, the temporal evolution of the signal, a fixed range of fundamental frequencies set (for particular music instrument) analyzing the contents of speech etc. However, this also adds constraints and leads to estimate more and more parameters. This fact has to be envisaged for more a particular application.

As the temporal method is based on a Kalman Filter, several contributions/modifications can be envisaged. A robust relaxation of the forgetting factor when sources are changing too fast and the use of sources dependent on forgetting factors can also help if one source is more sta- tionary than the others. For the Alternate estimation, we used different forgetting factors for the Short and the Long Term part. The model can also be reconsidered in a non-linear way and analyzed with an Extended Kalman Filter. This is already being analyzed by our team. A multiple system can also be envisaged as a dual system, one aiming at estimating the Short Term (ST) part and the other one the Long Term (LT) part, and why not as a preprocessing of the proposed algorithms. This will allow to estimate the ST parameters in the LT prediction error and the other way round. Along these lines, an EMKF algorithm can be designed to track the periods and optimize the forgetting factors. The forgetting factor relaxation is crucial when the periods are varying in order to allow a quicker adaptation to the sources. For more practical improvements, the algorithm should be able to change the number of sources and adapt the size of the involved matrices. However, when the periods of sources intersect, the algorithms can reverse the sources.

The frame based algorithm needs a lot of improvement. First of all, the representation we worked on with a more sophisticated representation of the Fourier Transform, as the Wigner-Ville Time Frequency distribution, can be envisaged. The Fourier Transform is attractive but suffers from limitations. Then the size of the used analysis window is fixed. Using multiple windows could help to model the transient (quick variation) part of the signals. However, if a window is optimal for a source, it is not necessary the case for the others. This idea has to be considered if we are interesting in a particular source from the mixture. The algorithm based on the minimization of the IS distance has to be finalized. The actual results, not stated in the thesis, show that a bad estimation of the LT parameters can lead to negative variances. Imposing a non-negativity constraint in the estimation can solve this problem. Also regarding the LT parameter, a time domain sparsity constraint is under investigation for the estimation of this part of the model. As it is done for the separation algorithm, the analysis window has to be considered during the derivation of the algorithms. An initialization procedure, using a Weighted Itakura Saito distance, is given in Appendix F.7 and is still under development. The procedure is done to allow a cold initialization or a robust initialization when one source (or all of them) disappears . In this weighted version, the weight focuses on a single source related to a specific period. The long-term part of the model is replaced by a frequency comb convolved with the spectrum of the window. Like this, we don't need to estimate the Long Term part and Short Term part of the model is estimated. A multipitch estimator is also proposed.

For both temporal and spectral methods, we did not use any pre-processing and the two kinds of methods suffer from the lack of information about the sources. The use of a psychoacoustic model could also be used to predict how the information is hidden by the most powerful audio components, adjacent and contiguous frequency over time. As aforementioned, the period of the source is the main feature used for the identification. If the period is wrong, the algorithm will fail the separation. We chose an un- constrained model for the long term but the use of a dictionary, as in the CELP model, can be envisaged. The number of present sources is also crucial. If it is not well estimated, the algorithms will try to find a source when there is none. As the algorithms are adap- tive and/or based on previously estimated parameters, it degrades the results and pollutes the next estimation. An Akaike criterium can be used in order to estimate the number of present sources whereas it may fail to estimate the short-term order as the input of the Short Term is not white.

Also, and as aforementioned, the algorithms are not necessary restricted to speech applications. Music applications are envisaged. In this case, several musicologic facts can be used. The model can be informed using more musical knowledge, such as tempo or rhythm. The sets of periods to find can be fixed for a certain family of instruments. The spectral shape can be restricted to be similar to predefined instruments spectral shapes etc. But in the considered work, we have made the choice not to use dictionaries, so that the algorithm is free to adapt to the data.

Part II

Annexe
Music Signal Processing

Appendix A

Instruments and Interpretation Effects: Description

In this chapter we review some instruments and their specific interpretation effects. Since the amount of possible interpretation effects is very large we focus on those we try to detect in next two chapters. Some general techniques can be performed on several instruments, but generally the tunning of their algorithms has to be changed.

A.1 Introduction

Music transcription is the process of creating a musical score (i.e. a symbolic representation, such as a MIDI file, of the music within) from an audio recording. In the traditional sense, automatic transcription implies the estimation of several features such as the pitch, position and duration of individual notes. However, music transcription is still an active field of research as shown by the large amount of publication [17, 65, 104, 129–142] and books [143, 144]. Polyphonic music transcription is generally called multiple fondamental frequency estimation or tracking and involved a part of the Music Information Retrieval (MIR) community.

But music can not be reduced to a succession of notes, and an accurate transcriptor should be able to detect other performance characteristics such as slow tempo variations or interpretations effects. The tools built for automatic transcription can also be used in a pedagogic way such as a student could want to improve his level with the help of a software. This means that the software should be able to detect some defects. For a violin it can be used for improving the use of the bow, for a wind intrument it can be the constancy of the blow etc.

The detection of ornamentation remains an open-field of research. Ornamentation techniques are utilised for giving more expression to the music by altering or embellishing small pieces of a melody. For the most part of traditional (ancient) instruments there is no general agreement in the use of specific symbols to transcribe ornamentation, where its notation and understanding has considerably varied across centuries [145]. For band instruments (guitar, bass, piano etc.) there are many different types of ornaments and for modern music it is generally instruments specific. For this study the state of the art is very poor, [146] present a method for detecting and modifying a tremolo from a monophonic recording which is not instrument specific. In [147] a plucking point position estimation (where the right hand play) system is presented for the guitar and in [148],

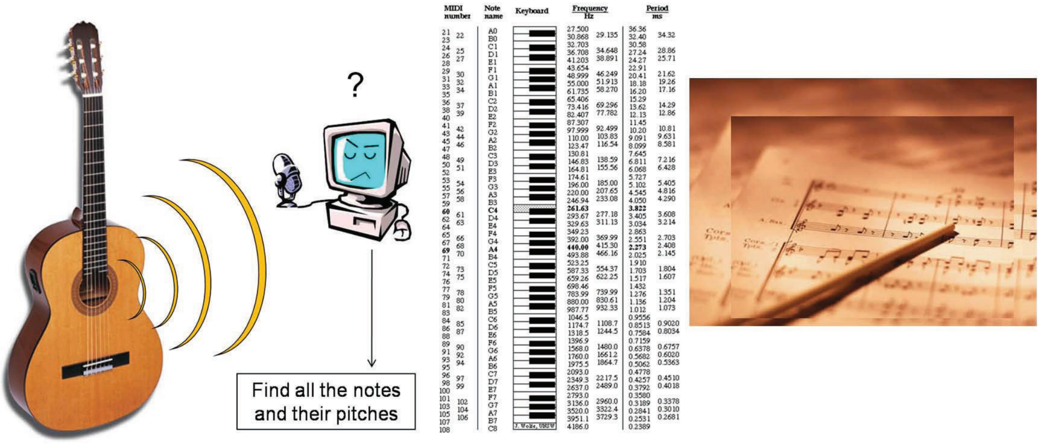


Figure A.1: *Music Transcription*

a single-note ornamentation detection system customised to the characteristics of the tin whistle is presented. A method that uses independent component analysis (ICA) to detect piano trills is presented in [149]. A more general approach to detect different types of ornamentation is presented in [150]. The model utilises an onset detector which detects very close events. A set of rules are defined and describe different types of ornamentation. The ornamentation transcription system is applied to the context of Irish traditional music.

In this thesis we focus on mono-instrumental, monophonic interpretation effects detection for a set of instruments (piano, guitar, bass guitar and violin). The criterium use instruments caracteristics to detect the presence of interpretation effects or playing defects.

A.2 Violin

The violin is a bowed string instrument with four strings usually tuned in perfect fifths. It is the smallest and highest-pitched member of the violin family of string instruments. Generally a violin is played with a bow which exerce a friction on the string but it also can be played with the finger (*pizzicato*).

A.2.1 Pizzicato

A note marked *pizz.* (abbreviation for pizzicato) in the written music is to be played by plucking the string with a finger of the right hand rather than by bowing.

A.3 Electric Bass Guitar

The electric bass guitar (also called electric bass or simply bass) is a stringed instrument played primarily with the fingers or thumb, or by using a plectrum. The bass guitar is similar in appearance and construction to an electric guitar, but with a larger body, a longer neck and scale length, and usually four strings tuned to pitches one octave lower than those of the four lower strings of a guitar (E, A, D, and G).

A.3.1 *Slap*

The slap is a very common technics in bass playing. The attack consist to hit the string with the thumb, in the begining of the fretboard, like a hammer. The resulting sound is almost completely percussive at the attack and after the sinusoidal regime appears. Note that a note play by slap have a small duration compared to a note played with the finger.

A.4 Piano

The piano is a musical instrument which is played by means of a keyboard. Widely used in Western music for solo performance, ensemble use, chamber music, and accompaniment, the piano is also very popular as an aid to composing and rehearsal. Although not portable and often expensive, the piano's versatility and ubiquity have made it one of the most familiar musical instruments. In a piano, the sound generation mechanism works as follows: when the musician presses a key, a hammer strikes the string (or actually between one and three strings, depending on the key) and this interaction triggers the note. When the key is released, a damper comes to stop the vibration of the strings and the note fades out.

A.4.1 *Forte* Pedal

The *Forte* Pedal is also called the *sustain* pedal or simply *The* pedal, since it's the most used pedal. When the sustain pedal is pressed, all the dampers of the piano are kept raised; this allows the strings to keep vibrating after the key is released, and allows strings associated to other keys to vibrate, due to sympathetic resonance, and coupling *via* the bridge. If several notes are played with the pedal, they will be mixed with a longer duration. A second effect has yet to be noticed. As a matter of fact, the two higher octaves of the piano do not have any damper, but the use of the pedal still has an influence on the sound. For this range of notes, the note does not last longer with or without the pedal, but a natural reverberation due to the resonance of the sound board appears and this sound leads to an additional floor noise.

Figure A.2: *A violin*

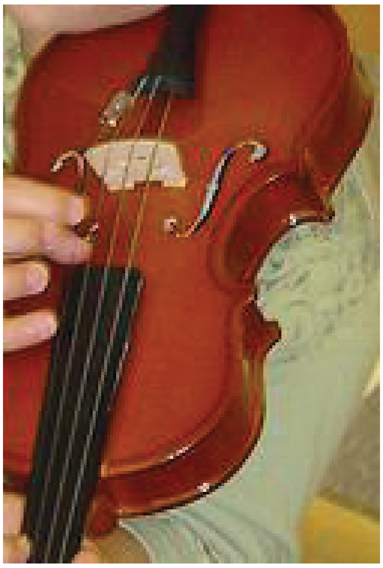


Figure A.3: *Pizzicato*

A.4.2 *Practice* Pedal

On many upright pianos, there is a middle pedal called the *practice* or celeste pedal. This drops a piece of felt between the hammers and strings, greatly muting the sounds. This is generally used for training but.

A.5 Guitar

The guitar is a musical instrument with ancient roots that was adapted readily to a wide variety of musical styles. It typically has six strings, but four-, seven-, eight-, ten-, eleven-, twelve-, thirteen- and eighteen-string guitars also exist. The size and shape of the neck and the base of the guitar also vary, producing a variety of sounds. The two main types of guitars are the electric guitar and the acoustic guitar (of which the three main types are the classical guitar, the steel-string flattop guitar, and the archtop guitar).

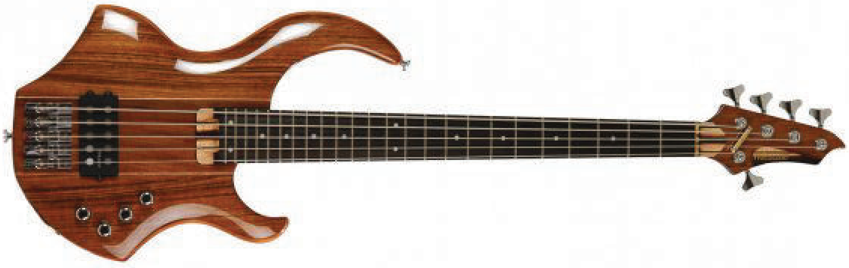


Figure A.4: *Electric Bass (5 strings)*

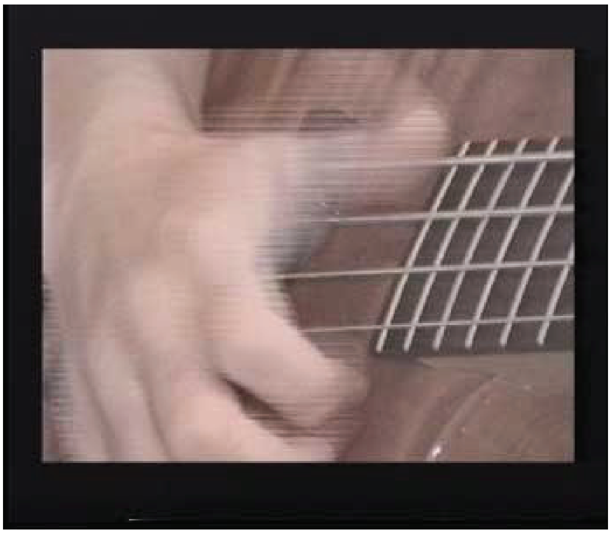


Figure A.5: *Slap example*

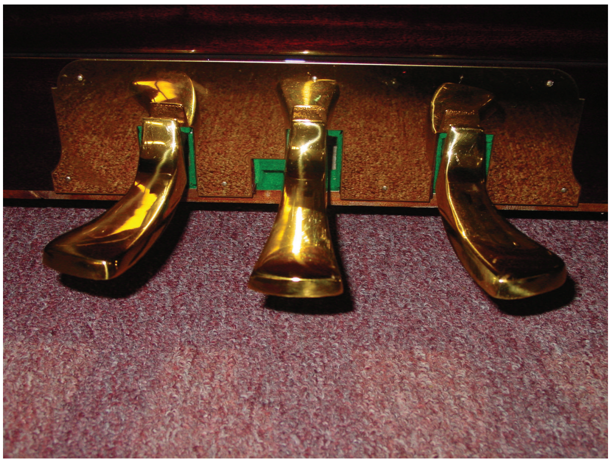


Figure A.6: *Soft (left), pratice (middle) and sustain pedals (right)*

Guitars are recognized as one of the primary instruments in flamenco, jazz, blues, country, mariachi, rock music, and many forms of pop. They can also be a solo classical instrument. Guitars may be played acoustically, where the tone is produced by vibration of the strings and modulated by the hollow body,

Figure A.7: *Electro-Classical Guitar*

A.5.1 ***Palm*** **mute**

Palm mutes are executed by placing the side of the picking hand across all of the strings and very close to the bridge before or during the attack. This produces a muted sound. While rare in classical guitar technique, palm muting is a standard technique on an electric guitar, Plam mute is more used when the musician play with a pick.

A.5.2 ***Slide***, ***Bend*** **and** ***Hammer***

There exists a lot of interpretation effects for the guitar, here we focus on three of them which have a similar caracteristic: the bend, the hammer and the slide. The bend is the action of deforming the string by pulling it up or down for increasing its length and changing the frequency. For the hammer, after a played note another finger come and strike another frette and become the new note. The slide, also called Glissando, is the action of sliding the finger to anoter frette.

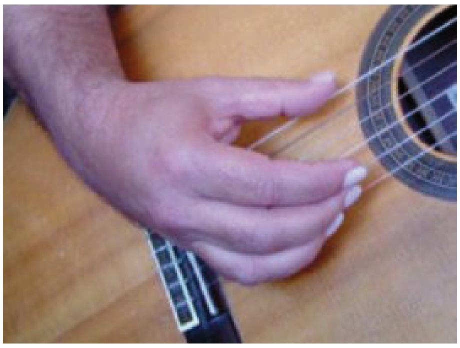


Figure A.8: *PalmMute*

Appendix B

Tools

In this chapter we present the tools used for the detection of playing defects and interpretation effects. A part of the tools (and associated results for the interpretation effects detection) were implemented on a real time simulator developed by the startup SigTone [1]. This simulator was presented during the Grand Colloque STIC 2006 (with a high background noise) in Lyon, and was ranked in the first eight best projects of the competition.

B.1 Sinusoidal modeling for *SNR* Estimation

B.1.1 Model

The estimation of the parameters of a sinusoidal signal has been dealt with extensively in the literature [54] [151], we consider the estimation of the parameters of a sinusoidal signal $s(t)$, given by :

$$s(t) = x(t) + n(t), \tag{B.1}$$

$$x(t) = \sum_{n=0}^{N-1} A_n(t) \cos(2\pi \frac{f_n(t)}{f_s} + \phi_n(t)) \tag{B.2}$$

Where $A_n(t)$, $f_n(t)$ and $\phi_n(t)$ are the amplitude, the frequency and the phase of the partial n of the signal at the time t, the sinusoidal part is defined by $x(t)$ and the noise part by $n(t)$, f_s is the sampling frequency. The noise is assuming to be zero mean additive white Gaussian noise with variance σ^2.

The musical signal which is by nature non stationary is analysed piece by piece. The synthesis method consists in estimating the parameters of each frame, generating each partial signal by using the purely sinusoidal model and then reforming the complete signal by using an overlap and add method. The noise is extracted by the subtraction of the synthesis signal to the original noisy signal.

$$\hat{x}(t) = \sum_{n=1}^{N} A_n^{est}(t) \cos(2\pi \frac{f_n^{est}(t)}{f_e} + \phi_n^{est}(t)) \tag{B.3}$$

$$n^{est}(t) = s(t) - \hat{x}(t), \tag{B.4}$$

The fact of only subtracting sinusoidal part lead to conserve into the noise all the residuals like true noise and, the quantities of interest, the other noise like instrumental noise which has received some interest in the musical domain [55, 152–154].

Fig B.1 and B.2 illustrate the overlap and add method, the total number of window depend on the overlap and can be defined as:

$$N_{Windows} = \lfloor(\frac{L}{L-ovr}\frac{N}{L})\rfloor - \lfloor(\frac{L}{L-ovr}-1)\rfloor \tag{B.5}$$

Where N is the total length of the data, L the size of the window and ovr the overlap. Note that in order to keep the sum of window equal to one we have to change its maxima.

B.1.2 Motivation, Analysis and synthesis method

The need of low complexity for real time applications often imposes the use of methods based on short term spectra, such as those obtained with the Short Time Fourier Transform (STFT) [155]. Thus, the signal is analyzed frame by frame by using a sliding window with overlap. The choice of the analysis window is very important in order to reduce the interference between adjacent FFT peaks [156]. The parameters of the analysis window have some constraints:

- First of all, we use a stationary model so the length of the window must be small for considering quasi-constant parameters inside a frame. However, this size is limited by the time-frequency incertitude.
- In the case of non-rectangular window the energy must be concentrated in the center of the frame for limited non-stationary effect.
- The window must be symmetric in order to allow a perfect reconstruction during the overlap and add method for the synthesis.
- For the spectral constraint, the signal is a sum of sinusoids and the analysis is temporally finished so the Fourier transform of this is the convolution of a sum of Dirac deltas by the Fourier transform of the analysis window. Since this has by nature an infinite length, then the FFT of the windows must have a significant attenuation from a certain frequency distance of the main lobe in order to have negligible effect on the interferences of far peaks.

The parameter estimation method must have a low complexity but a adequate precision. The maximum likelihood leads, in the case when all the parameters are unknown for a single tone, to finding the maximum of the periodogram [157], but the precision for all detected maxima k_m is given by $\hat{f}_0 = k_m \frac{f_s}{N}$, where N is the size of the data and the frequency resolution can be improved by zero padding.

In [158] it has been shown that spectral reassignment [159] derivative algorithm and phase vocoder are equivalent. The phase based method gives very good results. The last well-known method is based on interpolation [54], which has low complexity and gives reasonable result, but the bias is significant for a certain range of SNR. It is important to note that in our case the SNR can not be very high due to the presence of the instrumental noise added to the background noise in the noise part. The value is typically under $40dB$ and was determined by using the ESPRIT method [102, 103].

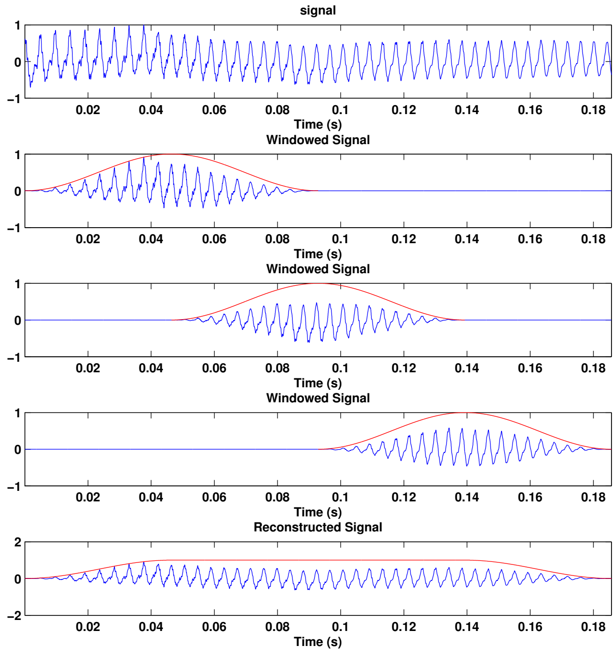


Figure B.1: *Example of the Overlap and Add Method*

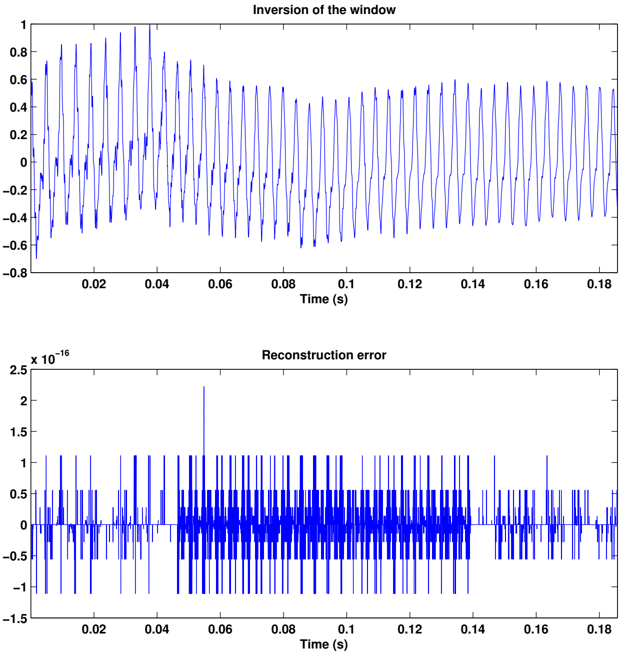


Figure B.2: *Example for the reconstruction error*

Figure B.3 shows the standard deviation of the frequency estimation error for the phase vocoder method and the parabolic interpolation for a cisoid compared to the Cramer Rao Lower bound (CRB) [157] [158] given by :

$CRB_c = \frac{6}{P^2\ N\ (N^2-1)}\ 10^{-\frac{SNR}{10}}$ where P is the amplitude of the cisoid (here $P = 1$).

As the parabolic interpolation is done on the log spectrum, the best window to use is the Gaussian window [53] and we can observe the error estimation from $40dB$. This is not the case for the phase vocoder for a single cisoid. In the case of multiple cisoids, and more generally for a real signal (which possesses negative frequency) we can observe in Figure B.5 that the two methods are limited by the interference. Note that the CRB for the real signal is $CRB_r = 2\ CRB_c$ [158].

Figure B.4 shows the estimation error of the parabolic interpolation of a cisoid which has a time varying frequency block (in each frame the frequency is constant) and for two cisoids one is fixed and one varies (assuming that the maximum position of the periodogram is known for the two). We have chosen a Hann window for the analysis and we can say that over a frequency $\Delta_f \simeq 500\ Hz$ of separation the estimation error is in the bias of the method. So, it is not necessary to clean the interference of peaks which are over this range.

Figure B.5 shows the result obtained with different methods. The frequency range is set to low frequency (under $2\ \Delta_f$) so that the tone interferes with itself (due to the negative frequency), for the PV and the PI. The first two lines are without correction and the others are with correction. We consider four combinations: estimate the parameter with the two methods for making the correction of the interferer and for estimating the peak of interest. We can conclude that the correction is necessary but, in the range of SNR we considered the methods are similar. The length of the window is fixed to $L = 1024$ samples for high frequency instruments like violin or guitar and to $L = 2048$ samples for low frequency instruments like piano or bass guitar. The sampling frequency is set to $f_s = 44100\ Hz$ so the duration of the signal is about 23.22 ms for $L = 1024$. We use a zero padding factor of four and the overlap is set to 50% for $L = 1024$. Another parameter to choose is the number of frequency peaks to search in each frame; we have taken it equal to $Nb = 32$.

B.1.3 Estimation - Interpolation - Amplitudes estimation

The first task is to find the Nb principle peaks in the spectrum. Taking a fixed value for peaks has some drawbacks: in the case of pure noise, the sinusoidal signal part will be estimated by the noise's dominant peaks and the resulting estimate SNR will be bounded at a lower value. Equivalently, for a rich spectrum there will be some harmonics in the noise. After we have found the peaks, we choose one and we calculate the interference of the peaks include into the ± Δ_f interval. For each peak of this interval, we estimate its frequency and amplitude by parabolic interpolation and then we calculate its phase by linear interpolation; as mentioned by [53], parabolic and linear interpolation leads to the same result. When the interference is cleaned from the peak of interest, we interpolate its parameters. For the parabolic interpolation we use:

$$Y_{m'} = S_{dB}(f_m + m'), \qquad m' = -1, 0, 1 \tag{B.6}$$

Where $S_{dB}(f) = 20\ log_{10}(|X(f)|)$, and $X(f)$ is the Fourier Transform of $x(t)$.

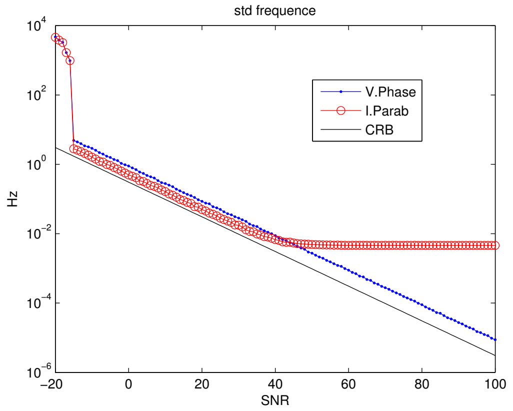


Figure B.3: *Standard deviation of the frequency error for the phase vocoder method and parabolic interpolation for a cisoid*

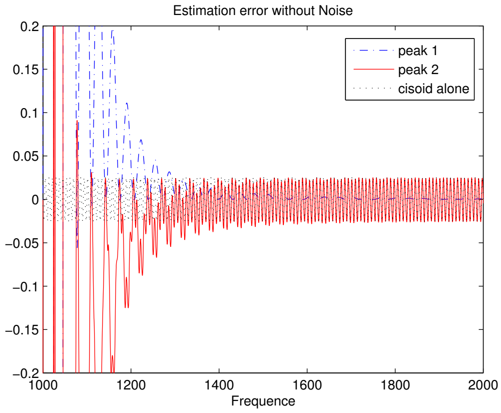


Figure B.4: *Estimation error of the parabolic interpolation, for one and two cisoids*

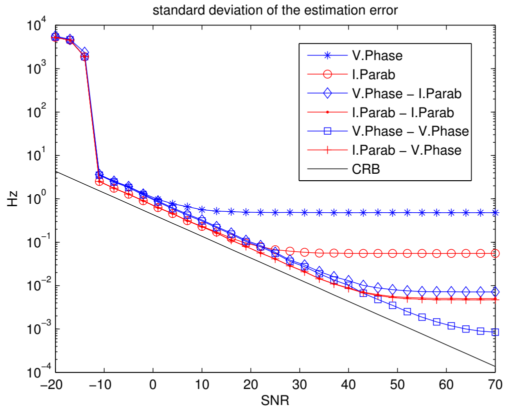


Figure B.5: *Estimation error of the parabolic interpolation and for the phase vocoder with the cleaning of the other peaks on the periodogramme*

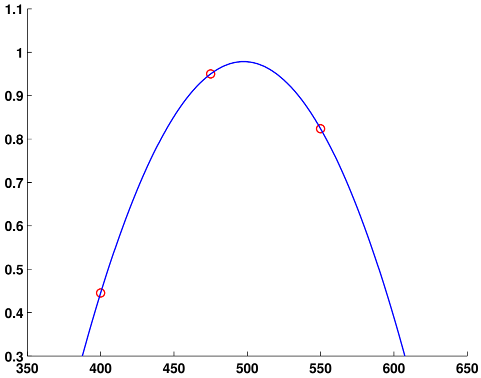


Figure B.6: *Example of a parabolic interpolation, The circles correspond to samples of the spectrum*

The estimated frequency is given by

$$f_m^{est} = f_m + \frac{1}{2}\ \frac{Y_{+1} - Y_{-1}}{Y_{-1} + Y_{+1} - 2Y_0} \tag{B.7}$$

and the corresponding amplitude, as in Figure B.6, by

$$S_{dB}^{est} = Y_0 - \frac{f_m^{est}}{4}\ (Y_{-1} - Y_{+1}),\ \ A_m^{est} = 10^{\frac{1}{20}\ S_{dB}^{est}} \tag{B.8}$$

For estimating the interference of the nearests peaks we have to obtain an expression of the pertubation due to their presence in the spectrum on the peak of interest. In our case we use a Hann window of size L given by :

$$w(n) = 0.5 - 0.5\ \cos(2\pi \frac{n}{L}), \qquad 0 \leq n < L \tag{B.9}$$

The Hann window is temporally finished, so we express it with the rectangular window :

$$r(n) = \begin{cases} 1 & ,0 \leq n < L \\ 0 & ,otherwise \end{cases} \tag{B.10}$$

we can rewrite :

$$\begin{aligned} w(n) &= [0.5 - 0.5\ \cos(2\pi \frac{n}{L})]\ r(n) \qquad \text{(B.11)} \\ &= 0.5\ r(n) - 0.25\ e^{2i\pi \frac{n}{L}}\ r(n) - 0.25\ e^{-2i\pi \frac{n}{L}}\ r(n) \end{aligned}$$

The DFT of the rectangular function is :

$$R(f) = \sum_{t=0}^{L-1} (e^{-2i\pi f t}) = e^{-i\pi f (L-1)}\ \frac{sin(\pi f L)}{sin(\pi f)} \tag{B.12}$$

So for the Hann window we obtain :

$$W(f) = 0.5\ R(f) - 0.25\ R(f - \frac{1}{L}) - 0.25\ R(f + \frac{1}{L}) \tag{B.13}$$

After the estimation of the parameters of each peaks we subtract their contribution given by :

$$\begin{aligned} W_m^{est}(f) &= \sum_{\substack{n=1 \\ n\neq m}}^{Nb_{\in\Delta f}} A_n^{est} W(f - f_n^{est}) e^{i\phi_n^{est}} + \sum_{n=1}^{Nb_{\in\Delta f}} A_n^{est} W(f + f_n^{est}) e^{i\phi_n^{est}} \\ f &= [f_m - 1, f_m, f_m + 1] \end{aligned} \tag{B.14}$$

Where A_n^{est},f_n^{est} and ϕ_n^{est} are the estimates parameters and f_m the frequency corresponding to a maximum of the periodogramme. The second term is only of use in the case of low frequency.
And ϕ_n is defined by :

$$\phi_n^{est} = \phi_{\lfloor f_n^{est} \rfloor} + (f_n^{est} - f_n)\,(\phi_{\lceil f_n^{est} \rceil} - \phi_{\lfloor f_n^{est} \rfloor}) \tag{B.15}$$

When the contributions are subtracted we interpolate the value of the parameters on the peak of interest.

B.1.4 Synthesis, Noise extraction and SNR estimation

For the synthesis signal $\hat{x}(t)$ we use the parameter estimated before and we create partial signal with the model. The total reconstruction of the signal is made by using the overlap and add method and by using the same window as for the analysis. Then we subtract the noise estimates to the original signal and we compute the SNR, given by :

$$SNR = 10\ log_{10}\left(\frac{\sum_t \hat{x}(t)^2}{\sum_t\ (x(t) - \hat{x}(t))^2}\right) \tag{B.16}$$

B.1.5 Discussion on the method

In the case of monophonic recording, this method gives very good results. But we have to mention that the overlapping of the partials due to different sources is not taken into acount. We can interpret the result of the analysis as follow: When we analyse the spectrum we subtract an energetic contribution and not necessary a contribution from one source. On the other hand, the size of the window is fixed so the system is not multiresolution, most of the estimated noise is due to the transient part of the signals which need a smaller window [160].

B.2 Onset Detection

Music onset detection plays an essential role in music signal processing and has a wide range of applications such as music transcription, beat-tracking, and tempo identification. Different sound sources (instruments) have different types of onsets that are often classified as soft or hard. Hard onsets are characterized by sudden increases in energy, whereas soft onsets show more gradual changes. Hard onsets can be well detected by energy-based approaches, but the detection of soft onsets remains a challenging problem. Let us suppose that a note consists of a transient, followed by a steady-state part, and the onset of the

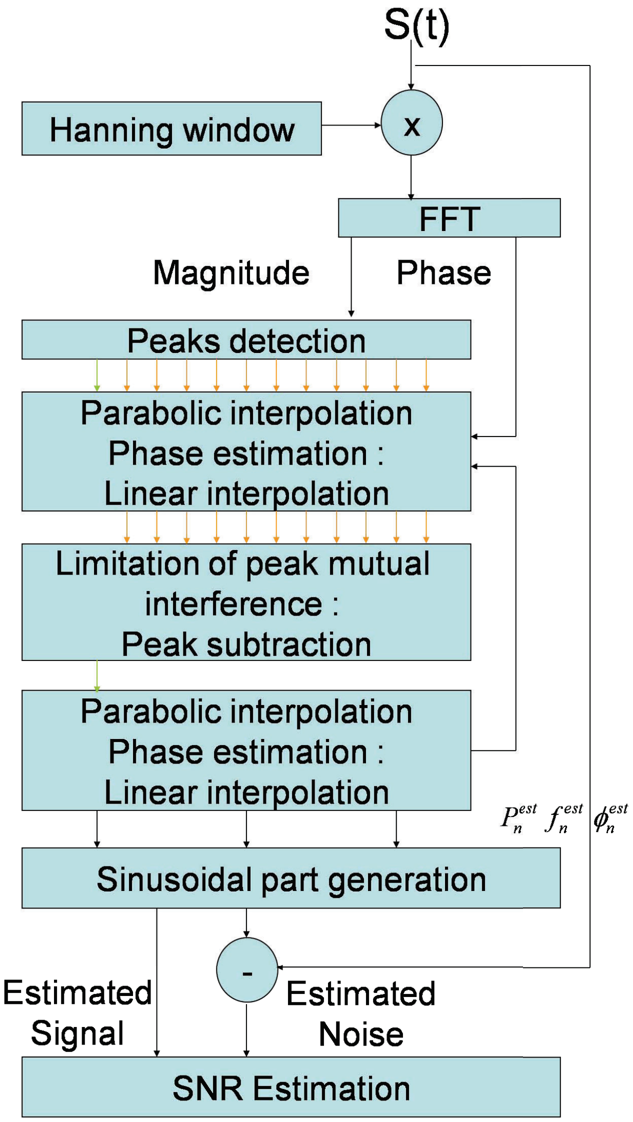

S(t)
Hanning window
x
FFT
Magnitude
Phase
Peaks detection
Parabolic interpolation
Phase estimation :
Linear interpolation
Limitation of peak mutual
interference :
Peak subtraction
Parabolic interpolation
Phase estimation :
Linear interpolation
P_n^{est} f_n^{est} ϕ_n^{est}
Sinusoidal part generation
-
Estimated
Signal
Estimated
Noise
SNR Estimation

Figure B.7: *General Diagram for the Harmonic+Noise Decomposition*

note is at the beginning of the transient. For hard onsets, usually, energy changes are significantly larger in the transients than in the steady-state parts. Conversely, when considering the case of soft onsets, energy changes in the transients and the steady-state parts are comparable, and they do not constitute reliable cues for onset detection anymore. Consequently, energy-based approaches fail to correctly detect soft onsets. This fact can be used to develop appropriate pitch-based methods that yield better performances, for the detection of soft onsets, than energy-based methods. Here we choose an approach based on a *Harmonic+Noise* Decomposition similar to [103]. The method can be resumed as follow:

- First of all, the signal passes through an uniform bank filter and is decimated.
- On each subband we use the algorithm described in the previous section.
- For all the synthesis (harmonic) and stochastic (Noise) component we perform a Spectral Flux.
- Then, on the detection function we search all the peaks and merge them for the noise and the synthesis separetely

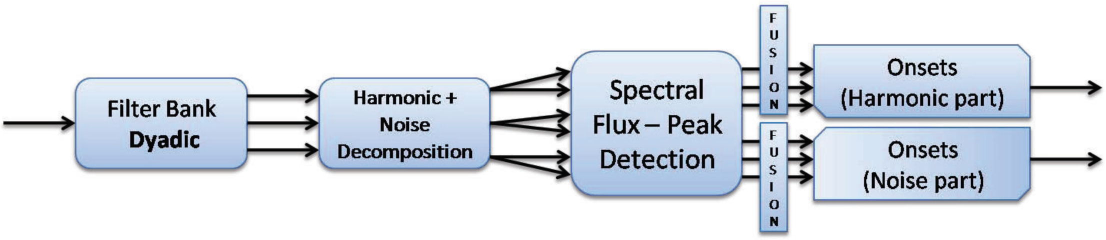


Figure B.8: *General Diagram for the Onsets Detection*

B.2.1 Example of onsets detection

Figure B.9 illustrates the onsets detection function; the song is composed of a series of *Hit Hat* songs. As we can see, the detection function gives a good result. The onsets are found by pick picking. Note that due to the *STFT* analysis the detection function or the found onsets has to be shifted and time scaled.

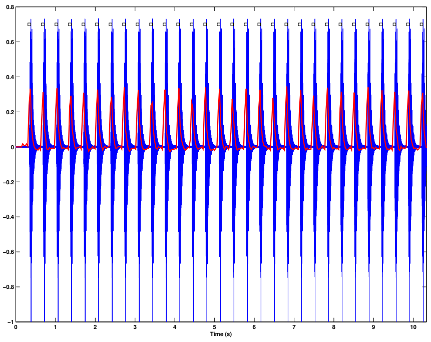


Figure B.9: *Example of detection function*

B.3 Pitch estimation

Pitch estimation algorithms (PEAs) have numerous applications in speech and music processing. Three steps are taken by these PEAs to estimate the pitch:

- The spectrum is estimated using a short-time Fourier transform (STFT).
- A set of pitch candidates is selected and a score is computed for each pitch candidates by computing an integral transform over a function of the spectrum.
- The pitch candidates with the largest score is selected as the estimated pitch.

The first algorithm to be presented is the Harmonic Product Spectrum (HPS) [161]. This algorithm estimates the pitch as the frequency that maximizes the product of the spectrum at n harmonics (i.e., multiples) of that frequency, or equivalently, as the frequency that maximizes the sum of the logarithm of the spectrum at n harmonics. Using an integral transform this can be written as

$$p = \arg\max_f \int_0^\infty \log\left|X(f')\right| \sum_{k=1}^{n} \delta(f' - kf)df' \tag{B.17}$$

where X is the spectrum of the signal and p is the estimated pitches. One problem with this algorithm is that if any of the harmonics is missing, the integral will be minus infinity for the candidate corresponding to the pitch. An algorithm that does not have problems with missing harmonics is Sub-Harmonic Summation (SHS) [128]. Instead of multiplying the spectrum, SHS adds it. This algorithm also introduces a decaying weighting factor that gives more emphasis to low order than high order harmonics,

$$p = \arg\max_f \int_0^\infty \left|X(f')\right| \sum_{k=1}^{n} c^{k-1}\delta(f' - kf)df' \tag{B.18}$$

The purpose of the weighting factor ($c > 0$ and $c < 1$)is to avoid subharmonics of the pitch having the same score as the pitch. A problem with the algorithms presented so far is that they suffer from a kind of blindness. That is, for each pitch candidates, they look at the spectrum only at its harmonics, ignoring the contents of the spectrum everywhere else. An example will illustrate why this is a problem. Suppose a signal has a flat spectrum. This signal is perceived by the ear as having no pitch. However, these algorithms will force each pitch candidate to have the same score, and therefore each candidate is a valid estimate for the pitch. This blindness problem is partially solved by an algorithm called Subharmonic to Harmonic Ratio (SHR) [162]; however, since it uses the logarithm of the spectrum, it has the problem shown for HPS. SHR adds the logarithm of the spectrum at harmonics of the pitch candidate but also subtracts it at the middle points in between, i.e.,

$$p = \arg\max_f \int_0^\infty \log\left|X(f')\right| \sum_{k=1}^{n} \delta(f' - kf) - \delta(f' - (k - 1/2)f)df' \tag{B.19}$$

Unlike previous algorithms, this algorithm produces a different score for the pitch of a pulse train than for a flat spectrum. However, this algorithm still suffers from blindness: it is not able to recognize the pitch of inharmonic signals.

For our mono-pitch estimation method we use the three method and we take the median of the three estimation.

B.4 Fundamental to Harmonics Energy Ratio (FHER)

Another tool that we use is a variation of the *Odd to Even Harmonic Energy Ratio* [163]. Instead of analyzing the odd to even energy ratio we compare the energy of the harmonics to the fundamental. We can do this by different way: the first one needs to find with precision all the harmonics, for a given note. We first search the fundamental frequency, then we search the harmonics one by one and give a tolerance to the theoritical position of the harmonic for finding the peaks maxima. For limiting the influence of the noise floor we use as cues the information related to the sub-harmonics (between two harmonics). We can express the criterium as follows:

$$FHER \;=\; \frac{a(1)}{\sum_{k=2}^{K}(a(k) - a(k-\frac{1}{2}))} \tag{B.20}$$

For the second method, if we know the fondamental frequency we can compute the spectrum with a number of points equal to twice the period. Like this, half of the points of the spectrum correspond to the harmonics and the other to the sub-harmonics. Note that for this we need a signal with low inharmonicity.

B.5 Slow variation fondamental frequency tracking

For tracking the evolution of the fondamental frequency (and its amplitude) we have developed a slightly modified version of the *Pisarenko* method. The traditional Pisarenko method assumes that a signal, x, consists of n complex exponentials in the presence of white noise. Since the number of complex exponentials must be known a priori, it is somewhat limited in its usefulness. In our case we assume that the fondamental frequency is known, so we fix the number of sinusoides to one, because we isolate it by modulation and low pass filtering. We construct the correlation matrix for two consecutive sample as

$$R = A_{n,n+1}\; A_{n,n+1}^{H} \tag{B.21}$$

where H denote the hermitian and A is the low pass filtered demodulated signal, we use a singular value decomposition of

$$R \;=\; UDU^{T} \tag{B.22}$$

with T the transpose operator, we apply a Vandermonde vector to the eigenvector

$$[U1\; U2]^{T}[1\; z^{-1}] = U1 \;+\; U2\; z^{-1} \;=\; 0 \tag{B.23}$$

we found the pole z and we estimate the instantaneous frequency. The amplitude of the fondamental frequency is computed by applying a half-wave rectification (for example [143]) and a low pass filtering.

B.5.1 Example

As an example, we use the modified pisarenko method on a single violin note. Figure B.10 shows the violin signal, the estimated fondamental frequency evolution and its amplitude. Note that we don't directly found the correct frequency, we found the variation of the frequency around the estimation used for the demodulation. In the plot the frequency used for the demodulation is added to the value.

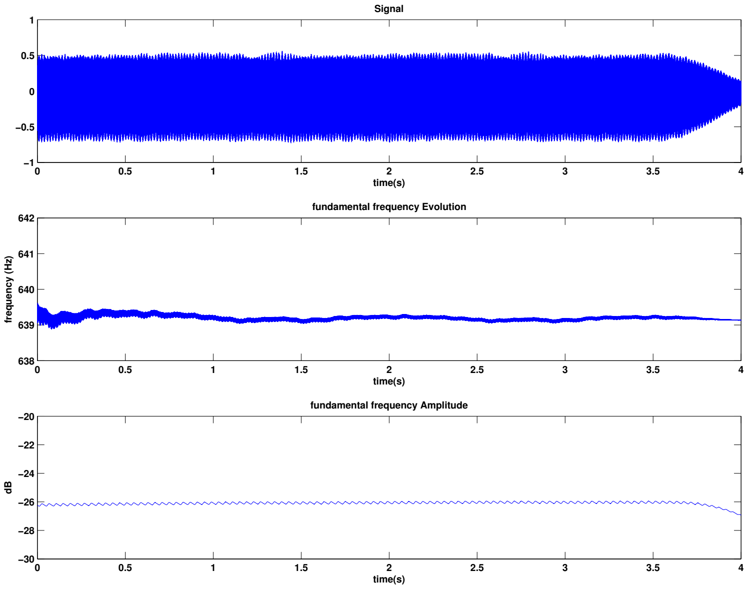


Figure B.10: *Exemple of Modified Pisarenko Method on a Violin Note*

Appendix C

Interpretation Effects and Playing Defects: Detection

C.1 Violin Playing defects detection

When a violinist plays, he moves the bow upon the strings and the sound is generated by the friction of this movement. A well played note is constrained by (at least) three parameters:

- The speed of the displacement of the bow.
- The pressure exerted on the string.
- The two orientations of the bow (on itself and on the string).

A good player exercises constant speed and pressure, keeps the bow completely parallel on the string and the displacement orthogonal to the string. When one or more of these constraints are not respected, the sound becomes more noisy. In the worst case, we only heard the displacement of the bow.

Figure C.1 show a succession of notes played by a student. The black part corresponds to the detected note, the line corresponds to the relative SNR and the SNR is given by this value. The SNR is a quantity which is independent of the volume and we can observe the difference in the estimation of each note. We can thus qualify a good note and a bad note. So, here, a good note has an SNR larger than 18 dB and a very bad note has an SNR smaller than 8 dB. In practice, all the thresholds have to be adjusted in accordance with the background noise.

C.1.1 Orientation defects

For illustrating the default introduced by an orientation default, we have made ten records by position, three of them ($[0, 15, 30]$ degree) are shown on FigureC.2, the six other records are done on the other orientation. The duration of the note are short, less than 1 s. The result of the $SyNR$ is given on TableC.1.

It appears clear that the angle default leads to a decrease of the SNR, so it can be detected.

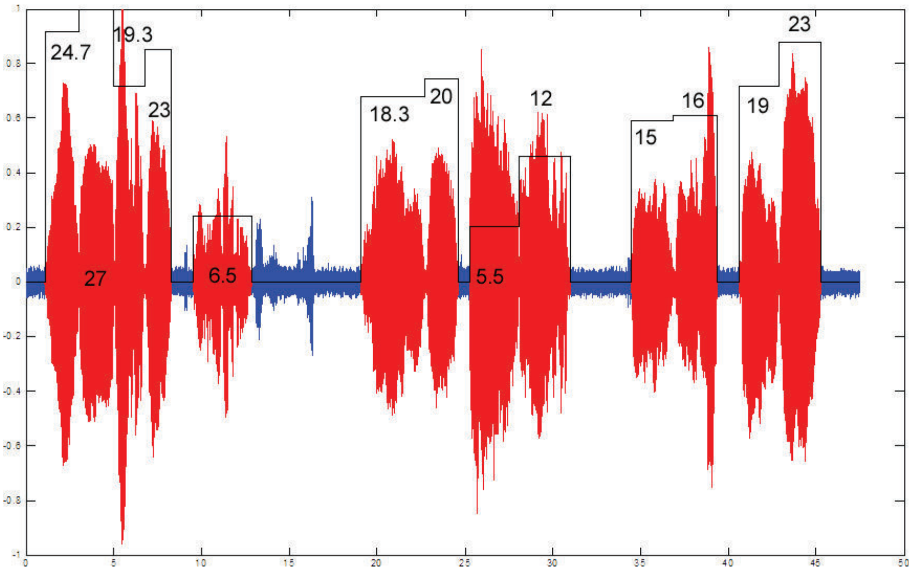


Figure C.1: *SNR Estimation for a Violin piece played by a student*

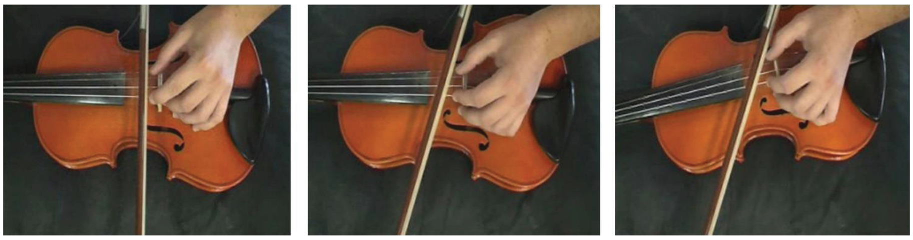


Figure C.2: *Orientation of the Bow*

Note	1	2	3	4	5	6
Angle (degree)	0	-15	15	-30	30	-45
SyNR exp.0	**13.5**	12.9	10.0	8.9	6.2	3.7
SyNR exp.1	**12.8**	8.6	12.5	7.9	7.1	4.2
SyNR exp.2	**12.9**	12.1	9.2	6.4	5.9	5.5
SyNR exp.3	**13.2**	9.4	12.6	8.1	6.8	5.2
SyNR exp.4	9.9	7.1	**10.7**	6.9	7.5	3.9
SyNR exp.5	**11.7**	11.2	8.8	7.2	6.8	4.2
SyNR exp.6	10.3	**10.8**	7.6	7.8	5.7	4.1
SyNR exp.7	11.5	**12.1**	8.4	6.8	7	3.3
SyNR exp.8	10.1	10.1	**10.9**	9.1	8.8	3.6
SyNR exp.9	**12.3**	11.6	8.4	8.2	7.9	3.6
Mean exp.	11.8	10.6	9.9	7.7	7	4.1

Table C.1: Resulting *SyNR* for the angle default

C.2 Instruments Interpretation Effects Detection

C.2.1 *Slap* Detection

The slap is a very common technique in bass playing. The strike consists of hitting the strings with the thumb, in the beginning of the fretboard, like a hammer. The resulting sound is almost completely percussive upon the strike, and afterwards the sinusoidal regime appears. Note that a note played by slap has a smaller duration compared to a note played with the finger. Figure C.3 shows the result of the SNR estimation on a bass sequence composed of two single notes. The first note is played with the finger (sweet) and the second is played by slap (percussive). As we expected, the SNR of the slap note is small compare to the note played with the finger.

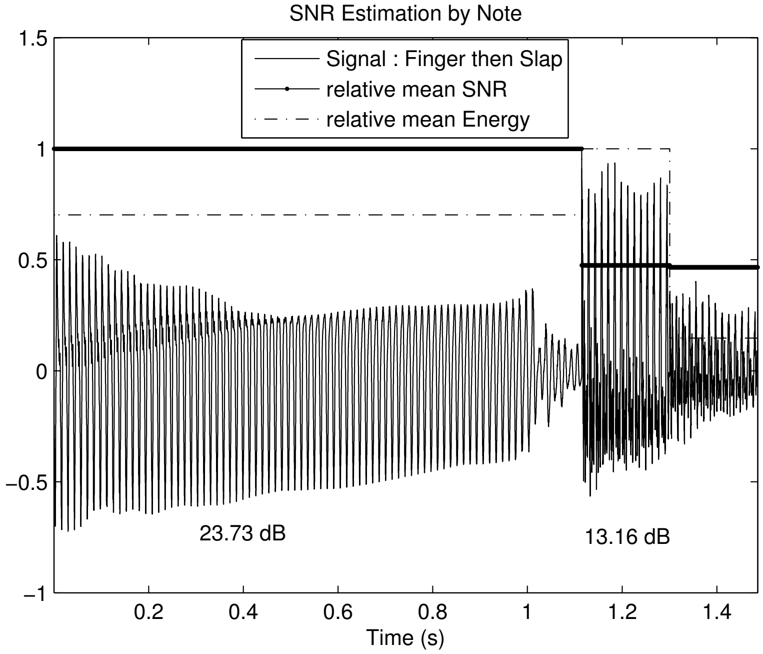


Figure C.3: *Estimation of the SNR for bass sequence of two notes, the first is play with the finger and the second is play by slap.*

In a polyphonic (mono-instrument) song, the criterium is still valid if the slap is played without the resonance of a note. In Figure C.4 the song contains a serie of chords and the associated estimation of the SNR and in Figure C.5 played using Slap. In this case first the Harmonic+Noise Decomposition is done. Then for the SNR estimation we analyse the synthesis and the residual signals with a sliding window: a hann window of duration $\approx$ 20 ms with an overlap of 50%. We can still observe that the SNR is (in mean) higher for the finger case, the minimum value is 10 dB.

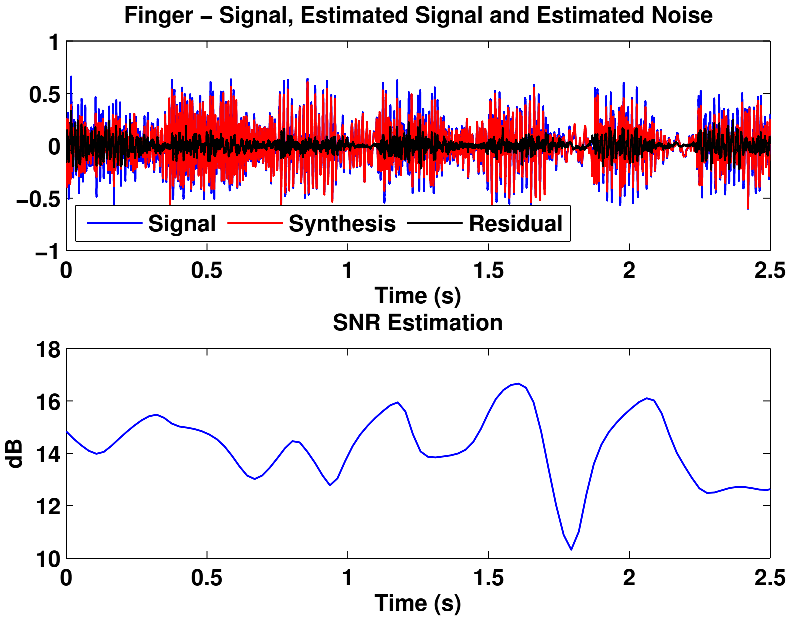


Figure C.4: *Bass played with Finger and SNR Estimation.*

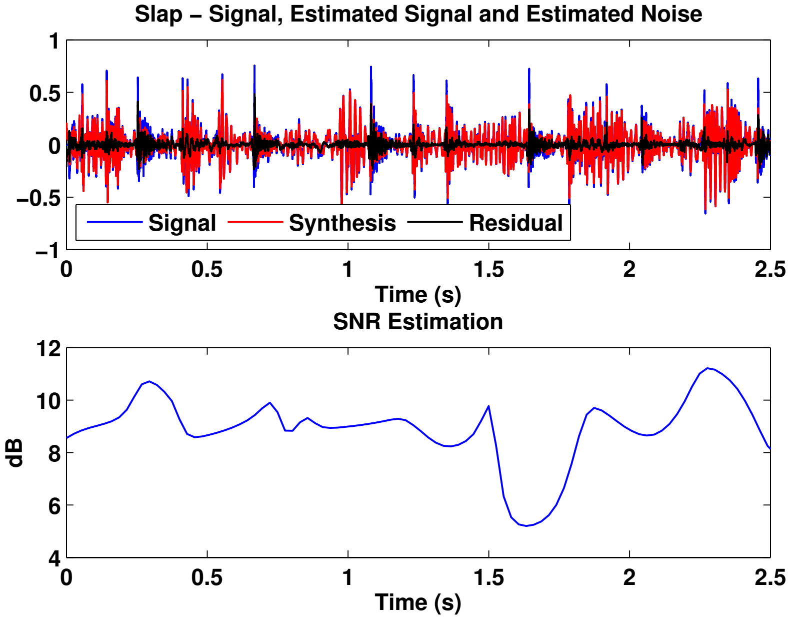


Figure C.5: *Slap Bass and SNR Estimation.*

C.2.2 *Forte* Pedal Detection

As said before, when a key is pressed the sound is generated by the striking of a hammer on the strings of the piano, the sound dies when a damper stop the vibration of the string after the release of the key. When the sustain pedal is pressed, all the dampers of the piano are kept raised; this allows the strings to keep vibrating after the key is released, and allows strings associated to other keys to vibrate, due to sympathetic resonance, and coupling *via* the bridge. If several notes are played with the pedal, they will be mixed with a longer duration. A second effect has yet to be noticed. As a matter of fact, the two higher octaves of the piano do not have any damper, but the use of the pedal still has an influence on the sound. For this range of notes, the note does not last longer with or without the pedal, but a natural reverberation due to the resonance of the sound board appears and this sound leads to an additional floor noise [164].

Similar observations can be found in previous work. [165] proposes a polyphonic piano transcription system which detects and takes into account the use of the pedal. The detection of the pedal is based on an estimation of the noise floor. It is estimated as the mean value of the Discrete Fourier Transform (DFT) magnitude over the analysis frame, but only on frequency bins considered as “not active” in the frame (not associated with an actually played note - these frequencies are determined by a varying threshold). Another modelling of the sustain pedal can be found in [166,167]. Through the analysis of middle-range piano notes, played *legato* with and without the pedal, the authors point out three features that should be able to discriminate between notes played with and without the pedal, and be useful for piano synthesis: noise floor, decay time of the partials and amplitude beating.

C.2.2.1 Database

Special recording was done in order to study the effect of the sustain pedal. Two identical microphones (omnidirectional electrostatic Shoeps) were placed on the right side of a grand piano (grand piano Yamaha C1) at one meter from the sound board and the sound was digitalized at a sampling rate of $44.1KHz$ and encoded with 16 bits, through an Edirol UA5 soundcard. This configuration was chosen in order to gather a maximum of the resonance generated by the sound board when the pedal is pressed.

We initially considered that the actual gesture of the musician could have an importance, and we decided to distinguish between notes played *staccato* (short strike on the key) and played *legato* (the key is kept pressed on). For the *staccato*, since the strike is short, the damper takes only very little time to go down, whereas we have the opposite for notes played *legato*. The database is thus composed of four categories of notes: *staccato* without pedal (*staccato* in the following), *staccato* with pedal (*staccato+ped.*), *legato* without pedal (*legato*) and with pedal (*legato+ped.*), some corresponding waveforms are illustrated in Figure C.6. We recorded single tones from low to high frequency range of the piano, with and without the use of the sustain pedal. The interval between each note is a fourth (C to F for example) and each note was played in the four configurations previously described. It lead to a database of all in all 200 note recordings.

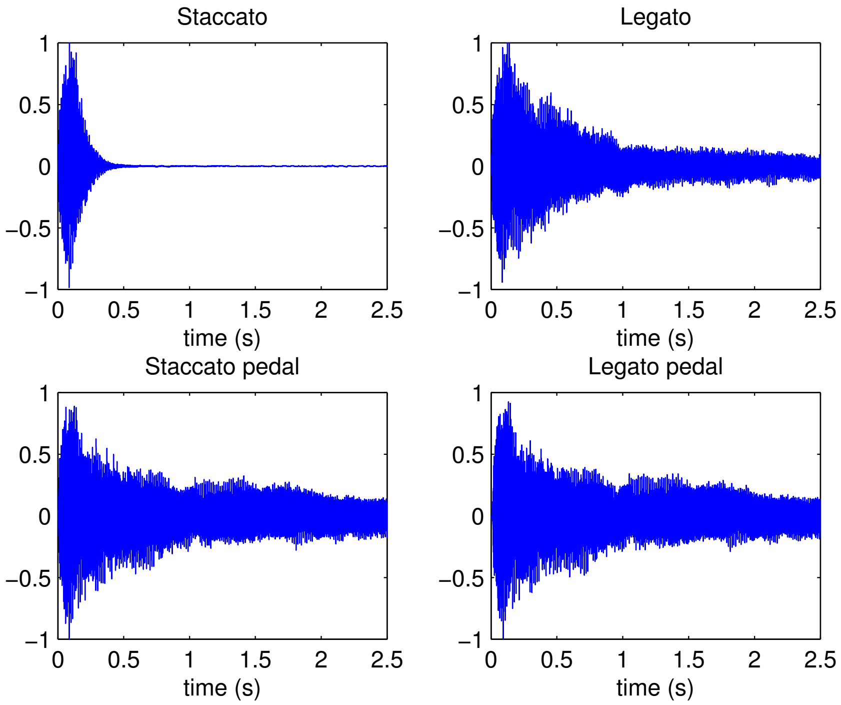


Figure C.6: Examples of waveforms of *staccato*, *legato*, *staccato+ped.* and *legato+ped.*, note D_2

C.2.2.2 Feature extraction

As we want to study two features, one being specifically linked with the sinusoidal part of the note (decay time of the partials) and the other concerning the noise (noise floor power), it seems natural to perform an "harmonic plus noise" decomposition [11] before feature extraction itself.

C.2.2.3 Pre-processing

Since for real recordings the background noise is seldom white a preprocessing step is applied. The original spectrum is whitened by means of autoregressive modeling (AR) of the background power spectral density. In order to increase the number of points in the spectrum we use zero padding. The background spectrum is obtained by median filtering and inversed by a Finite Impulse Response (FIR) filtering, at the end this operation is compensated by AR-Filtering. However the purpose is to study piano tones. The very large range of frequencies covered by the piano (*88* notes from *27.5Hz* to *4186Hz*) makes it difficult to have the same efficiency for all the notes. In order to increase the resolution each studied note is slightly decimated according to its range of frequencies. Since we are in a monophonic case we have used a correlation method for this purpose.

C.2.2.4 Harmonics amplitudes tracking

For the study of the envelopes of the partials we have used Fast Sequential LS Estimation [168]. This method is an adaptive algorithm used for the estimation of slowly varying amplitudes. It assumes the frequencies are known in advance and gives a continuous evolution of each partial. It takes into account the sinusoidal nature of the data and because it uses a rotational invariance technique, has a low complexity. First of all, the preprocessing procedure explained above is applied to the signal. Then in the whitened magnitude spectrum, we find the frequencies by searching for local maxima (peak picking). Finally we use them as inputs for Least Square Estimation. Figure C.7 shows an example of the evolutions of the first three harmonics for a note played with and without the sustain pedal. The Pedal has a dominating effect on the envelopes. During the attack the evolution is the same for the two cases but after $100ms$ the behavior changes. For this note the decay and release times change and a beating appears on the first harmonic.

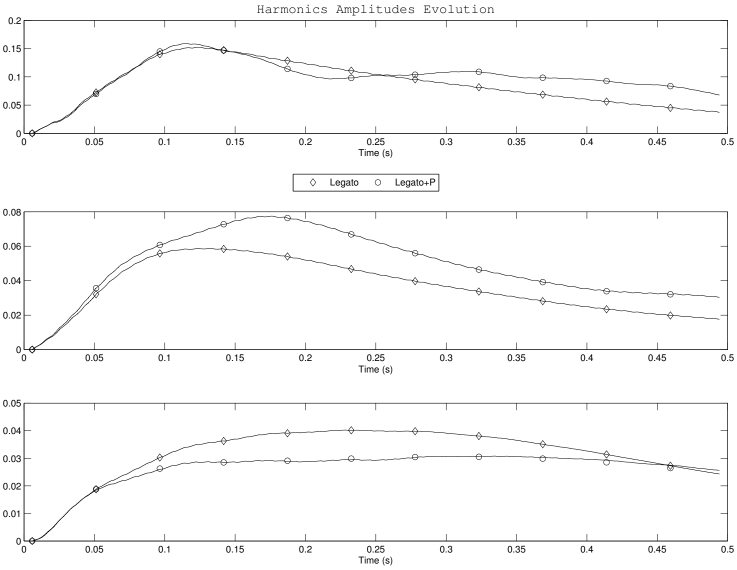


Figure C.7: Evolution of the amplitudes of the first three harmonics

C.2.2.5 Noise estimation

Figure C.8 shows the spectra of the two kind of playing and of the noise spectra with and without the sustain pedal. On the noise spectra, we can see the resonance of the sound board.

As it seems impossible to detect the resonance of the pedal before the end of the attack, we start the study at $200ms$ after the begining of the note. The duration studied is also $200ms$, the signal is then normalized in energy. As the presence of the noise of the Pedal is constrained to the low frequencies, we first decimate the signal by a factor 20 then we get the noise by the harmonic plus noise decomposition. We model the noise as an autoregressive process (AR) of the first order to obtain the shape of each noise spectrum.

Figure C.9 shows the result of the AR modelling of noise obtained. White lines separate the Pedal cases from the non Pedal cases. It appears that the AR has a flatter shape for

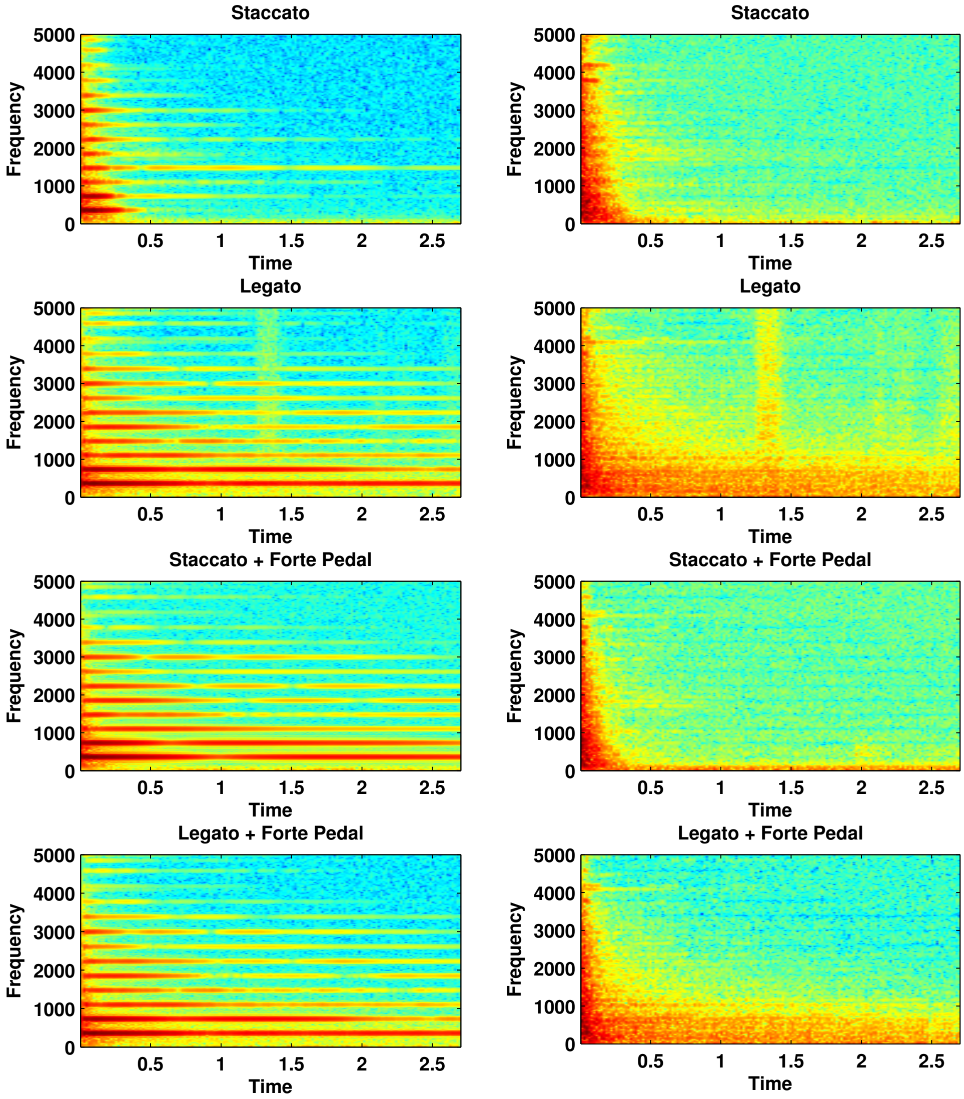


Figure C.8: Harmonic+Noise Decomposition for *staccato*, *legato*, *staccato+ped.* and *legato+ped.*

the pedal with a slightly lower power in the low frequency range. The bottom of Figure C.9 shows the total power in each AR spectrum. Using 30 measurement data we have trained a threshold that separates the two cases. The power of each AR of the training data was computed and the result shows a separation between the two cases. We applied the same threshold to the other data and put the results on the same figure with a point for the notes estimated to be played with the sustain pedal. We find that:

- 3 out of 85 pedal noises are interpreted as non pedal, around 96.5 percent.
- 21 out of 85 non pedal noises are interpreted as pedal, around 75 percent.

So a total error rate of 15 percent is obtained. In spite of the simplicity of the method used we achieve a good detection rate. Note that the results may be highly dependent on the harmonic plus noise decomposition used.

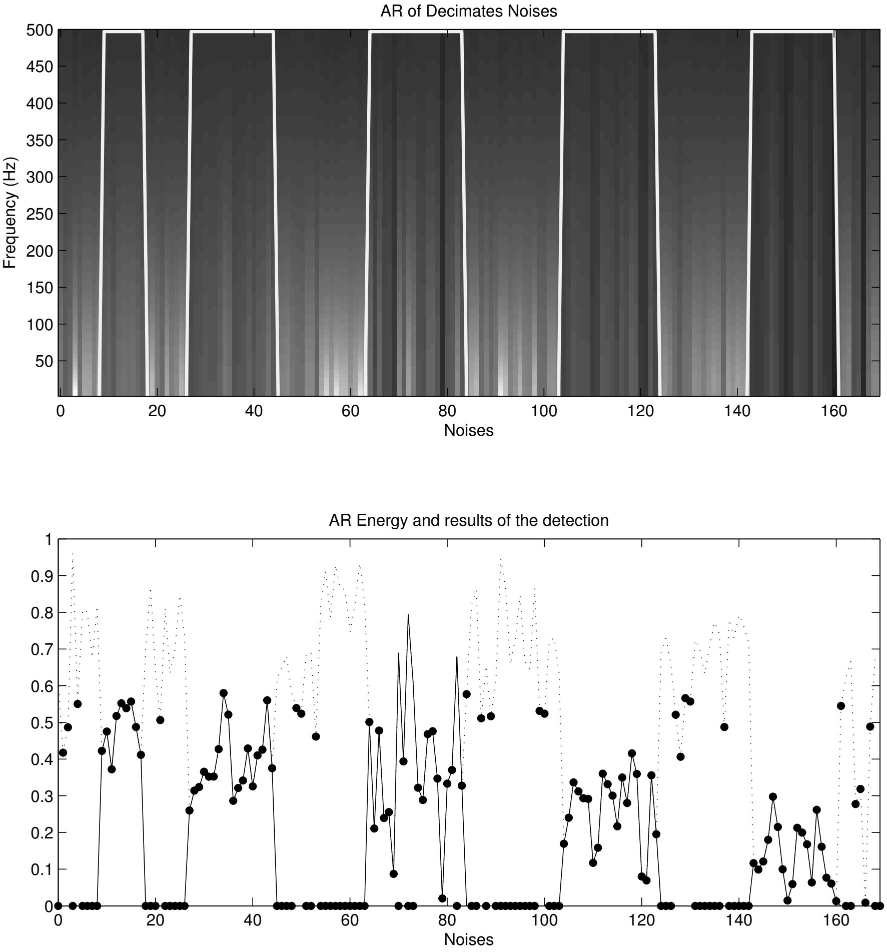


Figure C.9: Top : Autoregressive modeling of the Noise for 170 note recordings. A white line on top indicates notes with Pedal. Bottom : power of the AR model : For notes with Pedal (solid line) and notes without Pedal (dashed line). The dots indicate the notes that are estimated to be notes with Pedal.

C.2.3 *Palm* mute, *Pizzicato* and *Pratice* Pedal detection

C.2.3.1 Description

On bowed string instruments (violin, cello etc.) it's a method of playing which consist on plucking the strings with the fingers, rather than using the bow. The sound produce is very different, short and percussive rather than sustained. On the guitar, it's associated to a kind of plucking, which reach the sound of a pizzicato on a bowed string instrument. For the guitar, pizzicato is often called Palm mute and it's done differently. Palm mutes are executed by placing the side of the picking hand across all of the strings and very close to the bridge before or during the attack. This produces a muted sound. While rare in classical guitar technique, palm muting is a standard technique on an electric guitar, Plam mute is more used when the musician play with a pick. For more details, the hand operates a low pass filtering (as for the damper pedal of the piano). Figure C.10 illustrates this effect. The presented results show an attenuation of the power of the harmonics. Here we present some results for the guitar but the method can also be applied to instruments which can use a mute style like piano, bass guitar (rarely used) or obviously violin.

C.2.3.2 *Palm* mute detection result

Figure C.10 shows the short time Fourier transform of a piece of song played normaly and then played using palm mute. It is easy to see the low pass filtering operated by the hand. In Music Information Retrieval community, this effect is known as the Spectral Roll-Off (see App. F.9), just searching a features like the Roll-Off will not permit to detects the palm mute in a polyphonic context. For the analysis we use the FHER (described in section B.4) which will works in a polyphonic context (as far as the good fondamental frequencies are found). The song consists on a serie of 49 notes, the first 24 notes are played normaly and the last 25 notes are palm muted. The played notes are a repetition of six notes with four frequencies $[G2, B2, D3, E3, D3, B2]$.

Figure C.11 shows the waveform of the song, the associated onsets detection function which is very correctly performed in this case and the detection function for the Palm Mute detection, a dashed line shows the bordure between the two kind of playing. As we can see, the criterium gives a good separation between the two cases. In this example, it is defined with the used data, no training was done.

C.2.3.3 Violin *Pizzicato*

We did the same experiences on isolated violin notes, we took 21 (Major scale on 3 octaves) bowed notes and the same notes played pizzicato on four differents string. The result is shown on Figure C.12 we can observe that the separation is better on the violin, and it is more visible on the high frequency strings.

C.2.3.4 *Pratice* Pedal

Although we use it just for training, the practice pedal seems to lead to the same effect than the "palmute" for guitar. We have recorded a serie of five notes on a bright Piano (low, middle and high frequency range), each note is first played normally and then with the practice pedal. The STFT of this song is shown in Figure C.13 and we can make the same observation as for the palmute, the fondatemental seems to have the same amplitude for the two case but the harmonics are highly reduced with their order.

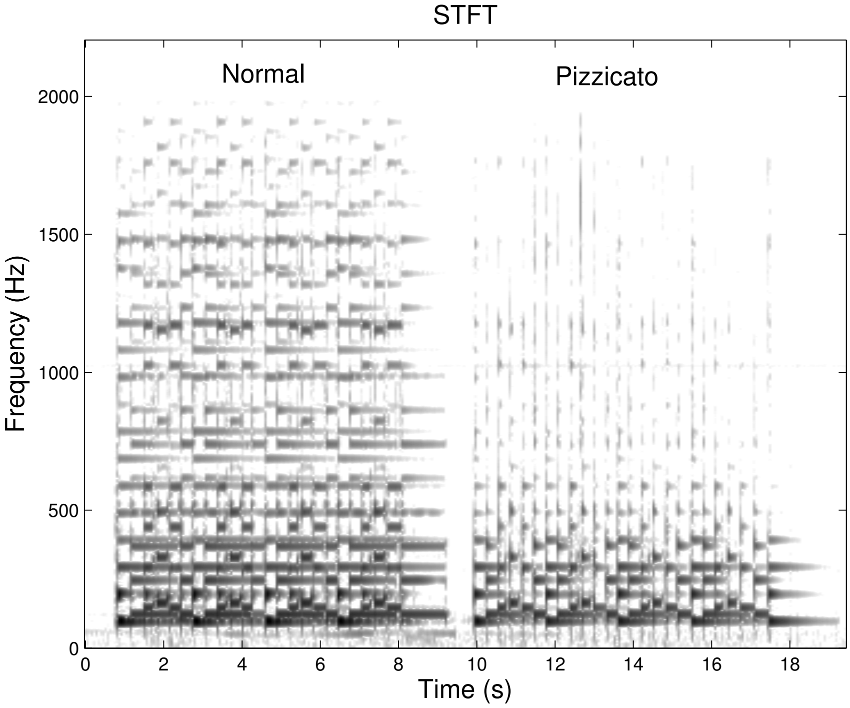


Figure C.10: *Short Time Fourier Transform, The song is composed of Non-Pizzicato and Pizzicato notes.*

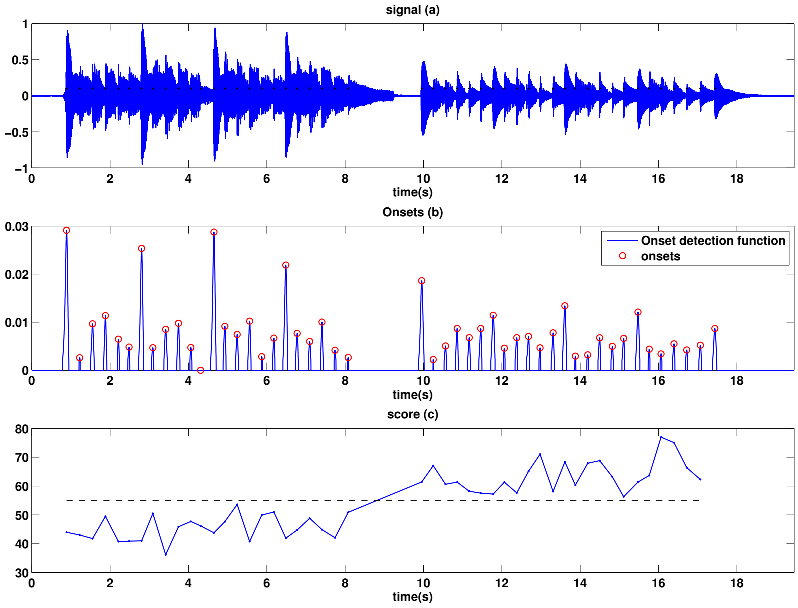


Figure C.11: *Signal and onsets (a), onsets detection function and onsets (b), result of the detection (c).*

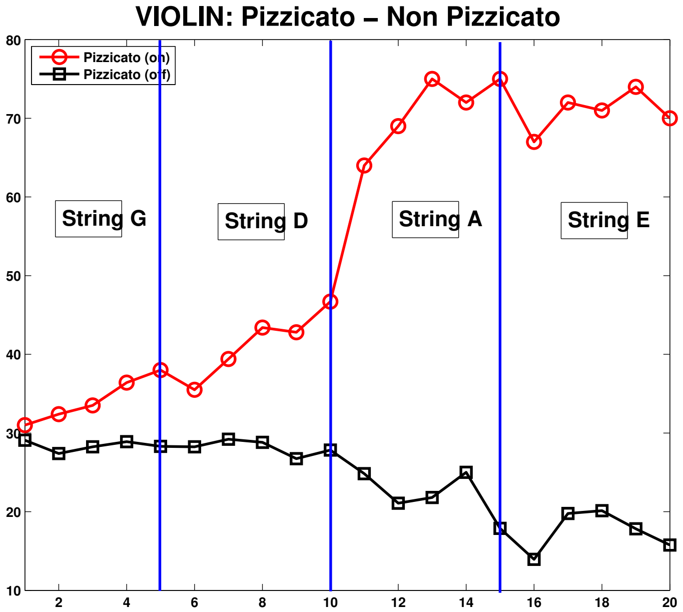


Figure C.12: *Results of the analysis for the detection of the Pizzicato for a Violin*

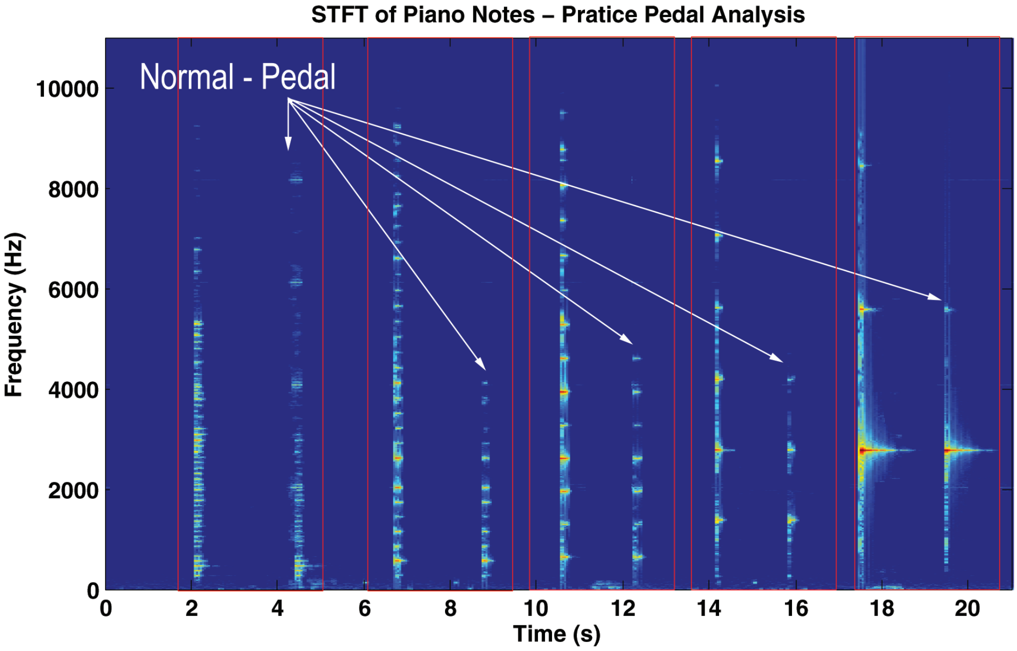


Figure C.13: *STFT of notes played with and without the practice pedal*

C.2.4 Guitar: *Bend, slide* and hammer

The bend, the slide and the hammer are the most common interpretation effects and can be used on all the strings instruments like guitar, bass, violin etc. The common characteristics of these effects is that they only have one attack for several frequency variation, and except for the bend, the variation is at least a half tone. The bend can have a continuous frequency variation and it is limited to at most two tones. The hammer is limited by the length of the hand but can also be performed with open string so there's no restriction about the frequency variaton range, the tapping which involved the right hand give also the same sound. Finally, for the slide the variation is limited by the length of the fretboard.

C.2.4.1 Fundamental Frequency and Amplitude tracking

As mentioned before, the effects seem to have some defined features about their frequency and amplitude variations. We can think than the frequency variation is smoother for the bend than for the other as we can think than the hammering will gives some extra-energy to the new notes. Figures C.14, C.15 and C.16 show some instantaneous tracking of the fundamentale frequency (which generaly is the stronger harmonic) and to its amplitude. The notes played are the same, with a demi ton of difference between the first and the second note. Unfortunaly, there is no big difference between the three cases, the hammer is the only one (on this example) which doesn't increase the energy of the fondamentale and the bend gives the smoother variation of frequency.

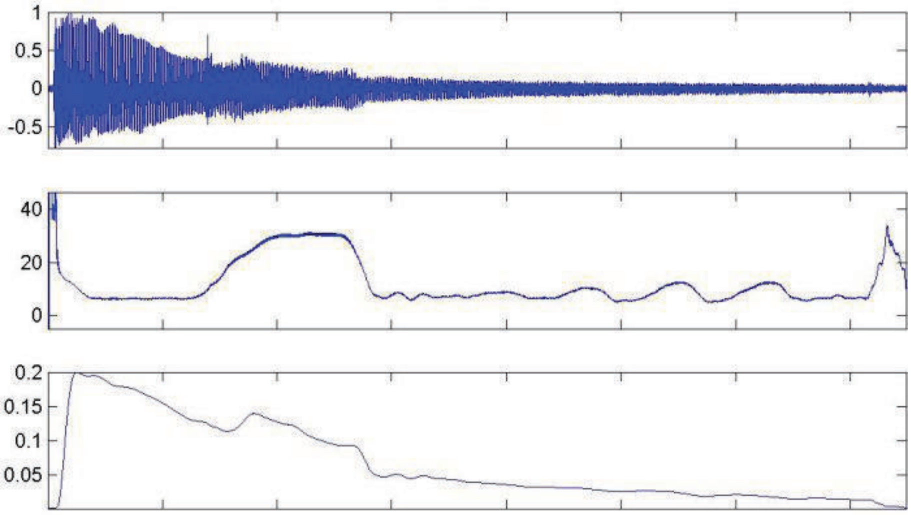


Figure C.14: *Instantaneous frequency and amplitude tracking of a Bend, guitar note.*

As we want to detect the attack, which is the transient part of the note, but only due to the attack and not to frequency variation, the onset detection is changed. Instead of using the spectral flux of the harmonic part of the signal and in its noise part for finding common onset (as mentioned in Section B.2) we keep hard onset which are present in the noise part and very low (or inexistant) in the harmonic part, if the noise onset is more or equal five times the harmonic onset we keep it.

The test data set includes some monophonic and mono-instrumental recording, for the rest we define the effects by B for the bend, H for the hammer, S for the slide and P for the played case. The first set is composed by four successions of two notes played alternatively, the first note is always played (P), the second is played or reached by one of the above effects and the last one is played or is the opposite (design by off) of the effect. The data are played on different strings and notes and follow this scheme: PPP-PHH_{off}-PSS_{off}-PBB_{off} (6*12 notes). The sond data set includes other notes (2 notes by set), the first is played and the second is an effect, and represents 24×2 notes.

Figure C.17 shows the results of the onsets detection (the detection function, the adaptive theshold and the onsets), the instantenaous frequency, and amplitude of the

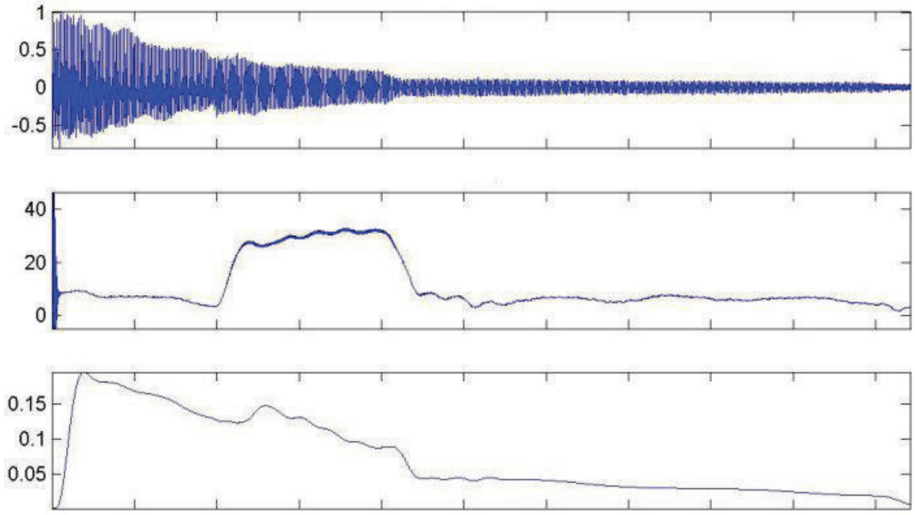


Figure C.15: *Instantaneous frequency and amplitude tracking of a Slide, guitar note.*

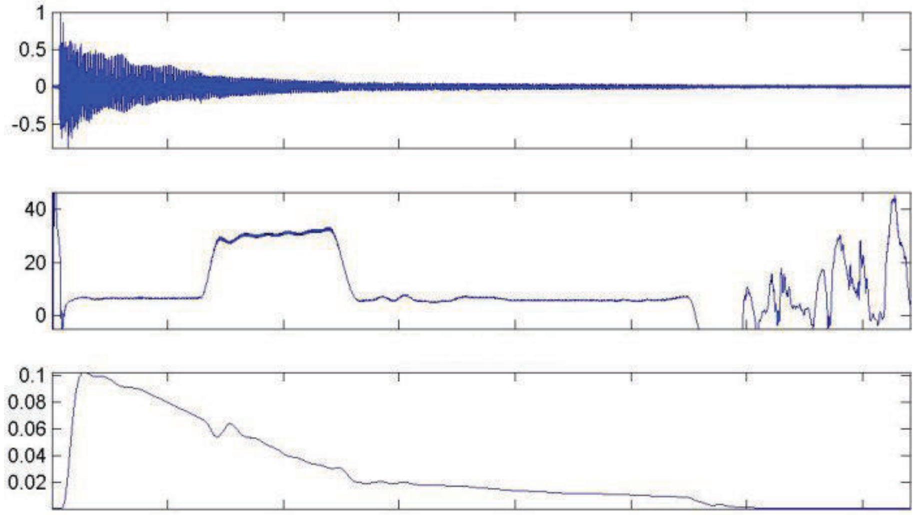


Figure C.16: *Instantaneous frequency and amplitude tracking of a Hammer, guitar note.*

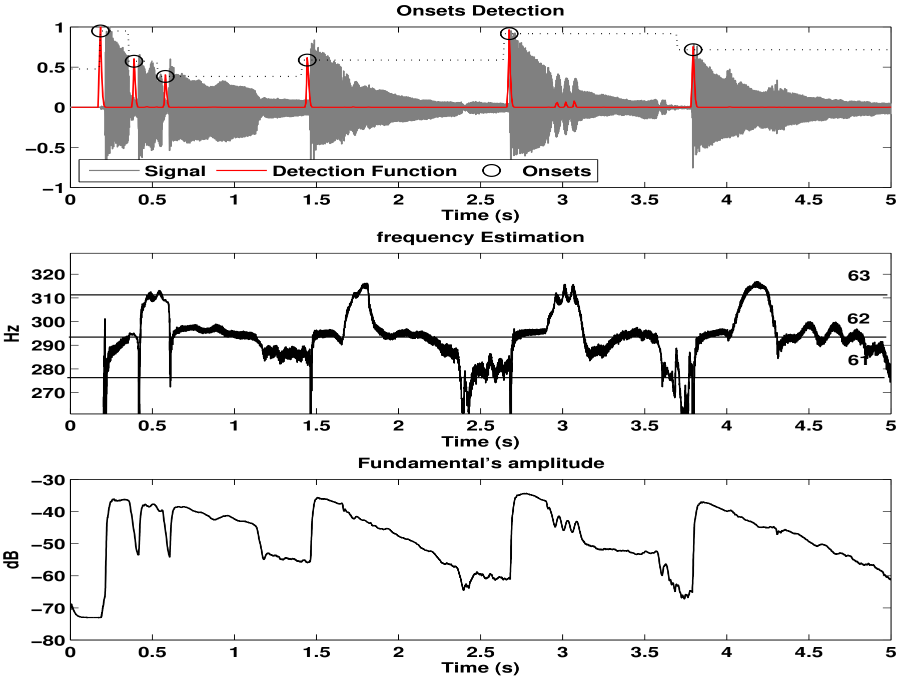


Figure C.17: *sound containing interpretation effects, onset detection function and instantaneous frequency and amplitude tracking*

fundamental. The adaptive threshold, that is a post-processing, takes as input the first onset, if the value of the detection function is upper than ten percents of the last maximum it becomes the new one. For the frequency three line show the position of the previous and next half tone (in Midi). On our dataset, the system find all the played note (60/60), it interprets some effect as played note (8/60) and it detects some artefact note (4/60). Note that the onset detector is not able to work in other case (Mono-phonic, Mono-Instrumental).

Appendix D

Periodic Signal Modeling

D.1 Problem introduction

Precise automatic music transcription requires accurate modeling and identification of the spectral content of the audio signal. Whereas a deterministic model in terms of modulated periodic signals allows to distinguish different notes, the presence of multiple notes separated by octaves poses a big problem since they share the same periodicity, and hence completely overlapping spectral content.

Here we depart from the theory of the method based on cyclic correlation analysis, extending it by using the even and odd part of the periodic signature of the signal. In section D.2.3 we apply the method as a pitch determination algorithm on both synthetic and acoustic signal. Then, in section D.2.3.2 we use it for solving the octave ambiguity problem and compare it to a more sophisticated spectal method and, finally we conclude.

D.1.1 Illustration

The Octave of a note has a fondamental frequency which is twice the note, $f_{octave} = 2\, f_{note}$. If the instrument is perfect, purely harmonic, the periodicity (temporal, Figure D.1) or the harmonics (spectral, Figure D.2) are completly shared. The contribution of the octave is shown in Figure D.3 for a guitar. In this condition, traditional pitch estimator failed to find the presence of the two notes, and an additional analysis must be used.

As for the interpretation effects detection, the state of the art is very limited and the Octave detection is, in the most case, ignored in automatic music transcription. In [169] an octave detector is proposed, the system use Support Vector Machine (SVM) and analyse the timbre of the mixed notes. Here we present a method which is equivalent of the *Odd to Even Harmonics Energy Ratio* (certainly used on several systems) [163] but the analyse is not completly performed in the spectral domain, we analyse the Odd and Even harmonics ratio in the time domain [170].

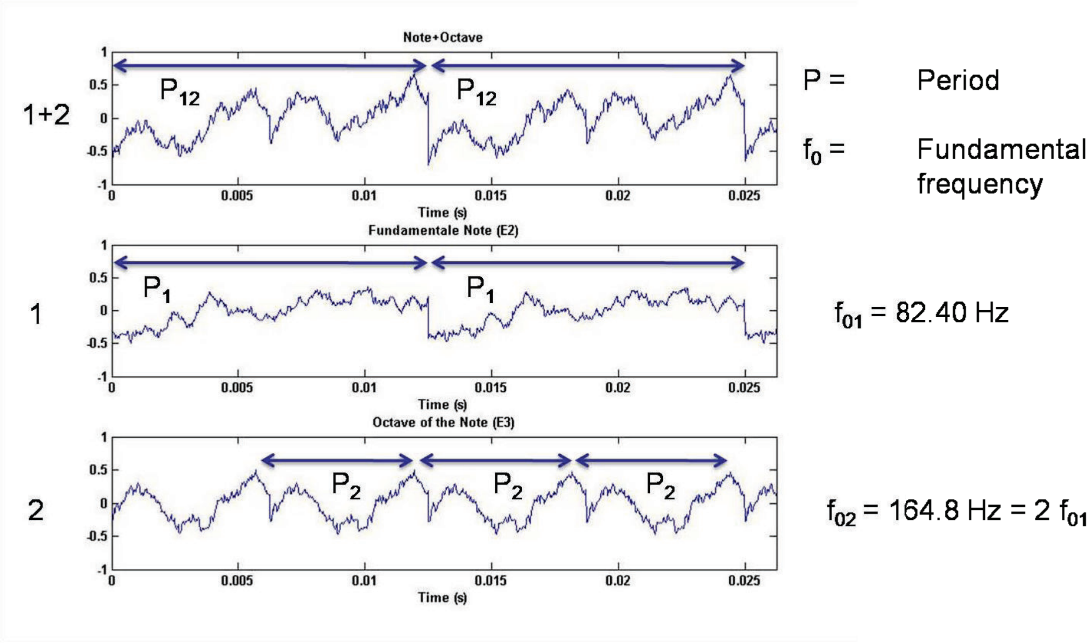


Figure D.1: *Illustration of the Octave Problem, Temporal point of view*

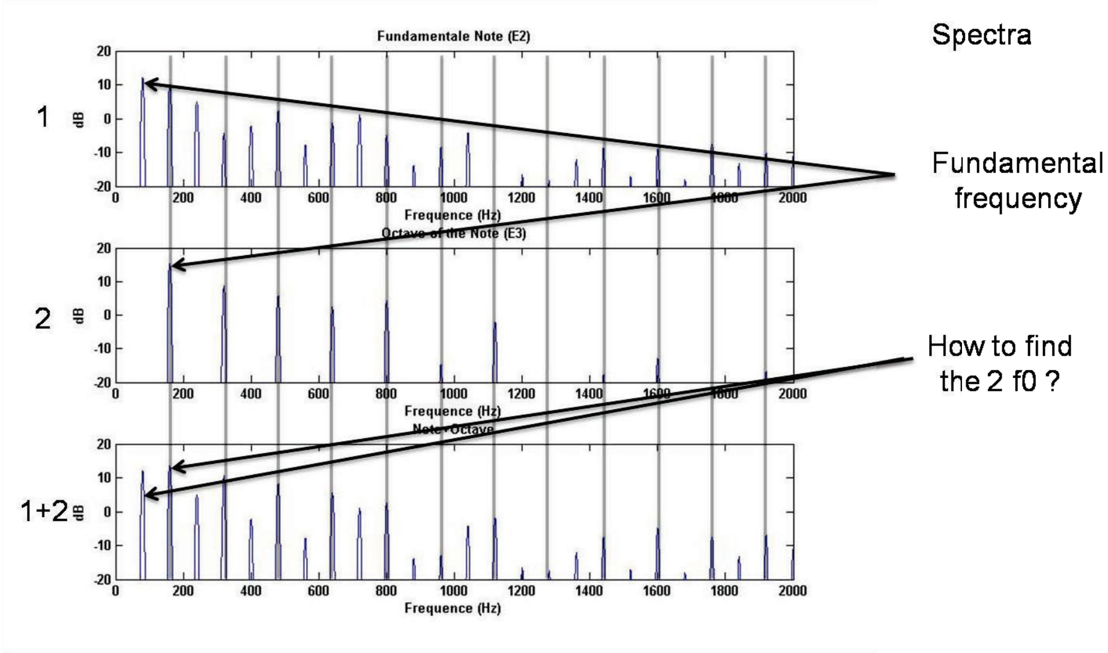


Figure D.2: *Illustration of the Octave Problem, Spectral point of view*

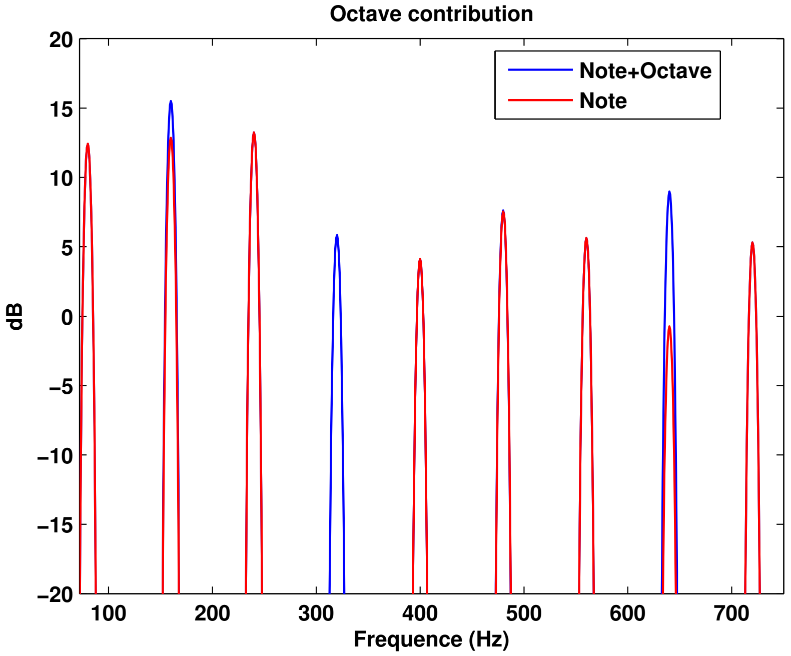


Figure D.3: *Difference between note and note+octave*

D.2 Periodic signal Modeling

Musical signals are often modeled as the sum of sinusoids with time varying parameters and an additional noise. For an instrumental or a speech signal, the signal is also harmonic with fundamental frequency equal to f_0.

$$\begin{aligned} s(t) &= x(t) + n(t), && \text{(D.1)} \\ x(t) &= \sum_{n=0}^{N-1} A_n(t) \cos(2\pi \frac{f_n(t)}{f_s} + \phi_n(t)) && \text{(D.2)} \\ f_n(t) &= n\ f_0(t) && \text{(D.3)} \end{aligned}$$

As defined in [143] the periodic signal can be expressed by its *generalized ACF*, which is cyclic.

$$r_k^P = r_k *\ \delta_{k,o,P}, \quad \delta_{k,n,P} = \sum_{i=-\infty}^{+\infty} \delta_{k,n+iP} \tag{D.4}$$

where $*$ denotes the convolution operator; and δ the Kroenecker delta. Its spectral expression is given by:

$$S^P(f) = S(f)\ \frac{1}{P}\ \delta_{\frac{1}{P}}(f), \quad \delta_{f_0}(f) = \sum_{k=-\infty}^{+\infty} \delta(f - kf_0) \tag{D.5}$$

If we define $S(f)$ as:

$$S(f) = \sum_{k=0}^{P-1} r_k^P \; e^{-j2\pi fk}, \text{with } r_k^P \; = r_{P-k}^P \tag{D.6}$$

$$\tag{D.7}$$

Then the spectral envelope of a such periodic signal can be written as:

$$S(f) = r_0 + 2\sum_{k=1}^{\frac{P}{2}-1} r_k \; \cos(2\pi fk) + \; r_{\frac{P}{2}} \; \cos(2\pi f\frac{P}{2}) \tag{D.8}$$

We can define the even and odd parts of the cyclic correlation:

$$\begin{aligned}
r_k^P &= r_k^{P,e} \; + r_k^{P,o}, && \text{(D.9)}\\
r_k^{P,e} &= \frac{1}{2}\,(r_k^P \; + r_{k\;+\frac{P}{2}}^P), && \text{(D.10)}\\
r_k^{P,o} &= \frac{1}{2}\,(r_k^P \; - r_{k\;+\frac{P}{2}}^P), && \text{(D.11)}\\
r_{k+\frac{P}{2}}^P &= r_{\frac{P}{2}-k}^P && \text{(D.12)}
\end{aligned}$$

The influence on the spectrum is expressed as follow:

$$\begin{aligned}
S(f) &= S^e(f) + S^o(f), && \text{(D.13)}\\
S^e(f) &= S(f)[\,\frac{1}{2}(\,\frac{1+e^{-j2\pi f\frac{P}{2}}}{2} + \frac{1+e^{j2\pi f\frac{P}{2}}}{2})], && \text{(D.14)}\\
S^e(f) &= S(f)(\,\frac{1}{2} + \,\frac{1}{2}\cos(2\pi f\frac{P}{2})) = S(f)\;F^e(f), && \text{(D.15)}\\
S^o(f) &= S(f)[\,\frac{1}{2}(\,\frac{1-e^{-j2\pi f\frac{P}{2}}}{2} + \frac{1-e^{j2\pi f\frac{P}{2}}}{2})], && \text{(D.16)}\\
S^o(f) &= S(f)(\,\frac{1}{2} - \,\frac{1}{2}\cos(2\pi f\frac{P}{2})) = S(f)\;F^o(f) && \text{(D.17)}
\end{aligned}$$

Figure D.4 shows the frequency selection of the even and odd parts. As the Fourier Transform is done with P points, with P the period of the signal, each point of the spectrum is a peak of the periodic signal and the Spectrum represents the spectral envelope if the signal is harmonic. If we define the fundamental frequency as the first harmonic, the even part cancels the odd harmonics and leaves the even harmonics unchanged and vice-versa for the odd part.

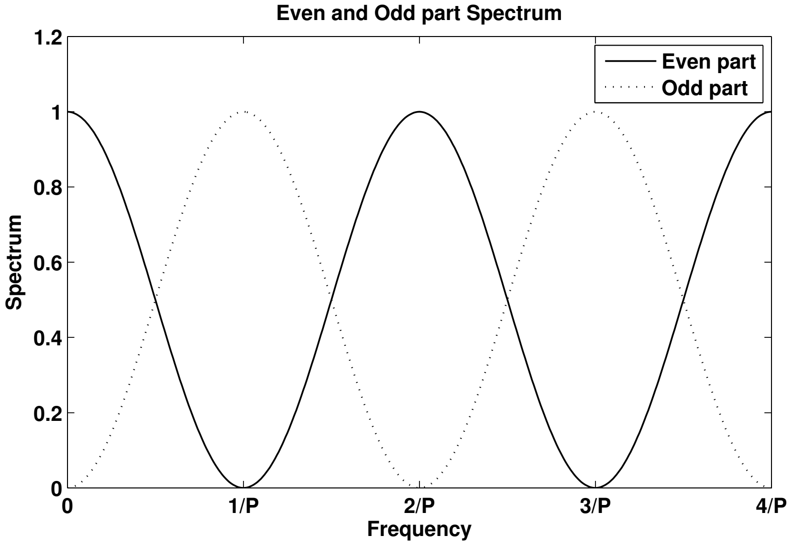


Figure D.4: Even and odd parts of the spectrum.

D.2.1 Definition of the periodic signature

The signal is first resampled to a power of two in order to avoid problems when the even and odd part are computed and for having an integer period. Then the signal is cut into frames of length P, the periodic signature is expressed by its *generalized ACF* :

$$R^P = IDFT(|DFT(X^P)|^p) \tag{D.18}$$

where R^P and X^P are two matrices for which each column represents a period of the signal and its cyclic representation respectively:

$$X^P = [x_1 \ . \ . \ . \ x_m] \tag{D.19}$$

$$x_m = \left[s_{(1+(m-1)P)} \ . \ . \ . \ s_{(mP)}\right]^T \tag{D.20}$$

where T denote the transpose operator, m is the number of period in the analysed signal and x is a signal vector containing P samples.

As the harmonics of an audio signal are time varying and non perfectly harmonic, we need to have a robust estimate of the periodic signal. This signature is estimated as the principal vector of the eigenvalue decomposition of R^P. We define u, the periodic signature, as the first column of $U = SVD(R^P)$.

Then the odd and even parts of the signature are computed as:

$$u_k^{P,e} = \frac{1}{2}\,(u_k^P + u_{k+\frac{P}{2}}^P), \tag{D.21}$$

$$u_k^{P,o} = \frac{1}{2}\,(u_k^P - u_{k+\frac{P}{2}}^P), \tag{D.22}$$

D.2.2 As a pitch detector

For estimating the pitch of the signal we reduce the set of fundamental frequencies to the first twelve frequencies of the first octave from a midi correspondance. For all of this sets we perform the algorithm described before and choose as candidate the one which maximize an energy criterium. Since the periodic signature is normalized in energy, we will work with its even part. The even part also represents the octave of the pitch, so we change the set of candidates to the previous octave. Working with the lower octave candidates didn't reduces the set of octaves to the first one. When a candidate is chosen, we compute the energy of its *Even To Odd Parts Ratio* (*EOR*); if it's more than a threshold, then we decide that its true octave is the next one and we continue on the next octave by keeping, as periodic signature, the even part.

Since the energy of the periodic signature is normalised to one, the energy of the Even and Odd Part are bounded to 0.5, the chosen threshold is compared to the Even to Odd Parts Ratio and set to 10.

D.2.3 Simulation for the pitch estimation

For this simulation we have generated light inharmonic signals, in fact all the parameters are randomly generated. The Inharmonicity coefficient is set to $\beta = 10^{-5}$, so the frequencies follows as a rule $f_n = n\, f_0 \sqrt{1 + \beta\, n^2}$. The amplitudes and phases are uniformly distributed in [0;1] and [0;2π] respectively. The amplitudes are also decreasing with the index and the sum of the amplitudes is normalized to 1. We have chosen the tessitura of the guitar for our analyse such that the set of midi code is [40;88].

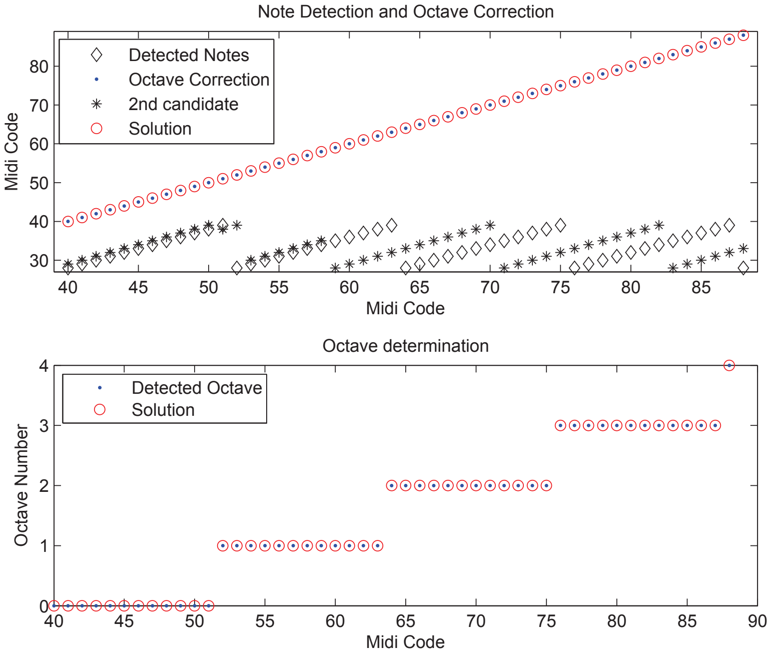


Figure D.5: Pitch detection and Octave Selection for a synthetic signal.

Figure D.5 shows the result of the analysis: as expected, the notes are correctly interpreted on the octave zero, and their true octaves are correctly found. The second possible candidate is also shown for each notes. As we can see for the first and a half octave, it has a semitone difference, while for the next octave it's a perfect fourth difference (5 semitones upper).

D.2.3.1 Diagram for the pitch determination

The following diagram (Figure D.6) summarizes the algorithm for the pitch detection algorithm.

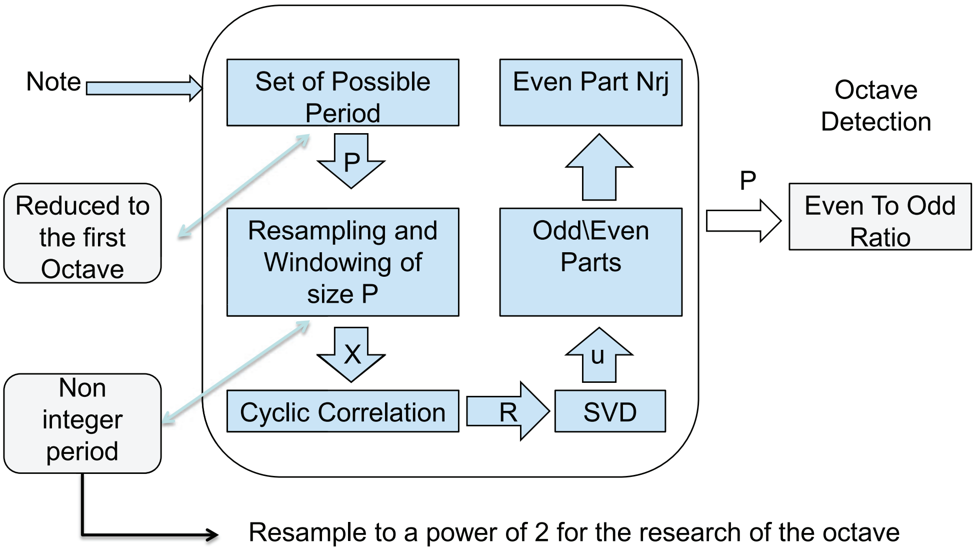

Note
Set of Possible Period
Even Part Nrj
Octave Detection
P
Reduced to the first Octave
P
Resampling and Windowing of size P
Odd\Even Parts
Even To Odd Ratio
X
u
Non integer period
Cyclic Correlation
R
SVD
Resample to a power of 2 for the research of the octave

Figure D.6: *Diagram of the pitch estimation algorithm*

D.2.3.2 Illustration for the octave determination

In order to illustrate the *good* octave selection, we give in Figure D.7, D.8 and D.9 some details. The true octave is the 3^{rd} one (from a midi correspondance) and the song is a guitar note, the lower note of a guitar is a E2 (82.4 Hz) and it begins at the second octave. As mentioned before we begin at the octave -1, and, for the guitar, the octave -1 is the octave 1 of the midi reference. We apply the algorithm described before for finding the pitch. The first plot shows the signal (resampled), the second and third ones are the signal cut into small frames and its cyclic correlation respectively. Then the signature and the decomposition into its odd and even parts. The bottom part presents when we are at a lower octave than the good one, we have, in the periodic signature, more than one period and this gives more energy to the even part. This method is philosophically equivalent to the subharmonic-to-harmonic ratio one [162].

D.2.4 Application to a true signal

For this analysis we have recorded all the first 37 notes of the guitar (midicode 40 to 76) on an acoustic guitar. The notes are played with a guitar pick and the guitar was plugged and linked to an external soundcard. The analysis is made on the first $250ms$ of the signal (including the attack). Note that the guitar was not perfectly tuned (impossible) and the used candidate are determined again by the midi reference frequency.

Figure D.5 shows the result of the analysis for the guitar, the result is not perfect, but we can see that if a note is not well detected its octave is false and the note found is the perfect fourth of the played note, the second candidat of the previous analysis. In this case the true note becomes the second choice. Note that the perfect fourth shares some harmonics in the even part but doesn't share its fundamental frequency.

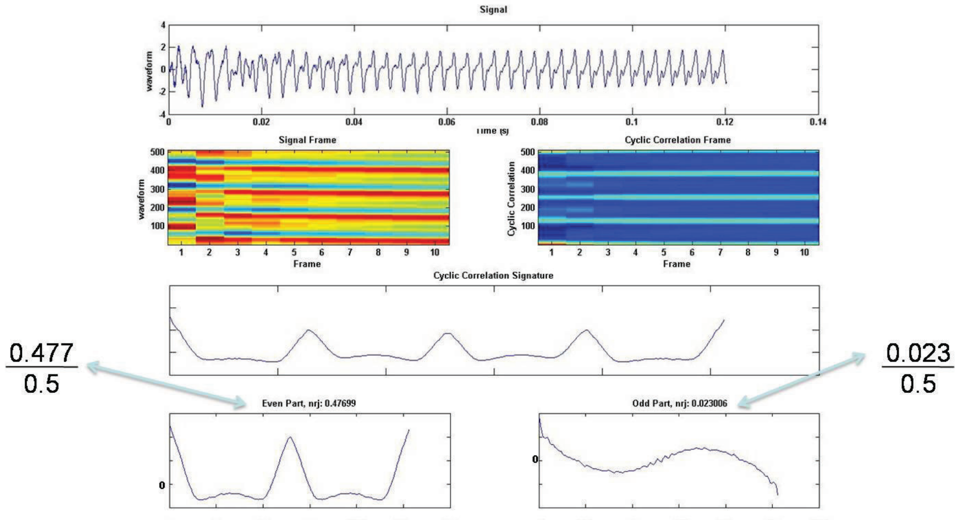


Figure D.7: *Octave* −2

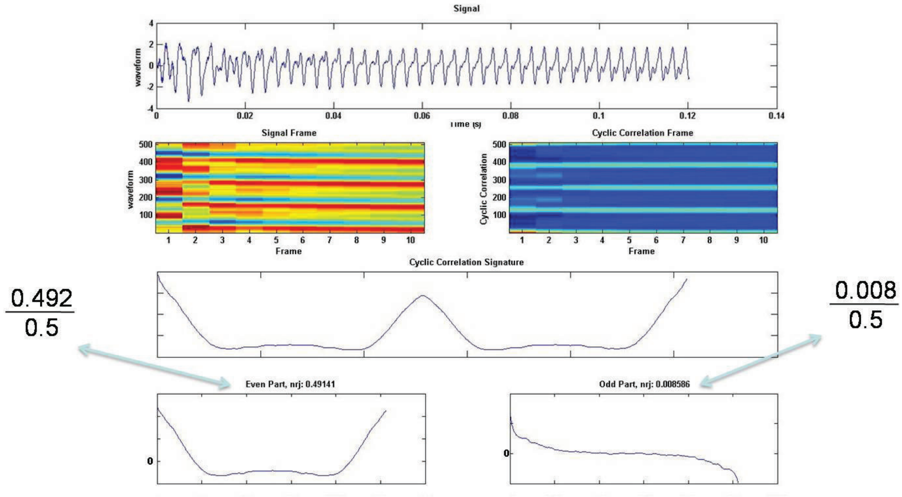


Figure D.8: *Octave* −1

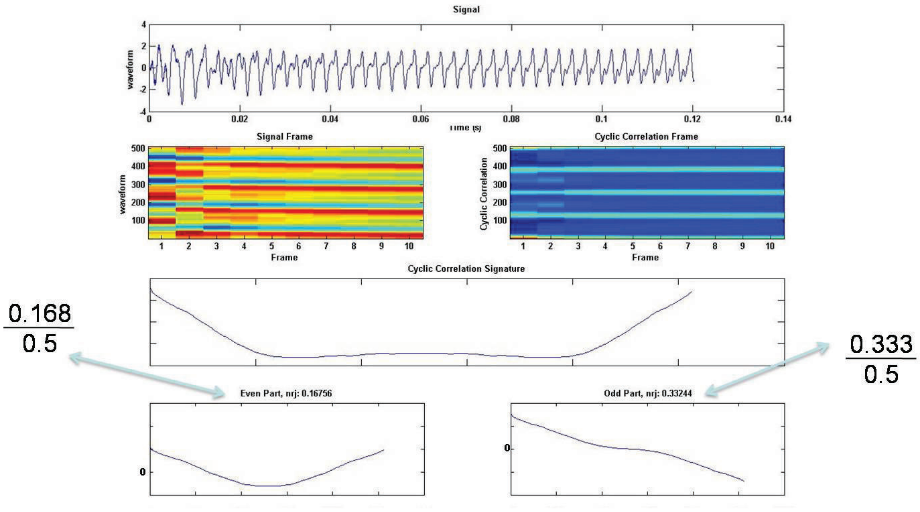


Figure D.9: *Good Octave*

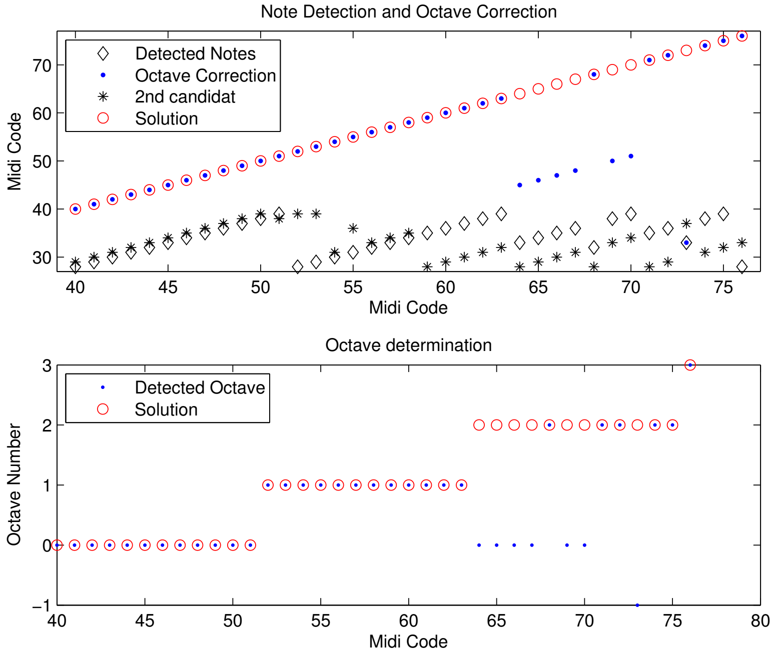


Figure D.10: Pitch detection and Octave Selection for guitar.

D.2.5 Application to the Octave Problem

In this section we analyse the octave problem. The octave problem rises when a note and its octave are played together. They share the same periodicity and the even harmonics of the played notes are amplified by the harmonics of the Octave. For the analysis we assume that the fundamental frequencies are known. In spectral analysis there are, at least, two ways to estimate the even and odd frequencies. The first one consiste in finding all the peaks in the spectrum, by peak picking, and by paying attention not to miss some of them otherwise an odd harmonic can become an even harmonic and vice-versa, another point is the inharmonicity of the signal. To find the peaks we have to adjust, from one peak to the next one, the distance and searching a local maximum around it. The second method is equivalent to the proposed method; it consists in computing the spectra of the matrix X^P, define before, and taking the average trough the time dimension. this is Welch's periodogram, then the even harmonics are the even samples of the spectrum.

D.2.6 Note plus its Octave

In this case, a note is played with and without its octave, recorded in the same condition as before with an acoustic guitar. We compare the results of the proposed method with the first spectral techniques (with peak picking). The second spectral method, explain before gives very similar results to the proposed one (temporal), so we just show our proposed method. The results (Figure D.11) are poor for the two methods due to the coloration of the spectra.

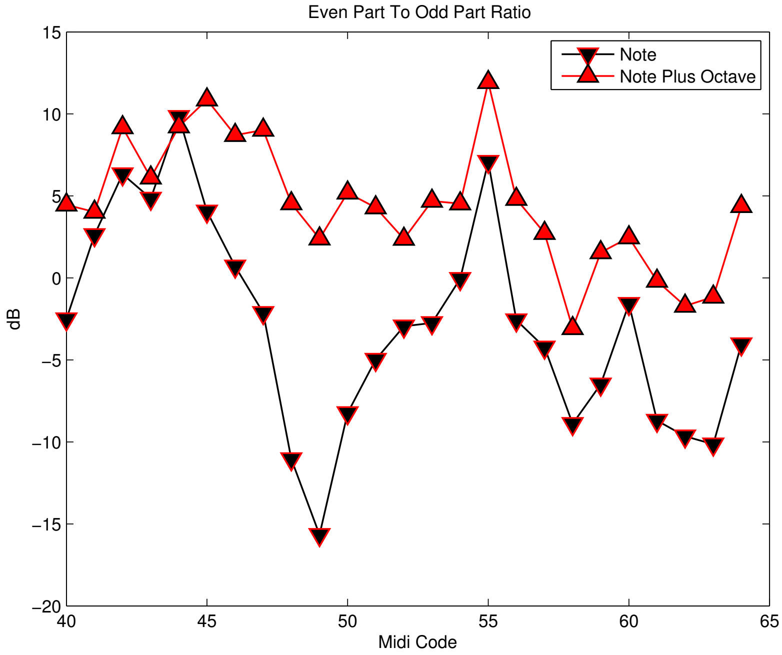


Figure D.11: Octave problem, a note with its octave.

We have decided to add in our framework another one preprocessing: for the rest of the simulation we worked in the prediction error of the signal. The signal is modeled as an autoregressive model of order ten. Morever, we defined that a note can't be interpreted as its octave, but a note with its octave can be interpreted as the note by itself.

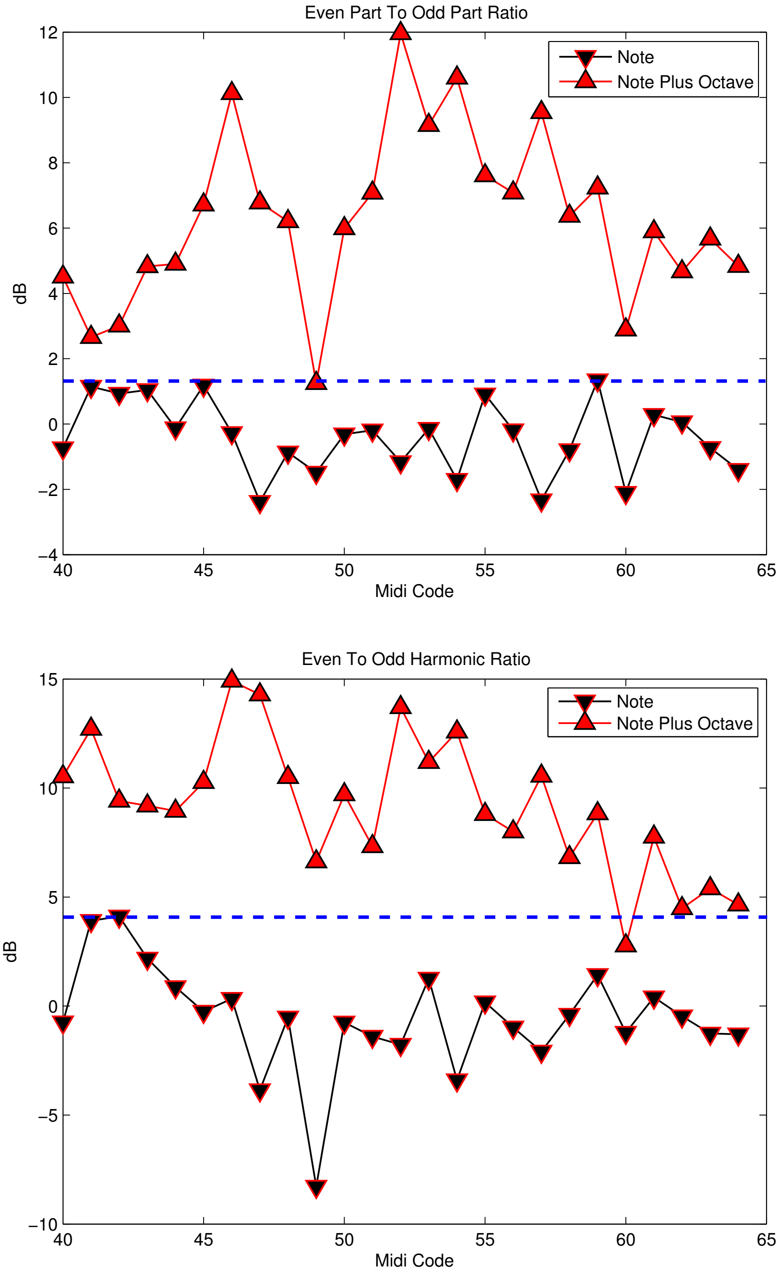


Figure D.12: Octave problem in the prediction error, a note with its octave with the temporal method (top) and the spectral method (bottom).

The results (Figure D.12) are better for the two methods. The dashed line is the upper value of the notes alone, in the two cases we make one error.

D.2.7 Note plus its first two octaves

In this part the notes are compared to the case where the first two octaves are present simultaneously. The analysis is performed at the fundamental frequency ($f0$), at twice and triple the frequency. For a visibility problem we don't show the result for the notes alone and, for an evident reason, the analysis is done on the first octave (midi code 40 to 52).

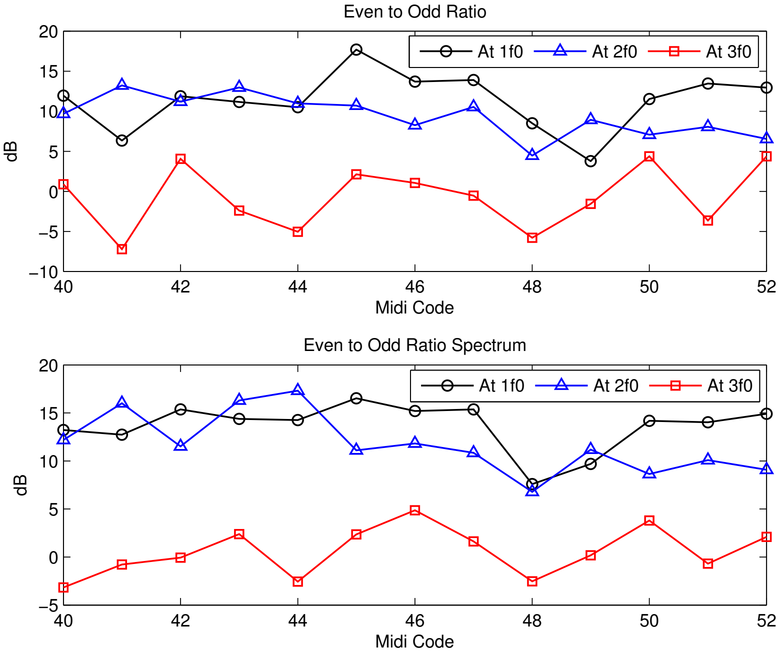


Figure D.13: Octave problem in the prediction error, a note with its first and second octaves. Temporal method (top) and Spectral method (bottom).

The results in Figure D.13 are also good for the two methods. The analysis at the fundamental frequency finds the next octave; at the first octave, we found the $2nd$ octave and after there is nothing.

D.2.8 Note plus its second octave

Now we compare the two methods for the case of a note with its second octave (an octave is missing). The second octave influences one harmonic over four from the fourth harmonic, so the result of the analysis should be slightly similar to the previous analysis. Figure D.14 shows the result, we know which octave is the last one but nothing between the note and the octave. The only possibility for solving this problem is to estimate the envelope of the individual component of the signal.

D.2.9 Parameters used

The records were performed with a sampling frequency of 44100 Hz with a normal acoustic guitar, the sound card used is a Firebox from Presonus. The period of each analysis is resampled to 512, which allows a significant number of decompositions for the Even and Odd ones. The parameter p of the *generalized ACF* is set to 1. The order of the predictor used for the prediction error is 10 and the time duration of each analysis is 250 ms.

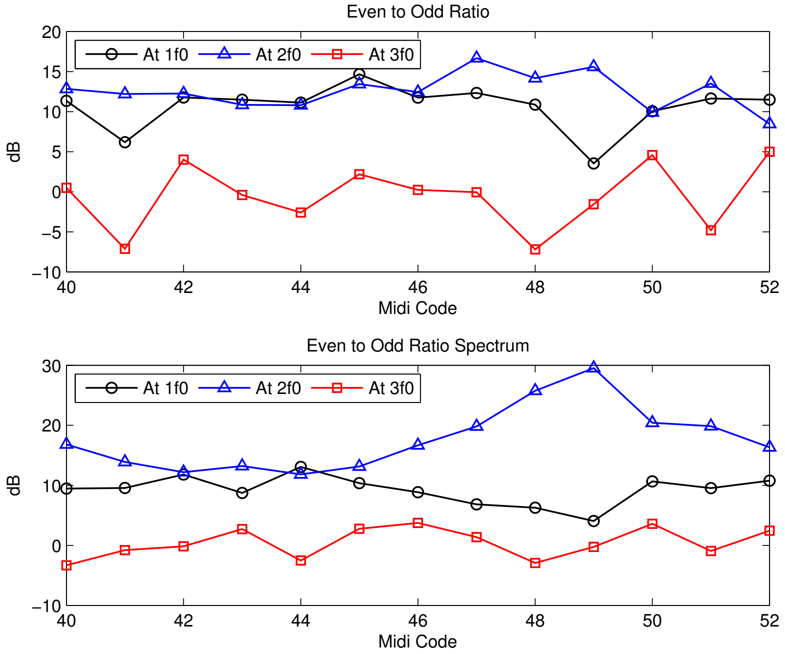


Figure D.14: Octave problem in the prediction error, a note with its second octave. Temporal method (top) and Spectral method (bottom).

D.3 Conclusion and Future work

A novel pitch determination algorithm is proposed using the separation of the Even and Odd parts of a cyclic signature of the signal. The ratio of the even and odd parts can determine the octave of the note. Simulations on synthetic and true signals show the potential of the proposed method, which can be improved by adding some constraints on the pitch candidate. A temporal view for the estimation of the present octave in the signal is proposed, and the results are compared to a more optimised method reaching similar results. Although the intermediate octave problem is not solved, we will extend our algorithm by including the estimation of the spectral envelope.

Appendix E

Audio Visual Guitar Transcription

E.1 Introduction

Written music is traditionally presented as a score, a musical notation which includes attack times, duration and pitches of the notes that constitute the song.

When dealing with the guitar this task is usually more complex. In fact, the only pitch of the note is not always enough to represent the movements and the positions that the performer has to execute to play a piece. A guitar can indeed chime the same note (i.e. a note with the same pitch) at different positions of the fretboard on different strings (See Figure E.1). This is why the musical transcription of a guitar usually takes form of a tablature.

A tablature is a musical notation which includes six lines (one for each guitar string) and numbers representing the position at which the string has to be pressed to perform a note with a given pitch (figure E.2). Special notations are added to represent particular effects like bend ('ˆ'), hammer on ('h'), pull off ('p'), etc.

The information about the movements which one player should execute to perform a piece can also be referred under the name of fingering. Burns and Wanderley [171] report few attempts that have been done to automatically extrapolate fingering information through computer algorithms:

1. real time processing using midi guitar
2. post processing using sound analysis
3. post processing using score analysis

[172] retrieves fingering information through the use of midi guitar. Using a midi guitar with different midi channels associated to each different string, Verner can, in real time, extract the complete tablature. The study points out that users of midi guitars reported false note detections, difficulties while playing, and problems of synchronization among the different strings. In any case this approach is not always applicable because it needs expensive equipment which, on top of that, is not usually used by performers on scene.

[173] suggests a solution based on the timbre of a guitar which only relies on the audio recording of a guitarist. Indeed, even if two notes have the same pitch they can have different timbre; with a-priori knowledge on the timbre of a guitar, it is therefore

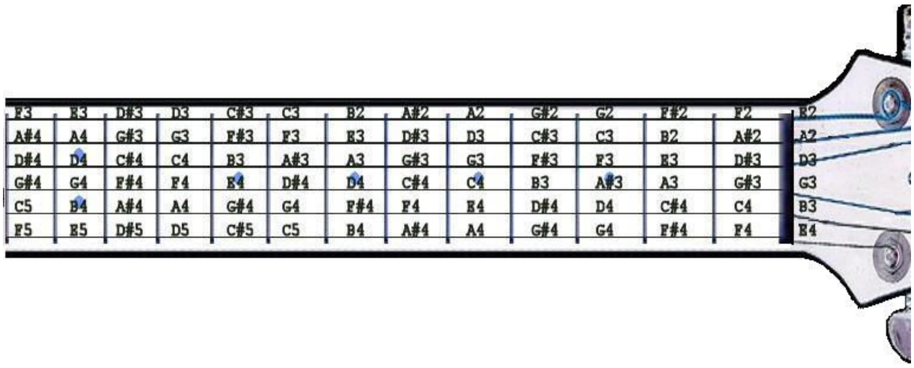

F3 E3 D#3 D3 C#3 C3 B2 A#2 A2 G#2 G2 F#2 F2 E2
A#4 A4 G#3 G3 F#3 F3 E3 D#3 D3 C#3 C3 B2 A#2 A2
D#4 D4 C#4 C4 B3 A#3 A3 G#3 G3 F#3 F3 E3 D#3 D3
G#4 G4 F#4 F4 E4 D#4 D4 C#4 C4 B3 A#3 A3 G#3 G3
C5 B4 A#4 A4 G#4 G4 F#4 F4 E4 D#4 D4 C#4 C4 B3
F5 E5 D#5 D5 C#5 C5 B4 A#4 A4 G#4 G4 F#4 F4 E4

Figure E.1: Notes on a guitar fretboard

possible to estimate the fingering position. Common issues are precision, needs for a-priori knowledge, and monophonic operation limitation.

Another possibility is to analyze the produced score and to extract the tablature by applying a set of rules based on physical constraints of the instrument, biomechanical limitations, and others philological analysis. This kind of methods can result [174] in tablatures which are similar to the one generated by humans, but hardly deal with situations in which the artistic intention or skill limitations are more important than the biomechanical movement.

Last but not least, [171] propose to use the visual modality to extract the fingering information. Their approach makes use of a camera mounted on the head of the guitar and extracts fingering information on the first 5 frets by using a circular Hough transformation to detect finger tips. Their system was positively evaluated in some preliminary studies but is not applicable to all cases because it needs ad hoc equipment, configuration, and it only returns information about the first 5 frets. Similarly Zhang et al. [175] track finger tips on a violin with a B-spline model of fingers contours.

This chapter presents a multimodal approach to address this issue [176, 177]. The proposed approach combines information from video (webcam quality) and audio analysis in order to resolve ambiguous situations.

E.2 Guitar Transcription

The typical scenario involved in the discussion of this chapter involves one guitarist playing a guitar in front of a web-cam (XviD 640x480 pixels at 25 fps). In the work presented here the entire fretboard of the guitar needs to be completely visible on the video.

E.2.1 Automatic Fretboard Detection

The first frame of the video is analyzed to detect the guitar and its position. The current version of our system presents few constraints: the guitarist is considered to play a right handed guitar (i.e. the guitar face on the right side) and to trace an angle with the horizontal which does not exceed 90°. The background is assumed to be less textured than the guitar. As a final result, this module returns the coordinates of the corner points defining the position of the guitar fretboard on the video (two outermost points for each detected fret). Guitar frets have some interesting characteristics: they are straight and usually have a different brightness compared to the wood.

The process for obtaining the position of the frets is the following. The Hough transform is employed to find the orientation of the fret board, while the edges are obtained thanks to the Canny algorithm on the original image. The image is then rotated according to the dominant edge orientation in order to align the fret board with the horizontal axis. Wavelet analysis upon the rotated image is performed for enhancing the frets. Then, horizontal projection is performed in order to crop the image to the fretboard only. At this point we have a good estimation of the frets' position but due to some perspective effect the frets may not be straight.

Skewing is applied to the image until the vertical projections are maximized. Candidates (peaks) are chosen on the projection and identified on the original image (by couple). Invalid candidate frets are further filtered out by searching for the maximum energy path between top/bottom and bottom/top extremities. Paths cannot be greater than the distance between the two extremities. Additionally, if the two paths are different then the candidate fret is discarded. At this stage, only valid frets should remain.

E.2.2 Fretboard Tracking

We have described how the fretboard position is detected on the first frame of the video. We make use of the Tomasi Lukas Kanade algorithm to follow the points along the video.

The coordinates of the end points of each fret are influenced by the movement of the hand. Therefore, some template matching techniques are applied to enforce points to stick to the fretboard. Two constraints were chosen to be invariant to scale, translation or 3D rotations of the guitar: 1) all the points defining the upper (as well as lower) bound of the fretboard must be aligned; 2) the lengths of the frets must comply to the rule $L_i = L_{(i-1)} * 2^{-1/12}$ where L_i represent the length of the i^{th} fret.

To enforce the first constraint a first line is computed that matches the highest possible number of points. The points apart from the line are filtered out and a linear regression (least squares) is computed. All points apart from this second line are filtered out and recomputed.

The second constraint is applied by comparing the positions of the points with a template representing the distances of all the frets from the nut (i.e. the fret at the head of the guitar). The best match is found for having the lowest possible number of errors. Points outside the template are removed and their positions are recomputed.

Every twenty seconds the tracking is re initialized to solve any kind of issues which may arise from a wrongful adrifts of the Lukas Kanade point tracking (see section E.3). Furthermore, sometimes it may happen that no match can be found because too many points are lost at the same time or because the guitar is not facing the camera. In this cases a new match is searched in the following frames trough the algorithms described in section E.2.1.

E.2.3 Hand Detection

In section E.2.2 the methodology employed to follow the position of the frets along the video has been described. Thanks to these coordinates it is possible to separate the region belonging to the fretboard into $n\_strings \times n\_frets$ cells corresponding to each string/fret intersection. Filtering is done on the frame to detect the skin color and the number of "hand" pixels is counted. A threshold can be applied to detect the presence of the hand (see figure E.3.a).

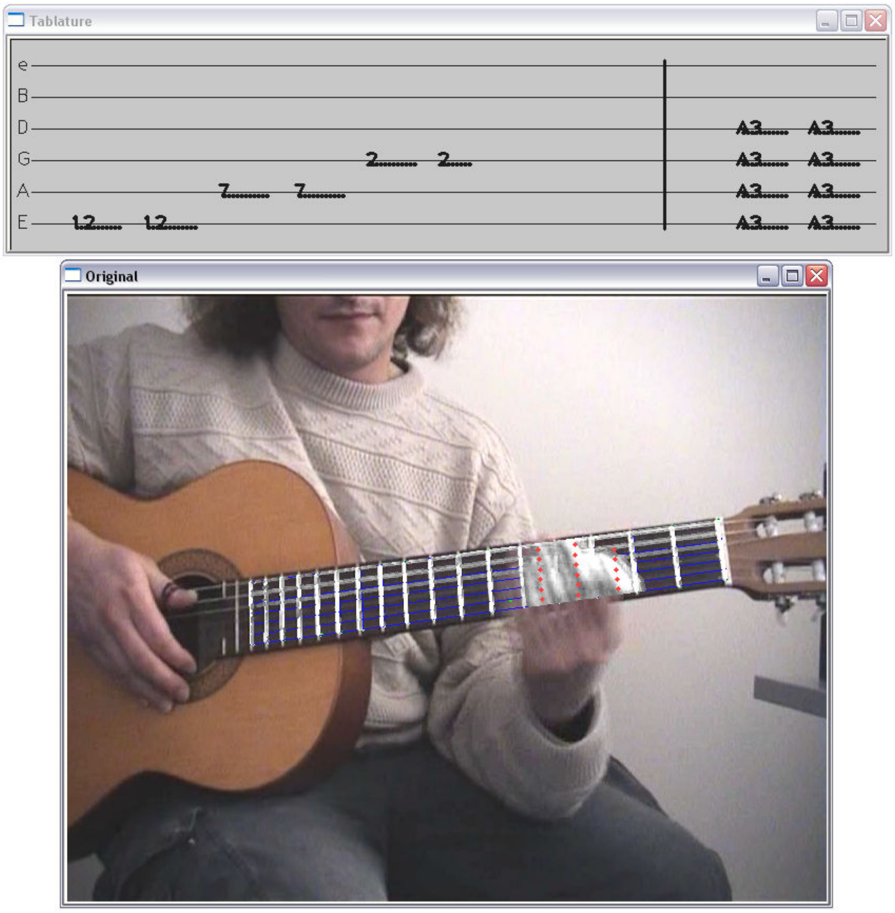


Figure E.2: Interface of the Automatic Transcription System

E.2.4 Audio Visual Information Fusion

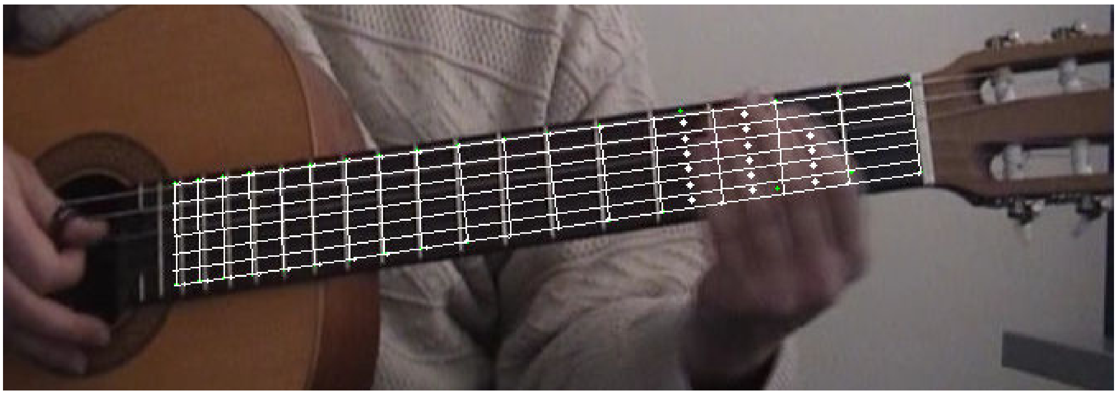
Thanks to the audio analysis and standard audio processing techniques [155] we can extrapolate the pitch of the performed notes. This information is converted to a midi file with the information of the note played and the information of the attack time and duration of the note. For each frame the information about the position of the hand is used to discriminate the correct fret-string couple producing a certain pitch.

Figure E.2 shows an example of the developed interface. We can see the interface incorporate two windows. The windows named "Tablature" shows the resulting tablature. The x axis represents the time and the six horizontal lines represent the six strings of the guitar. The vertical line at around 3/4 of the interface represent $t = 0$: at its right the information only comes from the audio analysis; at its left the information is fused together with the video information. At the right of the line $t = 0$, the same note is represented at the same time on several strings to represent the incertitude that audio brings about when dealing with instruments such as the guitar. Indeed that particular pitch can be played though all the tagged strings. At time 0 (the time represented in the windows named "Original") video is analyzed, the hand is detected at a certain fret and ambiguity is solved. At the left of the line $t = 0$ only one note at time is therefore represented. One may notice that all positions represented in the tablature at the left of the line $t = 0$ generate the same pitch (E3 = $164.81Hz$).

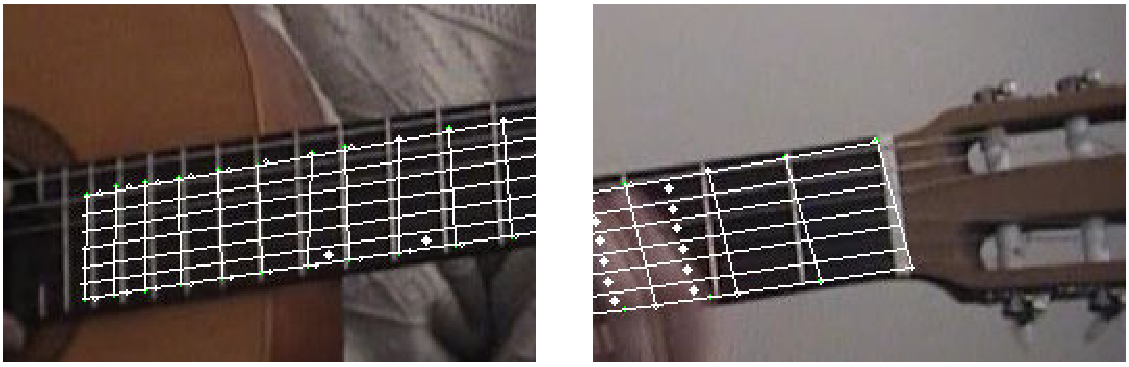
E.3 Prototype

We have tested the proposed algorithms on several short videos (around 30 seconds per video). In these videos the guitarist performs different pattern designed to test the algorithms on four different guitars (two classical, one Spanish, and one acoustic). Videos were taken in our laboratories with a DV camera placed on a tripod at less than 2 meters from the guitarist and converted to XviD 640x480 pixels, 25 frames per second at around 250 Kbps. Audio was taken with the integrated camera microphone as well as with a gun zoom microphone to reduce ambient noise.

The guitar tracking algorithm worked correctly all along all the videos. Nevertheless, issues may arise when dealing with fast hand movement which may significantly reduce the number of trackable points and/or slide a consistent number of tracked points in a specific direction. In these cases the two constraints that were described in section E.2.2 may not be sufficient to perform a good tracking.



(a) Correct Tracking



(b) Vertical adrift

(c) Horizontal adrift

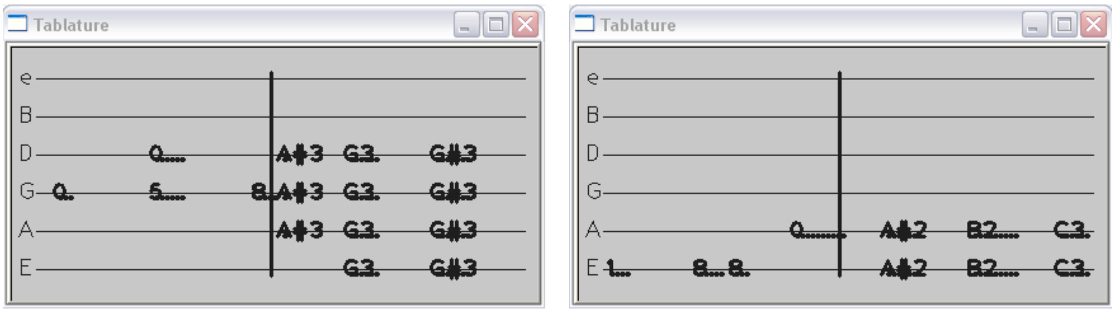
Figure E.3: Example of video errors

1) *alignment constraint.* If a significative number of points slide up or down the best fitting line may not be exactly parallel to the strings (see figure E.3 a).

2) *linear template constraint.* When a significative number of points slide horizontally or it is lost it can happen that the template matching matches better the wrong points than the correct ones. This may result in vertical lines which does not anymore match to the frets borders (see figure E.3 b).

With time both these phenomena may be amplified until the tracking is completely lost. We have empirically estimated both these phenomena to be sensible only after 30 to 40 seconds of videos and proceeded to re initialize the tracking algorithm every twenty seconds using the algorithm described in section E.2.

The hand detection was set to detect hand when at least 60% of the cell (i.e. the rectangle defining a fret and a string) contained the hand. This was found to be the minimum percentage allowing to have 0% false positives (which are due to the luminance of frets borders and strings). Setting the threshold at 60% was enough to solve 89% of the note ambiguities (see figure E.2).



(a) Ambiguity (b) Fragmentation

Figure E.4: Transcription Errors

In 11% of the cases a note which was played was assigned to two different possible positions. This corresponds to cases in which the played pitch matches with the fundamental pitch of a string (i.e. the pitch the string chime when played without pushing any fret; E2, A2, D3, G3, B3, E4). In this cases both possibilities are actually possible and our system did not disambiguate the note (see figure E.4 a).

In around the 3% of the cases one single long note was transcribed as two or more separate notes. This phenomenon was due to the artistic intention of the guitarist who slightly “bended” the string bringing both the hand and the string outside the cell. This will be addressed in future versions of the algorithm (see figure E.4 b).

E.4 Future Work

A prototype has been described in the former section which demonstrates how the adoption of simple video analysis can help the process of generation of a tablature for guitar music. The example pieces involved in this first prototype only contained a small subset of the possible techniques involved in guitar music. In this section we list some of the possible improvements upon our system.

In the former section we have seen that our system may lose a note when the guitarist “bends” the string. Future work will solve this issue by applying a probabilistic model for the position of the hand. For each cell on the fretboard a $P(h)$ will be computed representing the probability that the hand is both present on the cell and used to play (for example, a part from the case of “barre”, only finger tips are used).

Audio analysis will be extended to the polyphonic case allowing for chords and more complex pattern. To help the audio analysis dealing with polyphonic audio we will apply some machine learning techniques to learn prototypical hand positions and shapes (minor chords, major chords and principal variations).

Another system will explicitly perform right hand detection and following to estimate the string attack point to help both the audio and video processing units. Other system may be developed to detect guitar effects such as bending, tapping, slides, hammering on and pulling off, and others.

E.5 Conclusions and Future work

In this chapter we have overviewed a complete, quasi unconstrained, guitar tablature transcription system which uses low cost video cameras to solve string ambiguities in guitar pieces. A prototype was developed as a proof of concept demonstrating the feasibility of the system with today technologies. Results of our studies are positive and encourage further studies on many aspects of guitar playing.

Applications of this research include computer aided pedagogical system which may significantly help guitar students, automatic indexing of song videos through tablature indexing, computer software which may help guitarist create and share music, and many others.

For the future works, a lot of other cues can be analyzed as the right hand movement for automating style classification or refrain detection. With accurate methods, the fingering may be also analyzed and some of the most common interpretation effects may also be detected.

Appendix F

Miscellaneous

F.1 Short Term AR Coefficients Generation Using Levinson Algorithm

For obtaining a sequence of minimum phase coefficients we use the Levinson algorithm. The Levinson algorithm is a procedure to recursively calculate the solution to an equation involving a Toeplitz matrix, the algorithm runs in $O(n^2)$ time. If we have the following system, of order n, to solve:

$$R_{n+1}A_n = \begin{bmatrix} r_0 & r_1 & \cdots & r_n \\ r_1 & \ddots & \ddots & \\ \vdots & \ddots & \ddots & r_1 \\ r_n & & r_1 & r_0 \end{bmatrix} \begin{bmatrix} 1 \\ a_1 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} \sigma^2 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{F.1}$$

The prediction filter is calculated in a order-recursive way, alternating the estimation of the coefficients and of the variance $\begin{pmatrix} A_n \\ \sigma_n^2 \end{pmatrix} \Rightarrow \begin{pmatrix} A_{n+1} \\ \sigma_{n+1}^2 \end{pmatrix}$. The first estimates are $A_0 = 1$ and $\sigma_n^2 = r_0$

$$\Delta_{n+1} = [r_{n+1} \cdots r_1]\ A_n \tag{F.2}$$

$$K_{n+1} = -\frac{\Delta_{n+1}}{\sigma_n^2} \tag{F.3}$$

$$A_{n+1} = \begin{bmatrix} A_n \\ 0 \end{bmatrix} + K_{n+1} \begin{bmatrix} 0 \\ J\ A_n \end{bmatrix} \tag{F.4}$$

$$\sigma_{n+1}^2 = \sigma_n^2\ (1 - K_{n+1}^2) \tag{F.5}$$

Whith J:

$$J = \begin{bmatrix} 0 & \cdots & \cdots & 0 & 1 \\ \vdots & \ddots & \ddots & 1 & 0 \\ \vdots & \ddots & 1 & 0 & \vdots \\ 0 & 1 & 0 & 0 & \vdots \\ 1 & 0 & \cdots & \cdots & 0 \end{bmatrix} \tag{F.6}$$

From the algorithm we can see that a way for generating the coefficients is to generates the *ParCors* (K) coefficients using a uniform distribution ($[-1; 1]$) and to give the order.

It can be summarized as follow:
Initialize $A_0 = 1$

- generates a uniform number for K_{n+1}
- compute the coefficient $A_{n+1} = \begin{bmatrix} A_n \\ 0 \end{bmatrix} + K_{n+1} \begin{bmatrix} 0 \\ J\ A_n \end{bmatrix}$
- when we reach the good order, stop

F.2 Iterative Algorithm for estimating Short plus Long Term AR Model

Estimating the AR coefficients (Short Term and Long Term) can be a direct process if it is done by joint estimation. Generally, joint estimation involves one global filter [69], this approach can leads to non-minimum phase Short Term AR and it is not wanted. Other joint approach as in [78] needs to know the Pitch for working. In the approach considered here we want to uncouple the two aspects in order to estimate the Period.

F.2.1 Short Term AR Coefficients

The Short Term AR Coefficients, of order p, are estimated using Linear Prediction, the Short time correlation matrix is used for finding the coefficients:

$$\mathbf{R}\,\mathbf{a} \;=\; \mathbf{g} \tag{F.7}$$

$\mathbf{a}$, of size $p + 1$, contains the coefficients of interest. $\mathbf{R}$ is a Toeplitz Matrix constructed with the correlation sequence of the signal and $\mathbf{g} = [\sigma_e^2\ 0\ \cdots\ 0]$ contains the Short term prediction error variance σ_e^2. One can directly inverse the matrix, or use several known techniques as the Levinson-Durbin Recursion or others.

F.2.2 Long Term AR Coefficient

The Long term AR coefficient is related to period (τ) of the signal, it is a sparse vector. The pitch predictor has a small number of taps. The delays associated with these taps are bunched around a value which corresponds to the estimated pitch period in samples.

$$\mathbf{b} \;=\; [1,\ 0,\ \cdots,\ 0,\ -(1-\alpha)\times b,\ -\alpha\times b]^T \tag{F.8}$$

It is a two taps example, b is the Long term coefficient. Other number of taps can be used, generally, it is one, two or three taps. The Long term coefficients and the period are estimated in the correlation sequence. If the signal is also Short term AR, the Short term influence has to be removed. In this case we use the correlation sequence of the Short term error.

F.2.3 Iterative estimation

The iterative process consists in alterning the estimation of the coefficients. First an estimate of the Short term coefficients is done on the signal, then in the Short term prediction error the Long term coefficient and the period are estimated. Then we use estimated Long term parameters for estimating the Short term in the Long term prediction error. The algorithm can be summarised as follow:

- initialization
  - Estimation of the Short term coefficients in the signal
  - Estimation of the Long term coefficient and of the period in the Short term error
- Main process
  - Estimation of the Short term coefficients in the Long term error using the parameters estimated
  - Estimation of the Long term coefficients in the Short term error using the parameters estimated
  - Estimation of the variance in the Short plus Long term error
- stop condition

F.3 Short Term Fourier Transform (STFT)

The Short Term Fourier Transform (STFT) also called Short Time Fourier Transform, is a Fourier-related transform used to determine the spectral contents of Short segments of a signal. It analyses a signal by Short duration segments, because the analyzed signal is not stationary, so the signal is naturally weighted by a Short duration time window. If no explicit window is used, the signal is then weighted by a rectangular window. The uses of other window, as the raised cosine windows, which decrease to zero (or closed to) at the boundary, implies to overlapp the windows for a perfect output reconstruction of the total signal, it also reduces the artifacts at the boundary. A frame of a STFT, in the discret case, is the Fourier Transform of a part of the signal weighted by a window:

$$\mathbf{STFT}(x(n)) = X(m, f) = \sum_n w(n-m)x(n)\exp^{-2\ i\ pi\ f\ n} \tag{F.9}$$

With $x(n)$ the signal at time sample n, w is the analysis window. m refers to a time frame, f to a continuous frequency and X is the Fourier transform of x for the frame m. As this the frequency is continuous but in pratice it is replaced by a discrete Fourier Transform.

F.4 Evaluation Criteria

F.4.1 Decomposition

The principle of the performance measures described in [74] is to decompose a given estimate $\hat{x}_k(t)$ of a source $x_k(t)$ as a sum

$$\hat{x}_k(t) = x_{target}(t) + e_{interf}(t) + e_{noise}(t) + e_{artif}(t) \tag{F.10}$$

where $x_{target}(t)$ is an allowed deformation of the target source $x_k(t)$, $e_{interf}(t)$ is an allowed deformation of the sources which accounts for the interferences of the unwanted sources, $e_{noise}(t)$ is an allowed deformation of the perturbating noise (but not the sources), and $e_{artif}(t)$ is an *artifact* term that may correspond to artifacts of the separation algorithm such as musical noise, etc. or simply to deformations induced by the separation algorithm that are not allowed.

F.4.2 Global Criteria

Four global performance measures are defined :

F.4.2.1 Source to Distortion Ratio

$$SDR = 10log_{10} \frac{\|x_{target}(t)\|^2}{\|e_{interf}(t) + e_{noise}(t) + e_{artif}(t)\|^2} \tag{F.11}$$

F.4.2.2 Source to Interferences Ratio

$$SIR = 10log_{10} \frac{\|x_{target}(t)\|^2}{\|e_{interf}(t)\|^2} \tag{F.12}$$

The interference noise is due to the presence of the other sources in the reconstruction of one source.

F.4.2.3 the Sources to Artifacts Ratio

$$SAR = 10log_{10} \frac{\|x_{target}(t)\|^2}{\|e_{artif}(t)\|^2} \tag{F.13}$$

The artifacts noise (e_{artif}) represents the noise due to the separation algorithm. In fact the artifacts noise is not explained by the interference noise (e_{interf}) and the additive noise (e_{noise}).

F.4.3 Local Criteria

Sometimes, it is not very satisfying to summarize the performance by a single figure for the whole signal: it may happen that on some pieces of the estimated signal the interferences are very low because the target source is loud, but on other pieces the target source vanishes. This the case when analysing a Long speech signal. Global performance criteria can still be used as local if the signals are splitted into frames, it needs to introduce an window and allow overlap between consecutive frames.

F.5 Windows Properties

F.5.1 Notations

The length of a signal y is M, N is the length of the window w and N_{fft} the number of points of its Fourier Transform W. Generally M and N are a power of two because the Fourier Tansform is done using Fast Fourier Transform (FFT) which needs a power of two, if not the FFT add zeros for reaching the next power of two. f is the normalized frequency.

F.5.2 Typical Windows

The common windows are:

- The Rectangular window:

$$\begin{aligned} w(n) &= 1 \text{ if } \leq t \leq N-1, \ 0 \text{ otherwise} \\ &= r(n) \\ W(f) &= \sum_{n=0}^{N-1} e^{-2i\pi f n} \\ &= e^{-i\pi f(N-1)} \frac{\sin(\pi f N)}{\sin(\pi f)} \\ &= R(f) \end{aligned}$$

We denote the Fourier Transform of the Rectangular Window ($r(n)$) by $R(f)$

- The Triangular window:

$$\begin{aligned} w(n) &= \frac{2}{N-1}\left(\frac{N-1}{2} - |n - \frac{N-1}{2}|\right) \text{ if } \leq t \leq N-1, \ 0 \text{ otherwise} \\ &= r_{half}(n) * r_{half}(n) \\ W(f) &= R_{half}^2(f) \end{aligned}$$

where $r_{half}(n)$ is a rectangular window of size $N/2$

- The Hann window:

$$\begin{aligned} w(n) &= r(n) \times \frac{1}{2}\left(1 - \cos\left(\frac{2\pi n}{N-1}\right)\right) \text{ if } \leq t \leq N-1, \ 0 \text{ otherwise} \\ W(f) &= \frac{1}{2}R(f) - \frac{1}{4}R(f - \frac{1}{N-1}) - \frac{1}{4}R(f + \frac{1}{N-1}) \end{aligned}$$

The term Hanning window is sometimes used to refer to the Hann window.

- The Hamming window:

$$\begin{aligned} w(n) &= r(n) \times \left(0.54 - 0.46\cos\left(\frac{2\pi n}{N-1}\right)\right) \text{ if } \leq t \leq N-1, \text{ 0 otherwise} \\ W(f) &= 0.54\ R(f) - 0.23\ R(f - \frac{1}{N-1}) - 0.23\ R(f + \frac{1}{N-1}) \end{aligned}$$

The term Hanning window is sometimes used to refer to the Hann window.

- The Blackman window:

$$\begin{aligned} w(n) &= r(n) \times \left(0.42 - 0.5\cos\left(\frac{2\pi n}{N-1}\right) + 0.08\cos\left(\frac{4\pi n}{N-1}\right)\right) && \text{(F.14)} \\ &\rightarrow \text{ if } \leq t \leq N-1, \text{ 0 otherwise} \\ W(f) &= 0.42\ R(f) - 0.25\ R(f - \frac{1}{N-1}) - 0.25\ R(f + \frac{1}{N-1}) && \text{(F.15)} \\ &+ 0.04\ R(f - \frac{2}{N-1}) + 0.04\ R(f + \frac{2}{N-1}) \end{aligned}$$

F.5.3 Windows and Spectra

Figure F.1 shows the windows and their spectra, computed with zero padding. The rectangular window has the narrower main lobe but also the bigger side-lobes. The width of the main lobe and the difference (in dB) between the main lobe and the first secondary lobe are given in table F.1

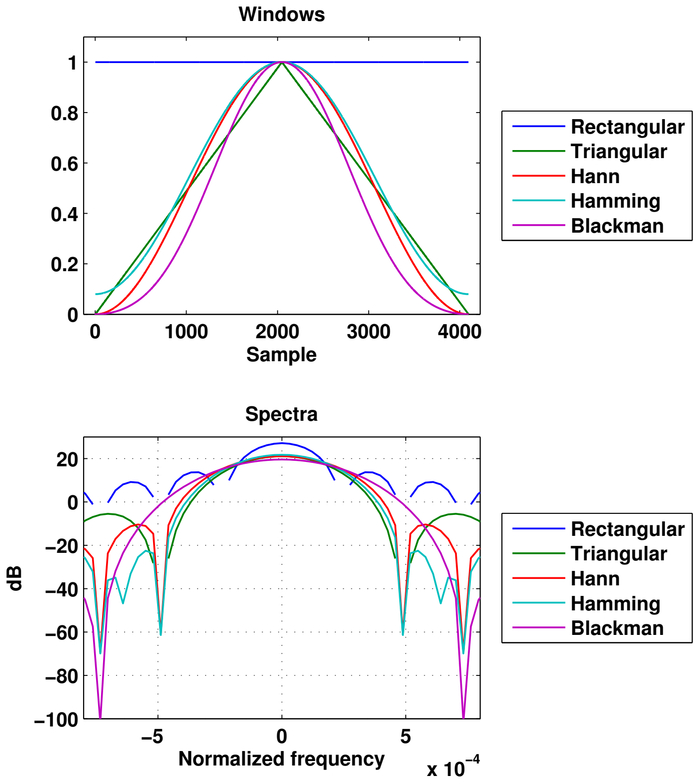


Figure F.1: Windows and Spectra.

Table F.1: Main Lobe Width.

Window	Main Lobe Width	Difference Main Lobe/First Secondary Lobe ($\approx$)
Rectangular	$2/N$	-13 dB
Triangular	$4/N$	-26 dB
Hanning	$4/N$	-31 dB
Hamming	$4/N$	-41 dB
Blackman	$6/N$	-57 dB

F.6 Circulant Matrix

In linear algebra, a circulant matrix is a special kind of Toeplitz matrix where each row vector is rotated one element to the right relative to the preceding row vector. In numerical analysis circulant matrices are important because they are diagonalized by a discrete Fourier transform, and hence linear equations that contain them may be quickly solved using a fast Fourier transform.

$$\mathbf{C} = \begin{bmatrix} c_0 & c_{n-1} & \dots & c_2 & c_1 \\ c_1 & c_0 & c_{n-1} & & c_2 \\ \vdots & c_1 & c_0 & \ddots & \vdots \\ c_{n-2} & & \ddots & \ddots & c_{n-1} \\ c_{n-1} & c_{n-2} & \dots & c_1 & c_0 \end{bmatrix}. \tag{F.16}$$

F.6.1 Circulant Matrix construction

Consider the Discrete Fourier Transform Matrix **F**, defined as:

$$F(p,q) = exp(-2\ i\ \pi\ p\ q\ /\ N_{fft}) \tag{F.17}$$

N_{fft} is the number of points used for the transform and can be greater thant the length of the analysed signal.
As a circulant matrix is equivalent to a circular convolution, its Fourier Transform is equivalent to a component-wise multiplication, this is represented by a diagonal matrix. So we can construct the circulant matrix of a vector **c** as:

$$\mathbf{C}\ =\ \mathbf{F}\ diag(\mathbf{F}\ \mathbf{c})\ \mathbf{F}^{-1} \tag{F.18}$$

In practice the Discrete Fourier Transform (DFT) Matrix is done using the Fast Fourier Transform (FFT) Algorithm. Note only that:

$$\mathbf{F}_{N_{fft}}\ \mathbf{F}^{-1}_{N_{fft}} = \mathbf{I}_{N_{fft}} \tag{F.19}$$

$$\mathbf{F}_{N_{fft}}\ \mathbf{F}^{*}_{N_{fft}} = N_{fft}\ \mathbf{I}_{N_{fft}} \tag{F.20}$$

$$\tag{F.21}$$

F.6.2 Circulant Matrix Properties

F.6.2.1 Product

Giving the circulant matrix construction it is easy to see some evident properties of circulant Matrices. Consider two circulant matrix **A** and **B**, constructed with vector **a** and

b respectively, then the product of this matrices is equal to:

$$\begin{aligned}
\mathbf{AB} &= \mathbf{F}\ diag(\mathbf{F}\ \mathbf{a})\ \mathbf{F}^{-1}\ \mathbf{F}\ diag(\mathbf{F}\ \mathbf{b})\ \mathbf{F}^{-1} && \text{(F.22)}\\
&= \mathbf{F}\ diag(\mathbf{F}\ \mathbf{a})\ diag(\mathbf{F}\ \mathbf{b})\ \mathbf{F}^{-1} && \text{(F.23)}\\
&= \mathbf{F}\ diag(\mathbf{F}\ \mathbf{b})\ diag(\mathbf{F}\ \mathbf{a})\ \mathbf{F}^{-1} && \text{(F.24)}\\
&= \mathbf{BA} && \text{(F.25)}
\end{aligned}$$

As a consequence of this we can permut the order of circulant matrices.

F.6.2.2 Inverse

Consider the inversion of a circulant matrix

$$\begin{aligned}
\mathbf{C}^{-1} &= \left(\mathbf{F}\ diag(\mathbf{F}\ \mathbf{c})\ \mathbf{F}^{-1}\right)^{-1} && \text{(F.26)}\\
&= \mathbf{F}\ diag(\mathbf{F}\ \mathbf{c})^{-1}\ \mathbf{F}^{-1} && \text{(F.27)}\\
\check{\mathbf{c}} &= \mathbf{F}\ \mathbf{c} && \text{(F.28)}\\
\mathbf{C}^{-1} &= \mathbf{F}\ diag(\check{\mathbf{c}})^{-1}\ \mathbf{F}^{-1} && \text{(F.29)}\\
&= \mathbf{F}\ diag\left(\frac{1}{\check{\mathbf{c}}}\right)\ \mathbf{F}^{-1} && \text{(F.30)}\\
& && \text{(F.31)}
\end{aligned}$$

Then the inverse of a circulant **C** can be easily constructed with its generative vector **c**.

F.7 Frame based algorithms Initialization

In the previous Chapter we have introduced two parameters estimation algorithms. The second one, namely the minimization of the Itakura Saito (IS) distance, is incomplete as the Long Term Auto Regressive (AR) model is not adapted during the process. This implies to know it, here we propose a per source initialization using the minimization of IS. The "per source" analyse is done by using a weighted version of the IS distance which leads, in an ideal case, to analyse the source separatly. We hope that the estimation will be sufficently good for giving a robust initialization for the differents parameters involved in the algorithm.

F.7.1 Per Source Weighted Itakura-Saito Distance Minimization

In order to find good initial estimates for the joint approach, we shall consider the minimization of a weighted Itakura-Saito distance for the spectrum of one source k, in which the weighting $C_k(f)$ focuses on the harmonics where the source spectrum is much stronger than that of the rest of the signal. The weighted Itakura Saito distance for source k is:

$$\int df\ C_k(f)\left[\frac{Y_w(f)}{S_k(f;\theta_k)} - 1 - ln\left(\frac{Y_w(f)}{S_k(f;\theta_k)}\right)\right]. \qquad \text{(F.32)}$$

At this point we acknowledge the effect of a window in the frame processing. The effect of a window on the spectrum of a short+long term AR model is roughly equivalent to the effect of the long-term correlation coefficient b. Hence, when the long-term correlation is mainly limited by the window, we shall take b arbitrarily close to 1, but incorporate the effect of the window on the spectrum. The source spectrum becomes a sum of harmonic peaks with a fundamental frequency f_0, convolved with the squared Fourier transform $W(f)$ of the (properly normalized) analysis window w_t :

$$\hat{S}_k(f) \; = \; \sum_n \alpha_n \, W(f \; - \; n \; f_0) \tag{F.33}$$

where the summation range can go up to $\lfloor \frac{0.5}{f_0} \rfloor$ (where f_0 is assumed to be expressed relative to the sampling frequency) or this initial spectral analysis may be limited to a limited frequency range. The spectral peak magnitudes α_n can be seen to be the samples (at frequencies $n \, f_0$) of the spectral envelope which can be modeled as (short-term) AR. The α_n can be estimated by a least-squares fit between $\hat{S}_k(f)$ and $Y_w(f) = \frac{1}{N} \, |y_w(f)|^2$, the periodogram of the windowed signal $w_t y_t$. The spectrum of the other signals in the mixture can be obtained as the residual spectrum $E_k(f) = \max(Y_w(f) - \hat{S}_k(f; \theta_k), \hat{\sigma}^2_v)$. To improve the spectral estimate w.r.t. a simple residual, we floor the residual at the noise level. The (white) noise level can be estimated from the sorted periodogram values $Y_w(f)$ (after some experimenting, we have taken the value at 20% from the minimum).

F.7.2 Pitch Estimation

The estimation of the α_n by a least-squares fit between $\hat{S}_k(f)$ and $Y_w(f)$ mentioned above leads to the α_n estimates as simple (scaled) samples of $\hat{S}_k(f) * \breve{W}(f)$ (convolution) at $f = n \, f_0$. The fundamental frequency estimate is then obtained from

$$\hat{f}_{0,k} \; = \; \arg\max_f \; \int df \, \frac{\hat{S}_k(f)}{E_k(f)} \; . \tag{F.34}$$

In other words, only the spectral peaks of a source that are less perturbed by the rest of the signal mixture are accounted for. The pitch estimation requires an exhaustive search over the useful frequency range. It can be carried out on a limited range of the spectrum. Multiple pitches are obtained if the cost function (F.34) shows multiple maxima.

F.7.3 AR coefficients estimation

An estimate of the short term AR spectral envelope model of source k can be obtained from (F.32) using the following weighting function:

$$C_k(f) \; = \; \frac{Y_w(f)}{E_k(f)} \; . \tag{F.35}$$

This weighting focuses the IS distance on frequencies where a single source model is valid. It is assumed though that the resulting subset of frequencies is sufficient to determine the AR spectral envelope correctly, although the estimation quality of the short-term parameters is less critical than that of the long-term parameters (mainly pitch). Minimizing the weighted IS distance leads to an algorithm similar to the one presented in section 4.6 of Chapter 4 but now both g_k and r_k involve the weighting function.

In the case of an appropriately chosen window (as in chapter 4 section 4.8.4 [178]), the windowing can be expected to dominate the long-term correlation, leading to the following modification of the short-long term AR model

$$S_k(f) = \frac{\sigma_k^2}{|A_k(f)|^2 |B_k(f)|^2} \rightarrow S_k(f) = \frac{\sigma_k^2}{|A_k(f)|^2} \sum_n W(f - n f_{0,k}). \tag{F.36}$$

So in this case the source parameters are limited to $f_{0,k}$, $\mathbf{a}_k$ and σ_k^2.

F.7.4 Multipitch Simulation Example

The multipitch algorithm is tested on a synthetic signal. As, in its formulation, it seems close to the Spectral Sum [128] we compare the detection function of the two estimators in Figure F.2. The results are quite similar, we can observe that the Spectral Sum is smoother and, also, that the octave problem is also presents.

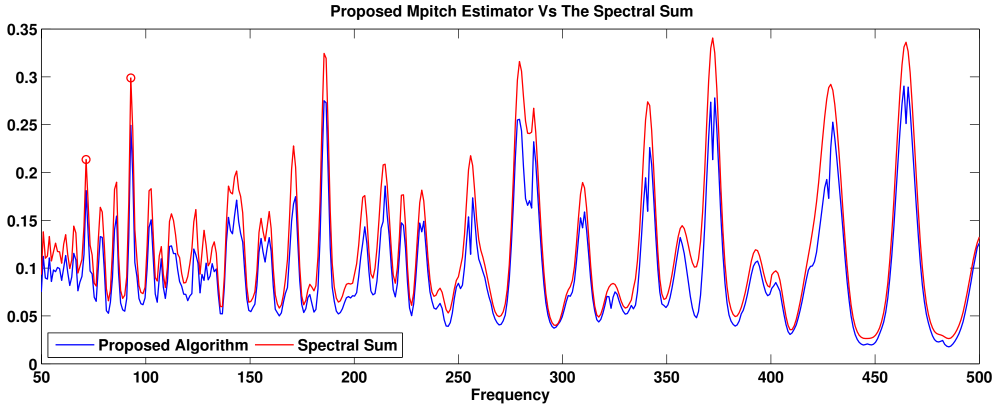


Figure F.2: MultiPitch Algorithm Versus Spectral Sum.

If the number of sources is known, a multidimensional research can also be done. In the case of two sources this leads to search the two frequencies related to the global maximum in a Bi-dimensional function. In Figure F.3 a black circle shows the good maximum, the estimated frequencies are the good one (here the axis are the index, not the frequencies)

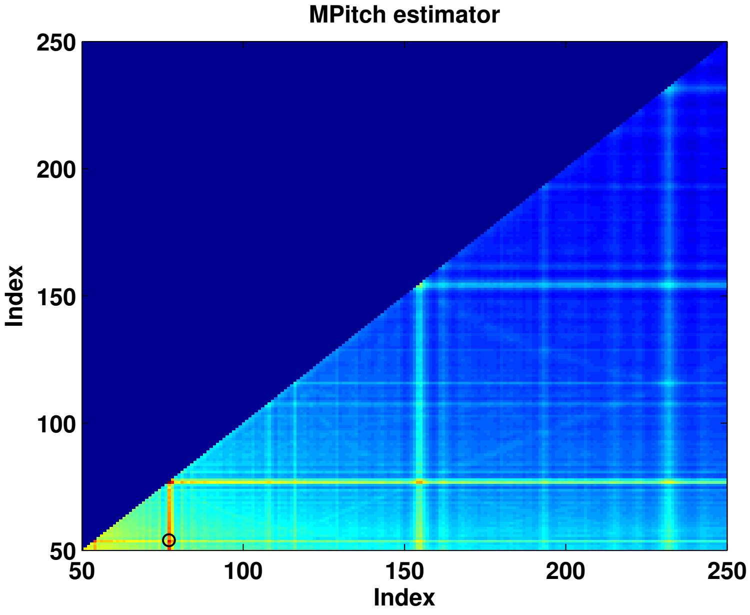


Figure F.3: MultiPitch Bi-Dimensional research.

F.8 Alt-EMK Versus Joint-EMK

In this section we show why we stopped to use the **Joint-EMK**, the analyzed signal is the same than in section 5.3.2.1. We just analyse the first seconde of the signal. As we can observe in Figure F.4 the **Joint-EMK** didn't separate the sources, the two extracted sources are almost the same. They share the observation except just before 0.4 s. During this time the periods were update several times (about 15 times).

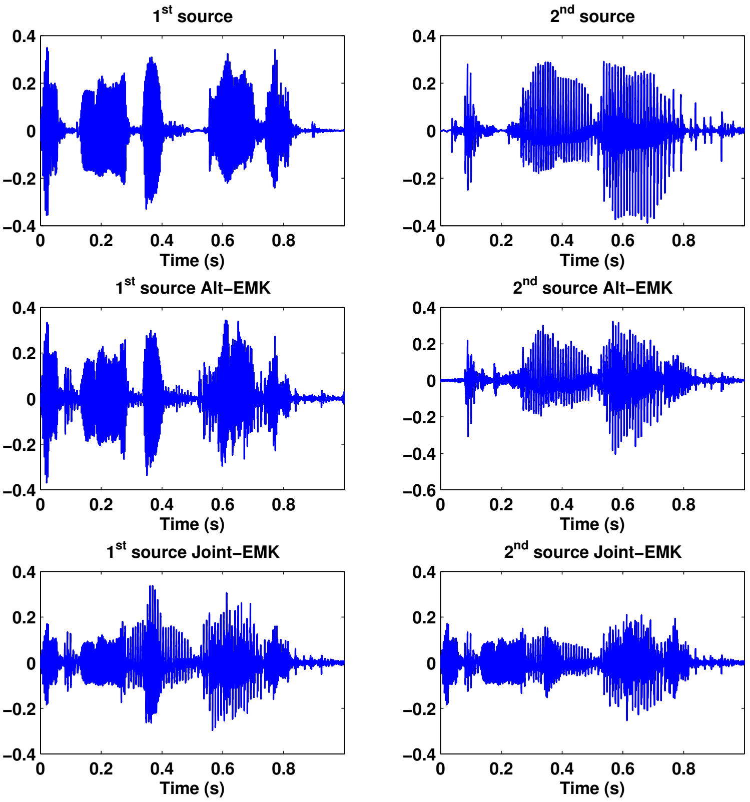


Figure F.4: Comparison **Alt-EMK** and **Joint-EMK** with non-stationary sources.

F.9 Spectral Roll Off

The spectral roll-off point is the frequency bin wich separated the spectrum in two, the first part contains 95% of the signal energy. It is somehow related to a harmonic to noise cutting frequency. we can defined it as:

$$\sum_{n=0}^{f_c} X_n^2 = 0.95 \sum_{n=0}^{Nfft/2} X_n^2 \tag{F.37}$$

where f_c is the cutting frequency (named roll off point), $Nfft$ is the number of FFT points and X_n is the amplitude spectrum at sample n.

F.10 Parameters Used for Sources Separation Simulations

In this section we give the missing parameters used in the simulations.

F.10.1 Simulations of Chapter 2

The STFT shown in Chapter 2 use the following parameters, each segments have a lenght of 1024 samples and are weighted with a Hann Window. 4096 fft points are used in the FFT algorithm and the overlap is about 75%. The signal used is "$female\_inst\_sim\_1.wav$" and is decimated to 8KHz. For the comparison the short term order is fixed to 8. On the Figure the intensity axis are limited from -40dB to 0dB with, as colormap, an opposed to gray scale.

F.10.1.1 Evaluation criteria Versus SNR

The synthetic signals used in this simulation are generated with the following parameters

$$\begin{aligned}
a_1, n &= [1, \ +0.1392, \ -0.4111, \ +0.0390, \ +0.1693, \ -0.0016]^T \\
a_1, n &= [1, \ +0.1178, \ +0.2060, \ -0.2593, \ -0.0199, \ -0.1534]^T \\
b_{1,2} &= [0.92, 0.98]^T \\
\sigma^2_{1,2} &= [0.3772, 0.2461]^T \\
T_{1,2} &= [48.2, 54.6]
\end{aligned}$$

F.10.2 Simulations of Chapter 3

F.10.2.1 Weighted sources

The weight used for the amplitude variations are shown in Figure F.5. For the first source a $1 - Hann$ window is used and for the second one a half Hamming window is applied at the end of the signal.

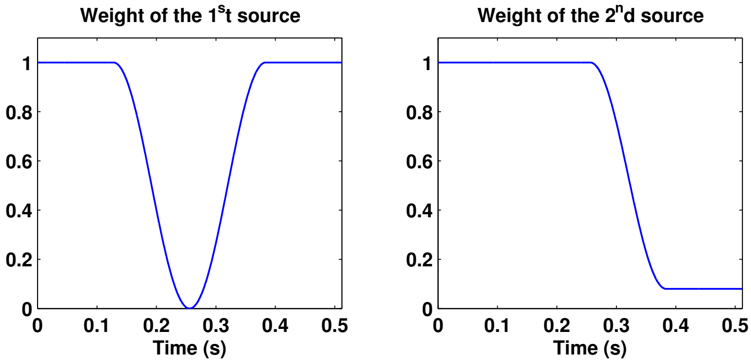


Figure F.5: The weight for each sources.

F.10.2.2 Fondamentals Frequencies variations

For the frequency variation we use a sinusoidal plus noise model [11], the signals are composed of non constant amplitudes harmonics, the intensity was pre-defined with an AR model randomly generated. The phases are also random.

F.10.3 Simulations of Chapter 4

F.10.3.1 LDU decomposition

For the LDU decomposition we use the Cholesky decomposition, that gives the LDU decomposition of matrix **P** as follow:

$$\begin{aligned} \mathbf{L}_1 &= chol(\mathbf{P}) && \text{(F.38)} \\ \mathbf{D}_1 &= diag(diag(\mathbf{L}_1)) && \text{(F.39)} \\ \mathbf{L} &= \mathbf{L}_1^H \ \mathbf{D}_1^{-1} && \text{(F.40)} \\ \mathbf{D} &= \mathbf{D}_1 \ \mathbf{D}_1^H && \text{(F.41)} \\ \mathbf{U} &= \mathbf{L}^H && \text{(F.42)} \end{aligned}$$

F.10.4 Simulations of Chapter 5

The STFT shown in Chapter 5 use the following parameters, each segments have a lenght of 1024 samples and are weighted with a Hann Window. 4096 fft points are used in the FFT algorithm and the overlap is about 75%. The mixture used is composed of "*female_inst_sim_1.wav*" and "*male_inst_sim_1.wav*" which are decimated to 8KHz. For the short duration simulations we use the first 4000 samples of active voice (done manually) of the respective files.

F.11 Noise Variance esimation

In section 3.8.2.2 we have used all the proposed algorithms on synthetic signals. All the parameters are estimated during the process, the last noise variance estimate for each input SNR value and for each algorithms are shown in Figure F.6.

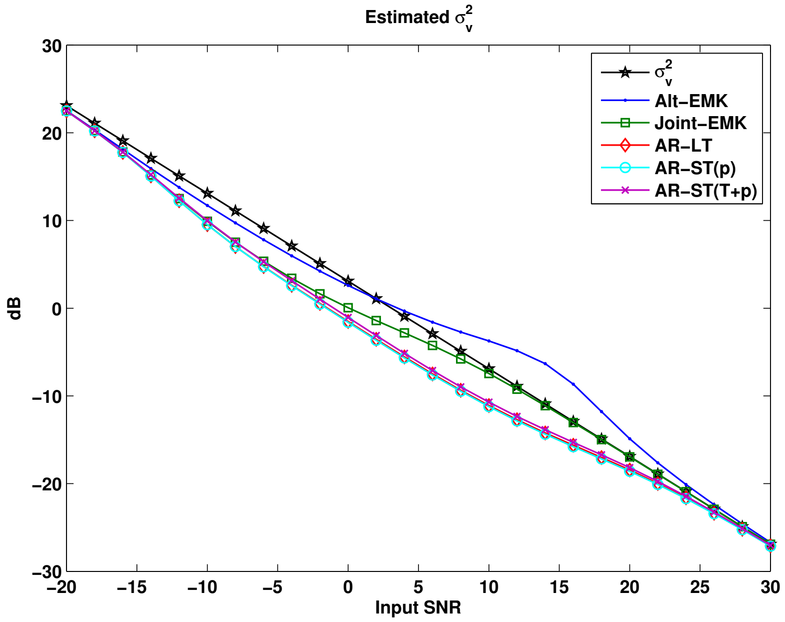


Figure F.6: Estimated Noise Variance for all algorithms.

Bibliography

[1] S. Vrazick and associate, “Sigtone,” *http://www.sigtone.com/*, 2000.

[2] M. Paleari, “Affective computing : display, recognition, and computer synthesis of emotions,” Ph.D. dissertation, Télécom Paris Tech - EURECOM, Paris, France, 10 2009.

[3] O. Gillet, “Transcription des signaux percussifs. application à l’analyse de scènes musicales audiovisuelles,” Ph.D. dissertation, Télécom Paris Tech, Paris, France, June 2007.

[4] J. Cardoso, “Blind signal separation: Statistical principles,” *IEEE Proc., 9(10)*, pp. 2009–2025, 1998.

[5] A. Hyvarinen and E. Oja, “A fast fixed-point algorithm for independent component analysis,” *Neural Computation*, pp. 1483–1492, 1997.

[6] O. Yilmaz and S. Rickard, “Blind separation of speech mixtures via time-frequency masking,” *IEEE Trans. Speech and Audio Processing*, pp. 1830–1847, 2004.

[7] N. Mitianoudis and T. Stathaki, “Overcomplete source separation using laplacian mixture models.” *IEEE Trans. Speech and Audio Processing*, 2005.

[8] C. Févotte and S. J. Godsill, “A bayesian approach to time-frequency based blind source separation,” *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, WASPAA*, 2005.

[9] W. C. Chu, *Speech coding algorithms-foundation and evolution of standardized coders.* John Wiley and Sons, NewYork, 2003.

[10] A. P. Dempster, N. M. Laird, and D. B. Rubin, “Maximum likelihood from incomplete data via the em algorithm,” *Journal of the royal statistical society, series B*, vol. 39, no. 1, pp. 1–38, 1977.

[11] X. Serra, “Musical sound modeling with sinusoids plus noise,” *Musical Signal Processing*, 1997.

[12] P. Comon and C. Jutten, *Handbook of Blind Source Separation: Independent Component Analysis and Applications.* Academic Press, 2010.

[13] M. Malcangi, “Source separation and beat tracking: A system approach to the development of a robust audio-to-score system,” in *Computer Music Modeling and Retrieval*, ser. Lecture Notes in Computer Science, U. K. Wiil, Ed. Springer Berlin / Heidelberg, 2005, vol. 3310, pp. 71–82.

[14] S. Gannot, D. Burshtein, and E. Weinstein, "Iterative and sequential kalman filter-based speech enhancement algorithms," *Speech and Audio Processing, IEEE Transactions on*, vol. 6, no. 4, pp. 373 –385, jul. 1998.

[15] P. Leveau, E. Vincent, G. Richard, and L. Daudet, "Instrument-specific harmonic atoms for mid-level music representation," *IEEE Transactions on Audio, Speech and Language Processing*, 2008.

[16] N. Bertin, "Les factorisations en matrices non-négatives. approches contraintes et probabilistes, application à la transcription automatique de musique polyphonique," Ph.D. dissertation, Télécom Paris Tech, Paris, France, 2009.

[17] V. Emiya, "Transcription automatique de la musique de piano," Ph.D. dissertation, Télécom Paris Tech, Paris, France, 2008.

[18] J. Durrieu, "Automatic transcription and separation of the main melody in polyphonic music signal," Ph.D. dissertation, Télécom Paris Tech, Paris, France, 2010.

[19] E. Vincent, "Musical source separation using time-frequency source priors," *IEEE Transactions on Audio, Speech & Language Processing*, vol. 14, no. 1, pp. 91–98, 2006.

[20] E. Vincent, M. Jafari, S. Abdallah, M. Plumbley, and M. Davies, "Model-based audio source separation," Tech. Rep., 2006.

[21] H. Viste and G. Evangelista, "Sound source separation: Preprocessing for hearing aids and structured audio coding," in *Proceedings of 4th International Conference on Digital Audio Effects (DAFx-01).*

[22] J. Hérault and B. Ans, "Circuits neuronaux à synapses modifiables: décodage de messages composites par apprentissage non supervisé," *C.R. de l'Académie des Sciences*, 1984.

[23] J. Hérault, C. Jutten, and B. Ans, "Détection de grandeurs primitives dans un message composite par une architecture de calcul neuromimétique en apprentissage non supervise," *GRETSI, Groupe d'Etudes du Traitement du Signal et des Images*, 1985.

[24] J. Hérault and C. Jutten, "Space or time adaptive signal processing by neural networks models," *neural networks models*, 1986.

[25] G. Burel, "Blind separation of sources: a nonlinear neural algorithm," *Neural Networks*, vol. 5, pp. 937–947, 1992.

[26] A. Hyvärinen and P. Pajunen, "Nonlinear independent component analysis: Existence and uniqueness results," *Neural Networks*, vol. 12, pp. 429–439, 1999.

[27] P. Comon, "Independent component analysis," *in Proc. Int. Sig. Proc. Workshop on Higher-Order Statistics*, 1991.

[28] ——, "Independent component analysis, a new concept?" *Signal Processing*, vol. 36, pp. 287–314, 1994.

[29] A. Hyvrinen, J. Karhunen, and E. Oja, "Independent component analysis," *Wiley-Interscience*, 2001.

[30] M. Zibulevsky and B. A. Pearlmutter, "Blind source separation by sparse decomposition in a signal dictionary," *Neural Comput*, 2001.

[31] C. Févotte and S. J. Godsill, "A bayesian approach to blind separation of sparse sources," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 14, no. 6, pp. 2174–2188, Nov. 2006.

[32] A. Jourjine, S. Rickard, and zgr Yilmaz, "Blind separation of disjoint orthogonal signals: Demixing n sources from 2 mixtures," in *In IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP00)*, 2000, pp. 2985–2988.

[33] S. Rickard, R. Balan, and J. Rosca, "Real-time time-frequency based blind source separation," in *in Proc. of International Conference on Independent Component Analysis and Signal Separation (ICA2001)*, 2001, pp. 651–656.

[34] M. Cooke and T. Lee, "Speech separation challenge," *http://www.dcs.shef.ac.uk/ martin/SpeechSeparationChallenge.htm*, 2006.

[35] E. Vincent, "Stereo audio source separation evaluation campaign," *http://www.irisa.fr/metiss/SASSEC07/?show=intro*, 2007.

[36] D. Barry and B. Lawler, "Sound source separation: Azimuth discrimination and resynthesis," *DAFX*, 2004.

[37] D. Barry, B. Lawler, and E. Coyle, "Real-time sound source separation: Azimuth discrimination and resynthesis," *AES*, 2004.

[38] S. Kurita, H. Saruwatari, S. Kajita, K. Takeda, and F. Itakura, "Evaluation of blind signal separation method using directivity pattern under reverberant conditions," *ICASSP*, 2000.

[39] M. Ikram and D. Morgan, "A beamforming approach to permutation alignment for multichannel frequency-domain blind speech separation," *ICASSP*, 2002.

[40] N. Murata, S. Ikeda, and A. Ziehe, "An approach to blind source separation based on temporal structure of speech signals," *NeuroComputer*, 2001.

[41] S. Winter, H. Sawada, S. Araki, and S. Makino, "Overcomplete bss for convolutive mixture based on hierarchical clustering," *SAPA*, 2004.

[42] A. Ozerov, P. Philippe, F. Bimbot, and R. Gribonval, "Adaptation of bayesian models for single-channel source separation and its application to voice/music separation in popular songs," *IEEE Transactions on Audio, Speech Language Processing*, pp. 1564–1578, 2007.

[43] S. Roweis, "Factorial models and refiltering for speech separation and denoising," in *In EUROSPEECH*, 2003, pp. 1009–1012.

[44] L. Benaroya and F. Bimbot, "Wiener based source separation with HMM/GMM using a single sensor," in *ica03*, Nara, Japan, apr 2003, pp. 957–961.

[45] L. Benaroya, F. Bimbot, and R. Gribonval, “Audio source separation with a single sensor,” *IEEE Transactions on Audio, Speech Language Processing*, pp. 191–199, 2006.

[46] S. T. Roweis, “One microphone source separation,” in *In Advances in Neural Information Processing Systems 13.* MIT Press, 2000, pp. 793–799.

[47] D. D. Lee and H. S. Seung, “Algorithms for non-negative matrix factorization,” in *NIPS*, 2000, pp. 556–562.

[48] H. Lantéri, C. Theys, C. Richard, and C. Févotte, “Split gradient method for non-negative matrix factorization,” in *Proc. 18th European Signal Processing Conference (EUSIPCO’10)*, Aalborg, Denmark, Aug. 2010.

[49] H. Lantéri, M. Roche, O. Cuevas, , and C. Aime., “A general method to devise maximum likelihood signal restoration multiplicative algorithms with non-negativity constraints. signal processing,” in *Signal Processing*, 2001.

[50] E. Aristidi, F. Vakili, L. Abe, A. Belu, B. Lopez, H. Lantéri, A. Schutz, and J. Menut, “Iran: interferometric remapped array nulling,” in *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, W. Traub, Ed., vol. 5491, 2004.

[51] E. Aristidi, F. Vakili, A. Schutz, H. Lantéri, L. Abe, A. Belu, P.-M. Gori, O. Lardiere, B. Lopez, J. Menut, and F. Patru, “Iran: Interferometric remapped array nulling,” in *in Astronomy with High Contrast Imaging III, EAS Publications Series*, vol. 22, 2006, pp. 103–107.

[52] R. McAulay and T. Quatieri, “Speech analysis/synthesis based on a sinusoidal representation,” *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 34, no. 4, pp. 744 – 754, aug. 1986.

[53] M. Abe and J. O. Smith, “Design criteria for the quadratically interpolated fft method,” *ICASSP*, 2005.

[54] F. Keiler and S. Marchand, “Survey on extraction of sinusoids in stationary sounds,” *DAFX*, 2002.

[55] A. Schutz and D. T. M. Slock, “Modèle sinusoïdale: Estimation de la qualité de jeu dun musicien, détection de certains effets dinterprétation,” in *GRETSI 2007, 21eme colloque traitement du signal et des images, September 11-14, 2007, Troyes, France*, 09 2007.

[56] A. P. Klapuri, “Multipitch estimation and sound separation by the spectral smoothness principle,” 2001, pp. 3381–3384.

[57] S. Mallat and Z. Zhang, “Matching pursuits with time frequency dictionaries,” *IEEE Transaction on Signal Processing*, vol. 41, pp. 3397–3415, 1993.

[58] M. Triki and D. T. M. Slock, “Music source separation via sparsified dictionaries vs. parametric models,” in *ISCCP 2006, International Symposium on Communications, Control, and Signal Processing , March 13-15 2006, Marrakech, Morocco*, 03.

[59] A. de Cheveigné, “Separation of concurrent harmonic sounds: Fundamental frequency estimation and a time-domain cancellation model for auditory processing,” *Journal of the Acoustical Society of America*, vol. 6, pp. 3271–3290, 1993.

[60] H.-M. Lehtonen, V. Välimäki, and T. I. Laakso, “Canceling and selecting partials from musical tones using fractional-delay filters,” *Comput. Music J.*, vol. 32, no. 2, pp. 43–56, 2008.

[61] E. L. Carpentier and C. Févotte, “Séparation de sources autorégressives gaussiennes par maximum de vraisemblance et filtrage de kalman,” in *Proc. 18e colloque GRETSI sur le Traitement du Signal et des Images*, Toulouse, France, Sep. 2001.

[62] R. J. Weiss, “Underdetermined source separation using speaker subspace models,” Ph.D. dissertation, Department of Electrical Engineering, Columbia University, 2009.

[63] L. Benaroya, “Séparation de plusieurs sources sonores avec un seul microphone,” Ph.D. dissertation, Université de Rennes 1, 2003.

[64] A. Ozerov, “Adaptation de modèles statistiques pour la séparation de sources monocapteur. application à la séparation voix / musique dans les chansons,” Ph.D. dissertation, Université de Rennes 1, 2006.

[65] T. Virtanen, “Sound source separation in monaural music signals,” Ph.D. dissertation, 2006.

[66] M. Triki, “Some contributions to statistical signal processing and applications to audio enhancement and mobile localization,” Ph.D. dissertation, Télécom Paris Tech - EURECOM, Paris, France, 2007.

[67] J. Makhoul, *Linear prediction: A tutorial review*, 1975.

[68] P. Kabal and R. Ramachadran, “Joint optimization of linear predictors in speech coders,” *IEEE Trans. Speech and signal Processing*, 1989.

[69] D. Giacobello, M. G. Christensen, J. Dahl, S. H. Jensen, and M. Moonen, “Joint estimation of short-term and long-term predictors in speech coders,” in *ICASSP ’09: Proceedings of the 2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2009, pp. 4109–4112.

[70] C. Couvreur and Y. Bresler, “Decomposition of a mixture of gaussian ar processes,” *Proc. IEEE Int. Conf. Acous. Speech Signal Process*, 1995.

[71] E. Weinstein, A. V. Oppenheim, M. Feder, and J. R. Buck, “Iterative and sequential algorithms for multisensor signal enhancement,” *IEEE Trans. Signal Proc.*, vol. 42(4), 1994.

[72] W. Gao and J. Lehnert, “Diversity combining for ds/ss systems with time-varying, correlated fading branches,” *IEEE Transactions on Communications*, vol. 51, pp. 284–295, 2003.

[73] N. J. Higham, *Accuracy and Stability of Numerical Algorithms.* Philadelphia, PA: Society for Industrial and Applied Mathematics., 2002.

[74] E. Vincent, C. Févotte, and R. Gribonval, "Performance measurement in blind audio source separation," *IEEE Trans. Audio, Speech and Language Processing*, vol. 14(4), pp. 1462–1469, 2006.

[75] F. Itakura and S. Saito, "Analysis synthesis telephony based on the maximum likelihood method," *Proc. 6th International Congress on Acoustics*, p. C17C20, 1968.

[76] Yo-Yoma, "official website," *http://www.yo-yoma.com/.*

[77] "Audionamix," *http://audionamix.com/Vuvuzela/index.html.*

[78] P. Kabal and R. P. Ramachadran, "Joint Optimization of Linear Predictors in Speech Coders," in *IEEE Transaction on acoustics speech and signal processing*, 1989.

[79] C. Févotte, B. Torrésani, L. Daudet, and S. J. Godsill, "Sparse linear regression with structured priors and application to denoising of musical audio," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 16, no. 1, pp. 174–185, Jan. 2008.

[80] R. Gribonval, L. Benaroya, E. Vincent, and C. Févotte, "Proposals for performance measurement in source separation," *Proc. Int. Conf. on Independent Component Analysis and Blind Source Separation (ICA)*, pp. 763–768, 2003.

[81] C. Févotte, R. Gribonval, and E. Vincent, "Bss eval toolbox user guide, revision 2.0," *IRISA Technical Report* 1706, vol. 14(4), 2005.

[82] L. Benaroya, R. Blouet, C. Févotte, and I. Cohen, "Single sensor source separation using multiple-window stft representation," in *Proc. International Workshop on Acoustic Echo and Noise Control (IWAENC'06)*, Paris, France, Sep. 2006.

[83] R. Blouet, G. Rapaport, I. Cohen, and C. Févotte, "Evaluation of several strategies for single sensor speech/music separation," in *Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP'08)*, Las Vegas, USA, Apr. 2008, pp. 37–40.

[84] A. Ozerov, C. Févotte, and M. Charbit, "Factorial scaled hidden markov model for polyphonic audio representation and source separation," in *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA'09)*, Mohonk, NY, USA, Oct. 2009.

[85] A. Pash, "How to silence vuvuzela horns in world cup broadcasts," *Lifehacker. Gawker Media, http://lifehacker.com/5564085/how-to-silence-vuvuzela-horns-with-an-eq-filter.*, 2010.

[86] J.-F. Cardoso, "Source separation using higher order moments," in *in Proc. ICASSP*, 1989, pp. 2109–2112.

[87] A. Belouchrani, K. Abed-meraim, J.-F. Cardoso, and E. Moulines, "A blind source separation technique using second order statistics," 1997.

[88] A. M. Bronstein, M. M. Bronstein, and M. Zibulevsky, *Blind Source Separation: Biomedical applications.* Wiley Encyclopedia of Biomedical Engineering, 2005.

[89] L. D. Lathauwer, B. D. Moor, and J. Vandewalle, "Fetal electrocardiogram extraction by blind source subspace separation," *IEEE Trans. Biomed. Eng*, vol. 47, pp. 567–572, 2000.

[90] S. Bozonnet, N. W. D. Evans, X. Anguera, O. Vinyals, G. Friedland, and C. Fredouille, "System output combination for improved speaker diarization," in *Interspeech 2010, September 26-30, Makuhari, Japan*, 09 2010.

[91] H. Yang, "On-line blind equalization via on-line blind separation," *Signal Processing*, vol. 68, pp. 271–281, 1998.

[92] P. Comon, "Analyse en composantes indépendantes et identification aveugle," *Traitement du Signal*, vol. 7, 1990.

[93] C. Jutten, "Calcul neuromimétique et traitement du signal : analyse en composantes indépendantes," *Thèse détat és sciences physiques*, 1987.

[94] C. Févotte, "Itakura-saito nonnegative factorizations of the power spectrogram for music signal decomposition," in *Machine Audition: Principles, Algorithms and Systems*, W. Wang, Ed. IGI Global Press, Aug. 2010, ch. 11.

[95] D. F. Rosenthal and H. G. Okuno, "Computational auditory scene analysis," *LEA Publishers, Mahwah NJ*, 1998.

[96] M. Mandel, D. Ellis, and T. Jebara, "An em algorithm for localizing multiple sound sources using variational em," *EUSIPCO*, 2005.

[97] L. Benaroya, L. M. Donagh, F. Bimbot, and R. Gribonval, "Non negative sparse representation for wiener based source separation with a single sensor," in *in Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP03)*, 2003, pp. 613–616.

[98] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, October 1999.

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

[100] E. Vincent and M. Plumbley, "Low bit-rate object coding of musical audio using bayesian harmonic models," pp. 1273 –1282, may 2007.

[101] P. Smaragdis, "Convolutive speech bases and their application to supervised speech separation," in *the IEEE Trans. on Speech and Audio Processing*, 2007.

[102] R. Badeau, G. Richard, and B. David, "Fast adaptive esprit algorithm," *IEEE Workshop on Statistical Signal Processing SSP'05*, 2005.

[103] R. Badeau, "Methodes a haute résolution pour l'éstimation et le suivi de sinusoides modulées," Ph.D. dissertation, Télécom Paris Tech, Paris, France, 2005.

[104] C. Yeh, "Multiple fundamental frequency estimation of polyphonic recordings," *Doctoral Thesis, Univ. Paris 6 - UPMC*, 2008.

[105] G. Meurisse, P. Hanna, and S. Marchand, "A New Analysis Method for Sinusoids+Noise Spectral Models," in *A New Analysis Method for Sinusoids+Noise Spectral Models Proceedings of the Digital Audio Effects (DAFx06) Conference*, Canada, 09 2006, pp. 139–144.

[106] L. Omlor and M. A. Giese, "Blind source separation for over-determined delayed mixtures." in *NIPS'06*, 2006, pp. 1049–1056.

[107] M. Joho, H. Mathis, and R. H. Lambert, "Overdetermined blind source separation: Using more sensors than source signals in a noisy mixture," in *in Proc. International Conference on Independent Component Analysis and Blind Signal Separation*, 2000, pp. 81–86.

[108] T. Virtanen and A. Klapuri, "Separation of harmonic sound sources using sinusoidal modeling," vol. 2, 2000, pp. 765 – 768.

[109] A. S. Bregman, *Auditory scene analysis: The perceptual organization of sound.* Cambridge, MA, USA: MIT Press, 1990.

[110] S. Godsill and M. Davy, "Bayesian harmonic models for musical pitch estimation and analysis," Tech. Rep., 2002.

[111] M. Goto, "A real-time music-scene-description system: Predominant-f0 estimation for detecting melody and bass lines in real-world audio signals," *Speech Communication (ISCA Journal)*, vol. 43, pp. 311–329, 2004.

[112] T. Virtanen and A. Klapuri, "Separation of harmonic sounds using linear models for the overtone series," in *Series, Proc. ICASSP2002*, 2002, pp. 1757–1760.

[113] ——, "Separation of harmonic sounds using multipitch analysis and iterative parameter estimation," 2001, pp. 83 –86.

[114] R. Gribonval and E. Bacry, "Harmonic decomposition of audio signal with matching pursuit," *IEEE Transaction on Signal Processing*, vol. 51, 2003.

[115] R. Balan, A. Jourjine, , and J. Rosca, "Ar process and sources can be reconstructed from degenerate mixtures," in *in Proc. of International Conference on Independent Component Analysis and Signal Separation*, 1999, pp. 467–472.

[116] Proakis, John G. and Manolakis, Dimitris K., *Digital Signal Processing (3rd Edition).* Prentice Hall, March 2006.

[117] N. H. Fletcher and T. D. Rossing, "The physics of musical instruments," *Springer Verlag*, 1991.

[118] M. Feder and E. Weinstein, "Parameter estimation of superimposed signals using the em algorithm," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 36, pp. 477–489, 1988.

[119] M. Beal, "Variational algorithms for approximate bayesian inference," *Ph.D. Thesis, Gatsby Computational Neuroscience Unit, Univ. College London*, 2003.

[120] E. A. Wan and A. T. Nelson, "Neural dual extended kalman filtering: Applications in speech enhancement and monaural blind signal separation," in *in Proc. of IEEE Workshop on Neural Networks and Signal Processing*, 1997, pp. 466–475.

[121] R. K. Olsson and L. K. Hansen, "Linear state-space models for blind source separation," *J. Mach. Learn. Res.*, vol. 7, pp. 2585–2602, 2006.

[122] A. V. Oppenheim and R. W. Schafer, "Discrete-time signal processing," *Prentice-Hall*, pp. 447–448, 1989.

[123] D. G. Tzikas, A. C. Likas, and N. P. Galatsanos, "The Variational Approximation for Bayesian Inference, Life After the EM Algorithm," *IEEE Signal Processing Magazine*, Nov. 2008.

[124] J. A. Fessler and A. O. Hero, "Space-alternating generalized expectation-maximization algorithm," *IEEE Trans. Signal Processing*, vol. 42, pp. 2664–2677, 1994.

[125] B. Hu, I. Land, R. Piton, and B. Fleury, "A bayesian framework for iterative channel estimation and multiuser decoding in coded ds-cdma," in *Global Telecommunications Conference, 2007. GLOBECOM '07. IEEE*, Nov. 2007, pp. 1582–1586.

[126] L. M. Bregman, "The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming," *USSR Computational Mathematics and Mathematical Physics*, vol. 7, pp. 200–217, 1967.

[127] R. Gray, A. Buzo, J. Gray, A., and Y. Matsuyama, "Distortion measures for speech processing," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 28, no. 4, pp. 367 – 376, aug. 1980.

[128] D. J. Hermes, "Measurement of pitch by subharmonic summation," *J. Acoust. Soc. Am.*, vol. 83, pp. 257–264, 1988.

[129] J. Moorer, "On the segmentation and analysis of continuous musical sound by digital computer," *Dept. of Music, Stanford University*, 1975.

[130] R. Maher, "A approach for the separation of voices in composite musical signals," *Doctoral Thesis, Univ. of Illinois at Urbana-Champaign*, 1989.

[131] D. Mellinger, "Event formation and separation in musical sound," *Doctoral Thesis, Stanford Univ.*, 1991.

[132] L. Rossi, "Identification de sons polyphoniques de piano," *Doctoral Thesis, Université de Corse*, 1998.

[133] D. Godsmark, "A computational model of the perceptual organisation of polyphonic music," *Doctoral Thesis, Univ. of Sheffield*, 1998.

[134] A. Sterian, "Model-based segmentation of time-frequency images for musical transcription," *Doctoral Thesis, Univ. of Michigan*, 1999.

[135] M. Marolt, "Transcription of polyphonic solo piano music," *Doctoral Thesis, Univ. of Ljubljana, Slovenia,*, 2002.

[136] L. I. Ortiz-Berenguer, "Identificatin automtica de acordes musicales," *Doctoral Thesis, Universidad Politécnica de Madrid*, 2002.

[137] J. Bello, "Towards the automated analysis of simple polyphonic music : A knowledgebased approach," *Doctoral Thesis, Univ. of London*, 2003.

[138] A. Klapuri, "Signal processing methods for the automatic transcription of music," Ph.D. dissertation, 2004.

[139] E. Vincent, "Modèles dinstruments pour la séparation de sources et la transcription denregistrements musicaux," *Doctoral Thesis, Univ. Paris 6 - UPMC*, 2004.

[140] A. Cemgil, "Bayesian music transcription," *Doctoral Thesis, SNN, Radboud University Nijmegen, the Netherlands*, 2004.

[141] A. Camacho, "Swipe : a sawtooth waveform inspired pitch estimator for speech and music," *Doctoral Thesis, Univ. of Florida*, 2007.

[142] H. Kameoka, "Statistical approach to multipitch analysis," *Doctoral Thesis, Univ. of Tokyo*, 2007.

[143] A. Klapuri and M. Davy, "Signal processing methods for music transcription," *Springer*, 2006.

[144] D. Wang and G. Brown, "Computational auditory scene analysis : principles, algorithms, and applications," *IEEE Press/Wiley-Interscience*, 2006.

[145] K. Kreitner, "Ornaments," *http://www.grovemusic.com*, 2006.

[146] S. Rossignol, X. Rodet, P. Depalle, J. Soumagne, and J. Collette, "Vibrato: detection, estimation, extraction, modification," *Conference on Digital Audio Effects (DAFx), Trondheim, Norvège*, 1999.

[147] C. Traube and J. O. Smith, "Estimating the plucking point position on a guitar string," *Proc. Of the Int. Conference on Digital Audio Effects (DAFx-00)*, 2000.

[148] M. Gainza, E. Coyl, and R. Lawlor, "Single-note ornaments transcription for irish tin whistle based on onset detection," *Proc. Of the Int. Conference on Digital Audio Effects (DAFx-04)*, 2004.

[149] J. Brown and P. Smaragdis, "Independent component analysis for automatic note extraction from musical trills," *JASA*, 2004.

[150] M. Gainza and E. Coyle, "Automating ornamentation transcription," *ICASSP*, 2007.

[151] R. Althoff, R. Keiler, and U. Zolzer, "Extracting sinusoids from harmonic signals," *DAFX*, 1999.

[152] M. Alonso, R. Badeau, B. David, and G. Richard, "Musical tempo estimation using noise subspace projection," *WASPAA*, 2003.

[153] A. Schutz and D. T. M. Slock, "Estimation of the parameters of sinusoidal signal components and of associated perceptual musical interpretation effects," in *Jamboree 2007: Workshop By and For KSpace PhD Students, September, 14th 2007, Berlin, Germany*, 09 2007.

[154] ——, “Toward the detection of interpretation effects and playing defects,” in *DSP 2009, 16th International Conference on Digital Signal Processing, July 05-07, 2009, Santorini, Greece*, 07 2009.

[155] X. Serra, “Musical sound modeling with sinusoids plus noise,” in *Musical Signal Processing*, Lisse, the Netherlands, 1997, pp. 91–122.

[156] J. S. Marques and L. B. Almeida, “A background for sinusoid based representation of voiced speech,” *acoustics, Speech and Signal Processing*, 1986.

[157] D. C. Rife and R. Boorstyn, “Single tone parameter estimation from discrete time observation,” *IEEE Transaction on information theory*, vol. IT-20, no. 5, Sept 1974.

[158] S. Marchand and M. Lagrange, “On the equivalence of phase-based methods for the estimation of instantaneous frequency,” *In Proceedings of the 14th European Conference on Signal Processing*, Sept 2006.

[159] F. Auger and P.Flandrin, “Improving the readibility of time frequency and time scale representations by the reassignement method,” *IEEE Transaction on Signal Processing*, vol. 43, no. 5, pp. 1068–1089, 1995.

[160] C. Févotte, B. Torresani, L. Daudet, and S. J. Godsill, “A perceptually motivated multiple-f0 estimation method,” *IEEE Trans. Audio, Speech and Language Processing, vol. 16, no 1, pp. 174-185, Jan.*, 2008.

[161] M. R. Schroeder, “Period histogram and product spectrum: new methods for fundamental frequency measurement,” *J. Acoust. Soc. Am.*, vol. 43, pp. 829–834, 1968.

[162] X. Sun, “A pitch determination algorithm based on subharmonic-to-harmonic ratio,” *the 6th International Conference of Spoken Language Processing*, 2000.

[163] G. Peeters, “A large set of audio features for sound description (similarity and classification) in the CUIDADO project,” IRCAM, Tech. Rep., 2004.

[164] A. Schutz, N. Bertin, D. Slock, B. David, and R. Badeau, “Piano forte pedal analysis and detection,” *AES124*, 2008.

[165] I. Barbancho, A. Barbancho, A. Jurado, and L. Tardon, “Transcription of piano recordings,” *Applied Acoustics*, 2004.

[166] H. M. Lehtonen, H. Penttinen, J. Rauhala, and V. Valimaki, “Analysis and modeling of piano sustain-pedal effects,” *JASA*, 2007.

[167] H. M. Lehtonen, “Analysis and parametric synthesis of the piano sound,” *Master thesis, Helsinski University of Technology, Finland*, 2005.

[168] B. David and R. Badeau, “Fast sequential ls estimation for sinusoidal modeling and decomposition of audio signals,” *2007 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics WASPAA2007, New Paltz, New York, USAoct. 2007, pp. 211-214.*

[169] Y.-R. Chien and S.-K. Jeng, “An automatic transcription system with octave detection,” in *Acoustics, Speech, and Signal Processing, 2002. Proceedings. (ICASSP ’02). IEEE International Conference on*, vol. 2, 2002, pp. 1865–1868.

[170] A. Schutz and D. T. M. Slock, “Periodic signal modeling for the octave problem in music transcription,” in *DSP 2009, 16th International Conference on Digital Signal Processing, July 05-07, 2009, Santorini, Greece*, 07 2009.

[171] A. Burns and M. M. Wanderley, “Visual methods for the retrieval of guitarist fingering,” in *NIME ’06: Proceedings of the 2006 conference on New interfaces for musical expression*, Paris, France, 2006, pp. 196–199.

[172] J. A. Verner, “Midi Guitar Synthesis: Yesterday, Today and Tomorrow,” *Recording Magazine*, vol. 8 (9), pp. 52–57, 1995.

[173] C. Traube, “A Interdisciplinary Study of the Timbre of th Classical Guitar,” Ph.D. dissertation, McGill University, 2004.

[174] D. P. Radicioni, L. Anselma, and V. Lombardo, “A Segmentation-Based Prototype to Compute String Instruments Fingering,” in *CIM04: Proceedings of the 1st Conference on Interdisciplinary Musicology*, Graz, Austria, 2004.

[175] B. Zhang, J. Zhu, Y. Wang, and W. K. Leow, “Visual Analysis of Fingering for Pedagogical Violing Transcription,” in *MM ’07: Proceedings of the 15th international conference on Multimedia*, Augsburg, Germany, 2007, pp. 521 – 524.

[176] M. Paleari, B. Huet, A. Schutz, and D. T. M. Slock, “A multimodal approach to music transcription,” in *1st ICIP Workshop on Multimedia Information Retrieval : New Trends and Challenges, October 12-15, 2008, San Diego, USA*, 10 2008.

[177] ——, “Audio-visual guitar transcription,” in *Jamboree 2008, Workshop By and For KSpace PhD Students, July, 25 2008, Paris, France*, 07 2008.

[178] A. Schutz and D. T. M. Slock, “Single-microphone blind audio source separation via Gaussian Short+Long Term AR Models,” in *ISCCSP 2010, 4th International Symposium on Communications, Control and Signal Processing, March 3-5, 2010, Limassol, Cyprus*, 03 2010.