HABILITATION À DIRIGER DES RECHERCHES

PRÉSENTÉE À

L'UNIVERSITÉ BORDEAUX 1

ÉCOLE DOCTORALE DE MATHÉMATIQUES ET D'INFORMATIQUE

par **Sylvain Marchand**

SPÉCIALITÉ: INFORMATIQUE

Avancées en modélisation spectrale du son musical (Advances in Spectral Modeling of Musical Sound)

Soutenue le: 5 décembre 2008

Après avis de:

MM.	Professeur Philippe Depalle ..... (McGill University, Montréal, Canada)	Rapporteurs
	Professeur Julius O. Smith III ..... (CCRMA, Stanford University, USA)	
	Professeur Udo Zölzer (Helmut Schmidt University, Hambourg, Allemagne)	

Devant la Commission d'examen formée de:

Mme	Jenny Benois-Pineau, Professeur ..... (LaBRI, Université Bordeaux 1)	Présidente et Rapporteur
Mme	Myriam Desainte-Catherine, Professeur (LaBRI, Université Bordeaux 1)	Examinateurs
MM.	Philippe Depalle, Professeur ..... (McGill University, Montréal)	
	Laurent Daudet, MdC HDR ..... (Université Pierre et Marie Curie, Paris)	
	Laurent Girin, Professeur ..... (Université Stendhal, Grenoble)	
	Gaël Richard, Professeur ..... (TELECOM ParisTech)	
	Robert Strandh, Professeur ..... (LaBRI, Université Bordeaux 1)	
	Udo Zölzer, Professeur ..... (Helmut Schmidt University, Hambourg)	

– 2008 –

This document was entirely written using LaTeX.

Contents

Introduction (version française)

Des battements du cœur aux cycles des planètes, notre univers est empli de phénomènes quasi-périodiques. C'est également vrai pour les sons musicaux, mon univers professionnel depuis le doctorat. Le présent manuscrit décrit ma recherche en modélisation informatique du son musical depuis cette thèse de doctorat, et peut ainsi être vu comme une suite logique au mémoire de thèse. Il décrit nos travaux avec les doctorants co-encadrés (Mathieu Lagrange, Matthias Robine, Martin Raspaud, et actuellement Joan Mouba et Guillaume Meurisse).

Nos contributions apparaissent dans le texte sous la forme de références numériques [1–54] (et sont listées dans la Section A.5) tandis que les autres références bibliographiques sont alphanumériques. Ces contributions sont programmées dans nos logiciels InSpect ("In(spect) Spect(rum)", voir Figure 1) et ReSpect ("Re(spect) Spect(rum)", voir Figure 2), logiciels conçus pour l'analyse / transformation / synthèse des sons spectraux, et initiés au début de ma thèse de doctorat (voir [14]).

Partie I. La première partie de ce manuscrit est une introduction au son musical, pluridisciplinaire par nature. Les notions de mathématiques, physique, traitement du signal, acoustique, psychoacoustique et musique y sont limitées au strict nécessaire pour la compréhension du reste de ce document. Ainsi, le lecteur familier avec ces notions peut sauter cette première partie. Toutefois, indiquons que nous y introduisons également des notations utiles pour la suite.

Le **Chapitre 1** commence par la représentation temporelle des signaux sonores, dans les cas continu et discret, et avec une une brève introduction à la théorie et à la pratique de l'échantillonnage et de la reconstruction. Une contribution avec Martin Raspaud [43] consiste d'ailleurs en une technique de reconstruction / ré-échantillonnage améliorée pour les signaux non centrés en zéro, ce qui est le cas des paramètres des modèles sinusoïdaux. La représentation spectrale est également introduite, avec la transformée de Fourier (et son inverse) dans les cas continu et discret. La dualité fondamentale entre les domaines temporel et spectral est illustrée pour une variété d'opérations et de fonctions. Cette dualité jouera un rôle clé dans la suite de ce document. En effet, un problème pouvant sembler difficile dans un domaine (temporel ou spectral) peut admettre une solution élégante dans l'autre domaine (spectral ou temporel). Après ces bases mathématiques et physiques pour le son numérique, sont brièvement décrites la production, la propagation et la réception (perception) du son. Des notions d'acoustique sont introduites, telles le délai et l'atténuation du son du fait de la propagation de l'onde acoustique dans l'air. D'importantes notions de psychoacoustique sont également présentées, comme des échelles perceptives, les seuils d'audibilité et les phénomènes de masquage utilisés pour la synthèse additive rapide (Section 3.3.3) ou le tatouage (Section 8.2).

Le **Chapitre 2** présente des éléments de musique, des notions d'acoustique musicale aux paramètres musicaux (hauteur, intensité et timbre) et leurs variations dans le temps (vibrato, trémolo). À nouveau, nous nous limitons dans ce chapitre aux notions strictement nécessaires à la bonne compréhension de la suite du document.

Partie II. La deuxième partie de ce manuscrit traite de mon principal sujet de recherche : la modélisation sinusoïdale des sons musicaux.

Le **Chapitre 3** présente la représentation sinusoïdale à court terme, avec les méthodes d'analyse et de synthèse associées. Côté modèle, nous nous intéressons aux *pics* spectraux et, contrairement à la thèse, nous considérons maintenant pleinement le cas non stationnaire, où les paramètres peuvent varier à l'intérieur même d'une petite fenêtre temporelle.

Le module d'extraction de pics d'InSpect implémente de nombreuses méthodes d'analyse à court terme. Trois principales méthodes d'analyse non stationnaire sont décrites dans ce document : l'interpolation quadratique, la réallocation spectrale et notre contribution – la méthode de la dérivée (Section 3.2.3.3) proposée initialement avec Myriam Desainte-Catherine [10, 1], améliorée ensuite avec Mathieu Lagrange et Jean-Bernard Rault [36] et récemment étendue au cas non stationnaire avec Philippe Depalle [46]. Nous avons initié une comparaison des méthodes d'analyse avec Florian Keiler [23] et nous avons fait une étude plus poussée avec Mathieu Lagrange [37, 6], en exhibant notamment des équivalences au sein des méthodes d'analyse basées sur la phase de la transformée de Fourier à court terme. Il apparaît que leur précision est quasi-optimale sous des hypothèses de non-stationnarité simples (premier ordre, linéaire).

En ce qui concerne la synthèse, nous nous intéressons aux méthodes temporelles comme le résonateur numérique (actuellement implanté dans le module SYN de ReSpect) et notre nouvelle approche polynomiale (Section 3.3.2.2) avec Matthias Robine et Robert Strandh [38, Rob06], qui utilise une structure de données très efficace. Nous envisageons l'implantation d'une méthode hybride (basée sur les deux précédentes), avec une complexité extrêmement intéressante. Afin de diminuer la taille du problème (le nombre de partiels à synthétiser), nous avons proposé avec Mathieu Lagrange [20, Lag04] (durant sont Master) de tirer parti de phénomènes psychoacoustiques (Section 3.3.3), de manière très efficace à nouveau grâce à une structure de données appropriée (module PSY de ReSpect). Lors du Master de Matthias Robine, nous avons également envisagé la possibilité d'utiliser des méthodes non linéaires.

Le **Chapitre 4** présente la représentation sinusoïdale à long terme, avec les méthodes d'analyse et synthèse associées. En ce qui concerne le modèle, nous nous intéressons aux *partiels*.

Le module de suivi de partiels d'InSpect implémente diverses méthodes d'analyse à long terme. Pour l'analyse à long terme, ma thèse [Mar00] a servi de point de départ pour les recherches faites avec Mathieu Lagrange lors de son doctorat [Lag04]. Ensemble, nous avons proposé un algorithme de suivi de partiels amélioré, utilisant la prédiction linéaire (Section 4.2.3, [24, 27, 7]) et le contrôle du contenu haute-fréquence (Section 4.2.4, [31, 7]). Ces techniques se sont révélées être très utiles respectivement pour la restauration du son musical (Section 4.2.3.3, [4]) et la détection des *onsets* des notes (Section 4.2.4.3). Les informaticiens ont un rôle clé à jouer dans la conception de ces algorithmes de suivi de partiels, qui ont plus à voir avec l'algorithmique que le traitement du signal. Probablement à cause de cela, et à cause du fait que le problème du suivi des partiels est encore mal posé, ce domaine de recherche pourtant vieux de plus de 20 ans apparaît encore balbutiant. Bien que nous ayons initié un effort conséquent pour l'évaluation des algorithmes d'analyse à long terme (Section 4.2.5, [41]), c'est un problème majeur toujours ouvert.

Côté synthèse, avec Laurent Girin *et al.* [25] nous avons étendu l'approche polynomiale par morceaux classique au cas non stationnaire (Section 4.3). Bien que les résultats paraissaient prometteurs en théorie, ils se révélèrent assez décevant en pratique et par conséquent nous sommes revenus au modèle le plus simple (le plus rapide) pour la synthèse.

Partie III. La troisième partie de ce manuscrit décrit des recherches en cours (dans le cadre de thèses co-encadrées) ainsi que des perspectives de recherche (de futurs sujets de thèse et de projets).

Le **Chapitre 5** présente notre approche à très long terme pour la modélisation avec Laurent Girin et Mohammad Firouzmand [29, 32, 5] et le modèle polynôme+sinusoïdes (Section 5.1.3) avec Martin Raspaud [35] aboutissant à une modélisation sinusoïdale hiérarchique (Section 5.2, [30, Ras07]) de grand intérêt pour l'étirement temporel avec préservation des modulations (vibrato, trémolo). Ces recherches ont été menées durant la thèse de Martin Raspaud [Ras07]. De plus amples recherches sont encore nécessaires : considérer l'étirement temporel non uniforme pour préserver les transitoires, trouver un moyen d'étendre la hiérarchie vers le niveau macroscopique (musical), tout du moins pour les musiques répétitive (quasi-périodiques), et proposer une meilleure méthode d'analyse pour le modèle polynôme+sinusoïdes au second niveau du modèle hiérarchique, en tirant parti des méthodes haute-résolution (collaboration en cours avec Roland Badeau) et une version adaptée de notre algorithme de suivi de partiels amélioré.

Jusqu'à 2004, je n'ai considéré que les sons avec un faible niveau de bruit (et avec des paramètres déterministes). Le **Chapitre 6** décrit notre travail en cours sur la modélisation stochastique avec Pierre Hanna et Guillaume Meurisse (Master et thèse en cours), et implanté dans les modules d'analyse / synthèse stochastiques d'InSpect. L'objectif de la thèse de Guillaume Meurisse est de proposer un modèle unifié où les signaux déterministes et stochastiques peuvent être analysés et synthétisés dans le même cadre : avec un modèle basé sur les distributions statistiques. Nous avons proposé une méthode d'analyse basée sur la distribution de Rice (Section 6.2, [39]), méthode particulièrement efficace pour des rapports signal sur bruit moyens (typiquement rencontrés dans les sons musicaux) mais nécessitant des améliorations pour les rapports plus faibles et le cas non stationnaire. Nous devons également proposer la méthode de synthèse correspondante.

Toujours en 2004, j'ai commencé à considérer les signaux binauraux (la plupart des gens écoutent maintenant la musique en utilisant des casques audio, et les studios d'enregistrement apportent un soin particulier aux effets spatiaux pour les sons stéréophoniques stockés sur les CD audio). Le **Chapitre 7** décrit notre travail en cours sur le son spatial avec Joan Mouba (thèse en cours), implanté dans les modules de localisation et de spatialisation d'InSpect, ainsi que dans le module SPA de ReSpect. L'objectif de la thèse de Joan Mouba est de proposer une multi-diffusion sur plusieurs haut-parleurs à partir d'un son binaural : les entités sonores sont extraites du mélange binaural et envoyés aux haut-parleurs, après avoir éventuellement subi des manipulations intermédiaires (changement de volume, de position dans l'espace, *etc.*). Premièrement, nous proposons un modèle binaural simplifié (Section 7.3, [45]) basé sur des indices spatiaux fonctions de l'azimut de la source. Deuxièmement, nous considérons également la distance de la source, avec une localisation basée sur la brillance. Nous proposons alors un algorithme de séparation de sources (Section 7.5, [40]) et des techniques de spatialisation binaurale et pour la multi-diffusion (Sections 7.4.1 et 7.6, [45]).

Pour finir, le **Chapitre 8** liste nos principales pistes de recherche pour le futur. La première est la décomposition de la musique polyphonique en entités sonores (voir Section 8.1), à l'aide de critères perceptifs. Une piste de recherche majeure est de trouver un critère unique et de remplacer le suivi de partiels par la structuration des entités. Pour l'instant, la structuration est faite de manière sous-optimale après l'analyse à long terme (voir Figure 1), et en utilisant chacun des critères suivants séparément : les *onsets* communs (doctorat de Mathieu Lagrange [Lag04]), la structure harmonique (Master de Grégory Cartier [Car03], doctorat de Mathieu Lagrange [Lag04]), les évolutions corrélées des paramètres (doctorat de Mathieu Lagrange [Lag04], doctorat de Martin Raspaud [Ras07]), et la position spatiale (thèse de Joan Mouba en cours).

Étant donné que l'extraction des entités sonores est une tâche extrêmement difficile, une solution pour faciliter cette extraction pourrait être de tirer parti d'informations annexes (inaudibles) par

exemple stockées (tatouées) à l'intérieur du son (Section 8.2). Avec Laurent Girin [28] nous avons montré l'utilité du modèle sinusoïdal pour le tatouage audio-numérique. Inclure une telle information inaudible à l'intérieur de chaque entité sonore avant le processus de mixage pourrait faciliter l'extraction des entités du mélange. Nous avons démarré une collaboration avec Laurent Girin sur le sujet de la « séparation informée ».

Notre but ultime comporte un enjeu sociétal : avec l'**« écoute active »** (Section 8.3, [44]), nous prenons le pari qu'en chaque auditeur un musicien sommeille, et qu'il est notre devoir de permettre de changer la façon dont les gens écoutent la musique. Nous souhaitons donner à l'auditeur la liberté d'interagir avec le son en temps réel lors de sa diffusion, au lieu d'écouter ce son de façon classique, passive.

Introduction

From heartbeats to planetary motion, our universe is full of quasi-periodic phenomena. This is also the case of musical sounds, my professional universe since the Ph.D. The aim of this manuscript is to describe our research in musical sound modeling since this Ph.D. thesis, and thus it can be seen as an upgrade of the corresponding manuscript. This document describes our work together with the Ph.D. students I co-supervise(d) (Mathieu Lagrange, Matthias Robine, Martin Raspaud, and currently Joan Mouba and Guillaume Meurisse). Our contributions appear in the text as numeric references [1–54] – and are listed in Section A.5 – whereas the other bibliographic references are alphanumeric. These contributions are being implemented in our InSpect ("In(spect) Spect(rum)", see Figure 1) and ReSpect ("Re(spect) Spect(rum)", see Figure 2) software projects, for the analysis / transformation / resynthesis of spectral sounds, and initiated at the beginning of the Ph.D. (see [14]).

Part I. The first part of this manuscript is an introduction to musical sound, which is pluridisciplinary by nature. The notions of mathematics, physics, signal processing, acoustics, psychoacoustics, and music are limited here to the concepts strictly necessary for the understanding of the remainder of this document. Thus, the reader familiar with these notions may skip this first part. However, note that we also introduce useful notations there.

Chapter 1 is starting from the temporal representation of sound signals, in the cases of both continuous and discrete time, and with a short introduction to the theory and practice of sampling and reconstruction. Our contribution with Raspaud [43] is an enhanced reconstruction / resampling technique for non zero centered signals, extremely useful when dealing with the parameters of sinusoidal modeling. The spectral representation is also introduced, with the Fourier transform (and its inverse) in both the continuous and discrete cases. Then the fundamental duality of temporal and spectral domains is illustrated for a variety of operations and functions. This duality will play a key role in the remainder of this document. Indeed, a problem that might seem difficult in one domain (temporal or spectral) can have an elegant solution in the other domain (spectral or temporal). After these mathematical and physical bases for digital sound, the production, propagation, and reception (perception) of the sound are briefly described. Notions of acoustics are introduced, such as the delay and attenuation of the sound due to the propagation of the acoustic wave in the air. Important notions of psychoacoustics are also presented, such as scales closer to perception, thresholds of hearing and masking phenomena used for fast additive synthesis (Section 3.3.3) or watermarking (Section 8.2).

Chapter 2 presents elements of music, from notions of musical acoustics to the musical parameters (pitch, loudness, and timbre) and their variations in time (vibrato, tremolo). Again, this chapter is limited to the notions strictly necessary for the remainder of the document.

Part II. The second part of this manuscript deals with my main research subject: sinusoidal modeling of musical sounds.

Chapter 3 presents the short-term sinusoidal (STS) representation, with the associated analysis and synthesis methods. Regarding the model, we focus on spectral *peaks* and, compared to the Ph.D., we now fully consider the non-stationary case, where the parameters can evolve even within a short time window.

The peak extractor module of InSpect implements various short-term analysis methods. Three main non-stationary analysis methods are described in this document: quadratic interpolation, spectral reassignment, and our contribution – the derivative method (Section 3.2.3.3) we proposed together with Desainte-Catherine [10, 1], enhanced together with Lagrange and Rault [36], and recently extended to the non-stationary case together with Depalle [46]. We started the comparison of analysis methods together with Keiler [23] and did a more extensive survey together with Lagrange [37, 6], pointing out some equivalences among these analysis methods based on the short-time Fourier transform (STFT). It turns out that their precision is close to optimal under simple (first-order linear) non-stationary conditions.

Regarding the synthesis, we focus on temporal methods such as the digital resonator (currently in the SYN module of ReSpect) and our efficient polynomial approach (Section 3.3.2.2) together with Robine and Strandh [38, Rob06], using a very efficient data structure. Part of our research projects is the implementation of a hybrid method (based on the two previous methods), with an extremely interesting complexity. To reduce the size of the problem (the number of partials to synthesize), we proposed together with Lagrange [20, Lag04] (during his Master) to take advantage of psychoacoustic phenomena (Section 3.3.3), in a very efficient way again thanks to an appropriate data structure (PSY module of ReSpect). During Robine's Master, we also investigated the possibility to use non-linear methods.

Chapter 4 presents the long-term sinusoidal (LTS) representation, with the associated analysis and synthesis methods. Regarding the model, we focus on *partials*.

The partial tracker module of InSpect implements various long-term analysis methods. Regarding the long-term analysis, my Ph.D. [Mar00] served as a starting point for the main research done with Lagrange during his Ph.D. [Lag04]. Together, we proposed an enhanced partial-tracking algorithm using linear prediction (Section 4.2.3, [24, 27, 7]) and the control of the high-frequency content (Section 4.2.4, [31, 7]). These techniques turned out to be extremely useful for musical sound restoration (Section 4.2.3.3, [4]) and note onset detection (Section 4.2.4.3), respectively. Computer scientists have a key role to play with these partial-tracking algorithms, dealing more with algorithmics than signal processing. Probably because of that, and because of the fact that the problem of partial tracking is still ill-posed, this more than 20-year old research area may seem in its infancy. Although we initiated a consequent effort on the evaluation of analysis algorithms in the long-term case (Section 4.2.5, [41]), this is still a major open issue.

Regarding the synthesis, together with Girin *et al.* [25], we extended the classic piecewise polynomial approach to the non-stationary case (Section 4.3). Although the results were promising in theory, they were quite disappointing in practice and thus we reverted to the simplest – and faster – model for the synthesis.

Part III. The third part of this manuscript describes ongoing research subjects (PhDs in progress) and research perspectives (subjects for future PhDs and projects).

Chapter 5 presents our very-long term modeling approach together with Girin and Firouzmand [29, 32, 5] and the polynomials+sinusoids model (Section 5.1.3) together with Raspaud [35] leading to hierarchic sinusoidal modeling (Section 5.2, [30, Ras07]) of great interest for time scaling while preserving the modulations (vibrato, tremolo). This research was done during the Ph.D. of

Raspaud [Ras07]. Further work is still required: considering non-uniform time scaling to preserve the transients, finding a way to extend the hierarchy towards the macroscopic (musical) level – at least for repetitive (quasi-periodic) music, and proposing a better analysis method for the polynomials+sinusoids model for the second level of the hierarchic model – taking advantage of high-resolution methods (ongoing collaboration with Badeau) and an adapted version of our enhanced partial-tracking algorithm.

Until 2004, I have considered only sounds with a low noise level (and with deterministic parameters). **Chapter 6** describes our work in progress on stochastic modeling together with Hanna and Meurisse (Master and ongoing Ph.D.), and implemented in the stochastic analyzer / synthesizer modules of InSpect. The aim of Meurisse's Ph.D. is to propose an unified model where deterministic and stochastic signals can be analyzed and synthesized in the same framework – with a model based on probability density functions. We have proposed an analysis method based on the Rice probability density function (Section 6.2, [39]), particularly efficient for medium signal-to-noise ratios (SNRs) – typically encountered in musical sounds – but still to be enhanced for low SNRs as well as in the non-stationary case. We still have to propose the corresponding synthesis method.

Also in 2004, I started to consider binaural signals (most people now listen to music using headphones, and a special effort is made in recording studios on spatial effects for the stereophonic – yet binaural – signals stored on standard compact discs). **Chapter 7** describes our work in progress on spatial sound together with Mouba (ongoing Ph.D.), implemented in the localizer and spatializer modules of InSpect, as well as in the SPA module of ReSpect. The aim of Mouba's Ph.D. is to propose a binaural to multi-diffusion ("upmixing") technique: the sound entities are extracted from the binaural mix and send to loudspeakers, possibly with intermediate manipulations (volume change, sound relocation, *etc.*). First, we propose a simplified binaural model (Section 7.3, [45]) based on spatial cues as functions of the azimuth of the source. Second, we also consider the distance of the source, with a localization based on the brightness. We then propose a source separation algorithm (Section 7.5, [40]) and binaural and multi-diffusion synthesis techniques (Sections 7.4.1 and 7.6, [45]).

Finally, **Chapter 8** lists our main research directions for the future. The first one is the decomposition of polyphonic music in sound entities (see Section 8.1), according to perceptive criteria. A major research direction is to find a way to get a unique criterion and replace partial tracking by entity structuring. For now, the structuring is done in a sub-optimal way after the long-term analysis stage (see Figure 1), and by using each criterion separately: the common onsets (Ph.D. of Lagrange [Lag04]), the harmonic structures (Master's thesis of Cartier [Car03], Ph.D. of Lagrange [Lag04]), the correlated evolutions (Ph.D. of Lagrange [Lag04], Ph.D. of Raspaud [Ras07]), and the spatial location (ongoing Ph.D. of Mouba).

Since extracting the sound entities is an extremely difficult task, a solution to ease this extraction could be to take advantage of extra (inaudible) information stored (watermarked) in the sound (Section 8.2). Together with Girin [28] we have shown the suitability of spectral modeling for audio watermarking. Including such inaudible information in each sound entity prior to the mixing process should ease the extraction of the entities from the mix. We initiated a collaboration with Girin on this subject, called "informed separation".

Our ultimate goal has a social aspect: with **"active listening"** (Section 8.3, [44]), we bet that a musician lies within any listener, and that we should change the way people can listen to music. We aim at providing the listener with the freedom to interact with the sound in real-time during its diffusion, instead of listening to this sound in the usual – passive – way.

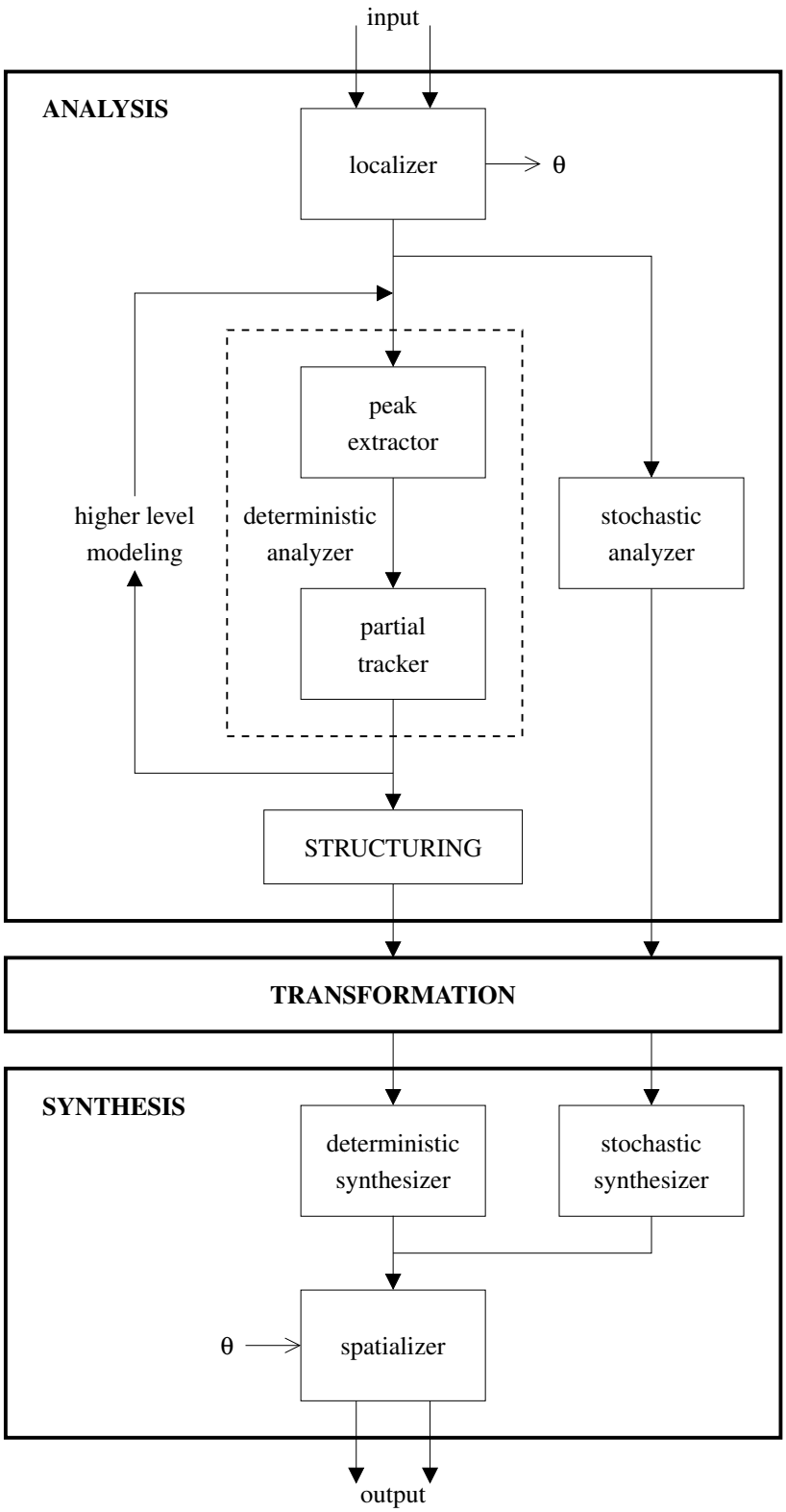


Figure 1: Diagram of the InSpect software program.

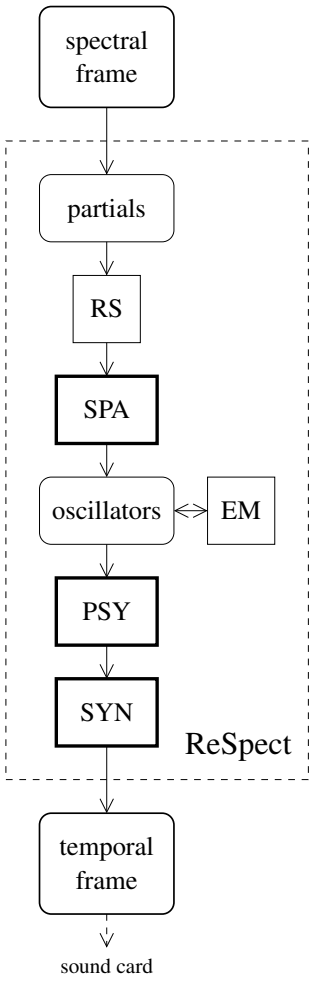


Figure 2: Diagram of the ReSpect software library. ReSpect is a real-time version of the deterministic synthesizer of InSpect (see Figure 1). The parameters of the partials are upsampled by the resampling (RS) module. Then the spatialization (SPA) module adds the spatial information to these partials, which control a set of software oscillators updated by an event manager (EM). The psychoacoustic (PSY) module selects only the audible partials for the synthesis (SYN) module.

Part I

Introduction to Musical Sound

The first part of this manuscript is an introduction to musical sound, which is pluridisciplinary by nature. The notions of mathematics, physics, signal processing, acoustics, psychoacoustics, and music are limited here to the concepts strictly necessary for the understanding of the remainder of this document. Thus, the reader familiar with these notions may skip this first part. However, note that we also introduce useful notations there.

Chapter 1 is starting from the temporal representation of sound signals, in the cases of both continuous and discrete time, and with a short introduction to the theory and practice of sampling and reconstruction. Our contribution with Raspaud [43] is an **enhanced reconstruction / resampling technique** for non zero centered signals, extremely useful when dealing with the parameters of sinusoidal modeling (see Part II). The spectral representation is also introduced, with the Fourier transform (and its inverse) in both the continuous and discrete cases. Then the fundamental duality of temporal and spectral domains is illustrated for a variety of operations and functions. This duality will play a key role in the remainder of this document. Indeed, a problem that might seem difficult in one domain (temporal or spectral) can have an elegant solution in the other domain (spectral or temporal). After these mathematical and physical bases for digital sound, the production, propagation, and reception (perception) of the sound are briefly described. Notions of acoustics are introduced, such as the delay and attenuation of the sound due to the propagation of the acoustic wave in the air. Important notions of psychoacoustics are also presented, such as scales closer to perception, thresholds of hearing and masking phenomena used for fast additive synthesis (Section 3.3.3) or watermarking (Section 8.2).

Chapter 2 presents elements of music, from notions of musical acoustics to the musical parameters (pitch, loudness, and timbre) and their variations in time (vibrato, tremolo). Again, this chapter is limited to the notions strictly necessary for the remainder of the document. Musical aspects are clearly part of our future research directions. For example, the scientific study of the spectral envelope (color) of musical sounds, with perception and encoding aspects, is an extremely interesting subject.

Chapter 1

Elements of Sound

The fast fluctuations – from tens to thousands per second – of the air pressure at the level of the ears generate an auditory sensation. The word "sound' is ambiguous, since it is used for both the physical vibration and the sensation this vibration produces. In order to fully understand the objective and subjective aspects of sound, basic knowledge of elements of acoustics and psychoacoustics is required. Notions of mathematics, physics, and signal processing are prerequisites. The aim of this chapter is to provide the reader with this basic pluridisciplinary knowledge, to introduce useful notations, and to list interesting properties. We limit ourselves to the notions required for the remainder of this manuscript, sometimes appealing simplifications. Thus, we only give hints for the mathematical demonstrations of the signal processing aspects, which can be found in [Pap77, Smi07]. Reference books dealing with the fundamentals of acoustics are numerous. And regarding psychoacoustics, we recommend [FZ06].

1.1 Temporal Domain

The simplest way to represent a sound is to consider its amplitude as a function of time (see Figure 1.1). This sound representation is often called the time domain or temporal domain.

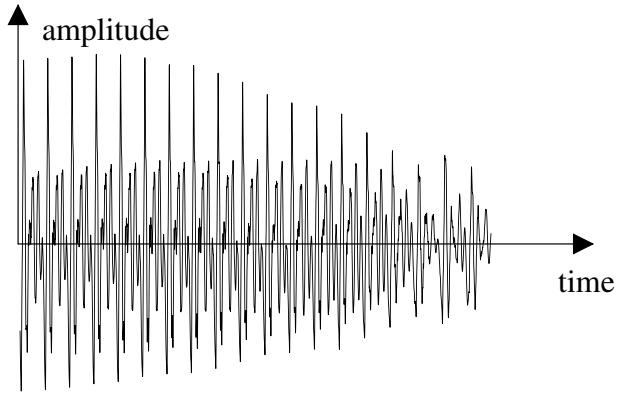


Figure 1.1: A sound represented in the temporal model.

1.1.1 Continuous Time

When t is the time considered as continuous and expressed in seconds, the signal is $s(t)$. Because of the limited amount of memory, computers cannot deal with continuous time $t \in \mathbb{R}$.

1.1.2 Discrete Time

Computers deal with sampled sounds (of finite length). Here we consider signals sampled uniformly, and we note $s[n] = s(nT_s)$ the corresponding discrete-time signal where the sampling period T_s (in seconds, s) is the inverse of the sampling frequency F_s (in Hertz, Hz), and n is the sample number ($n \in \mathbb{Z}$).

1.1.3 Sampling and Reconstruction

With the previous notation for $s[n]$, we are in fact doing a rather crude sampling, which is valid if and only if $s(t)$ is band-limited in frequency to $F_s/2$ (the Nyquist frequency).

Whis this condition, the Shannon-Nyquist theorem ensures that a (perfect) reconstruction of $s(t)$ can be computed from its sampled version $s[n]$, using a convolution (see Section 1.3.1.1)

$$s(t) = \sum_{n=-\infty}^{+\infty} s[n] \cdot r(t - nT_s) \tag{1.1}$$

with the (ideal) reconstructor

$$r(t) = \mathrm{sinc}(F_s t) \tag{1.2}$$

where sinc is the cardinal sine function

$$\mathrm{sinc}(x) = \begin{cases} \frac{\sin(\pi x)}{\pi x} & (x \neq 0), \\ 1 & (x = 0). \end{cases} \tag{1.3}$$

In practice, the ideal reconstructor cannot be used because of its infinite time support. To limit this support, we use the classic "window method": we multiply the ideal reconstructor r by a window w of finite length N samples ($N \in \mathbb{N}$) to obtain a practical reconstructor

$$r_w(t) = w(t) \cdot r(t). \tag{1.4}$$

For w, we use the Hann window (see Equation (3.41)), although other windows can be used [SG84]. The window length N allows us to tune the trade-off of reconstruction quality versus computation time in the resampling process.

Using this reconstructor in Equation (1.1), we obtain (an approximation of) the continuous signal $s(t)$ from its sampled version. Upsampling by a factor u ($u \geq 1$) the signal s is straightforward since we can compute this function at any time, all the more at multiples of the new sampling period T_s/u (upsampling is like considering the $s(t/u)$ function). Downsampling s by a factor d ($d \geq 1$) is slightly more complicated, since high frequencies have to be filtered out in order to fulfill the Nyquist condition. We then have to use $F_m = \min(F_s, F_s')$, where F_s' is the destination sampling rate, instead of F_s in Equation (1.2) to define the appropriate reconstructor.

Although in theory resampling works on every kind of band-limited signal, in practice, using the finite version of the reconstructor of Equation (1.4), artifacts are likely to occur, especially if the signal is not zero-centered. Indeed, an infinite number of samples are left out of the computation by the finite reconstructor, which is problematic if these values are not zeroes, as shown on Figure 1.2. The

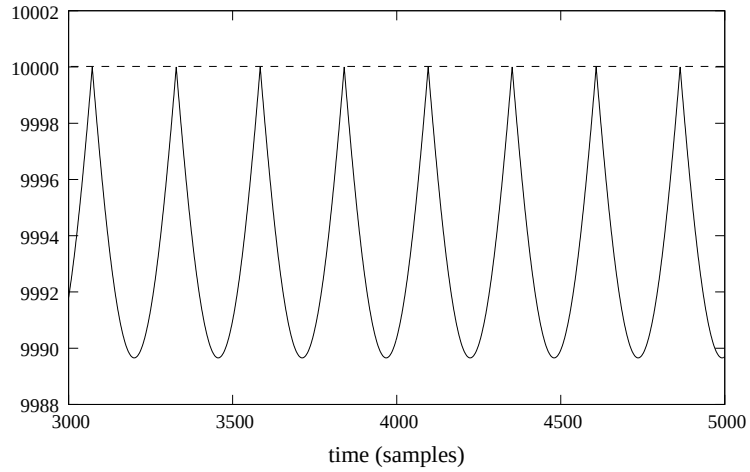


Figure 1.2: The classic resampling technique is not adapted for the resampling of non zero-centered signals. Here is an example of the resampling of a constant signal at 10000 (represented by a dashed line) – which is a typical value for the frequency parameter (in Hz) of partials in sinusoidal modeling (see Part II). The resampling ratio is 256. The result of this resampling is given as a solid line, far from the constant we would have expected. Moreover, if we had added a sinusoidal modulation with an amplitude of 5 on the signal – which is typical for a periodic modulation (vibrato) of the frequency – the resampling would have completely lost the modulation in the artifacts.

sinusoidal modeling parameters (see Chapter 4) are control signals that fall into this category of signals that are not zero-centered, with a slow time-varying envelope, and often modulated (for example by some vibrato for the frequency control signal). Whereas the modulations are zero-centered, this is not the case for the envelope. In order to enhance the resampling, together with Raspaud [43] we perform the centering on parts of the signal that are about to be used at a given reconstruction time t. More precisely, we model the envelope of the signal as a piecewise polynomial of low degree d ($d \leq 3$) estimated using the well-known least-square method (see [35] for details) under the span of the practical reconstructor. Then we remove the estimated envelope and the reconstruction is done on the zero-centered version. The contribution of the polynomial envelope to the resampled result is easy to compute and is re-added afterwards. As shown in [43], removing the local mean ($d = 0$) significantly enhances the result of the resampling. (For the example of Figure 1.2 – where the envelope is a constant – it gives a perfect result.) Increasing the degree gives even better results (for time-varying envelopes), although d must remain low enough not to capture any period of the modulations.

Moreover, in practice the sampled signal is also of finite length. To reduce transients at the beginning and at the end, the reflection method can be used. More precisely, the signal can be mirrored:

$$\begin{cases} s[-n] & = & 2\,s[0] - s[n] \\ s[(L-1)+n] & = & 2\,s[L-1] - s[(L-1)-n] \\ & \text{for} & 0 < n < L \end{cases} \tag{1.5}$$

with L being the length of the sampled signal, the first sample being $s[0]$ and the last $s[L-1]$. This linear extrapolation by mirroring the signal ensures the continuity of the signal and of its first derivative at the beginning and at the end, and provided that r_w is symmetric the values of the reconstructed signal match the values of the sampled signal at these positions. Other extrapolation techniques are possible, though.

1.2 Spectral Domain

Another – extremely convenient – way of representing a sound is to consider its spectrum, which is a complex-valued function of the frequency. This sound representation is called the frequency domain or spectral domain.

1.2.1 Complex Numbers

In practice, with sounds, we deal with real-valued signals. However, their spectra are complex-valued. Moreover, complex number often makes mathematical derivations much simpler...

Regarding complex numbers, we denote by j the imaginary unit, thus $j^2 = -1$. We can use the Cartesian representation $c = x + jy$, where $x = \mathrm{Re}(c)$ and $y = \mathrm{Im}(c)$ are the real and complex parts of the complex number c, respectively. We denote by $\bar{c} = x - jy$ the conjugate of c. We often use the Euler notation (polar representation) for complex numbers, that is $c = a\,e^{j\phi}$, where $a = |c|$ and $\phi = \angle c$ are the amplitude (magnitude) and the phase of the complex number c, respectively. For the sake of readability of its argument, we might also denote by $\exp(x) = e^x$ the exponential function. Finally, we will often use the Euler's formula

$$\exp(j\theta) = \cos(\theta) + j\sin(\theta) \tag{1.6}$$

together with its consequences

$$\cos(\theta) = \frac{\exp(+j\theta) + \exp(-j\theta)}{2}, \tag{1.7}$$

$$\sin(\theta) = \frac{\exp(+j\theta) - \exp(-j\theta)}{2j}. \tag{1.8}$$

1.2.2 Complex Amplitude, Magnitude, and Phase Spectra

The (continuous) spectrum of signal $s(t)$ (lowercase notation) will be denoted by $S(\omega)$ (uppercase notation), and to highlight this temporal / spectral correspondence, we write

$$s(t) \leftrightarrow S(\omega) \tag{1.9}$$

where ω is the frequency is radians per second and

$$\omega = 2\pi f \tag{1.10}$$

where f is the frequency in Hz (Hertz, *i.e.* cycles per second).

As mentioned before, the spectrum is a complex-valued function of the frequency, which gives in polar representation

$$S(\omega) = ae^{j\phi} \tag{1.11}$$

and thus

$$a = |S(\omega)|, \tag{1.12}$$

$$\phi = \angle S(\omega). \tag{1.13}$$

Furthermore, we will also consider the log-amplitude (or magnitude)

$$\mu = \log(a). \tag{1.14}$$

Thus, since

$$\log(S(\omega)) = \underbrace{\log(a)}_{\mu} + j\phi \tag{1.15}$$

we have

$$\mu = \Re(\log(S(\omega))), \tag{1.16}$$
$$\phi = \Im(\log(S(\omega))). \tag{1.17}$$

Among other properties, we recall that

- the spectrum of a real signal s exhibits an Hermitian symmetry: $S(-\omega) = \text{conj}(S(\omega))$;
- the spectrum of an even function is real;
- the spectrum of an odd function is imaginary;

which can be easily demonstrated using the definition of the Fourier transform (see [Smi07]).

1.2.3 Fourier Transform

The Fourier transform and its inverse transform are mathematical transforms allowing to switch from the time domain to the frequency domain and to do the opposite, respectively.

1.2.3.1 Continuous Fourier Transform (FT)

If s and S are the expressions of the same sound in the temporal and spectral domains, respectively, then the continuous Fourier transform and its inverse are given by the following equations:

$$S(\omega) = \int_{-\infty}^{+\infty} s(t)\, e^{-j\omega t}\, dt, \tag{1.18}$$
$$s(t) = \int_{-\infty}^{+\infty} S(\omega)\, e^{+j\omega t}\, d\omega. \tag{1.19}$$

Here we have omitted the classic $1/(2\pi)$ factor in Equation (1.19) to ease the correspondence between the continuous and discrete cases in Section 1.3. For a given frequency, the Fourier transform can be thought of as a correlation with a (complex) sinusoid of this frequency.

1.2.3.2 Discrete-Time Fourier Transform (DTFT)

For discrete-time signals, the expressions of the discrete-time Fourier transform (DTFT) and its inverse transform (IDTFT) are, respectively

$$S(\omega) = \sum_{n=-\infty}^{+\infty} s[n]\, e^{-j\omega n T_s}, \tag{1.20}$$
$$s[n] = \frac{1}{2\pi F_s} \int_{-\pi F_s}^{+\pi F_s} S(\omega)\, e^{+j\omega n T_s}\, d\omega. \tag{1.21}$$

Because of the sampling, $S(\omega)$ is now a $2\pi F_s$-periodic function.

It is often convenient to get rid of the sampling frequency by considering the normalized frequency (in radians per sample)

$$\underline{\omega} = \omega/F_s = \omega T_s \tag{1.22}$$

so that, for example, Equation (1.20) becomes

$$S(\omega) = \sum_{n=-\infty}^{+\infty} s[n]\, e^{-j\underline{\omega}n}.$$

Moreover, f (the frequency in Hz) can help getting rid of constants in equations. For example, with the normalized frequency $\underline{f} = f/F_s$, Equation (1.21) becomes

$$s[n] = \int_{-1/2}^{+1/2} S(2\pi \underline{f} F_s)\, e^{+j2\pi \underline{f} n}\, d\underline{f}.$$

1.2.3.3 Discrete Fourier Transform (DFT)

If the discrete-time signal s is of finite length $N = 2H+1$, then we use the discrete Fourier transform (DFT) and its inverse transform (IDFT), which are special cases of the DTFT and IDTFT, respectively.

As we considered discrete-time signals, we consider discrete-frequency spectra and note $S[k] = S(k\Omega_{F_s,N})$ where

$$\Omega_{F_s,N} = 2\pi F_s/N \tag{1.23}$$

is the width of one bin (sample) of the discrete spectrum. With this notation, the DFT and IDFT can be defined using the following equations, respectively:

$$S[m] = \sum_{n=-H}^{+H} s[n] e^{-jnmT_s\Omega_{F_s,N}}, \tag{1.24}$$

$$s[n] = \frac{1}{N} \sum_{m=-H}^{+H} S[m] e^{+jnmT_s\Omega_{F_s,N}}. \tag{1.25}$$

The discrete Fourier transform can be thought of as a correlation with an orthonormal basis of discrete sine functions. The magnitude spectrum $|S|$ shows the frequency content of the signal (see Figure 1.3).

1.2.3.4 Fast Fourier Transform (FFT)

The rapid computation of the DFT and its inverse is a crucial point in many applications (not only in computer music). For example, the famous radix-2 algorithm [CT65] is the among the most used algorithms. The length N has to be a power of 2, thus an even number. However, if $s[-N/2] = 0$ (which is the case if a Hann window of size N is used, see Section 3.2.2.2) and if $\tilde{s}$ denotes the version of s circularly-shifted by $N/2$ samples, we have

$$\sum_{n=-(N/2-1)}^{+(N/2-1)} s[n] e^{-jnmT_s\Omega_{F_s,N}} = \underbrace{\exp\left(j\frac{N}{2}mT_s\Omega_{F_s,N}\right)}_{(e^{-j\pi})^m=(-1)^m} \sum_{n=0}^{N-1} \tilde{s}[n] e^{-j2\pi nm/N} \tag{1.26}$$

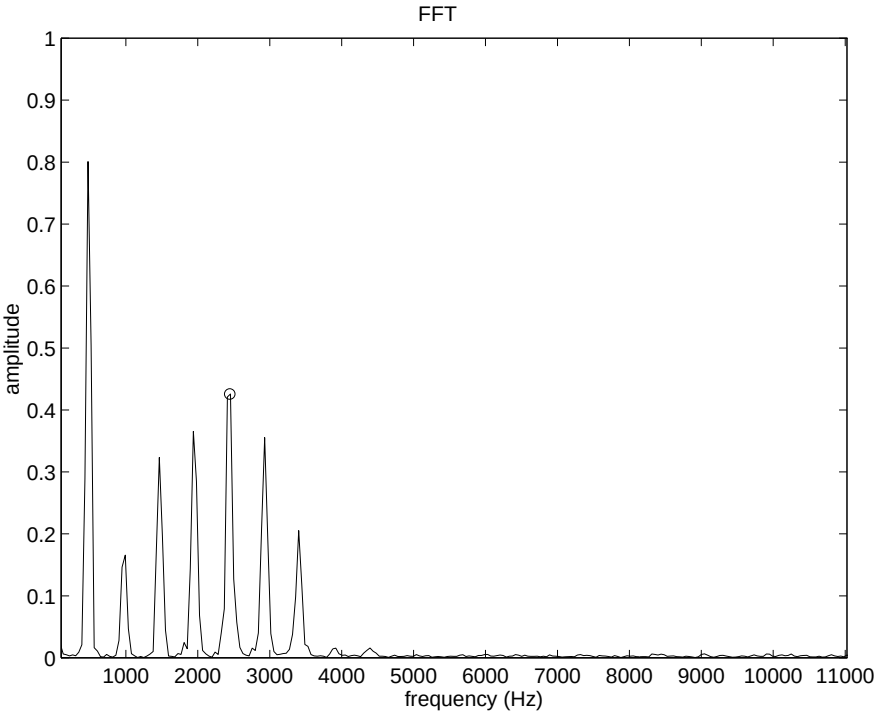


Figure 1.3: Magnitude spectrum resulting from a Fast Fourier Transform (FFT).

and noticing that

$$\begin{aligned}\sum_{n=0}^{N-1} \tilde{s}[n] e^{-j2\pi nm/N} &= \left(\sum_{n=0}^{N/2-1} \tilde{s}[2n]\ e^{-j\frac{2\pi}{N}(2n)m} + \sum_{n=0}^{N/2-1} \tilde{s}[2n+1]\ e^{-j\frac{2\pi}{N}(2n+1)m} \right) \\ &= \sum_{n=0}^{N/2-1} \tilde{s}[2n]\ e^{-j\frac{2\pi}{N/2}nm} + \left(e^{-j\frac{2\pi}{N}}\right)^m \cdot \sum_{n=0}^{N/2-1} \tilde{s}[2n+1]\ e^{-j\frac{2\pi}{N/2}nm}\end{aligned}$$

we recognize the combination of 2 DFTs of size $N/2$ (over $\tilde{s}[2n]$ and $\tilde{s}[2n+1]$, respectively). This way, when N is a power of 2, by repeating $\log_2(N)$ times the same scheme, we can go down to N Fourier transforms of size 1, very easy to compute. The complexity of this radix-2 FFT is thus $O(N\log(N))$, whereas the complexity of the original DFT (see Equation (1.24)) was $O(N^2)$.

1.3 Duality of Temporal and Spectral Domains

Because of the linearity of the Fourier transform (and its inverse), for any signals $s_1 \leftrightarrow S_1$ and $s_2 \leftrightarrow S_2$, and for any scalars (real or complex numbers) α_1 and α_2, we have

$$\alpha_1 s_1 + \alpha_2 s_2 \leftrightarrow \alpha_1 S_1 + \alpha_2 S_2. \tag{1.27}$$

Thus, the addition is the same operation in both domains (temporal and spectral).

1.3.1 Basic Operations

1.3.1.1 Convolution

The convolution $*$ is an operation whose definition in the continuous-time and discrete-time cases are, respectively

$$x(t) * y(t) = \int_{-\infty}^{+\infty} x(\tau)\, y(t-\tau)\, d\tau, \tag{1.28}$$

$$x[n] * y[n] = \sum_{k=-\infty}^{+\infty} x[k]\, y[n-k]. \tag{1.29}$$

Similarly, the convolution among spectra in the continuous and discrete cases are, respectively

$$X(\omega) * Y(\omega) = \int_{-\infty}^{+\infty} X(\nu)\, Y(\omega-\nu)\, d\nu, \tag{1.30}$$

$$X[m] * Y[m] = \sum_{k=-\infty}^{+\infty} X[k]\, Y[m-k]. \tag{1.31}$$

The convolution in one domain (temporal or spectral) is equivalent to the multiplication in the other domain (spectral or temporal). This is the convolution theorem:

$$x * y \leftrightarrow X \cdot Y, \tag{1.32}$$

$$x \cdot y \leftrightarrow X * Y. \tag{1.33}$$

(The proof can be done in both continuous and discrete time cases, by computing the spectrum corresponding to $x * y$ and the signal corresponding to $X * Y$, using the definitions of the Fourier transform and its inverse, respectively, see [Smi07]. The fact that Equation (1.33) is still valid in the continuous case is due to our choice in Equation (1.19) for the definition of the inverse Fourier transform.)

1.3.1.2 Differentiation

We denote by s' and S' the derivative of s and S, respectively. Then, the differentiation theorem and its dual are, respectively

$$s'(t) \leftrightarrow j\omega S(\omega), \tag{1.34}$$

$$-jts(t) \leftrightarrow S'(\omega). \tag{1.35}$$

(The proof can be done using the principle of integration by parts and the definitions of the Fourier transform and its inverse, applied to s' and S', respectively, see [Smi07].)

1.3.1.3 Translation

A translation in time (or time shift) is equivalent to a modification of the phase of the spectrum. More precisely, the shift theorem states

$$s(t-\tau) \leftrightarrow e^{-j\omega\tau} S(\omega), \tag{1.36}$$

$$e^{j\nu t} s(t) \leftrightarrow S(\omega-\nu). \tag{1.37}$$

(The proof is straightforward using the definitions of the Fourier transform and its inverse, applied to $s(t-\tau)$ and $S(\omega-\nu)$, respectively, see [Smi07].)

In the discrete case, we deal with circular shifts. Considering Equation (1.36) with $\omega = m\Omega_{F_s,N}$ and $\tau = (N/2)T_s$, if N is even then we have

$$e^{-j\omega\tau} = e^{-jm\pi} = \left(e^{-j\pi}\right)^m = (-1)^m$$

and thus a circular time-shift of $N/2$ samples of the signal $s[n]$ corresponds to a spectrum $(-1)^m S[m]$, as seen in Equation (1.26). This simple transformation can be used to cancel the linear phase of the FFT spectrum to revert to the zero-phase definition of our DFT.

1.3.1.4 Interpolation

"Zero padding" consists in adding extra $N-M$ samples with zero value to a discrete signal s of finite length M. The FFT is then computed on the resulting signal (consisting of N samples):

$$Z[m] = \sum_{n=0}^{M-1} s[n]\, e^{-j2\pi mn/N} = S\left[m \cdot \frac{M}{N}\right] \quad \text{with} \quad S[m] = \sum_{n=0}^{M-1} s[n]\, e^{-j2\pi mn/M}. \tag{1.38}$$

This does not increase the resolution of the spectrum, but gives an interpolated (resampled) version of this spectrum, which can be useful for some spectral analysis methods (see Section 3.2). Similarly, zero padding in the spectral domain is equivalent to resampling the time signal. However, zero padding increases the length of the data, thus the computation time of the FFT. Moreover, as we show in [6], many spectral analysis methods reach a nearly-optimal precision regardless to the zero-padding factor (N/M). For these reasons, we will avoid zero padding as much as possible in the remainder of this manuscript.

1.3.2 Basic Functions

1.3.2.1 Complex Sinusoid

A complex sinusoid of frequency ω_0, amplitude 1, and initial phase 0 has a spectrum which is an impulse of amplitude 1 located at frequency ω_0. More precisely, we can define the following function as a very crude simplification of the Dirac impulse:

$$\delta(x) = \begin{cases} 1 & \text{for } x = 0, \\ 0 & \text{otherwise} \end{cases} \tag{1.39}$$

and thus we can write

$$e^{j\omega_0 t} \leftrightarrow \delta(\omega - \omega_0) \tag{1.40}$$

although we are conscious that this is a mathematical shortcut, this will be sufficient for the mathematical derivations in the remainder of this manuscript.

Thus, the spectrum of a sinusoid is an impulse. And an impulse has a constant magnitude spectrum. This is a case where a continuous function in one domain (temporal or spectral) corresponds to a discontinuous function in the other domain (spectral or temporal). The Gaussian function is an example of continuous function whose temporal and spectral expressions are of the same kind (Gaussian), see Section 3.2.2.3. For a discontinuous function with the same behavior, let us consider impulse trains.

1.3.2.2 Impulse Trains

We define the train of impulses periodically spaced (with period T):

$$\delta_T(t) = \sum_{k=-\infty}^{+\infty} \delta(t - kT). \tag{1.41}$$

The spectrum of an impulse train is an impulse train:

$$\delta_T(t) \leftrightarrow \delta_{2\pi F}(\omega) \quad \text{where } F = 1/T. \tag{1.42}$$

1.3.2.3 Box and Sinc Functions

We define the (zero-centered) box function of width $W \geq 0$:

$$\text{box}_W(x) = \begin{cases} 1 & \text{for } -W/2 \leq x \leq +W/2, \\ 0 & \text{otherwise} \end{cases} \tag{1.43}$$

whose spectrum is a sinc function:

$$\text{box}_T(t) \leftrightarrow \text{sinc}\left(\frac{\omega}{2\pi F}\right) \quad \text{where } F = 1/T \tag{1.44}$$

because

$$\text{box}_T(t) \leftrightarrow \int_{-T/2}^{+T/2} e^{-j\omega t}\, dt = \left[\frac{e^{-j\omega t}}{-j\omega}\right]_{t=-T/2}^{t=+T/2} = \frac{e^{+j\omega T/2} - e^{-j\omega T/2}}{j\omega} = \frac{\sin(\omega T/2)}{\omega T/2} = \text{sinc}\left(\frac{\omega}{2\pi F}\right).$$

Moreover, the spectrum of a sinc function is a box function, and in the discrete-time case we have

$$\text{sinc}(F_s t) \leftrightarrow \text{box}_{2\pi F_s}(\omega) \tag{1.45}$$

because

$$\frac{1}{2\pi F_s} \int_{-\pi F_s}^{+\pi F_s} e^{+j\omega t}\, d\omega = \frac{1}{2\pi F_s} \left[\frac{e^{+j\omega t}}{jt}\right]_{\omega=-\pi F_s}^{\omega=+\pi F_s} = \frac{e^{+j\pi F_s t} - e^{-j\pi F_s t}}{2j\pi F_s t} = \frac{\sin(\pi F_s t)}{\pi F_s t} = \text{sinc}(F_s t).$$

This justifies the choice for the reconstructor in Section 1.1.3. Indeed, sampling a continuous signal $s(t)$ at a sampling frequency F_s can be regarded as multiplying this signal with an impulse train $\delta_{T_s}(t)$. Thus the spectrum of the sampled version $\tilde{s}$ is the convolution of the spectrum $S(\omega)$ by an impulse train $\delta_{2\pi F_s}(\omega)$, resulting of periodic copies of S centered around frequencies which are multiples of $\Omega_s = 2\pi F_s$ (see Figure 1.4). To reconstruct the continuous signal from its discrete version, we must recover its (non periodic) spectrum by filtering out all the copies but the one centered at frequency 0. Since the width of the spectrum S is $\Omega_s = 2\pi F_s$, it can be done by multiplying the spectrum by a box of this width. Thus the continuous signal $s(t)$ is the convolution of the sampled signal $s[n]$ (with zeroes between the known samples) with the signal corresponding to the box of width $2\pi F_s$, that is the ideal reconstructor $\text{sinc}(F_s t)$ of Equation (1.2). This also explains the Nyquist frequency $F_s/2$ (in Hz), since the reconstruction is possible only if the copies do not overlap (see Figure 1.5).

1.4 Acoustics

Objectively, sound is a physical phenomenon with a mechanical origin – an acoustic source – producing a local perturbation of the pressure which propagates into the air, thus giving birth to an acoustic wave. The sound signal in the temporal domain can be seen as the recording of the pressure level in time at some point (*e.g.* a microphone).

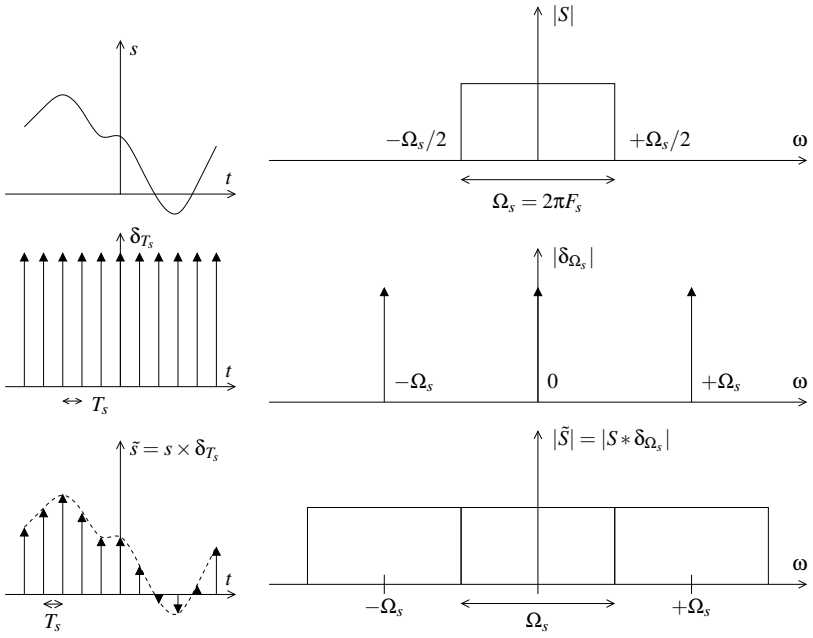


Figure 1.4: Uniform sampling with sampling period T_s, showing the temporal (left) and schematic spectral (right) representations of the continuous-time signal s (top), the sampling signal δ_{T_s} (middle), and the sampled signal $\tilde{s}$ (bottom).

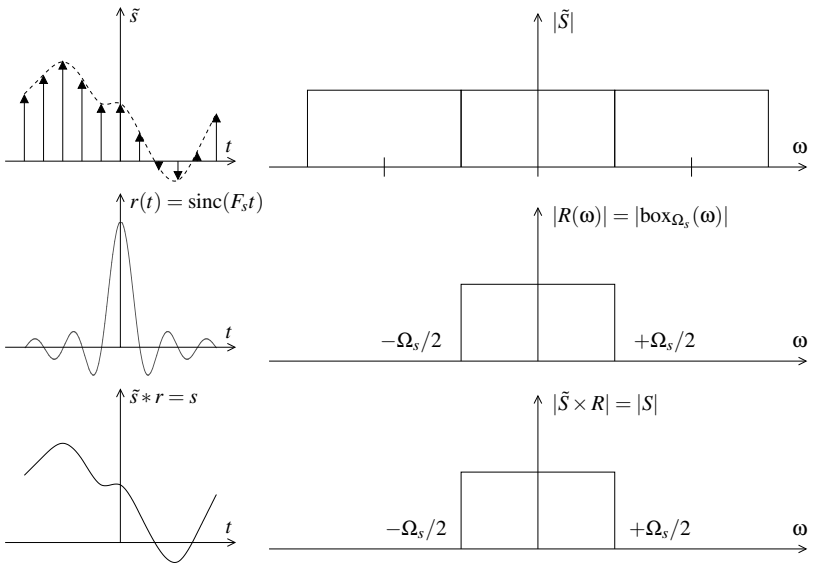


Figure 1.5: Uniform reconstruction of the sampled signal of Figure 1.4, showing the temporal (left) and schematic spectral (right) representations of the sampled signal $\tilde{s}$ (top), the reconstructor signal r (middle), and the reconstructed signal s (bottom).

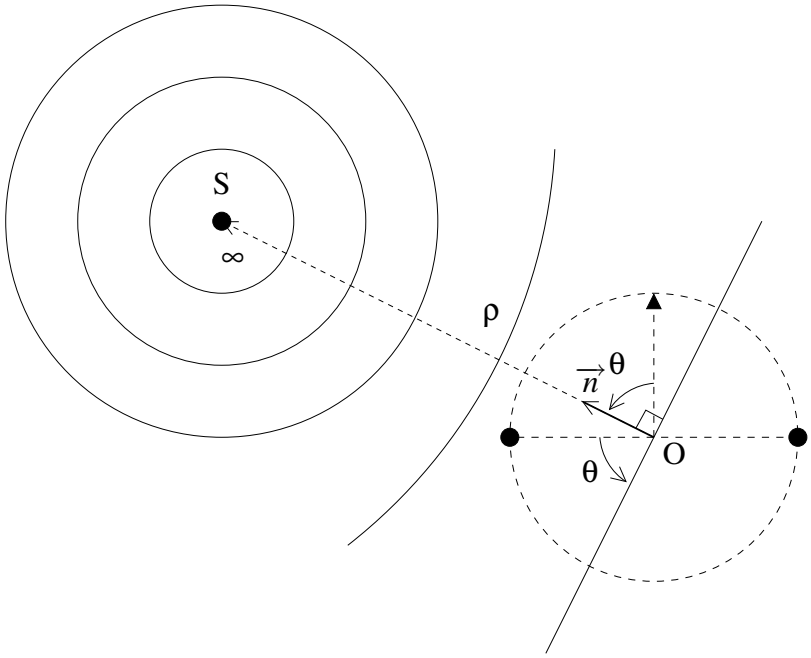


Figure 1.6: Punctual source S located at a distance ρ of the head center O, with the azimuth θ, and no elevation, propagating acoustic waves to the head.

1.4.1 Sound Source

We consider the sound source as punctual. In a polar coordinate system (see Figure 1.6), the source point is localized given its (ρ, θ, ϕ) coordinates, where ρ is the distance between the head center (O) and the source (S), θ is the azimuth angle, and ϕ the elevation angle. However, we will not consider any elevation ($\phi = 0$), and thus the notation ϕ can be kept for the phase. Indeed, as a first approximation in most musical situations, both the listeners and instrumentalists are standing on the (same) horizontal ground, with no relative elevation. Moreover, we consider the sound source as omni-directional. The sound source radiates spherical acoustic waves, that propagate to the ears through an energy transmission between air particles of the surrounding environment. We consider that the distance ρ is large enough for the acoustic wave to be regarded as planar when reaching the listener.

1.4.2 Wave Propagation

We consider that we are in outdoors conditions. We consider neither any room nor any obstacle (free-field case). We consider that acoustic waves are propagating in the air, at temperature T_c, pressure P_s, and relative humidity H_r. Standard values for these parameters are $T_c = 20°\text{C}$ (degrees Celsius), $P_s = 101325$ Pa (Pascals), and $H_r = 50\%$ (percents).

1.4.2.1 Delay

With these conditions, the acoustic waves propagate in the air at a speed $c \approx 343$ m/s, "the speed of sound" (Mach 1). As a consequence, covering the distance ρ takes a time $\Delta_t = \rho/c$, and the listener hears the sound that was produced by the source Δ_t seconds before.

1.4.2.2 Attenuation

In addition to this delay, the acoustic wave is subject to some attenuation.

Inverse Square Law

For spherical acoustic waves, the sound intensity is inversely proportional to ρ^2 and thus its amplitude is inversely proportional to ρ: the amplitude of the source is divided by 2 (decreases by ≈ -6dB) when measured at a distance ρ multiplied by 2. This is known as the inverse square law, applying to any point source which spreads its influence equally in all directions without a limit to its range. This comes from strictly geometrical considerations: the intensity of the influence at any given radius ρ is the source strength divided by the area of the sphere.

Air Attenuation

Moreover, there is also a frequency-selective attenuation by the air, roughly proportional to f^2, where f is the frequency in Hz. Since we consider that ρ is large enough for the waves to be regarded as planar when reaching the ears, the attenuation factor is given by the ISO 9613-1 norm [ISO93] for given air temperature, humidity, and pressure conditions. The total absorption $\mathcal{A}(\rho, f)$ in dB is

$$\mathcal{A}(\rho, f) = 20 \log_{10}\left(e^{\alpha(f)}\right) \cdot \rho \tag{1.46}$$

where α is the the absorption coefficient in nepers per meter, given by

$$\begin{aligned}\alpha(f) = F^2 P_s \Big\{ & 1.84 \cdot 10^{-11} (T_K/T_0)^{1/2} P_0 + (T_K/T_0)^{-5/2} \times \\ & \Big[0.01275 e^{-2239.1/T_K} / \left(F_{r,O} + (F^2/F_{r,O})\right) \\ & + 0.1068 e^{-3352/T_K} / \left(F_{r,N} + (F^2/F_{r,N})\right) \Big] \Big\}\end{aligned} \tag{1.47}$$

where $F = f/P_s$ is the frequency scaled by the atmospheric pressure P_s, P_0 is the reference atmospheric pressure (1 atm), T_K is the atmospheric temperature in degrees Kelvin (K), T_0 is the reference atmospheric temperature (293.15K), $F_{r,O}$ and $F_{r,N}$ are the scaled relaxation frequency for molecular oxygen and nitrogen, respectively given by

$$F_{r,O} = 24 + 4040 H_a (0.02 + H_a)/(0.391 + H_a), \tag{1.48}$$

$$F_{r,N} = (T_0/T)^{1/2} \left(9 + 280 H_a \exp\left(-4.17 (T_0/T)^{1/3} - 1\right)\right) \tag{1.49}$$

with

$$T_K = T_c + 273.15, \tag{1.50}$$

$$P_{\text{sat}} = 10^{\left(-6.8346 (T_1/T)^{1.261} + 4.6151\right)}, \tag{1.51}$$

$$H_a = H_r \frac{P_{\text{sat}}}{P_s} \tag{1.52}$$

where P_{sat} is the saturation pressure, $T_1 = 273.16$K is the triple-point isotherm temperature, and H_a is the absolute humidity. See [BSZ+95] for further details.

1.4.3 Doppler Effect

If either the source or the listener move, the Doppler effect may affect the perceived frequency of the sound source (see Figure 1.7). To obtain the perceived frequency, the following Doppler factor has to be applied (multiplied) to the frequency of the source:

$$\text{Doppler} = \frac{c + \overrightarrow{n} \cdot \overrightarrow{v}_{\text{O}}}{c + \overrightarrow{n} \cdot \overrightarrow{v}_{\text{S}}} \tag{1.53}$$

where $\overrightarrow{n}$ is the normalized listener-to-source orientation vector, $\overrightarrow{n} = \overrightarrow{\text{OS}}/|\text{OS}|$ (see Figure 1.6), $\overrightarrow{v}_{\text{O}}$ and $\overrightarrow{v}_{\text{S}}$ are the speed vectors of respectively the listener and the source, and $\cdot$ denotes the scalar product among vectors.

1.5 Psychoacoustics

Finally, the acoustic wave reaches the receptor: the ear. Figure 1.8 shows a schematic view of the human ear. The temporal domain is well-suited for the outer ear, that is until the tympanum. Indeed, a microphone placed in the ear canal at the level of the tympanum can record the amplitude of the sound as a function of time (see Section 7.7). But when the sound enters the inner ear, then the spectral domain is more convenient. Indeed, the cochlea is a key organ that performs a spectral decomposition. The sound enters the cochlea by the superior oval window, then propagates until it reaches the end of the chamber, then bounces and goes in the other direction until its reaches the inferior round window. The interference between the ingoing and outgoing waves is bending the basilar membrane inside the cochlea, at a position depending on the frequency of the sound (tonotopy), and squeezing there hair cells that transmit the information to the brain *via* the auditory nerve.

1.5.1 Perceptive Scales

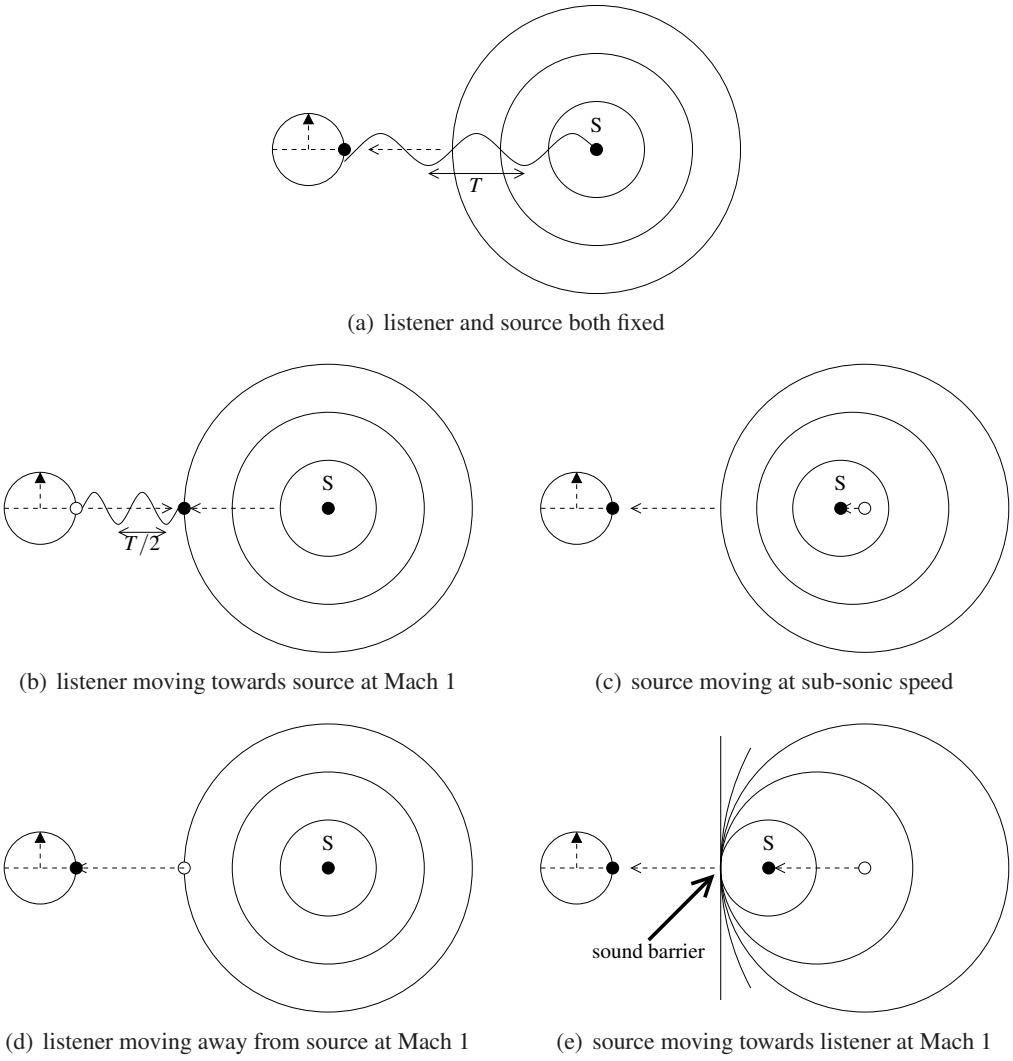
The Weber-Fechner law applies to every sensory organ and claims that the sensation is proportional to the logarithm of the excitation. As a consequence, human beings perceive the parameters (amplitude and frequency) of the spectral components on logarithmic scales. We are conscious that the subjective measures for amplitude and frequency are in fact loudness and pitch, respectively. Here we will introduce the objective decibel and Bark scales, close to the perception and well-suited to describe psychoacoustic phenomena.

1.5.1.1 Decibel Scale

The decibel (dB) scale is commonly used to represent the volume. The relation between the volume in dB and the linear amplitude is given by

$$V(a) = 20 \log_{10}\left(\frac{a}{A_{0\text{dB}}}\right). \tag{1.54}$$

If we consider that the maximal amplitude (1 in the linear scale) should correspond to a volume of 120dB in order to match the standard dB SPL (sound pressure level) scale, then we set $A_{0\text{dB}} = 10^{-6}$ corresponding to an acoustic pressure of $P_{0\text{dB}} = 2 \cdot 10^{-5}$ Pa (Pascals). Anyway, amplitude and pressure being proportional, it is just a matter of translation of the volume origin (0dB) in the logarithmic scale.



(a) listener and source both fixed

(b) listener moving towards source at Mach 1

(c) source moving at sub-sonic speed

(d) listener moving away from source at Mach 1

(e) source moving towards listener at Mach 1

Figure 1.7: Cases of moving listener or source: (a) reference case of fixed listener and source of frequency $F = 1/T$, (b) when the listener moves towards the fixed source at speed c, the observed period is halved and thus the heard frequency is doubled (Doppler effect); (c) when the source is moving, the same effect occurs, but the spherical waves are not concentric anymore – since the source point changes with time; (d) when the listener moves away from the source at speed c, then nothing is heard (silence); (e) when the source moves towards the listener at speed c, the spherical waves become tangent and form the sound barrier, heard as a sonic boom by the listener.

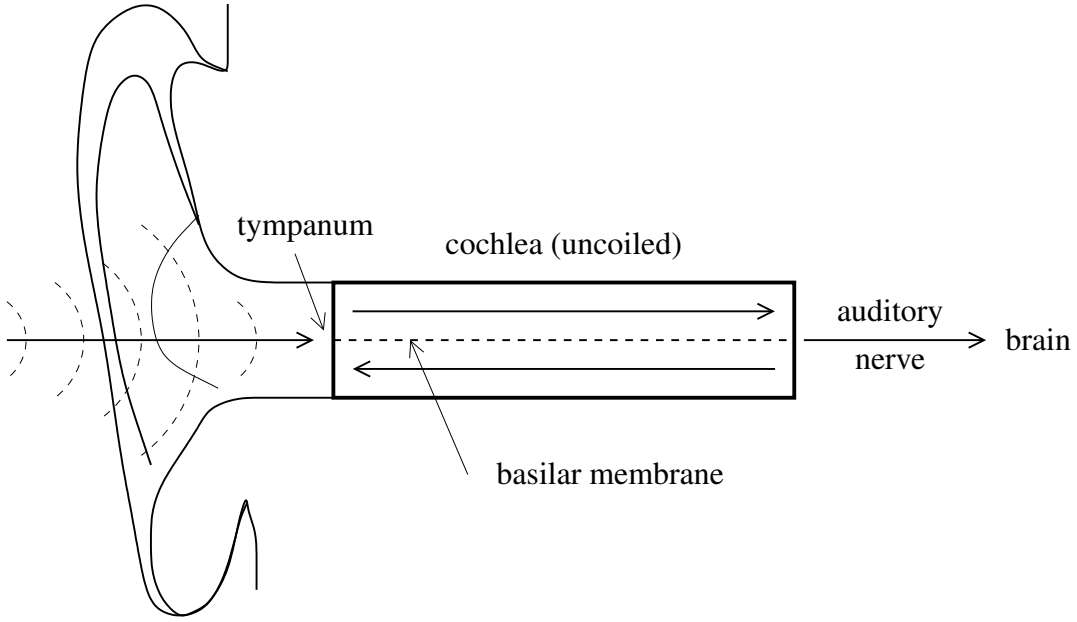


Figure 1.8: Schematic view of the section of the right ear. The middle ear has been omitted (because its role is mostly limited to the transmission and amplification of the sound to the inner ear). The cochlea has been uncoiled. Inside the cochlea is the basilar membrane that performs the frequency decomposition of the sound signals.

1.5.1.2 Bark Scale

A very convenient scale for representing frequencies is the Bark scale (after Barkhausen), which is very close to our perception [ZF81]. Equation (1.55) allows us to go from Hertz to Bark scales:

$$B(f) = \begin{cases} f/100 & \text{if} \quad f \leq 500, \\ 9 + 4\log_2(f/1000) & \text{if} \quad f > 500. \end{cases} \tag{1.55}$$

This is a simplification of the classic arctan-based formula for Hertz-to-Bark conversion, having the nice property of being easily invertible.

1.5.2 Threshold of Hearing

Human beings can hear frequencies in the range of 20Hz to 20kHz approximatively, but the sensibility threshold in amplitude $\mathcal{T}_a$ is a function of frequency (see Figure 1.9). Equation (1.56) provides us with a good approximation for this threshold. Spectral components with volumes below this threshold (expressed in dB) will not be heard:

$$\mathcal{T}_a(f) = 3.64(f/1000)^{-0.8} - 6.5e^{-0.6(f/1000-3.3)^2} + 10^{-3}(f/1000)^4. \tag{1.56}$$

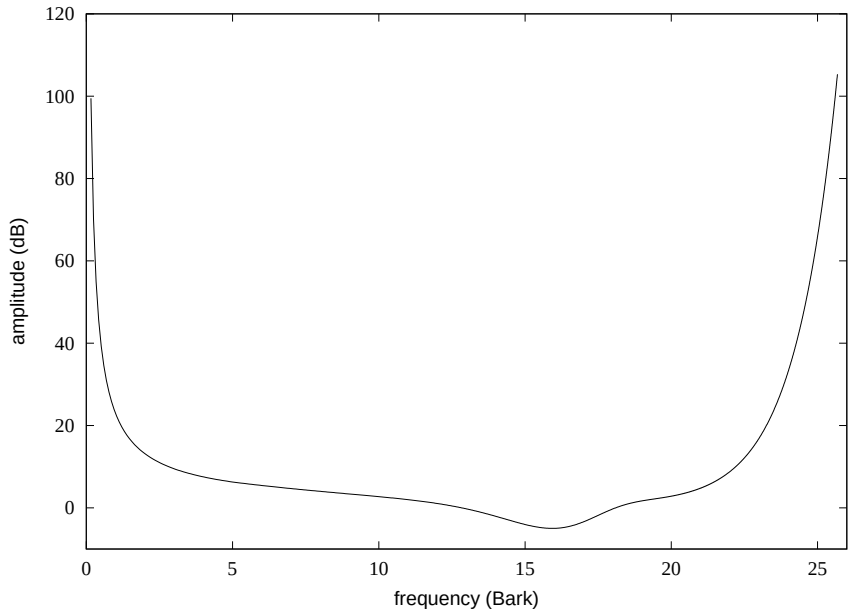


Figure 1.9: Threshold of hearing $\mathcal{T}_a$: a frequency at a given amplitude will not be heard if this amplitude is below the threshold.

1.5.3 Masking Phenomena

1.5.3.1 Frequency Masking

Physically, the addition of two signals of the same amplitude is ruled by a nonlinear addition law and gives a maximum of $+6$dB. However, from a perceptive point of view, there is a modification of the perception threshold for a sound m (masked sound) when it is played together with a louder sound M (masking sound). This phenomenon is known as simultaneous masking or frequency masking. Consider the case of M and m being two sinusoids of frequencies f_M and f_m, and amplitudes a_M and a_m, respectively. Assume that $a_M > a_m$. If f_m is close to f_M, the sound m is masked by the sound M and thus becomes inaudible.

As a first approximation we can consider that the masking threshold is close to a triangle in the Bark-dB scales. After Garcia and Pampin [GP99], we use a simple masking model to evaluate the signal-to-mask ratio (SMR) of each frequency component (see Figure 1.10). This model consists of

- the difference Δ between the level of the masker and the masking threshold (-10dB);
- the masking curve towards lower frequencies (left slope: 27dB/Bark);
- the masking curve towards higher frequencies (right slope: -15dB/Bark).

In fact, the left slope might change as a function of the volume (see [Moo98]). But this might also depend on frequency, which will make the model more complex. However, it turns out that with this value of Δ the simple model is very efficient.

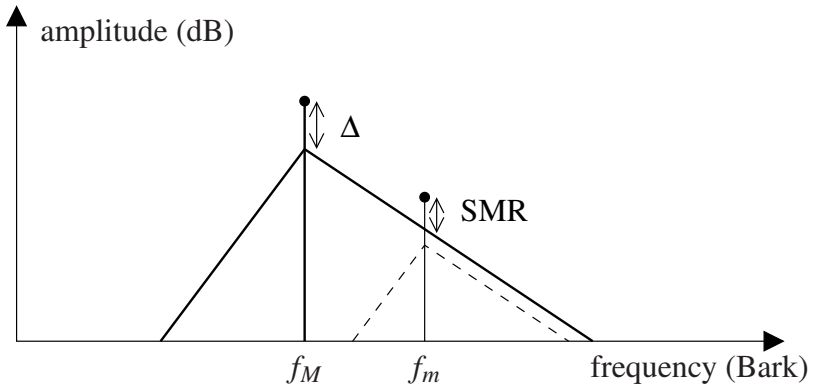


Figure 1.10: Frequency masking: the masking frequency f_M raises the threshold of hearing (triangular masking pattern), and here the frequency f_m is heard because its signal-to-mask (SMR) ratio is positive. Also notice that with our simplified triangular model, the masking pattern of f_m is fit into the masking pattern of f_M.

1.5.3.2 Temporal Masking

Another interesting phenomenon is temporal masking. There are two kinds of temporal masking. The post-masking occurs when the masking sound disappears. In fact, its effect persists in time during some milliseconds. As a consequence, even if the masking sound is not present anymore the masking effect is still present, although it decreases with time. Perhaps more surprisingly, pre-masking also exists. More precisely, the masking effect is active a few milliseconds before the masking sound really appears (our memory acting like a "buffer"). However this phenomenon is less important.

For now, we have not taken advantage of temporal masking. However, this phenomenon may explain why transients could be modeled in a rather rough way. For example, when a spectral frame is identified as containing a transient, Röbel [Röb03] sets all its phases in synchrony to zero. This produces a strong broadband masker, which will be heard as a transient, and the milliseconds of sound signal around this masker are of little importance since they will not be heard.

1.5.4 Modulation Threshold

The last psychoacoustic phenomenon we will consider in this document is the fact that we cannot hear the difference between two pure tones if they differ only in frequency with a difference below the frequency modulation threshold $\Delta_\omega = \max(2\text{Hz}, 0.0035\omega)$ (see [ZF81]).

Chapter 2

Elements of Music

When the acoustic sources are musical instruments, the vibrations caused by strings (violins, guitars, pianos, *etc.*), bars or rods (xylophones, metallophones, *etc.*), membranes (drums), plates or shells (gongs, bells, *etc.*), or air in a tube or pipe (trumpets, horns, pipe organs, *etc.*) are often quasi-periodic, and consist of the superpositions of vibration modes being the solutions of the equations for these vibrating systems.

2.1 Musical Parameters

The spectra of the resulting sounds are often harmonic, as shown in Figure 2.1. The frequency components are forming a comb: the first frequency is at F (the fundamental frequency), and the other frequency components are multiples of this fundamental frequency ($f_h = hF$), and are called harmonics (h being the rank of the harmonic with frequency f_h). An harmonic spectrum can thus be regarded as an impulse train in the spectral domain, multiplied by some spectral envelope C (or color) resulting from both the source and the resonant body of the instrument acting as a filter.

In case of sliding friction or viscous drag, the amplitudes corresponding to these frequencies are damped exponentially with time, and more precisely

$$a_h(t) = a_h(0) \cdot e^{-\alpha t} \tag{2.1}$$

where $\alpha \geq 0$ is the damping factor [FR98].

In the case of real non-ideal physical vibrators, the harmonic partials are often not in exact integral ratios. For example for stretched strings the frequency of an overtone obeys

$$f_h = hF\sqrt{1 + (h^2 - 1)\beta} \tag{2.2}$$

where $\beta \geq 0$ is the inharmonicity factor [FR98]. Equation (2.2) means that the frequency components are not found at harmonic positions anymore, but are gradually shifted upwards in the spectrum.

We call *partials* these spectral structures, each partial corresponding to a frequency component together with its associated amplitude. In the case of harmonic sounds, these partials correspond to the harmonics.

2.1.1 Pitch

Pitch is not an objective physical property, but a subjective psychophysical attribute of sound. However, in the case of harmonic sounds, it is strongly related to the fundamental frequency F. Again,

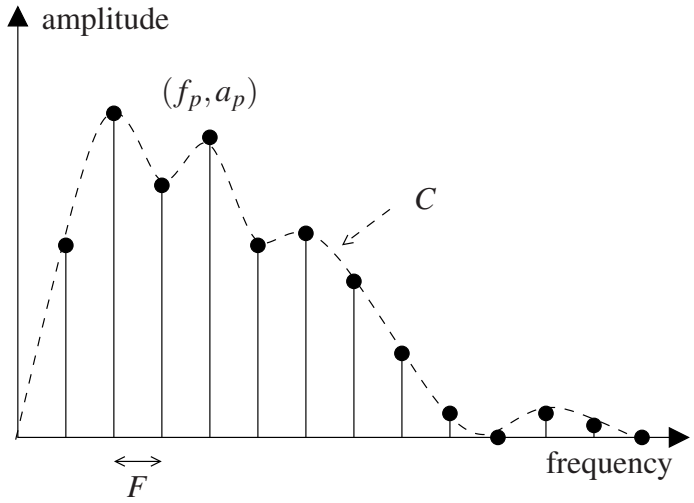


Figure 2.1: An harmonic sound (at time t) with its frequency F and color C.

pitch is perceived on a log scale. For a given fundamental frequency F, the corresponding pitch is

$$\mathcal{P}(F) = \mathcal{P}_{\text{ref}} + O \log_2\left(\frac{F}{F_{\text{ref}}}\right) \tag{2.3}$$

where $\mathcal{P}_{\text{ref}}$ and F_{ref} are, respectively, the pitch and the corresponding frequency of a tone of reference. We will use the standard values $\mathcal{P}_{\text{ref}} = 69$ and $F_{\text{ref}} = 440$Hz. The constant O is the division of the octave. An usual value is $O = 12$, leading to the classic dodecaphonic (twelve-tone) musical scale. With these values, $\mathcal{P}$ is the MIDI pitch [IMA88], where 69 corresponds to the `A4` note, 70 to `A#4`, *etc.*

2.1.2 Loudness

Loudness is not an objective physical property, but a subjective psychophysical attribute of sound. In this manuscript, we will not consider the loudness, but the volume $V(A)$ which is the root mean square (RMS) amplitude on the dB scale (see Equation (1.54)). For a single partial p, in the complex case (see Section 3.1.1), this amplitude is a_p. In Section 2.4.1 of the Ph.D. manuscript [Mar00], we have shown that the RMS amplitude of the set of P partials (with distinct frequencies) is

$$\mathcal{A} \approx \sqrt{\sum_{p=1}^{P} (a_p)^2}. \tag{2.4}$$

2.1.3 Timbre

By definition, timbre is what allows us to differentiate two sounds with the same loudness and pitch. Although this definition is not clear, it becomes tractable in the harmonic case. Indeed, pitch imposes a certain fundamental frequency, thus the spacing of the comb of partials; and amplitude imposes the global scale. The only (multi-dimensional) degree of freedom is the amplitude ratio of the different harmonics, captured by the spectral envelope (see Figure 2.1).

2.1.3.1 Spectral Envelope or Color

In Section 2.4.1 of the Ph.D. manuscript [Mar00], we use the word "color" for the spectral envelope. This seems natural for musicians who often characterize timbre simply as "tone color", with adjectives such as "sharp", "dull", "bright". However, a lot of work is still needed to scientifically define and characterize this color. Indeed, the spectral envelope is a function that interpolates the partials, but there are many candidates fulfilling this requirement. Moreover, the perception and coding of the sound color is still an open issue. Quoting the Ph.D. manuscript:

> For example, the video color is often encoded – by taking perception into account – as a Red+Green+Blue (RGB) value. It could be possible to encode in the same way the audio color by using 24 critical bands (...), that is with a set of 24 values instead of 3 for RGB.

Since then, some attempts have been done, *e.g.* by Teresawa *et al.* [TSB05] who propose to use the 13 first MFCCs (Mel-Frequency Cepstral Coefficients). The perceptual encoding of the color is an extremely important problem.

The *formants* correspond to the local maxima of the spectral envelope (for example, on Figure 2.1 there are 4 formants). These formants play a key role in speech for the recognition of vowels. They are also very important for instrumental sounds.

2.1.3.2 Spectral Centroid or Brightness

Since color is multi-dimensional and without any standardized encoding yet, we have to deal with some characteristics ("descriptors" or "features") that can summarize it. The brightness is probably the most important one [Gre75, Wes79]. It is related to the spectral centroid, given (in Hz) for a set of partials by

$$\mathcal{C} = \frac{\sum_p f_p \cdot a_p}{\sum_p a_p} \tag{2.5}$$

and for a discrete spectrum S by

$$\mathcal{C} = \frac{\sum_m m \cdot |S[m]|}{\sum_m |S[m]|} \cdot \Omega_{F_s,N}. \tag{2.6}$$

2.2 Modulations

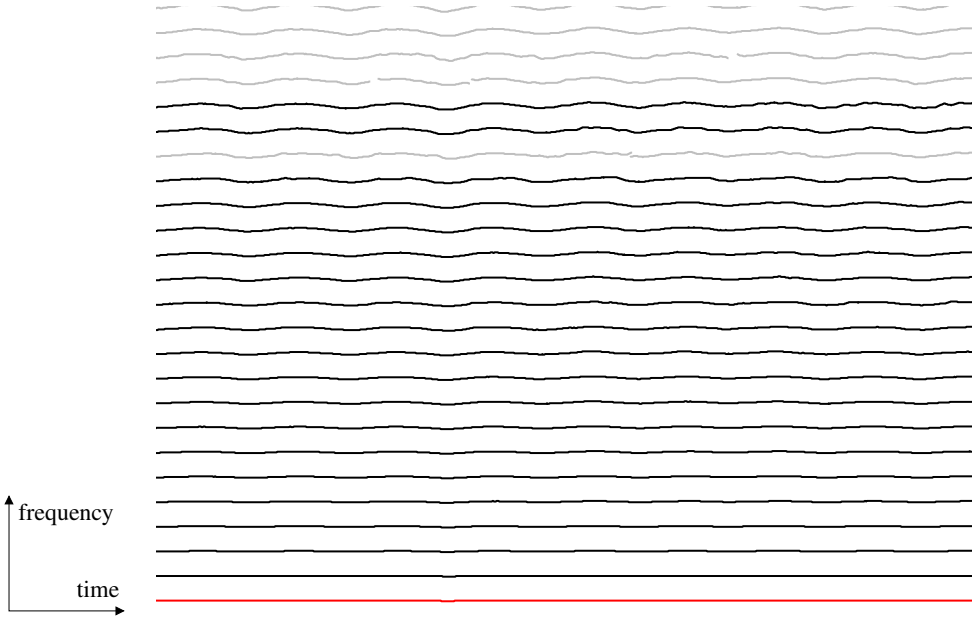
As shown in Figure 2.2, the parameters (frequencies and amplitudes) of the partials evolve in time. Moreover, they are often modulated.

2.2.1 Vibrato

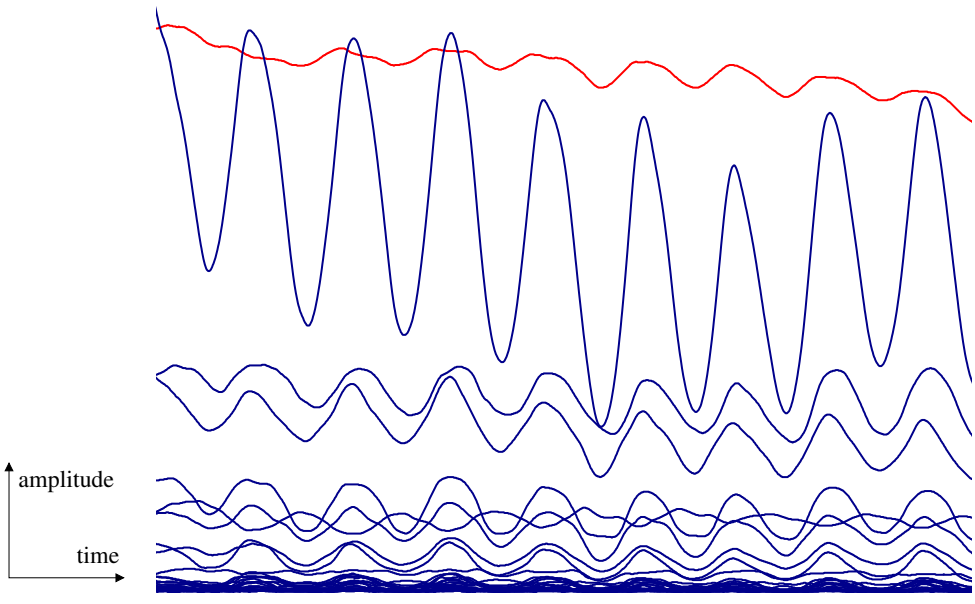
The macroscopic variations of the frequency over time constitute the melody of the music. When the variation is a sinusoid with a frequency around 10Hz, the musical effect is a vibrato. When this variation is monotonous (mathematically speaking), then the musical effect is a glissando or a portamento.

2.2.2 Tremolo

The variations of the amplitude over time constitute the dynamic of the music. When the variation is a sinusoid with a frequency around 10Hz, the musical effect is a tremolo. When this variation is



(a) frequencies



(b) amplitudes

Figure 2.2: The evolutions of the partials of an alto saxophone during ≈ 1.5 second. The frequencies (a) and amplitudes (b) are displayed as functions of time (horizontal axis).

monotonous (again mathematically speaking), then depending on whether it is increasing or decreasing, the musical effect is either a fade-in or a fade-out.

2.3 Note Onset / Offset

The onset and offset of a note correspond to the starting and ending times for this note, respectively. Onset detection is an important point for music transcription (see Chapter 8).

Part II

Sinusoidal Modeling

The second part of this manuscript deals with my main research subject: sinusoidal modeling of musical sounds[1].

Chapter 3 presents the short-term sinusoidal (STS) representation, with the associated analysis and synthesis methods. Regarding the model, we focus on spectral *peaks* and we fully consider the non-stationary case, where the parameters can evolve even within a short time window.

Three main non-stationary analysis methods are then described: quadratic interpolation, spectral reassignment, and our contribution – the **derivative method** (Section 3.2.3.3) we proposed together with Desainte-Catherine [10, 1], enhanced together with Lagrange and Rault [36], and recently extended to the non-stationary case together with Depalle [46]. We started the comparison of analysis methods together with Keiler [23] and did a more extensive survey together with Lagrange [37, 6], pointing out some equivalences among these analysis methods based on the short-time Fourier transform (STFT). It turns out that their precision is close to optimal under simple (first-order linear) non-stationary conditions. In the future, we plan to study the behavior of these analysis methods with more complex amplitude / frequency modulations (sinusoidal variations, tremolo / vibrato), and to continue the exploration of equivalences between methods and the exhaustive comparison of the STFT-based estimators, in order to find the most efficient one in terms of both precision and complexity. To reduce the complexity, we need an efficient way to compute an approximation of the Γ_w function of Equation (3.31) for any analysis window w, that should be derivable thanks to the properties of the Gaussian.

Regarding the synthesis, we focus on temporal methods such as the digital resonator and our **efficient polynomial approach** (Section 3.3.2.2) together with Robine and Strandh [38, Rob06], using a very efficient data structure. Part of our research projects is the implementation of a hybrid method (based on the two previous methods), with an extremely interesting complexity. To reduce the size of the problem (the number of partials to synthesize), we proposed together with Lagrange [20, Lag04] (during his Master) **to take advantage of psychoacoustic phenomena** (Section 3.3.3), in a very efficient way again thanks to an appropriate data structure. During Robine's Master, we also investigated the possibility to use non-linear methods, without real success. However, the non-linear source / linear filter approach presented at the very end of this chapter might deserve further research.

Chapter 4 presents the long-term sinusoidal (LTS) representation, with the associated analysis and synthesis methods. Regarding the model, we focus on *partials*.

Regarding the analysis, my Ph.D. [Mar00] served as a starting point for the main research done with Lagrange during his Ph.D. [Lag04]. Together, we proposed an **enhanced partial-tracking algorithm** using **linear prediction** (Section 4.2.3, [24, 27, 7]) and the control of the **high-frequency content** (Section 4.2.4, [31, 7]). These techniques turned out to be extremely useful for **musical sound restoration** (Section 4.2.3.3, [4]) and **note onset detection** (Section 4.2.4.3), respectively. Computer scientists have a key role to play with these partial-tracking algorithms, dealing more with algorithmics than signal processing. Probably because of that, and because of the fact that the problem of partial tracking is still ill-posed, this more than 20-year old research area seems in its infancy. Although we initiated a consequent effort on the evaluation of analysis algorithms in the long-term case (Section 4.2.5, [41]), this is still a major open issue and part of our future research subjects.

Regarding the synthesis, together with Girin *et al.* [25], we **extended the classic piecewise polynomial approach to the non-stationary case** (Section 4.3). Although the results were promising in theory, they were quite disappointing in practice and thus we reverted to the simplest – and faster – model for the synthesis. However, these practical results were obtained at a time where the precise analysis methods (see above) were not available. Thus, we should now proceed to new experiments.

[1] With contributions to the French ANR DESAM project – see *Curriculum Vitæ* for details.

Chapter 3

Short-Term Sinusoidal Modeling

Sinusoidal modeling has solid mathematical, physical, and physiological bases. It derives from Helmholtz's research and is rooted in the Fourier's theorem, which states that any periodic function can be modeled as a sum of sinusoids at various amplitudes and harmonically related frequencies. Here we will consider the sinusoidal model under its most general expression, which is a sum of complex sinusoids (the *partials*) with time-varying amplitudes and frequencies not necessarily harmonically related. The associated representation is in general orthogonal (the partials are independent) and sparse (a few amplitude and frequency parameters can be sufficient to describe a sound consisting of many samples), thus very computationally efficient. Each partial is a pure tone, part of the complex sound. The partials are sound structures very important both from the production (acoustics) and perception (psychoacoustics) points of views. In acoustics, they correspond to the modes of musical instruments, the superpositions of vibration modes being the solutions of the equations for vibrating systems (*e.g.* strings, bars, membranes). In psychoacoustics, the partials correspond to tonal components of high energy, thus very important for masking phenomena (see Section 1.5). The fact that the auditory system is well adapted to the acoustical environment seems quite natural.

With short-term modeling, the validity of the model (see below) is limited to a short time interval (here around time 0). This time interval is also called a time *frame*. For a given frame, the partials appear as peaks in the corresponding magnitude spectrum. Then, by sliding the time origin, we would have a frame-by-frame approach and then switch to long-term modeling (see Chapter 4).

3.1 Model

When partials are present at frame k, spectral peaks (or atoms) can be observed (see Figure 1.3). The i-th peak of frame k can be described by a set of instantaneous phase, amplitude, and frequency parameter values (measured at the center of the frame number k)

$$\mathcal{P}_{i,k} = (k, \phi_{i,k}, a_{i,k}, \omega_{i,k}) \tag{3.1}$$

and the k-th frame is then the set of peaks

$$\mathcal{S}_k = \bigcup_i \{\mathcal{P}_{i,k}\}. \tag{3.2}$$

Finally, the short-term sinusoidal (STS) representation is the set of the peaks of all the frames

$$\mathcal{S} = \bigcup_k \mathcal{S}_k. \tag{3.3}$$

For now, let us focus on a single frame centered at time $t = 0$, where the sound signal s is given by

$$s(t) = \sum_{p=1}^{P} a_p(t)\cos(\phi_p(t)) \tag{3.4}$$

$$\text{with} \quad \phi_p(t) = \phi_p(0) + \int_0^t \omega_p(u)\,du \tag{3.5}$$

$$\textit{i.e.} \quad \omega_p(t) = \frac{d\phi_p}{dt}(t) \tag{3.6}$$

where the functions a_p, ω_p, and ϕ_p are the instantaneous amplitude, frequency, and phase of the p-th partial, respectively. (The amplitude and frequency are positive.) The terminology "sinusoidal modeling" comes from the fact that the sine function is often used in Equation (3.4). However, we prefer to use the cosine function instead, for the sake of simplicity in mathematical derivations (see below). These functions are indeed the same apart a phase difference of $\pi/2$, since

$$\sin(\theta) = \cos\left(\theta - \frac{\pi}{2}\right), \tag{3.7}$$

$$\cos(\theta) = \sin\left(\theta + \frac{\pi}{2}\right). \tag{3.8}$$

3.1.1 Real / Complex Cases

In practice, sounds are real-valued signals. In order to analyze them (see Section 3.2), we will switch to the spectral domain using a short-time Fourier transform. For example, a real sinusoid of frequency $\omega > 0$ corresponds to two complex exponentials of frequencies $+\omega$ and $-\omega$ (see Equation (1.8)). And because of the Hermitian symmetry of the spectrum of a real signal, only the half (*e.g.* corresponding to the positive frequencies) of the spectrum will be of practical interest.

Maintaining the Hermitian symmetry throughout the whole analysis / transformation / synthesis chain can be heavy. But a more important problem is that the negative frequencies can interfere in the spectrum with the positive frequencies, especially for frequencies close to 0. Indeed, the DC component (corresponding to frequency 0) of the Fourier transform is the sum of the signal. If the signal is real, then this term is also real, thus with a null phase (modulo π, since this real number can be either positive or negative). More generally, the phases of low-frequency partials are attracted to the value 0 (or π), thus causing a bias in the estimation.

The DC component will always be problematic, but fortunately it can either be neglected (when we are considering zero-mean signals) of modeled separately (*e.g.* using polynomials, see Section 5.1.3). Apart from this special DC case, we can limit the bias for low-frequencies and forget about the Hermitian symmetry by considering complex signals with only positive frequencies (complex numbers being also much simpler for mathematical derivations):

$$s(t) = \sum_{p=1}^{P} a_p(t)\exp(j\phi_p(t)). \tag{3.9}$$

Switching back to the real case of Equation (3.4) is just a matter of considering the real part of $s(t)$ in Equation (3.9). But, in practice, we have to deal with real signals and thus we face the inverse problem: the signals are real, and we have to switch to the complex case using an "analytic" signal, which has no negative frequency. This can be done using the Hilbert transform.

For example, let us consider a pure tone consisting of a real cosine. It may be converted into a positive-frequency complex sinusoid by adding a quarter-cycle shifted (sine) imaginary part, thanks

to Equation (1.6). For more complex signals, after Depalle we propose to use a filter which shifts each sinusoidal component by a quarter cycle. Ideally, this filter has magnitude 1 at all frequencies and introduces a phase shift of $-\pi/2$ at each positive frequency and $+\pi/2$ at each negative frequency. Given this frequency response of such an ideal phase shifter, its impulse response can be derived using the IDTFT (see Equation (1.21)):

$$\begin{aligned} h[n] &= \frac{1}{2\pi}\int_{-\pi}^{0} \underbrace{e^{+j\pi/2}}_{+j} \cdot e^{+j\underline{\omega}n} d\underline{\omega} + \frac{1}{2\pi}\int_{0}^{+\pi} \underbrace{e^{-j\pi/2}}_{-j} \cdot e^{+j\underline{\omega}n} d\underline{\omega} \\ &= \frac{j}{2\pi}\left[\frac{e^{j\underline{\omega}n}}{jn}\right]_{\underline{\omega}=-\pi}^{\underline{\omega}=0} - \frac{j}{2\pi}\left[\frac{e^{j\underline{\omega}n}}{jn}\right]_{\underline{\omega}=0}^{\underline{\omega}=+\pi} \\ &= \frac{1-\cos(\pi n)}{\pi n} \quad (n \neq 0) \end{aligned} \tag{3.10}$$

and $h[0] = 0$. It is easy to verify that this filter turns a cosine function into a sine function:

$$\cos(\omega t) = \frac{\exp(+j\omega t) + \exp(-j\omega t)}{2} \xrightarrow{h} \underbrace{-j}_{1/j} \frac{\exp(+j\omega t) - \exp(-j\omega t)}{2} = \sin(\omega t).$$

Computing the analytic signal $\tilde{s}$ from the real signal s is then done using

$$\tilde{s} = s + jh * s. \tag{3.11}$$

In practice, we use the classic "window method" (multiplying h with the Hann window, see Section 3.2.2.2) to obtain an approximation of the impulse response with a finite time support, and the convolution is efficiently computed as a multiplication in the spectral domain.

3.1.2 Stationary / Non-Stationary Cases

As a first approximation, for each partial p we can consider the amplitude and frequency (and phase) as polynomials regarded as Taylor expansions of more general functions of time.

In the stationary case, the polynomials are limited to their first (constant) coefficient:

$$a_p(t) = a_p(0) \tag{3.12}$$

$$\omega_p(t) = \omega_p(0) \quad \textit{i.e.} \quad \phi_p(t) = \phi_p(0) + \omega_p(0)t. \tag{3.13}$$

In the non-stationary case, to ease the mathematical derivations, the log-amplitude

$$\lambda_p(t) = \log(a_p(t)) \tag{3.14}$$

is considered and a linear term is added to the polynomials:

$$\lambda_p(t) = \lambda_p(0) + \mu_p(0)t \tag{3.15}$$

$$\omega_p(t) = \omega_p(0) + \psi_p(0)t \quad \text{i.e.} \quad \phi_p(t) = \phi_p(0) + \omega_p(0)t + \frac{\psi_p(0)}{2}t^2. \tag{3.16}$$

The μ term is also physically motivated by the fact that the modes of musical instruments are often exponentially damped in amplitude.

This Taylor-expansion approach could be generalized for higher-order amplitude / frequency modulations. However, increasing the order of the polynomials makes the estimations of the parameters

more complex (see Section 3.2) and this polynomial approach is somewhat useless for sinusoidal oscillations commonly observed in tremolo / vibrato (since no polynomial of finite degree can approximate a sine function exactly). Investigating sinusoidal modulations, and more generally periodic modulations (which can be decomposed into sum of sinusoids, thanks to the Fourier's theorem) is part of our future research projects. For now, we will present the latest results in the state-of-the-art linear log-amplitude / frequency modulation model. Thus, we consider peaks of the form

$$\mathcal{P}_{i,k} = (t_k, \phi_{i,k}, \lambda_{i,k}, \omega_{i,k}, \mu_{i,k}, \psi_{i,k})\,. \tag{3.17}$$

In the remainder, since we will focus on estimation precision and not on frequency resolution, we will consider the case of only one partial ($P = 1$) – one peak – and also one frame, and thus we can simplify the mathematical notations by using, for any parameter ν, $\nu_0 = \nu_p(0)$. Then the (short-term) elementary signal we will consider corresponds to one complex partial in the non-stationary case:

$$s_{\phi_0,\lambda_0,\omega_0,\mu_0,\psi_0}(t) = \exp\left(\underbrace{(\lambda_0 + \mu_0 t)}_{\lambda(t)=\log(a(t))} + j\underbrace{\left(\phi_0 + \omega_0 t + \frac{\psi_0}{2}t^2\right)}_{\phi(t)}\right). \tag{3.18}$$

3.2 Analysis

In order to efficiently model real existing sounds, the problem is now to be able to estimate the model parameters from these sounds with the best precision (and within a reasonable amount of time, although real-time analysis is not our priority).

3.2.1 Short-Term Fourier Spectrum

For each frame, the signal will be a sum of elementary signals (see Equation (3.18)) plus some noise. To isolate the corresponding frequency components, we must switch to the spectral domain using a short-time Fourier transform (STFT). Although the (continuous) STFT is defined in theory by

$$S_w(t,\omega) = \int_{-\infty}^{+\infty} s(\tau)\; w(\tau - t)\; e^{-j\omega\tau}\; d\tau = e^{-j\omega t}\int_{-\infty}^{+\infty} s(\tau)\; w(\tau - t)\; e^{-j\omega(\tau - t)}\; d\tau$$

we will use instead a slightly modified definition:

$$S_w(t,\omega) = \int_{-\infty}^{+\infty} s(\tau)\; w(\tau - t)\; e^{-j\omega(\tau - t)}\; d\tau \tag{3.19}$$

where the phase modification factor has been omitted. Indeed we let the time reference slide with the window, which results in a phase shift of ωt. This corresponds to the practical case, where the discrete STFT is implemented frame-by-frame using an FFT. This also simplifies the comparison of the analysis methods, without altering the quality of the estimations, all made at time $t = 0$.

The complex spectra resulting from this transform are, in polar and log-polar forms:

$$S_w(t,\omega) = a(t,\omega)\exp\left(j\phi(t,\omega)\right), \tag{3.20}$$

$$S_w(t,\omega) = \exp\left(\lambda(t,\omega) + j\phi(t,\omega)\right) \tag{3.21}$$

where the instantaneous amplitude a (positive) and log-amplitude λ, as well as the instantaneous phase ϕ, are real-valued functions of time t and frequency ω. The dependence on s and w will be omitted in the notations for the sake of simplicity. These functions obey the following relations:

$$\begin{aligned} a(t,\omega) &= |S_w(t,\omega)|, && (3.22)\\ \lambda(t,\omega) &= \log(a(t,\omega)) && (3.23)\\ &= \Re(\log(S_w(t,\omega))), && (3.24)\\ \phi(t,\omega) &= \angle(S_w(t,\omega)) && (3.25)\\ &= \Im(\log(S_w(t,\omega))). && (3.26) \end{aligned}$$

In practice, the STFT will be discrete, an we will consider

$$S_w[n,m] = S_w(nT_s, m\Omega_{F_s,N}) \tag{3.27}$$

where T_s is the sampling period and $\Omega_{F_s,N}$ is the width of one frequency bin, see Equation (1.23). Moreover, the transform will often hop by intervals of I samples (the hop size). An FFT of size N will be then applied of each frame centered on a given sample n (multiple of I).

3.2.2 Analysis Windows

As seen in Equation (3.19), the short-term spectrum S_w of the signal s involves an analysis window w, with a finite time support for S_w to be computable in practice. But the multiplication among time signals is equivalent to a convolution in the frequency domain (see Section 1.3.1.1). The analysis window w must also be band-limited in frequency in such a way that for any frequency corresponding to one specific partial (corresponding to some local maximum in the magnitude spectrum), the influence of the other partials can be neglected (in the general case when $P > 1$).

In practice, to compute the STFT around time $t = 0$, the signal $s_{\phi_0,\lambda_0,\omega_0,\mu_0,\psi_0}$ (see Equation (3.18)) is multiplied by the window w, thus leading to the signal frame

$$x(t) = s_{\phi_0,\lambda_0,\omega_0,\mu_0,\psi_0}(t) \cdot w(t) \tag{3.28}$$

and an FFT is then applied. From the expression of the Fourier transform (see Equation (1.18)) of x, taking out of the integration the constants in this expression, we obtain a spectrum X of the form

$$X(\omega) = s_0 \cdot \Gamma_w(\omega_0 - \omega, \mu_0, \psi_0) \tag{3.29}$$

where the complex amplitude is

$$s_0 = s_{\phi_0,\lambda_0,\omega_0,\mu_0,\psi_0}(0) = \exp(\lambda_0 + j\phi_0) \tag{3.30}$$

and the effects of the window and modulations are embedded in a complex function

$$\Gamma_w(\omega,\mu_0,\psi_0) = \int_{-\infty}^{+\infty} w(t) \exp\left(\mu_0 t + j\left(\omega t + \frac{\psi_0}{2}t^2\right)\right) dt \tag{3.31}$$

which, for almost every window w, seems to have no simple analytic form (except for the Gaussian window, see Section 3.2.2.3).

In the stationary case ($\mu_0 = 0$ and $\psi_0 = 0$), things are much simpler. Indeed, the stationary signal

$$s_{\phi_0,\lambda_0,\omega_0,0,0}(t) = \exp(\lambda_0 + j(\phi_0 + \omega_0 t)) \tag{3.32}$$

has the following spectrum:

$$S_{\phi_0,\lambda_0,\omega_0,0,0}(\omega) = \exp(\lambda_0 + j\phi_0) \cdot \delta_{\omega_0}(\omega) = s_0 \cdot \delta_{\omega_0}(\omega) \tag{3.33}$$

thus, for any window w, the windowed signal has the following spectrum:

$$X(\omega) = S_{\phi_0,\lambda_0,\omega_0,0,0}(\omega) * W(\omega) = s_0 \cdot \delta_{\omega_0}(\omega) * W(\omega) = s_0 \cdot \underbrace{W(\omega - \omega_0)}_{\Gamma_w(\omega_0 - \omega,0,0)} \tag{3.34}$$

which shows that the spectrum of the analysis window W is simply centered on the frequency ω_0 and scaled by the complex amplitude s_0.

If in the stationary case the Γ_w function is simply the spectrum W of the analysis window w shifted in frequency, in the non-stationary case this function is much more complex. Figure 3.1 shows the magnitude and phase spectra of the Hann window under different amplitude / frequency modulation conditions, that is the magnitude and phase of the Γ_w function for w being the Hann window. After Masri [MC98], we can notice the phenomenon of spectral distortion: amplitude modulation[1] changes the magnitude of the spectral peak and appears in the phase spectrum as an odd function of frequency; frequency modulation does not seem to change the value of the maximum in magnitude, and appears in the phase spectrum as an even function of frequency. Moreover, Masri noticed that these two phase modulations seem orthogonal and thus should combine linearly. Some preliminary attempts in non-stationary sinusoidal analysis considered the distortion of the resulting phase spectrum. The idea was then to consider the difference and sum of the phases of two frequency components symmetrically around the peak (*e.g.* ± 1 bin) to derive the amplitude and frequency modulations. Masri first proposed to derive them from direct measurements [MC98], and then together with Lagrange and Rault we proposed to use polynomial approximations [22]. But these were very preliminary attempts. More sophisticated non-stationary methods were proposed since, and will be presented in Section 3.2.3.

Peak Picking. Each partial has to be located in the spectrum. This is classically done by finding a local maximum in the magnitude spectrum $|X|$. Thus, the magnitude spectrum of the window $|W|$ (or $|\Gamma_w|$ in the non-stationary case) must exhibit a global maximum at frequency $\omega = 0$ (or at least close to 0).

One-Peak Property. Also, in theory, in order to have a bijection between the partials and the peaks, it is suitable that the window spectrum causes only one peak per partial in the discrete spectrum. This is trivially verified by the Gaussian window (see Section 3.2.2.3), because its spectrum is also a Gaussian, thus with only one maximum. For a discrete spectrum of N bins, this "one-peak property" can be expressed in the stationary case this way (W being then the continuous spectrum of the discrete window w):

$$\forall \omega \in \left[-\frac{\Omega_{F_s,N}}{2}, +\frac{\Omega_{F_s,N}}{2} \right], \text{ the 2 series } \{|W(\omega - k\Omega_{F_s,N})|\}_k \text{ and } \{|W(\omega + k\Omega_{F_s,N})|\}_k \text{ are decreasing} \tag{3.35}$$

[1] The amplitude modulation was then applied as a linear variation on the amplitude a, and not on the log-amplitude λ. But in practice this makes little difference for small modulation values, especially if the initial amplitude $a(0) = a_0 = 1$, since

$$\log(a_0 + b_0 t) = \underbrace{\log(a_0)}_{\lambda_0} + \log\left(1 + \frac{b_0}{a_0} t\right) \approx \lambda_0 + \underbrace{\frac{b_0}{a_o}}_{\mu_0} t$$

thus, for a small modulation and close to time $t = 0$, a linear amplitude modulation of b_0 is equivalent to a log-amplitude modulation with $\mu_0 = b_0/a_0$.

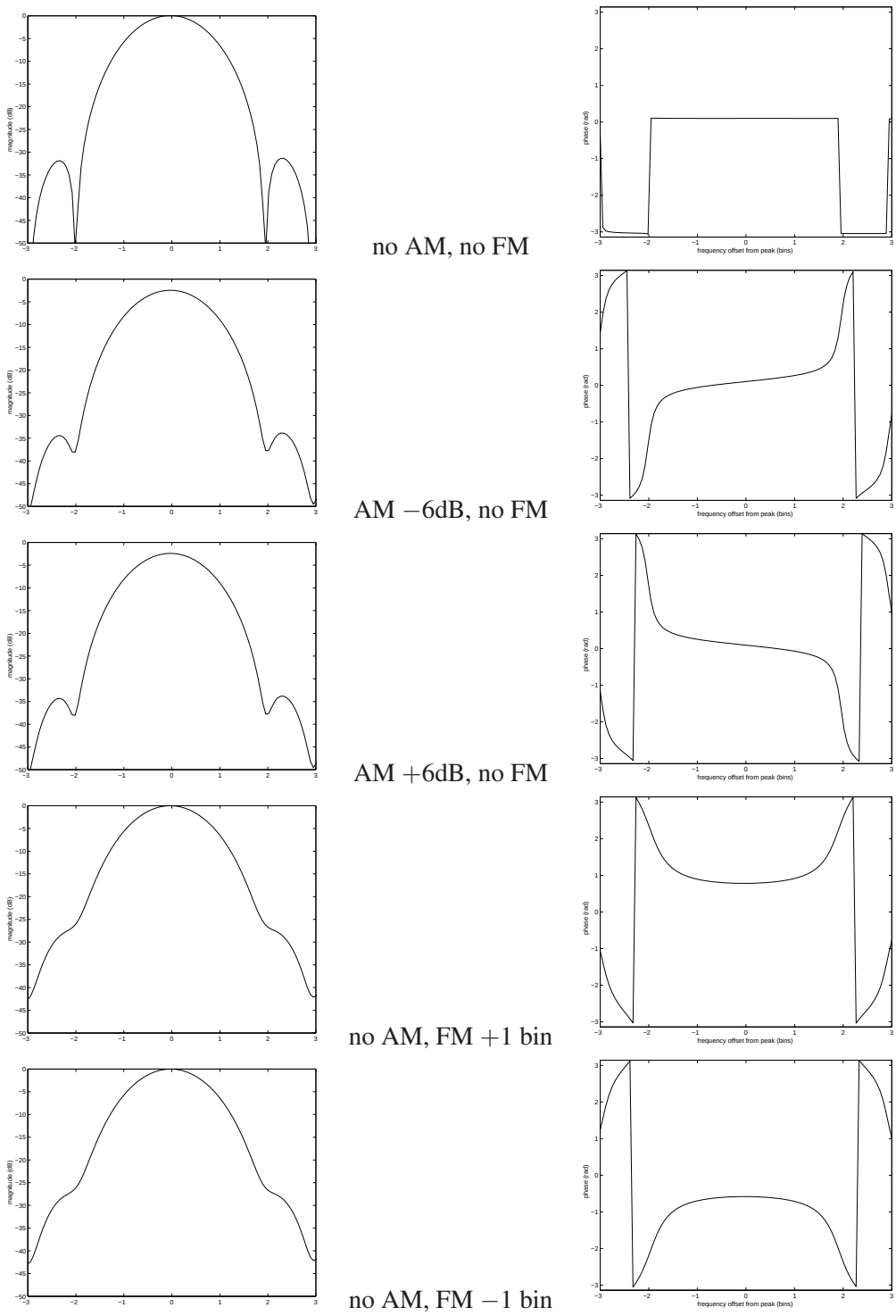


Figure 3.1: Magnitude (left) and (wrapped) phase (right) spectra of a sinusoid using the Hann window, with from top to bottom: no modulation, (linear) amplitude modulation of -6dB and $+6$dB per analysis frame, and frequency modulation of -1 and $+1$ bin per analysis frame (from the Ph.D. manuscript [Mar00]).

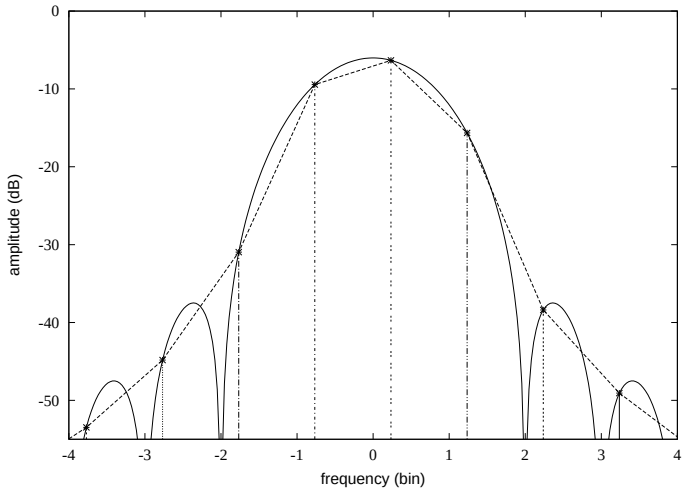


Figure 3.2: One-peak property of the Hann window (see Equation (3.35), here with $\omega = \frac{\Omega_{F_S,N}}{4}$).

and is also verified by the rectangular (see Section 3.2.2.1) and Hann (see Section 3.2.2.2 and Figure 3.2) windows (but not by other classic windows such as Barlett, Hamming, or Blackman). However, in practice, it is only required that the window causes no other local maximum above a certain magnitude threshold (which is the case for all these windows).

Peak Selection. The (modulated) window spectrum is often used for peak selection, to avoid spurious peaks caused by noise (stochastic signals, see Chapter 6). Indeed, if the peak corresponds to a sinusoid, then the magnitude spectrum around its location should follow the shape of the window spectrum. More precisely, a correlation of the spectrum X with the modulated window spectrum Γ_w locally around the peak can tell how much the peak is really close to a sinusoid.

In the remainder, we will require the window to be symmetric (thus with a real spectrum), and with a discrete version of odd size $N = 2H + 1$ (except for the Hann window, where an even size N can be regarded as a symmetric window of odd size $N - 1$ plus one extra zero – a kind of implicit zero-padding, see below).

3.2.2.1 Rectangular Window

Extracting a frame (of finite time support) from a signal (of infinite time support) without special care can be seen as considering the signal resulting from the multiplication of s by a rectangular window whose value is 1 for the considered time interval and 0 outside.

More precisely, the zero-centered (or zero-phase) rectangular window of length T (seconds) is defined by

$$w_{R,T}(t) = \begin{cases} 1 & \text{for } -T/2 \leq t \leq +T/2, \\ 0 & \text{otherwise.} \end{cases} \tag{3.36}$$

Since this window is not continuous, it is not differentiable (at least not at times $t = \pm T/2$).

The rectangular window is suitable for overlap-add with both deterministic and stochastic signals. More precisely, overlap-add is a well-known technique used for frame-by-frame processing (for details and references, see for example Section 4.1.2 of the Ph.D. manuscript [Mar00]). Frames of N samples (the frame size) are taken every I samples (the hop size), then processed, and the local result

is accumulated (added) at the corresponding time in the global result, using a window w (which can then be regarded as a weighting function). In case of linear transformations, the analysis window will be preserved in the local result and thus can also play the role of a synthesis window. If $I < N$, then the frames overlap, which produces a suitable interpolation effect among consecutive frames, to avoid discontinuities due to the processing. For the amplitude to be preserved, the condition is that the shifted versions of the window sum up to the constant 1. For the power (or variance for zero-mean signals) to be preserved – which is suitable for stochastic signals, as shown by Hanna [Han03], the condition is that the square of the shifted versions of the window sum up to the constant 1. Both conditions are trivially verified by the rectangular window for $I = N$. For other values of $I < N$, it is just a matter of scaling the result by some normalization coefficient.

The discrete-time version of the rectangular window is

$$w_{R,N}[n] = w_{R,NT_s}(nT_s) \tag{3.37}$$

where T_s denotes the sampling period (the inverse of the sampling rate F_s). The spectrum of this rectangular window can easily be obtained from the DTFT (see Equation (1.20)):

$$W_{R,N}(\omega) = \sum_{n=-\infty}^{+\infty} w_{R,N}[n] \cdot \exp(-j\underline{\omega}n) = \sum_{n=-H}^{+H} e^{-j\underline{\omega}n} = e^{-j\underline{\omega}H} \frac{1-e^{-j\underline{\omega}N}}{1-e^{-j\underline{\omega}}} \tag{3.38}$$

since we recognize the sum of the terms of a geometric series. Then, by rewriting the previous equation, since $H = (N-1)/2$ (N being odd), and by using Equation (1.8), we obtain

$$W_{R,N}(\omega) = \frac{e^{-j\underline{\omega}/2}}{e^{-j\underline{\omega}/2}} \frac{e^{+j\underline{\omega}N/2} - e^{-j\underline{\omega}N/2}}{e^{+j\underline{\omega}/2} - e^{-j\underline{\omega}/2}} = \frac{\sin(N\underline{\omega}/2)}{\sin(\underline{\omega}/2)} = N\mathrm{asinc}_N(\underline{\omega}) \tag{3.39}$$

where

$$\mathrm{asinc}_N(\underline{\omega}) = \frac{\sin(N\underline{\omega}/2)}{N\sin(\underline{\omega}/2)} \tag{3.40}$$

is the "aliased sinc" function (after Smith, see [Smi07]), which approaches the sinc function (see Equation (1.3)) as the sampling rate goes to infinity (the continuous-time case).

This spectrum has a very narrow main lobe of width $2\Omega_{F_s,N}$, but the side lobes are quite high (see Figure 3.3). Indeed, the first (and most prominent) side lobe is only at -13dB below the main lobe. Moreover, the side lobes roll off slowly, approximately 6dB per octave. For frequency resolution issues, it is preferable to use a window with side lobes which are lower and rolling off faster.

3.2.2.2 Hann Window

The zero-centered (or zero-phase) Hann window of length T (seconds) has also a simple analytic expression:

$$w_{H,T}(t) = \frac{1}{2}\left(1 + \cos(2\pi t/T)\right) \cdot w_{R,T}(t). \tag{3.41}$$

Moreover, unlike the rectangular window, it is differentiable (two times):

$$w'_{H,T}(t) = -\frac{\pi}{T}\sin(2\pi t/T) \cdot w_{R,T}(t), \tag{3.42}$$

$$w''_{H,T}(t) = -2\left(\frac{\pi}{T}\right)^2 \cos(2\pi t/T) \cdot w_{R,T}(t) \tag{3.43}$$

which is mandatory for the spectral reassignment method (see Section 3.2.3.2).

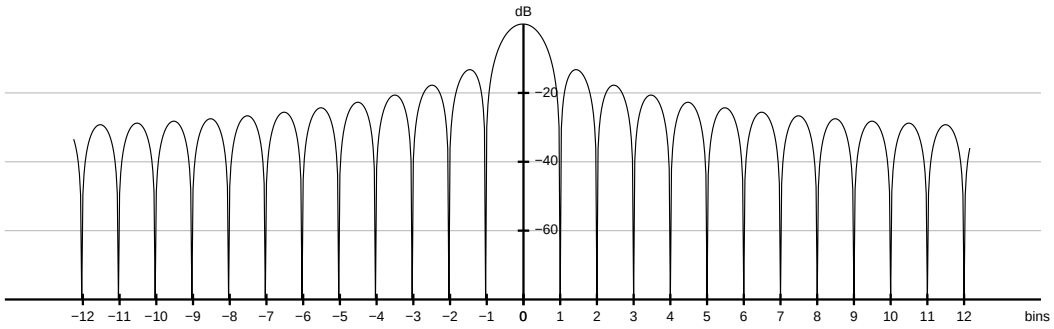


Figure 3.3: Magnitude spectrum of the rectangular window.

Like the rectangular window, the Hann window has a finite support, corresponding to one period of a trigonometric function (a property shared by the members of the family of trigonometric windows, such as Hamming or Blackman).

The Hann window is suitable for overlap-add, since it preserves the amplitude in the case of $I = N/2$ (see Section 4.1.2 of the Ph.D. manuscript [Mar00]), because

$$\sum_{k=-\infty}^{+\infty} w_{H,T}(t - kT/2) = 1. \tag{3.44}$$

An elegant way of demonstrating this is to remark that

$$\frac{1}{2}\left(1 + \cos(2\pi t/T)\right) = \left(\cos(\pi t/T)\right)^2 \tag{3.45}$$

and that shifting the time t by $T/2$ rotates the phase in the $\cos^2$ by $\pi/2$ rad, thus turning $\cos^2$ into $\sin^2$. The fact that $(\cos(\theta))^2 + (\sin(\theta))^2 = 1$ ends the demonstration.

Now we easily see that the square root of the Hann window is the (co)sine window, which is suitable for the overlap-add of stochastic signals since it preserves the power / the variance of zero-mean signals.

The discrete version of the Hann window (with sampling period T_s) is given by

$$w_{H,N}[n] = w_{H,NT_s}(nT_s) = \underbrace{\frac{1}{2}\left(1 + \cos(2\pi n/N)\right)}_{f_N(n)} \cdot w_{R,N}[n]. \tag{3.46}$$

With this definition, $N/2$ has to be an integer, thus N has to be even (which is interesting in practice, since a power of 2 is required by the classic implementation of the FFT, see Section 1.2.3.4). This way, this window can be regarded as an odd-length symmetric window preceded by some extra zero (zero padding, see Section 1.3.1.4). Also, since $f_N(N) = 0$, this makes no difference if a rectangular window of odd size $N+1$ is used in Equation (3.46).

The Hann window has also interesting spectral properties. Its spectrum has a simple analytic expression. Indeed, with the preceding remark and from Equations (3.46) and (1.7) – a cosine function being the sum of to complex exponentials, we have

$$w_{H,N}[n] = \frac{1}{2} w_{R,N+1}[n] + \frac{1}{4} w_{R,N+1}[n] \cdot \exp\left(+j\frac{2\pi n}{N}\right) + \frac{1}{4} w_{R,N+1}(t) \cdot \exp\left(-j\frac{2\pi n}{N}\right) \tag{3.47}$$

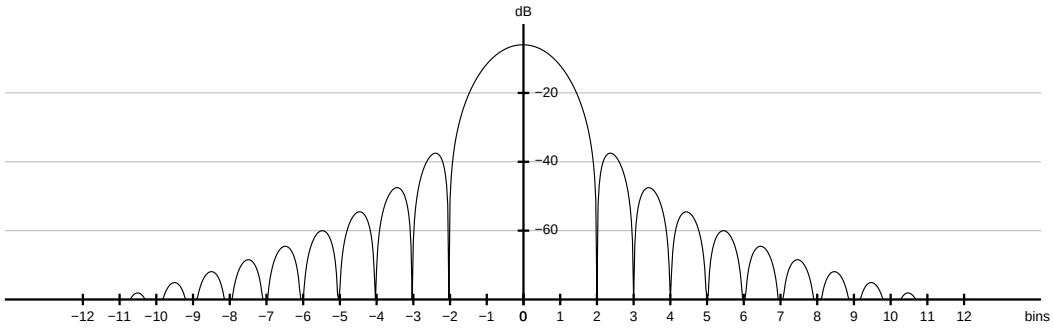


Figure 3.4: Magnitude spectrum of the Hann window.

and by switching to the spectral domain, we recognize a sum of shifted and scaled versions of 3 rectangular windows, and from Equation (1.37) we obtain

$$W_{H,N}(\omega) = \frac{1}{2}W_{R,N+1}(\omega) + \frac{1}{4}W_{R,N+1}\left(\omega - \Omega_{F_s,N}\right) + \frac{1}{4}W_{R,N+1}\left(\omega + \Omega_{F_s,N}\right). \tag{3.48}$$

It has a rather narrow main lobe of width $4\Omega_{F_s,N}$. And the first (and most prominent) side lobe is now at -31.5dB below the main lobe (see Figure 3.4). Moreover, the side lobes roll off approximately 18dB per octave. The Hann window has a remarkable discrete spectrum (of the form $[1/2, 1, 1/2]/2$): very compact and triangular[2].

Of course, there are many other windows in the literature (for a survey, see [Har78]), mostly designed for specific tasks. For example, the Bartlett (triangular) window is suitable for overlap-add, but is not differentiable and has strong side lobes, the Hamming window reduces the first side lobe level but misses the one-peak property (because the second side lobe is higher), the Blackman window has lower side lobes at the expense of a broader main lobe (thus a poor frequency resolution), and the family of Kaiser windows can tune the main-lobe width / side-lobe level trade-off, but then misses a simple analytic expression.

In our opinion, the Hann window is the best general-purpose window in practice. However, we will consider one more window, the Gaussian window, which has great properties and thus is very useful, at least in theory.

3.2.2.3 Gaussian Window

The (normalized) Gaussian window has a simple analytic expression:

$$w_{G,p}(t) = \sqrt{\frac{p}{\pi}}\exp\left(-pt^2\right) \quad (\forall t) \tag{3.49}$$

and is easily differentiable, since

$$w'_{G,p}(t) = -2ptw_{G,p}(t). \tag{3.50}$$

It is easy to derive the analytic expression of its spectrum using the definition of the Fourier transform (as proposed by Papoulis [Pap77]). However, Smith [Smi07] proposes a more elegant proof taking advantage of the duality of the temporal and spectral domains.

[2]This is indeed the "triangular frequency" window used by Keiler and Zölzer [AKZ99], with a parameter $S = 2$. Its triangular spectrum can be regarded as the convolution in frequency of two boxes. A dual is the Bartlett window, which is triangular in time, and can be regarded as the convolution in time of two rectangular windows.

Indeed, from the differentiation theorem (see Section 1.3.1.2), we have

$$w'_{G,p}(t) \leftrightarrow j\omega W'_{G,p}(\omega). \tag{3.51}$$

But from Equation (3.50) and the differentiation theorem dual (again, see Section 1.3.1.2), we have

$$w'_{G,p}(t) = \frac{2p}{j} \cdot (-jtw_{G,p}(t)) \leftrightarrow \frac{2p}{j} W'_{G,p}(\omega). \tag{3.52}$$

Therefore, from the two previous equations we deduce

$$j\omega W_{G,p}(\omega) = \frac{2p}{j} W'_{G,p}(\omega) \tag{3.53}$$

leading to

$$(\log(W_{G,p}(\omega)))' = \frac{W'_{G,p}(\omega)}{W_{G,p}(\omega)} = -\frac{\omega}{2p} = \left(-\frac{\omega^2}{4p}\right)'. \tag{3.54}$$

Integrating both sides (with respect to ω) yields

$$\log(W_{G,p}(\omega)) = -\frac{\omega^2}{4p} + \log(W_{G,p}(0))$$

and since $W_{G,p}(0) = 1$ (because it is the sum of the normalized Gaussian function), we have

$$W_{G,p}(\omega) = \exp\left(-\frac{\omega^2}{4p}\right). \tag{3.55}$$

Thus a remarkable property is that the spectrum of a Gaussian is a Gaussian. Another very important property is that we can derive an analytic expression for the Γ_w function of Equation (3.31) if w is a Gaussian window (see below).

The practical drawback of the Gaussian window is that it has an infinite time support, and thus is not directly suitable for the DFT. It has to be truncated first, or windowed...

3.2.3 Analysis Methods

The spectral analysis methods are also numerous, and the exhaustive review of all these methods could be the subject of a – rather big – book. Bester has studied many of them in his Ph.D. [Bet08]. Some of them were presented in the Ph.D. manuscript [Mar00] in the stationary case. Since then, the main analysis methods were extended to the non-stationary case.

3.2.3.1 Quadratic Interpolation

The quadratic interpolation method was proposed by Abe and Smith [AS05]. This can be seen as an extension to the non-stationary case of the parabolic method by Smith and Serra [SS87], where the parabolic shape of the magnitude spectrum of the Gaussian window was used the estimate the frequency and amplitude of the signal (see also the Ph.D. manuscript [Mar00] for a description and references). This extends the results by Marques and Almeida [MA86], Peeters and Rodet [PR99].

We start from the frame signal, using the Gaussian window (see Equation (3.28)):

$$x(t) = s_{\phi_0,\lambda_0,\omega_0,\mu_0,\psi_0}(t) w_{G,p}(t). \tag{3.56}$$

Let us first consider the frequency modulation:

$$\begin{aligned} x(t) &= s_{\phi_0,\lambda_0,\omega_0,\mu_0,0}(t) \cdot \exp\left(j\frac{\psi_0}{2}t^2\right) w_{G,p}(t) \\ &= s_{\phi_0,\lambda_0,\omega_0,\mu_0,0}(t) \cdot w_{G,z}(t) K^1_{p,z} \end{aligned} \tag{3.57}$$

$$\text{with} \quad z = p - j\frac{\psi_0}{2} \quad \text{and} \quad K^1_{p,z} = \sqrt{\frac{p}{z}} \tag{3.58}$$

thus it appears that the frequency chirp turns the real Gaussian into a complex one.

Let us then consider the amplitude modulation:

$$\begin{aligned} x(t) &= s_{\phi_0,\lambda_0,\omega_0,0,0}(t) \cdot \exp(\mu_0 t)\, w_{G,z}(t) \cdot K^1_{p,z} \\ &= s_{\phi_0,\lambda_0,\omega_0,0,0}(t) \cdot w_{G,z}(t-t_0) K^2_{z,t_0} \cdot K^1_{p,z} \end{aligned} \tag{3.59}$$

$$\text{with} \quad t_0 = \frac{\mu_0}{2z} \quad \text{and} \quad K^2_{z,t_0} = \exp\left(z{t_0}^2\right) = \exp\left(\frac{{\mu_0}^2}{4z}\right) \tag{3.60}$$

thus it appears that the amplitude ramp causes a translation in time of the Gaussian.

When switching to the spectral domain, the constants are unchanged, $s_{\phi_0,\lambda_0,\omega_0,0,0}(t) \leftrightarrow S_{\phi_0,\lambda_0,\omega_0,0,0}(\omega)$, and from Equation (1.36) we have $w_{G,z}(t-t_0) \leftrightarrow W_{G,z}(\omega)\exp(-j\omega t_0)$, thus

$$X(\omega) = S_{\phi_0,\lambda_0,\omega_0,0,0}(t) * \left(K^1_{p,z} K^2_{z,t_0} W_{G,z}(\omega) e^{-j\omega t_0}\right) \tag{3.61}$$

$$= e^{\lambda_0 + j\phi_0} K^1_{p,z} K^2_{z,t_0} W_{G,z}(\omega - \omega_0) e^{-j(\omega-\omega_0)t_0}. \tag{3.62}$$

Finally

$$\begin{aligned} X(\omega) &= \exp(\lambda_0 + j\phi_0)\sqrt{\frac{p}{z}}\exp\left(\frac{{\mu_0}^2}{4|z|^2}\bar{z}\right)\exp\left(-\frac{(\omega-\omega_0)^2}{4|z|^2}\bar{z}\right)\exp\left(-j(\omega-\omega_0)\frac{\mu_0}{2|z|^2}\bar{z}\right) \\ &= \underbrace{\exp(\lambda_0 + j\phi_0)}_{s_0} \cdot \underbrace{\sqrt{\frac{p}{z}}\exp\left(\frac{p + j\psi_0/2}{4p^2 + {\psi_0}^2}(j(\omega-\omega_0) - \mu_0)^2\right)}_{\Gamma_{w_{G,p}}(\omega_0-\omega,\mu_0,\psi_0)} \end{aligned} \tag{3.63}$$

thus we have an expression for the Γ_w function when w is the Gaussian window.

Switching to the polar representation of complex numbers, it is easy to show that

$$\log\sqrt{\frac{p}{z}} = -\frac{1}{2}\log\frac{z}{p} = -\frac{1}{2}\left(\log\left|\frac{z}{p}\right| + j\angle\frac{z}{p}\right) \tag{3.64}$$

$$\text{with} \quad \left|\frac{z}{p}\right| = \sqrt{1 + \left(\frac{\psi_0}{2p}\right)^2} \quad \text{and} \quad \angle\frac{z}{p} = \arctan\left(-\frac{\psi_0}{2p}\right) \quad (\text{since } p > 0). \tag{3.65}$$

Then, from Equations (3.24) and (3.63), after simplifications we obtain

$$\lambda(\omega) = \lambda_0 + \frac{{\mu_0}^2}{4p} - \frac{1}{4}\log\left(1 + \left(\frac{\psi_0}{2p}\right)^2\right) - \frac{p}{4p^2 + {\psi_0}^2}\left(\omega - \omega_0 - \frac{\mu_0\psi_0}{2p}\right)^2 \tag{3.66}$$

and from Equations (3.26) and (3.63), after simplifications we obtain

$$\phi(\omega) = \phi_0 + \frac{{\mu_0}^2}{2\psi_0} + \frac{1}{2}\arctan\left(\frac{\psi_0}{2p}\right) - \frac{\psi_0/2}{4p^2 + {\psi_0}^2}\left(\omega - \omega_0 + \frac{2p\mu_0}{\psi_0}\right)^2. \tag{3.67}$$

Equations (3.66) and (3.67) show that the log-amplitude and phase are both order-2 polynomials of the frequency ω. As a consequence, in practice, in order to estimate the model parameters from the spectrum, order-2 polynomials $u(\omega)$ and $v(\omega)$ are fitted for the log-amplitude $\lambda(\omega)$ and phase[3] $\phi(\omega)$ spectra (see Equations (3.23) and (3.25)) for estimation time $t = 0$, respectively, thus

$$u(\omega) \approx \lambda(0,\omega), \tag{3.68}$$
$$v(\omega) \approx \phi(0,\omega). \tag{3.69}$$

u are v polynomials, thus very easy to differentiate. Moreover, we have

$$u'' = \frac{-2p}{4p^2+{\psi_0}^2}, \tag{3.70}$$
$$v'' = \frac{-\psi_0}{4p^2+{\psi_0}^2}. \tag{3.71}$$

Using these equations, we can obtain an estimate of the parameter of the Gaussian

$$\hat{p} = -\frac{u''}{2(u''+v'')} \tag{3.72}$$

which is useful in practice if the analysis window used is not a Gaussian (and thus p is unknown).

We also obtain an estimate of the frequency modulation

$$\hat{\psi}_0^O = 2\hat{p}\frac{v''}{u''}. \tag{3.73}$$

Since $p > 0$, we have $u'' < 0$ and thus the root of u' corresponds to a local maximum (peak) in the magnitude spectrum, at frequency

$$\omega_M = \omega_0 + \frac{\mu_0\psi_0}{2p} \tag{3.74}$$

meaning that the peak in the magnitude spectrum is shifted (by $\frac{\mu_0\psi_0}{2p}$) when (log-)amplitude and frequency modulations (μ_0 and $\psi_0/2$) are both present.

Since

$$v'(\omega_M) = -\frac{\mu_0}{2p} \tag{3.75}$$

we obtain an estimate of the amplitude modulation

$$\hat{\mu}_0^O = -2\hat{p}v'(\omega_M). \tag{3.76}$$

The estimate of the frequency is then given by

$$\hat{\omega}_0^O = \omega_M - C_0\frac{\hat{\mu}_0\hat{\psi}_0}{2\hat{p}} \tag{3.77}$$

where $C_0 = 1$ (for the practical interest of introducing a constant here, see below).

[3] In practice, since the phases are measured modulo 2π, the measured phase spectrum has to be unwrapped on the frequency dimension, which can be a difficult task without *a priori* knowledge. Unwrapping the phase on the time dimension is easier, since we know at least that the phase is increasing with time since the frequency – its derivative – is positive.

Finally, since

$$u(\omega_M) = \lambda_0 + \frac{{\mu_0}^2}{4p} - \frac{1}{4}\log\left(1+\left(\frac{\psi_0}{2p}\right)^2\right), \tag{3.78}$$

$$v(\omega_M) = \phi_0 - \frac{{\mu_0}^2\psi_0}{8p} + \frac{1}{2}\arctan\left(\frac{\psi_0}{2p}\right) \tag{3.79}$$

we can easily obtain estimates for the log-amplitude and phase

$$\hat{\lambda}_0^Q = u(\omega_M) - C_1\frac{\hat{\mu}_0^2}{4\hat{p}} + C_2\log\left(1+\left(\frac{\hat{\psi}_0}{2\hat{p}}\right)^2\right), \tag{3.80}$$

$$\hat{\phi}_0^Q = v(\omega_M) + C_3\frac{\hat{\mu}_0^2\hat{\psi}_0}{8\hat{p}} - C_4\arctan\left(\frac{\hat{\psi}_0}{2\hat{p}}\right) \tag{3.81}$$

where in theory the constants should be $C_1 = 1$, $C_2 = 1/4$, $C_3 = 1$, and $C_4 = 1/2$.

However, in practice, the true Gaussian window cannot be used (see Section 3.2.2.3). The estimates for the amplitude and frequency modulations, $\hat{\mu}_0^Q$ and $\hat{\psi}_0^Q$, are then to be adjusted in function of $\Delta = \omega_M - \omega_m$ (where $\omega_m = m\Omega_{F_s,N}$ is the discrete frequency of the local maximum under investigation) and then a window-dependent set of adaptation coefficients C_i has to be used. This way, this method can be adapted for example to the Hann window (see [AS05]), using zero-padding though (which is also mandatory to limit phase-wrapping problems, see footnote 3).

After Smith, we believe that the reason why this method works so well for other windows than the true Gaussian is probably that for a smooth symmetric (even) function (such as the main lobe of the window spectrum), the Taylor expansion can often be limited to order 2 ($W(\omega) \approx W(0) + W''(0)\omega^2/2$) and thus can be regarded locally as a parabola, especially if zero-padding is used...

3.2.3.2 Spectral Reassignment

Reassignment was first proposed by Kodera *et al.* [KdVG76, KGdV78] and was generalized by Auger and Flandrin [AF95] to improve time-frequency representations. Usually, the values obtained when decomposing the signal on the time-frequency atoms are assigned to the geometrical center of the cells (center of the analysis window and bins of the Fourier transform). The reassignment method assigns each value to the center of gravity of the cell's energy. The method uses the knowledge of the first derivatives w' – obtained by analytical differentiation – of the analysis window w in order to adjust the frequency inside the Fourier transform bin (and the generalization of this method to the non-stationary case also requires the knowledge of the second derivative w''). Moreover, Hainsworth [Hai03] shows that the reassignment can be easily generalized for the amplitude modulation.

Indeed, starting from the definition of the STFT given in Equation (3.19), and considering Equations (3.24) and (3.26) giving the instantaneous log-amplitude and phase from the log-polar representation of the short-term spectrum, one can derive the instantaneous (log-)amplitude modulation and frequency

$$\hat{\mu}(t,\omega) = \frac{\partial}{\partial t}\lambda(t,\omega) = \Re\left(\frac{\partial}{\partial t}S_w(t,\omega)\right) = -\Re\left(\frac{S_{w'}(t,\omega)}{S_w(t,\omega)}\right), \tag{3.82}$$

$$\hat{\omega}(t,\omega) = \frac{\partial}{\partial t}\phi(t,\omega) = \Im\left(\frac{\partial}{\partial t}S_w(t,\omega)\right) = \omega - \underbrace{\Im\left(\frac{S_{w'}(t,\omega)}{S_w(t,\omega)}\right)}_{-\Delta_\omega} \tag{3.83}$$

where w' denotes the derivative of w, because

$$\frac{\partial}{\partial t}\log\left(S_w(t,\omega)\right) = \frac{\frac{\partial}{\partial t}S_w(t,\omega)}{S_w(t,\omega)} = j\omega - \frac{S_{w'}(t,\omega)}{S_w(t,\omega)} \tag{3.84}$$

since the first partial derivative (with respect to t) of $S_w(t,\omega)$ can be easily derived:

$$\begin{aligned}
\frac{\partial}{\partial t}S_w(t,\omega) &= \frac{\partial}{\partial t}\int_{-\infty}^{+\infty} s(\tau)\, w(\tau-t)\, e^{-j\omega(\tau-t)}\, d\tau \\
&= \int_{-\infty}^{+\infty} s(\tau)\, \underbrace{\frac{\partial}{\partial t}\left(w(\tau-t)\right)}_{-w'(\tau-t)}\, e^{-j\omega(\tau-t)}\, d\tau + \int_{-\infty}^{+\infty} s(\tau)\, w(\tau-t)\, \underbrace{\frac{\partial}{\partial t}\left(e^{-j\omega(\tau-t)}\right)}_{j\omega e^{-j\omega(\tau-t)}}\, d\tau \\
&= -S_{w'}(t,\omega) + j\omega S_w(t,\omega).
\end{aligned} \tag{3.85}$$

In practice, for a partial corresponding to a local maximum of the (discrete) magnitude spectrum at the (discrete) frequency ω_m, the estimates of the frequency and the amplitude modulation are respectively given by

$$\hat{\mu}_0^R = \hat{\mu}(0,\omega_m) \quad \text{and} \quad \hat{\omega}_0^R = \hat{\omega}(0,\omega_m). \tag{3.86}$$

Moreover, these instantaneous parameters are given for the reassigned time

$$\hat{t}_0 = \hat{t}(0,\omega_m) \quad \text{with} \quad \hat{t}(t,\omega) = t - \frac{\partial}{\partial\omega}\phi(t,\omega) = t + \underbrace{\Re\left(\frac{S_{tw}(t,\omega)}{S_w(t,\omega)}\right)}_{-\Delta_t} \tag{3.87}$$

(tw being the window w multiplied by a time ramp), obtained thanks to Equation (3.26) and the second partial derivative (with respect to ω) of $S_w(t,\omega)$:

$$\begin{aligned}
\frac{\partial}{\partial\omega}S_w(t,\omega) &= \frac{\partial}{\partial\omega}\int_{-\infty}^{+\infty} s(\tau)\, w(\tau-t)\, e^{-j\omega(\tau-t)}\, d\tau \\
&= \int_{-\infty}^{+\infty} s(\tau)\, w(\tau-t)\, \underbrace{\frac{\partial}{\partial\omega}\left(e^{-j\omega(\tau-t)}\right)}_{-j(\tau-t)\,e^{-j\omega(\tau-t)}}\, d\tau \\
&= -jS_{tw}(t,\omega).
\end{aligned} \tag{3.88}$$

As a consequence, these parameters might normally be reassigned back from time $\hat{t}_0$ to the estimation time $t=0$, using:

$$\hat{\mu}_0^R = \hat{\mu}, \tag{3.89}$$
$$\hat{\lambda}_0^R = \hat{\lambda} + \hat{\mu}\Delta_t, \tag{3.90}$$
$$\hat{\psi}_0^R = \hat{\psi}, \tag{3.91}$$
$$\hat{\omega}_0^R = \hat{\omega} + \hat{\psi}\Delta_t, \tag{3.92}$$
$$\hat{\phi}_0^R = \hat{\phi} + \hat{\omega}\Delta_t + \frac{\hat{\psi}}{2}{\Delta_t}^2 \tag{3.93}$$
$$\text{where} \quad \Delta_t = t - \hat{t}_0 = -\hat{t}_0 \tag{3.94}$$

omitting in the notations, for the sake of readability, the parameters (t,ω_m) for $\hat{\lambda}$, $\hat{\mu}$, $\hat{\phi}$, $\hat{\omega}$, and $\hat{\psi}$. However, it appears in the experiments described in Section 3.2.5 that Δ_t can be neglected. Indeed,

except for very low signal-to-noise ratios (SNRs) – in which case the estimation is not precise anyway – Δ_t is less than the sampling period (thus very low when compared to ω_0, μ_0, or ψ_0). Nevertheless, Röbel [Röb02] shows that, thanks to $\hat{t}$, an estimation of the frequency derivative ψ_0 is

$$\hat{\psi}_0^R = \hat{\psi}(0,\omega_m) \quad \text{with} \quad \hat{\psi}(t,\omega) = \frac{\partial\hat{\omega}}{\partial\hat{t}} = \frac{\partial\hat{\omega}}{\partial t}\Big/\frac{\partial\hat{t}}{\partial t} \tag{3.95}$$

with (again, omitting the parameters (t,ω) for the sake of readability)

$$\frac{\partial\hat{\omega}}{\partial t} = \underbrace{\frac{\partial\omega}{\partial t}}_{0} - \Im\left(\frac{\frac{\partial}{\partial t}S_{w'}\cdot S_w - S_{w'}\cdot\frac{\partial}{\partial t}S_w}{{S_w}^2}\right) = \Im\left(\frac{S_{w''}}{S_w}\right) - \Im\left(\left(\frac{S_{w'}}{S_w}\right)^2\right), \tag{3.96}$$

$$\frac{\partial\hat{t}}{\partial t} = \underbrace{\frac{\partial t}{\partial t}}_{1} + \Re\left(\frac{\frac{\partial}{\partial t}S_{tw}\cdot S_w - S_{tw}\cdot\frac{\partial}{\partial t}S_w}{{S_w}^2}\right) = \Re\left(\frac{S_{tw}S_{w'}}{{S_w}^2}\right) - \Re\left(\frac{S_{tw'}}{S_w}\right) \tag{3.97}$$

obtained from Equations (3.83) and (3.87), since from Equation (3.85) we have

$$\frac{\partial}{\partial t}S_{w'}(t,\omega) = -S_{w''}(t,\omega) + j\omega S_{w'}(t,\omega)$$

and because

$$\begin{aligned}
\frac{\partial}{\partial t}S_{tw}(t,\omega) &= \frac{\partial}{\partial t}\int_{-\infty}^{+\infty} s(\tau)\ (\tau-t)w(\tau-t)\ e^{-j\omega(\tau-t)}\ d\tau \\
&= \int_{-\infty}^{+\infty} s(\tau)\ \underbrace{\frac{\partial}{\partial t}\left((\tau-t)w(\tau-t)\right)}_{-w(\tau-t)-(\tau-t)w'(\tau-t)}\ e^{-j\omega(\tau-t)}\ d\tau \\
&+ \int_{-\infty}^{+\infty} s(\tau)\ (\tau-t)w(\tau-t)\ \underbrace{\frac{\partial}{\partial t}\left(e^{-j\omega(\tau-t)}\right)}_{j\omega e^{-j\omega(\tau-t)}}\ d\tau \\
&= -S_w(t,\omega) - S_{tw'}(t,\omega) + j\omega S_{tw}(t,\omega).
\end{aligned} \tag{3.98}$$

Amplitude and phase are eventually to be estimated. Since these estimations are not included in the original reassignment method, and since we know the estimated modulations $\hat{\mu}_0$ and $\hat{\psi}_0$, we propose to use the Γ_w function of Equation (3.31), thus

$$\hat{a}_0^R = \hat{a}^R(0) \quad \text{with} \quad \hat{a}^R(t) = \left|\frac{S_w(t,\omega_m)}{\Gamma_w(\Delta_\omega,\hat{\mu}_0,\hat{\psi}_0)}\right|, \tag{3.99}$$

$$\hat{\phi}_0^R = \hat{\phi}^R(0) \quad \text{with} \quad \hat{\phi}^R(t) = \angle\left(\frac{S_w(t,\omega_m)}{\Gamma_w(\Delta_\omega,\hat{\mu}_0,\hat{\psi}_0)}\right) \tag{3.100}$$

with $\Delta_\omega = \omega_m - \hat{\omega}_0^R$.

As noted by Hainsworth, the discrete version of the reassignment method introduces a bias in the estimations, since it is done in a rather crude way: the spectra are replaced by their discrete versions, the time t being replaced by the time sample n and the frequency ω_m being replaced by the corresponding bin index m.

However, the reassignment method seems currently the best STFT-based method in terms of estimation precision, at least regarding frequency (see [BCRD08]). But the derivative method we propose (see below) can perform even better.

3.2.3.3 Derivative Algorithm

We proposed this method in 1998 [10, 1] in the stationary case. The name "derivative algorithm" was proposed later by Keiler and used in 2002 our common survey [23]. After a discussion in 2003 with Quatieri, we were already convinced that this method could also work in the non-stationary case. This generalization was done in 2008 together with Depalle [46].

The idea behind this technique is extremely simple: differentiating a complex exponential gives a complex exponential of the same frequency. More precisely, considering Equation (3.18), we have

$$s'(t) = \frac{ds}{dt}(t) = (\mu_0 + j(\omega_0 + \psi_0 t)) \cdot s(t) \tag{3.101}$$

and thus

$$\Im\left(\frac{s'}{s}(t)\right) = \omega_0 + \psi_0 t \quad \text{and} \quad \Re\left(\frac{s'}{s}\right) = \mu_0. \tag{3.102}$$

For this method to work in the case of a signal made of several partials, we have to switch to the spectral domain. Again, we consider only the spectrum values close to a given local magnitude maximum m that represents the partial under investigation in order to be able to neglect the influence of the other partials. Since the differentiation is a linear operation, for a complex sound the spectra of its derivative will be the sum of the contributions of each sinusoid.

We have to check first the contribution of $r(t) = \psi_0 t$ in the spectral domain. Fortunately, it has nice properties: since $r(t)$ is an odd function, its spectrum $R(f)$ is imaginary, thus $j\psi_0 t$ only contributes to the real part of the spectrum of s'/s. Even-though its amplitude can be very high at extreme frequencies, $R(f)$ is null at frequency zero, and exhibits small values around 0.

Thus, the spectrum of s' involves a convolution sum between R, which equals 0 at frequency 0, with S_w, which energy is still essentially located around frequency ω_0 (it is exactly the case in the stationary case; it is only an approximation in the non-stationary case because of Equation (3.31) and the fact that the local maximum gets slightly shifted in frequency, see Section 3.2.3.1). This convolution sum results in a negligible contribution when compared to $\omega_0 S_w$. The complete theoretical investigation of these properties is beyond the scope of this document. However, in practice, evaluating S'_w/S_w close to the local maximum (discrete) frequency ω_m yields to an excellent estimation of the frequency:

$$\hat{\omega}_0^D = \hat{\omega}^D(0) \quad \text{with} \quad \hat{\omega}^D(t) = \Im\left(\frac{S'_w}{S_w}(t, \omega_m)\right). \tag{3.103}$$

Here, beware that S' stands for the spectrum of the signal derivative s'. It is not the derivative of the spectrum (which would have been ambiguous, since the short-term spectrum has two partial derivatives, see Section 3.2.3.2). Now that we have gotten an estimate of ω_0, we can evaluate S'_w at this frequency, where $R * S_w$ contribution equals zero, and thus we obtain an estimate for the amplitude modulation:

$$\hat{\mu}_0^D = \hat{\mu}^D(0) \quad \text{with} \quad \hat{\mu}^D(t) = \Re\left(\frac{S'_w}{S_w}(t, \hat{\omega}_0)\right). \tag{3.104}$$

In order to get the estimate of the frequency modulation ψ_0, we have to consider s'', the second derivative of s. More precisely, we have

$$\begin{aligned} s''(t) &= \frac{d^2 s}{dt^2}(t) \\ &= (\mu_0 + j(\omega_0 + \psi_0 t)) \cdot s'(t) + j\psi_0 \cdot s(t) \\ &= \left((\mu_0 + j(\omega_0 + \psi_0 t))^2 + j\psi_0\right) \cdot s(t) \\ &= \left((\mu_0{}^2 - \omega_0{}^2 - 2\omega_0\psi_0 t - \psi_0{}^2 t^2) + j(\psi_0 + 2\mu_0\omega_0 + 2\mu_0\psi_0 t)\right) \cdot s(t). \end{aligned} \tag{3.105}$$

We then use the same kind of properties that we just used for the spectrum of the first derivative. Even functions (*e.g.* proportional to t^2) in one part (real or imaginary) of the signal will contribute to the same part (real or imaginary) of the spectrum, whether odd functions (*e.g.* proportional to t) in one part (real or imaginary) of the signal will contribute to the opposite part (imaginary or real) of the spectrum. Moreover the effects of the convolution sums are negligible for $\hat{\omega}_0 \approx \omega_0$. Finally, we get the estimate of the frequency modulation:

$$\hat{\psi}_0^D = \hat{\psi}^D(0) \quad \text{with} \quad \hat{\psi}^D(t) = \Im\left(\frac{S_w''}{S_w}(t,\hat{\omega}_0)\right) - 2\hat{\mu}_0\hat{\omega}_0. \tag{3.106}$$

Let us consider now the estimation of the initial amplitude and initial phase of the signal. Since we know the estimated modulations $\hat{\mu}_0$ and $\hat{\psi}_0$, we propose to use the Γ_w function of Equation (3.31) as we did for the extension of the reassignment method in Section 3.2.3.2, but this time with $\Delta_\omega = \omega_0 - \hat{\omega}_0 \approx 0$, thus

$$\hat{a}_0^D = \left|\frac{S_w(0,\hat{\omega}_0)}{\Gamma_w(0,\hat{\mu}_0,\hat{\psi}_0)}\right|, \tag{3.107}$$

$$\hat{\phi}_0^D = \angle\left(\frac{S_w(0,\hat{\omega}_0)}{\Gamma_w(0,\hat{\mu}_0,\hat{\psi}_0)}\right). \tag{3.108}$$

Signal Derivatives

Unless the derivatives of the signal can be recorded together with the signal itself, in practice these (discrete-time) signal derivatives have to be estimated from the (discrete-time) signal s.

The first derivative $s'(t) = ds/dt$ is defined mathematically by

$$s'(t) = \lim_{\varepsilon\to 0}\frac{s(t+\varepsilon)-s(t)}{\varepsilon}. \tag{3.109}$$

Our first idea [1, Mar00] was thus to set ε to the sampling period T_s in the previous equation. In fact, the left and right derivatives for ε negative or positive, are respectively

$$s'_-(t) = \lim_{\varepsilon\to 0_-}\frac{s(t+\varepsilon)-s(t)}{\varepsilon} = \lim_{\eta\to 0_+}\frac{s(t)-s(t-\eta)}{\eta} \quad (\text{with } \eta = -\varepsilon), \tag{3.110}$$

$$s'_+(t) = \lim_{\varepsilon\to 0_+}\frac{s(t+\varepsilon)-s(t)}{\varepsilon}. \tag{3.111}$$

Since the signal s is supposed to be a $\mathcal{C}^\infty$ function, these left and right derivatives are equal, that is

$$s'_-(t) = s'_+(t) = s'(t). \tag{3.112}$$

In practice, the signal s is discrete, $s[n]$ representing $s(nT_s)$, and the smallest (non-zero) possible $|\varepsilon|$ for the direct computation of the derivative is the sampling period T_s. So let us consider the following approximations (first-order differences):

$$\hat{s}'_-[n] = (s[n]-s[n-1])/F_s, \tag{3.113}$$

$$\hat{s}'_+[n] = (s[n+1]-s[n])/F_s. \tag{3.114}$$

This time, $\hat{s}'_+[n] \neq \hat{s}'_-[n]$, but

$$\hat{s}'_+[n] = (s[n+1]-s[n])/F_s = \hat{s}'_-[n+1], \tag{3.115}$$

$$\hat{s}'_-[n] = (s[n]-s[n-1])/F_s = \hat{s}'_+[n-1]. \tag{3.116}$$

So, apart from a one-sample time translation, they are the same discrete functions (and this small translation has a negligible impact on the power spectrum of the signal derivative, which is the only expression considered by this method). Let us arbitrarily take the left derivative (resp. right derivative) approximation as an approximation for the derivative itself:

$$\hat{s}'[n] = F_s(s[n] - s[n-1]) \quad (\text{resp. } \hat{s}'[n] = F_s(s[n+1] - s[n])) \tag{3.117}$$

whose short-term spectrum is given by

$$\hat{S}'[n,m] = F_s(S[n,m] - S[n-1,m]) \quad (\text{resp. } \hat{S}'[n,m] = F_s(S[n+1,m] - S[n,m])). \tag{3.118}$$

In fact, Equation (3.117) defines a linear-phase high-pass filter whose transfer function is

$$H(z) = F_s(1 - z^{-1}) \quad (\text{resp. } H(z) = F_s(z-1)). \tag{3.119}$$

Finally (in any case), its gain is (see Figure 3.5)

$$\left|H\left(e^{j\omega}\right)\right| = F_s\sqrt{2\left(1-\cos(\omega)\right)} = 2F_s \sin\left(\frac{\omega}{2}\right) \tag{3.120}$$

and

$$|\hat{S}'(t,\omega)| = \left|H\left(e^{j\omega}\right)\right| \cdot |S(t,\omega)| = 2F_s \sin\left(\frac{\omega}{2}\right)|S(t,\omega)| \tag{3.121}$$

thus a frequency estimate is

$$\hat{\omega}^d(t,\omega) = 2\arcsin\left(\frac{1}{2F_s}\frac{|\hat{S}'(t,\omega_m)|}{|S(t,\omega_m)|}\right) \tag{3.122}$$

and, in the discrete case, with $t = n/F_s$ and $\omega_m = m\frac{2\pi F_s}{N}$:

$$\hat{\omega}^d[n,m] = 2\arcsin\left(\frac{|S[n,m] - S[n-1,m]|}{2|S[n,m]|}\right) = 2\arcsin\left(\frac{|S[n+1,m] - S[n,m]|}{2|S[n,m]|}\right) \tag{3.123}$$

thus the estimate of the frequency a time 0 is

$$\hat{\omega}_0^d = \hat{\omega}^d(0,\omega_m) = \hat{\omega}^d[0,m]. \tag{3.124}$$

Thus, we lose the simplicity of the theory and get an expression quite different for the frequency estimate. However, this expression is equivalent to the one obtained by the difference method (see Section 3.2.3.4) used by the phase vocoder. Also, with this approach, obtaining the new expressions for the other parameter estimates is not straightforward.

Our next idea was to keep with the original expressions for the estimates and to use our very accurate resampling method (see Section 1.1.3) to set ε much closer to 0 in Equation (3.109). The two reconstructor filters (for $s(t+\varepsilon)$ and $s(t)$) were then combined into a differentiator filter.

But there are other ways to design differentiator filters. No matter the practical differentiator filter, as the order of this filter increases, its impulse response shall converge to the theoretical one. Indeed, differentiation is a linear operation, that can also be regarded as a filter of complex gain $j\omega$ (where ω is the frequency of the input sinusoid). The discrete-time response of this filter could be obtained by the inverse Fourier transform of its frequency response. However, we keep with the mathematical

definition. The continuous-time signal $s(t)$ can be reconstructed (see Section 1.1) from its samples $s[m] = s(m/F_s)$ using the following equation:

$$s(t) = \sum_{k=-\infty}^{+\infty} s[k] \operatorname{sinc}(\underbrace{F_s t - k}_{u(t)}), \tag{3.125}$$

meaning that the signal $s'(t)$ is

$$s'(t) = F_s \sum_{k=-\infty}^{+\infty} s[k] \left(\frac{\cos(\pi u(t))}{u(t)} - \frac{\sin(\pi u(t))}{\pi u(t)^2} \right) \tag{3.126}$$

which samples are (given that the multiplicative term of $s[k]$ in Equation (3.126) equals 0 when $k = n$):

$$s'[n] = s'(n/F_s) = F_s \sum_{k \neq n} s[k] \left(\frac{(-1)^{(n-k)}}{(n-k)} \right). \tag{3.127}$$

Thus, the discrete derivative s' can be obtained by convolving the discrete signal s by the following differentiator filter[4]:

$$h[n] = F_s \frac{(-1)^n}{n} \text{ for } n \neq 0, \text{ and } h[0] = 0 \tag{3.128}$$

of infinite time support. In practice, we use the classic "window method" and multiply h by the Hann window (see Equation (3.41)) to obtain an approximation of the impulse response with a finite time support. This works quite well for high filter orders. With an order of 1023, the bias introduced by the approximation of the discrete derivative can be neglected. And the required convolution by an order-1023 impulse response is quite fast on nowadays computers. Figure 3.6 shows the maximal relative error for the first derivative, when the signal is a pure sinusoid of constant (reduced) frequency $\underline{\omega}$ and amplitude 1. The relative error is the maximal absolute error divided by the norm of the gain of the theoretic derivative (ω). These results show that the errors remain acceptable for most frequencies, except very high frequencies close to the (reduced) Nyquist frequency (π).

To obtain the second derivative, the differentiation is simply applied twice.

3.2.3.4 Difference Method

The last analysis method we will describe was in fact the first to be proposed, probably the simplest one, and yet very efficient. The difference estimator is indeed used in the classic phase vocoder approach [Dol86, Puc95, AKZ02]. Considering that the frequency is constant ω_0 during the time-interval between two successive short-term spectra, with a hop size of I samples, Equation (3.6) shows that the frequency can be estimated from the phase difference:

$$\omega_0 = \Delta_\phi / \Delta_t \quad \text{with} \quad \Delta_t = I T_s. \tag{3.129}$$

The assumption of constant frequency is particularly valid if the short-term spectra are separated by only $I = 1$ sample. In practice, the phase difference is obtained from the short-term spectra, by taking care of unwrapping the phase so that this difference is never negative (the frequency being positive). The resulting estimator is known as the difference estimator:

$$\hat{\omega}_0^\Delta = \hat{\omega}^\Delta[0, m] \quad \text{with} \quad \hat{\omega}^\Delta[n, m] = F_s \underbrace{\left(\angle_{\text{unwrap}} S[n+1, m] - \angle S[n, m] \right)}_{\Delta_\phi} \tag{3.130}$$

[4]Thanks to Badeau for the simplified mathematical derivations leading to this expression.

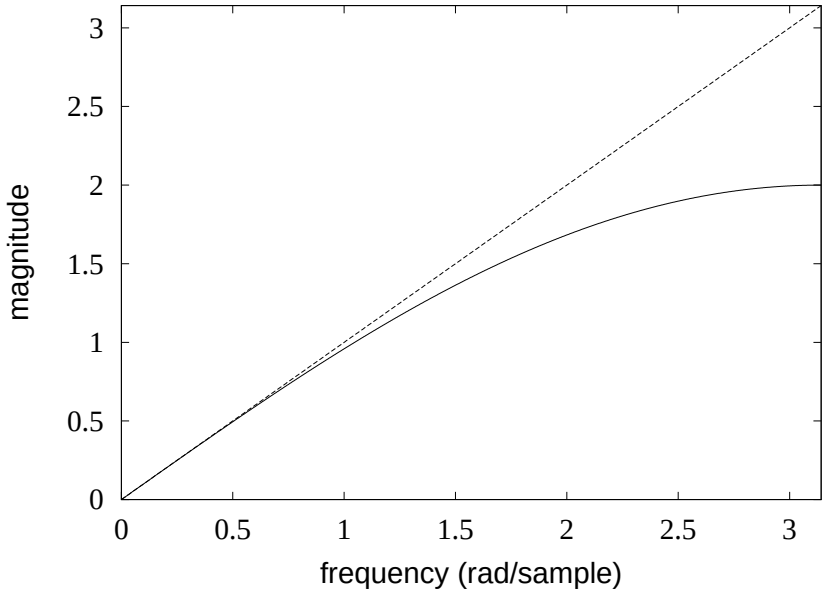


Figure 3.5: First-order difference (solid) versus theoretical differentiation (dashed) gains.

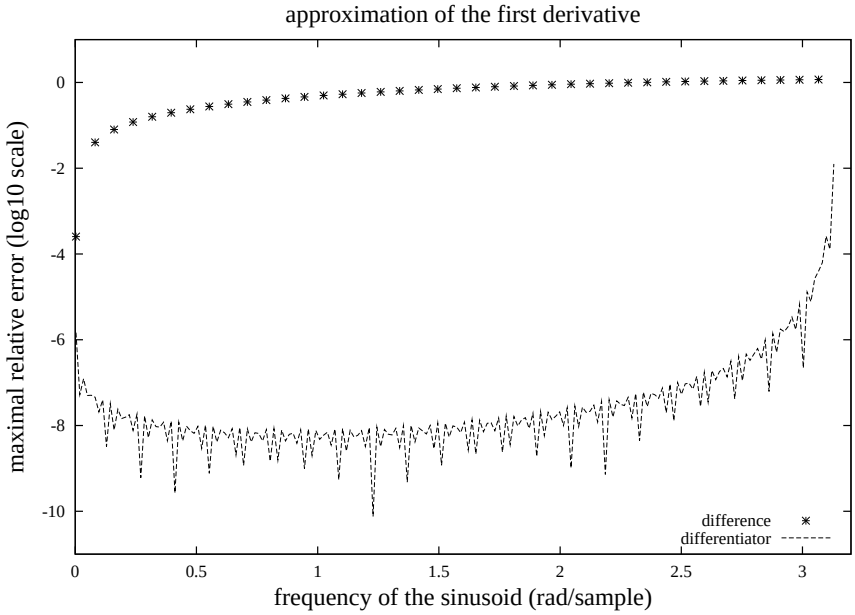


Figure 3.6: Maximal estimation error when approximating the (first) derivative of a sinusoid using the order-1023 differentiator. The error is acceptable for all but very high frequencies. As a comparison, the error obtained with the classic first-order difference approximation (equivalent to choosing $\varepsilon = 1/F_s$ in the expression of Equation (3.109)) is plotted with stars $*$.

which can also be formulated:

$$\hat{\omega}^{\Delta}[n,m] = F_s \underbrace{\angle \frac{S[n+1,m]}{S[n,m]}}_{\theta} + \alpha 2\pi \quad \text{with } \alpha = 1 \text{ if } \theta < 0, \text{ and else } 0. \tag{3.131}$$

(Note that the arctan function can be used to compute the angle.)

Using the Hann window for the short-term spectra, a very nice analytic formula for the $\hat{\omega}^{\Delta}$ frequency estimator can be derived for the ratio of these spectra in Equation (3.131). In fact, we shall evaluate this ratio one sample before, that is $S[n,m]/S[n-1,m]$. Indeed, because the window value is 0 for the first sample of the frame, a time-shift of 1 sample in the past is equivalent to a circular shift of the original signal frame by one sample to the right. The resulting spectrum is $S[n,m] \cdot \exp(-j2\pi m/N)$. Moreover, the discrete spectrum of the Hann window is very compact (see Section 3.2.2.2), and evaluating the spectral convolution at some bin m is just a matter of combining the value of this bin with the ones of its left $(m-1)$ and right $(m+1)$ neighbors. The derivation of the analytic formula is not crucial for this manuscript and thus is left as an exercise. The practical interest is that this way we can obtain an equivalent estimator which can now been implemented using only one FFT. Carrying on the investigation of these equivalences among estimators, for example to lower their complexity, is part of our future research directions.

We have an estimator of the frequency ω. But what about the other model parameters? The complex amplitude s_0 can be derived in the stationary case from the window spectrum:

$$\hat{s}_0^{\Delta} = S[0,m]/W(\omega_m - \hat{\omega}_0^{\Delta}) \tag{3.132}$$

thus

$$\hat{\lambda}_0^{\Delta} = |\hat{s}_0^{\Delta}| = \Re\left(\log\left(\hat{s}_0^{\Delta}\right)\right) \tag{3.133}$$

$$\hat{\phi}_0^{\Delta} = \angle\hat{s}_0^{\Delta} = \Im\left(\log\left(\hat{s}_0^{\Delta}\right)\right) \tag{3.134}$$

and, in the non-stationary case, the amplitude and frequency modulations might be derived using the same stationarity assumption between two consecutive estimates of the amplitude and frequency estimates, respectively

$$\mu_0 = \Delta_\lambda/\Delta_t, \tag{3.135}$$

$$\psi_0 = \Delta_\omega/\Delta_t. \tag{3.136}$$

The complete study and generalization of this difference method to the non-stationary case is part of our future research projects.

3.2.4 Theoretic Equivalences

Together with Lagrange [37, 6] we have already identified many equivalences among phase-based estimators, first in theory then in practice.

3.2.4.1 Phase-Based Frequency Estimators

Another way of deriving the expression for the frequency estimator using the difference method $\hat{\omega}^{\Delta}$ is to note that, for the discrete frequency of the peak ω_m close to its true frequency ω_0, we have

$$S_w[n+1,m] \approx S_w[n,m] \cdot \exp(j\Delta_\phi) \tag{3.137}$$

with the same Δ_ϕ as in Equation (3.129), and a geometrical interpretation in the complex plane is that the spectrum is rotated by Δ_ϕ radians (see Figure 3.7).

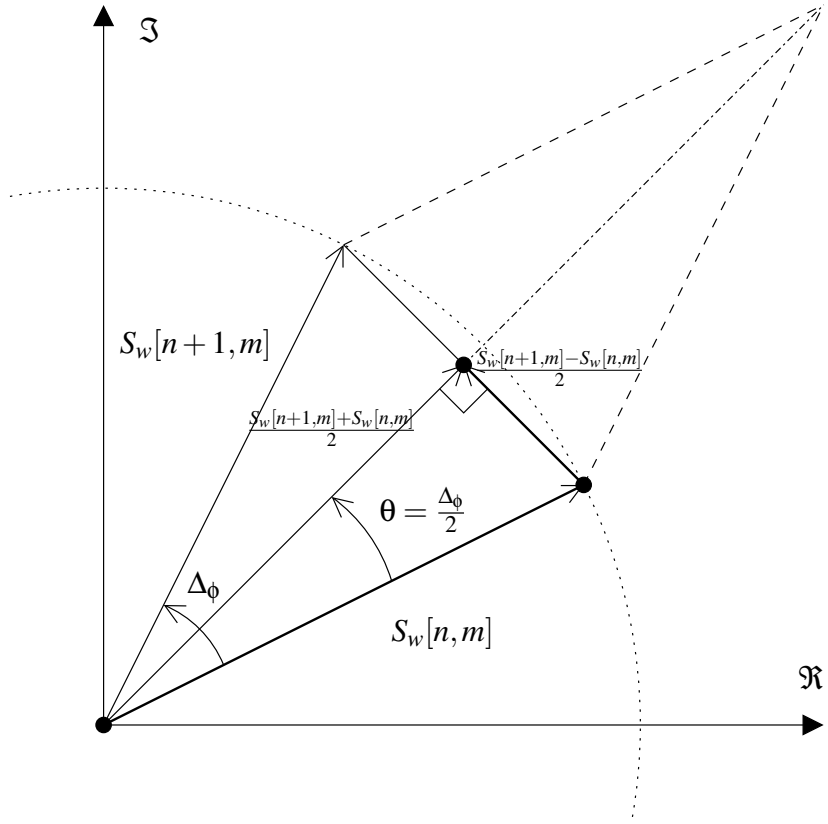


Figure 3.7: Vector relationships for phase difference and discrete derivative methods.

Trigonometric Estimator

The following identity:

$$\frac{S_w[n+1,m]+S_w[n,m]}{2} - S_w[n,m] = \frac{S_w[n+1,m]-S_w[n,m]}{2} \tag{3.138}$$

and simple geometric considerations lead us to consider a right-angled triangle, see Figure 3.7 (lower triangle, delimited by 3 dots). Another of its angles measures $\theta = \Delta_\phi/2$ radians. Simple trigonometric considerations give

$$\sin(\theta) = \left| \frac{S_w[n+1,m]-S_w[n,m]}{2} / S_w[n,m] \right|, \tag{3.139}$$

$$\cos(\theta) = \left| \frac{S_w[n+1,m]+S_w[n,m]}{2} / S_w[n,m] \right|. \tag{3.140}$$

By considering again Equation (3.129) with $H = 1$, but this time with Equations (3.139) and (3.140) – more precisely their inverses – to compute $\Delta_\phi = 2\theta$, we define

$$\hat{\omega}^-[n,m] = 2\arcsin\left(\left| \frac{S_w[n+1,m]-S_w[n,m]}{2S_w[n,m]} \right| \right), \tag{3.141}$$

$$\hat{\omega}^+[n,m] = 2\arccos\left(\left| \frac{S_w[n+1,m]+S_w[n,m]}{2S_w[n,m]} \right| \right). \tag{3.142}$$

The first of these estimators $\hat{\omega}^-$ is exactly the discrete derivative estimator $\hat{\omega}^d$ of Section 3.2.3.3 (see Equation (3.123)). We show in [36] that the behavior of the derivative estimator is closely linked to the properties of the mathematical function arcsin of Equations (3.141) and (3.123), which is not a linear transfer function. If the argument gets close to 1, a small error will lead to a non-negligible error on the estimated frequency. As shown in Figure 3.8, the errors of the estimator $\hat{\omega}^-$ – the discrete derivative one – and the estimator $\hat{\omega}^+$ behave symmetrically with respect to the frequency of the analyzed tone. More precisely, the error of $\hat{\omega}^+$ is high in the low frequencies and low in the high frequencies whereas the error of the discrete derivative estimator is low in the low frequencies and grows as the frequency grows. Indeed, the argument of the arccos function of the $\hat{\omega}^+$ estimator gets close to 1 when the frequency is close to 0. The $\hat{\omega}^+$ estimator can therefore be used in order to improve the precision of the discrete derivative one in the high frequencies. The resulting estimator, proposed in [36] and named the trigonometric estimator, combines the best of the arcsin ($\hat{\omega}^-$) and arccos ($\hat{\omega}^+$) estimators:

$$\hat{\omega}_0^t = \begin{cases} \hat{\omega}^-[0,m] & \text{if } m/N < 0.25, \\ \hat{\omega}^+[0,m] & \text{otherwise.} \end{cases} \tag{3.143}$$

This estimator and the discrete derivative one are of equivalent complexity since they both require the computation of only two FFTs: $S_w[n,m]$ and $S_w[n+1,m]$.

Arctan Estimator

The third and last way of computing the angle θ from the triangle plotted in Figure 3.7 leads to another estimator proposed by Betser [BCRD06]. This estimator, called the arctan estimator, is the last one we consider in this document:

$$\hat{\omega}_0^a = \hat{\omega}^a[0,m] \quad \text{with} \quad \hat{\omega}^a[n,m] = 2\arctan\left(\left| \frac{S[n+1,m]-S[n,m]}{S[n+1,m]+S[n,m]} \right| \right). \tag{3.144}$$

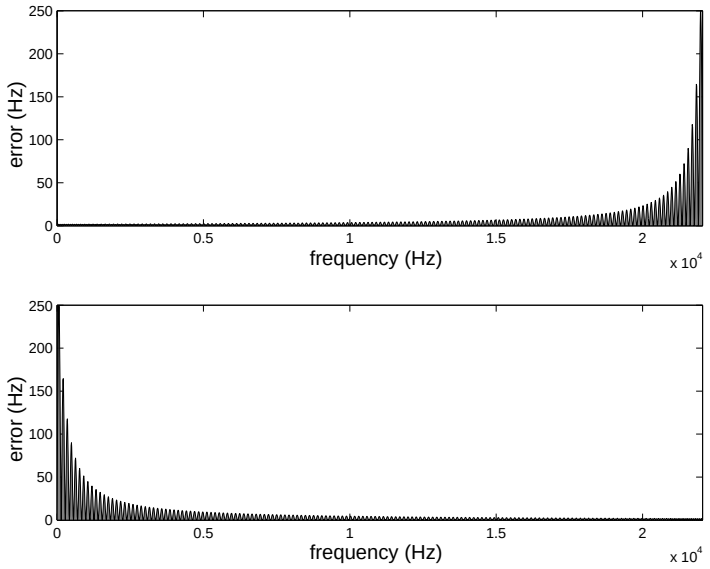


Figure 3.8: Estimation errors of $\hat{\omega}^-$ (top) and $\hat{\omega}^+$ (bottom) versus the frequency of the analyzed sinusoidal signal. It appears that the errors of these two estimators are symmetrically distributed around frequency $F_s/4$ (the half of the Nyquist frequency), here with $F_s = 44100$Hz.

Together with Equations (3.123) and (3.129), and because of the trigonometric relation on θ given in Equation (3.139), Figure 3.7 shows that the estimators $\hat{\omega}^{\Delta}$, $\hat{\omega}^{d}$, $\hat{\omega}^{t}$, and $\hat{\omega}^{a}$ are equivalent, at least in theory, since $\Delta_\phi = 2\theta$. Despite the theoretical equivalence of these phase-based estimators, the different mathematical operations used in their discrete versions influence their performance.

3.2.4.2 Spectral Reassignment and Derivative Algorithm

In [37, 6], the reassignment (Section 3.2.3.2) and derivative (Section 3.2.3.3) methods were proven to be theoretically equivalent, at least as regards the estimation of the frequency in the stationary case. In [46], we generalized the proof of the equivalence to the non-stationary case, and for the estimation of both the frequency and the amplitude modulation. More precisely, we introduce $\rho = \tau - t$ which gives another (equivalent) expression for the STFT (see Equation (3.19)):

$$S_w(t,\omega) = \int_{-\infty}^{+\infty} s(t+\rho)w(\rho)\exp(-j\omega\rho)\,d\rho \tag{3.145}$$

from which we can derive

$$\frac{\partial}{\partial t}\log(S_w(t,\omega)) = \int_{-\infty}^{+\infty}\left(\frac{d}{dt}s(t+\rho)\right)w(\rho)\exp(-j\omega\rho)\,d\rho = \frac{S'_w(t,\omega)}{S_w(t,\omega)}. \tag{3.146}$$

By considering Equation (3.146) instead of Equation (3.84) in Section 3.2.3.2, we would have obtained

$$\hat{\mu} = \Re\left(\frac{S'_w}{S_w}\right), \tag{3.147}$$

$$\hat{\omega} = \Im\left(\frac{S'_w}{S_w}\right) \tag{3.148}$$

instead of Equations (3.82) and (3.83), respectively. When also considering Equations (3.104) and (3.103), we conclude

$$\hat{\mu}(t,\omega) \quad = -\Re\left(\frac{S_{w'}}{S_w}\right) = \Re\left(\frac{S'_w}{S_w}\right) = \quad \hat{\mu}^D(t,\omega), \tag{3.149}$$

$$\hat{\omega}(t,\omega) \quad = \omega - \Im\left(\frac{S_{w'}}{S_w}\right) = \Im\left(\frac{S'_w}{S_w}\right) = \quad \hat{\omega}^D(t,\omega) \tag{3.150}$$

thus the reassignment and derivative methods are equivalent, at least in theory and for the estimation of the frequency and the amplitude modulation. However, Section 3.2.5.2 will show some differences in practice, since their practical implementations are very different.

In the stationary case, $\mu = 0$; considering Equations (3.150) and (3.149) (since for an imaginary number $x = j\omega$ where ω is real and positive, we have $|x| = \Im(x)$)[5], we have also $\hat{\omega}^R = \hat{\omega}^D = |S'_w/S_w|$.

3.2.5 Practical Evaluation

In this section, we present criteria and methods to quantitatively evaluate the estimators. Thus, we can see how the theoretic equivalences described below (Section 3.2.4) behave in practice.

3.2.5.1 Frequency Resolution

In this document, we will not consider frequency resolution issues. In fact, we will keep with the uniform resolution of the discrete STFT, that depends on the frame size N and the lobes of the window.

We are aware that high-resolution (HR) methods exist, such as the ESPRIT algorithm that outperforms the Fourier-based methods in terms of resolution and precision [BRD08]. However, there are at least two reasons not to use these HR methods for audio signals. First, these methods are very sensitive to the choice of the model order and require the number of partials P to be known in advance for an optimal estimation (although Badeau [Bad05] propose techniques to alleviate this). Second, the quest for high resolution might be unjustified. Indeed, let us consider the following signals s_1 and s_2:

$$s_1(t) = \cos(2\pi 5t)\cos\left(2\pi 4000t - \frac{\pi}{2}\right) \tag{3.151}$$

that is Equation (3.4) with $P = 1$, $a_1(t) = \cos(2\pi 5t)$, $\omega_1(t) = 2\pi 4000$, $\phi_1(0) = -\pi/2$, and

$$s_2(t) = \frac{1}{2}\cos\left(2\pi 3995t - \frac{\pi}{2}\right) + \frac{1}{2}\cos\left(2\pi 4005t - \frac{\pi}{2}\right) \tag{3.152}$$

that is Equation (3.4) with $P = 2$, $a_1(t) = 1/2$, $\omega_1(t) = 2\pi 3995$, $\phi_1(0) = -\pi/2$, $a_2(t) = 1/2$, $\omega_2(t) = 2\pi 4005$, $\phi_2(0) = -\pi/2$. At first sight, one might think that s_1 corresponds to a pure tone of frequency 4000Hz with a 5Hz tremolo, whereas s_2 corresponds to a complex sound made of two tones of steady frequencies 3995 and 4005Hz. In fact, from a strictly mathematical point of view, these sounds are the same ($s_1 = s_2$), because of the following equations:

$$\cos(p) + \cos(q) \quad = \quad 2\cos\left(\frac{p-q}{2}\right)\cos\left(\frac{p+q}{2}\right), \tag{3.153}$$

$$\cos(a)\cos(b) \quad = \quad \frac{1}{2}\left[\cos(a-b) + \cos(a+b)\right]. \tag{3.154}$$

But according to perception, Equation (3.151) must be preferred here, because the two partials of Equation (3.152) are so close in frequency – much less than 1 Bark – that they are perceived as a

[5] This note was missing in my Ph.D. manuscript [Mar00], leading to an inexact conclusion there in Equation (3.22).

unique partial. The ear decides when the mathematics cannot (see Section 1.5). In this case, HR methods would have made the wrong choice. And as shown in Figure 4.1, making the right decision in terms of time / frequency resolution is not an easy task. Moreover, since the HR methods do not consider frequency modulation (more precisely, they assume that $\psi = 0$), the ESPRIT algorithm may find many peaks for a single partial with vibrato.

3.2.5.2 Estimation Precision

We focus here on the precision of the estimation of the parameters of one sinusoid. Fortunately, the measure of the precision of the estimation of each parameter – taken separately – is well defined. However, we would need a distance measure between the original and estimated frames. But frames are sets of peaks, and each peak has a lot of parameters. Thus the high dimensionality and the heterogeneity of the dimensions leave us for now without such a distance.

The analysis frames we consider are of odd length $N = 2H + 1$ samples (using a Hann window of even size $N + 1$), with the estimation time 0 set at their center. In Equations (3.19) and (3.31), the continuous integrals turn into discrete summations over N values, with indices from $-H$ to $+H$.

To compare the estimators presented in Section 3.2.3, we consider discrete-time signals s (with sampling rate F_s), each consisting of a signal x (with fixed parameters to be estimated) embedded in some noise y. The comparison with the quadratic interpolation (Section 3.2.3.1) is left apart and is part of our future work. However, the comparison of reassignment with quadratic interpolation in various situations can be found in [BCRD08], but only for the estimation of the frequency.

The power of the noise is chosen to achieve a desired signal-to-noise ratio (SNR):

$$\mathrm{SNR}(x,y) = \frac{\mathrm{var}(x)}{\mathrm{var}(y)} \tag{3.155}$$

often expressed in dB ($10\log_{10}(\mathrm{SNR})$). In our experiments, the SNR ranges from -20 to 100dB.

Let us consider a complex sinusoid x generated according to Equation (3.18) with an initial amplitude $a = 1$ (*i.e.* initial log-amplitude $\lambda = 0$), mixed with a Gaussian white noise y of variance σ^2:

$$x[n] = s_{\phi,0,\omega,\mu,\psi}[n] \quad \text{(see Equation 3.18)}, \tag{3.156}$$

$$y[n] = \sqrt{\mathrm{SNR}} \cdot z[n] \tag{3.157}$$

where z is a Gaussian white noise of variance 1. In this complex case, the SNR is a^2/σ^2 (because the variance of a complex sinusoid of amplitude a is a^2 and the variance of the noise part is $\mathrm{var}(y) = \sigma^2$). The analyzed signal is $s = x + y$.

Musical applications consider real sinusoids rather than complex ones. In the real case, we consider instead the sum of

$$x_{\mathrm{real}} = \Re(x) \quad \text{and} \quad y_{\mathrm{real}} = y/\sqrt{2} \tag{3.158}$$

where the $1/\sqrt{2}$ normalizing factor is here to ensure the validity of Equation (3.155), because in the real case the variance of a real sinusoid of amplitude a is $a^2/2$.

As mentioned in Section 3.1.1, in practice we can also use estimators designed in the complex case with analytic signals, and more precisely complex signals obtained from real signals using the Hilbert transform.

For a given parameter ν, the estimates $\hat{\nu}$ are normally distributed. Thus, we focus on the mean and variance of the error.

Error Bias

The estimator is said to be without bias (or unbiased) if this distribution is centered on the reference value ν, *i.e.* the mean of the estimation error $\hat{\nu} - \nu$ is zero.

In practice, we could compare the estimation bias of estimators by considering the log-bias, defined as the base-10 logarithm of the absolute value of the mean error achieved at a given frequency so that this error is maximal in the considered frequency range:

$$b_{\log} = \log_{10}\left(\max_{\omega}|\hat{\omega} - \omega|\right). \tag{3.159}$$

Ideally, the estimators shall be unbiased (*i.e.* $b_{\log} = -\infty$). In practice, it is only required that this bias is low enough, which is (almost) always the case.

Error Variance

An efficient unbiased estimator should then minimize the variance of the estimation error. When evaluating the performance of an estimator in the presence of noise and in terms of the variance of the estimation error, an interesting element to compare with is the Cramér-Rao bound (CRB). The CRB is defined as the limit to the best possible performance achievable by an unbiased estimator given a data set.

For the model of Equation (3.18), for the five model parameters, the CRBs bounds have been derived by Zhou *et al.* [ZGS96]. We will consider their asymptotic versions (for a large N and a high number of observations), and give their expressions in the complex case.

We will see that the expressions of the CRBs are all inversely proportional to the SNR (a^2/σ^2), and thus each CRB will appear in function of the SNR as a line of slope -1, in the log scales. In the real case, the variance of the sinusoid is $a^2/2$ instead of a^2, and the CRBs in the real case are 2 times the complex CRBs. This will appear as a small translation of the line upwards, in the log scales. In the case of analytic signals, that is complex signals obtained from real signals by an Hilbert transform, we use the CRBs of the real case since the input signal is still the real one and the transform does not add any information for the estimator (the combination of the transform and the complex estimator results in a real estimator).

We will consider reduced parameters, *i.e.* made independent of the sampling frequency F_s. We thus define $\underline{\mu} = \mu/F_s$, $\underline{\omega} = \omega/F_s$, and $\underline{\psi} = \psi/{F_s}^2$.

Djurić and Kay [DK90] have shown that the CRBs depend on the time sample n_0 at which the parameters are estimated, and that the optimal choice in terms of lower bounds is to set n_0 at the center of the frame, *i.e.* $n_0 = H$, since the CRBs depend on

$$\varepsilon_k(\underline{\mu}, N) = \sum_{n=0}^{N-1} \left(\frac{n - n_0}{N}\right)^k \exp\left(2\underline{\mu}\frac{n - n_0}{N}\right). \tag{3.160}$$

Theoretical Bounds for Amplitude and Amplitude Modulation:

After Zhou *et al.* [ZGS96], we define:

$$D_1(\underline{\mu}, N) = 2(\varepsilon_0\varepsilon_2 - {\varepsilon_1}^2) \tag{3.161}$$

and give the expressions of the bounds for the amplitude a and amplitude modulation $\underline{\mu}$:

$$\mathrm{CRB}_{a,N}(\sigma,\underline{\mu}) \approx \frac{\sigma^2\varepsilon_2}{D_1}, \tag{3.162}$$

$$\mathrm{CRB}_{\underline{\mu},N}(\sigma,a,\underline{\mu}) \approx \frac{\sigma^2\varepsilon_0}{a^2N^2D_1}. \tag{3.163}$$

Theoretical Bounds for Phase, Frequency, and Frequency Modulation:

As explained by Zhou *et al.* [ZGS96], the expressions of the bounds are different whether there is a frequency modulation or not (because this changes the degree of the polynomial associated to the phase).

In the absence of frequency modulation ($\psi = 0$), the bounds for the phase ϕ and frequency $\underline{\omega}$ are given by

$$\mathrm{CRB}_{\phi,N}(\sigma,a,\underline{\mu}) \approx \frac{\sigma^2\varepsilon_2}{a^2D_1}, \tag{3.164}$$

$$\mathrm{CRB}_{\underline{\omega},N}(\sigma,a,\underline{\mu}) \approx \frac{\sigma^2\varepsilon_0}{a^2N^2D_1}. \tag{3.165}$$

In the presence of frequency modulation, the expressions of the bounds for the phase (ϕ), frequency ($\underline{\omega}$), and frequency modulation ($\underline{\psi}$) are given by

$$\mathrm{CRB}_{\phi,N}(\sigma,a,\underline{\mu}) \approx \frac{\sigma^2(\varepsilon_2\varepsilon_4-{\varepsilon_3}^2)}{a^2D_2}, \tag{3.166}$$

$$\mathrm{CRB}_{\underline{\omega},N}(\sigma,a,\underline{\mu}) \approx \frac{\sigma^2(\varepsilon_0\varepsilon_4-{\varepsilon_2}^2)}{a^2N^2D_2}, \tag{3.167}$$

$$\mathrm{CRB}_{\underline{\psi},N}(\sigma,a,\underline{\mu}) \approx 4\frac{\sigma^2(\varepsilon_0\varepsilon_2-{\varepsilon_1}^2)}{a^2N^4D_2} \tag{3.168}$$

where $D_2(\underline{\mu},N)$ is defined as:

$$D_2(\underline{\mu},N) = 2(\varepsilon_0\varepsilon_2\varepsilon_4-{\varepsilon_1}^2\varepsilon_4-\varepsilon_0{\varepsilon_3}^2+2\varepsilon_1\varepsilon_2\varepsilon_3-{\varepsilon_2}^3). \tag{3.169}$$

A very concise measure for the precision of the estimation is the log-efficiency (see [Bad05]), defined as the base-10 logarithm of the ratio between the variance of the error and the CRB for a given SNR:

$$e_{\log}(\nu) = \log_{10}\left(\frac{\mathrm{var}(\hat{\nu}-\nu)}{\mathrm{CRB}_\nu}\right). \tag{3.170}$$

This measure is meaningful when the variance of the estimation is almost always proportional to the CRB (with the same factor for all SNRs). However, this is often not exactly the case in practice, so in the remainder we will prefer the classic plot of the error variance as a function of the SNR, much less concise but very informative.

3.2.5.3 Phase-Based Frequency Estimators

Section 3.2.4 gave theoretic equivalences among phase-based estimators. However, practical implementation is likely to influence the performance of these estimators. Together with Lagrange [37, 6]

we compare most of these phase-based frequency estimators: the difference method ($\hat{\omega}^{\Delta}$, see Equation (3.130)), the spectral reassignment ($\hat{\omega}^{R}$, see Equation (3.86)), the discrete derivative algorithm ($\hat{\omega}^{d}$, see Equation (3.123)), the trigonometric estimator ($\hat{\omega}^{t}$, see Equation (3.143)), and the arctan estimator ($\hat{\omega}^{a}$, see Equation (3.144)).

The comparison is done in the cases of complex, real, or analytic signals (see Section 3.1.1). For each type of signal, two frequency ranges are considered: a narrow frequency range around 0.25 normalized frequency and the whole frequency range $(0, 0.5)$. The lower bound of the narrow frequency range is set to 0.24 and its upper bound to 0.26. The lower bound of the whole frequency range is set to 0 and its upper bound to 0.5. These bounds are exclusive, so that the first evaluated frequency in the whole range is 0.0025. Two SNR ranges are also considered: a high SNR range $\{20, 40, 60, 80, 100\}$ and a low SNR range $\{-20, -10, 0, 10\}$. We use frames of $N = 127$ samples. We consider 400 different frequencies in each considered range. For each frequency, 30 different phases are evaluated from $-\pi$ to $+\pi$. At each evaluation, the noise is randomized. For all the tested methods, the detection picks the greatest local maximum in the magnitude spectrum.

The results are summarized in Figures 3.9 and 3.10. It turns out that the reassignment method is biased at high SNRs (see Figures 3.10(a) and 3.9(a)). This bias is significantly reduced in the analytic case (see Figures 3.10(c) and 3.9(e)). However, the fact that this estimator performs (slightly) better in the analytic case than in the complex case is really strange. In fact, the real case is always worse than the complex case, because of the bias caused by interferences at extreme frequencies (see Figure 3.10). The analytic case should show intermediate results. But since it only slightly improves the precision (expect for the strange case of $\hat{\omega}^{R}$), its usefulness is questionable.

Also, we verify that the discrete derivative estimator $\hat{\omega}^{d}$ is biased when the whole frequency range is considered (see Figure 3.9(b)), because of the bias caused by the imprecision at high frequencies mentioned in Section 3.2.4 (see Figure 3.10(a)).

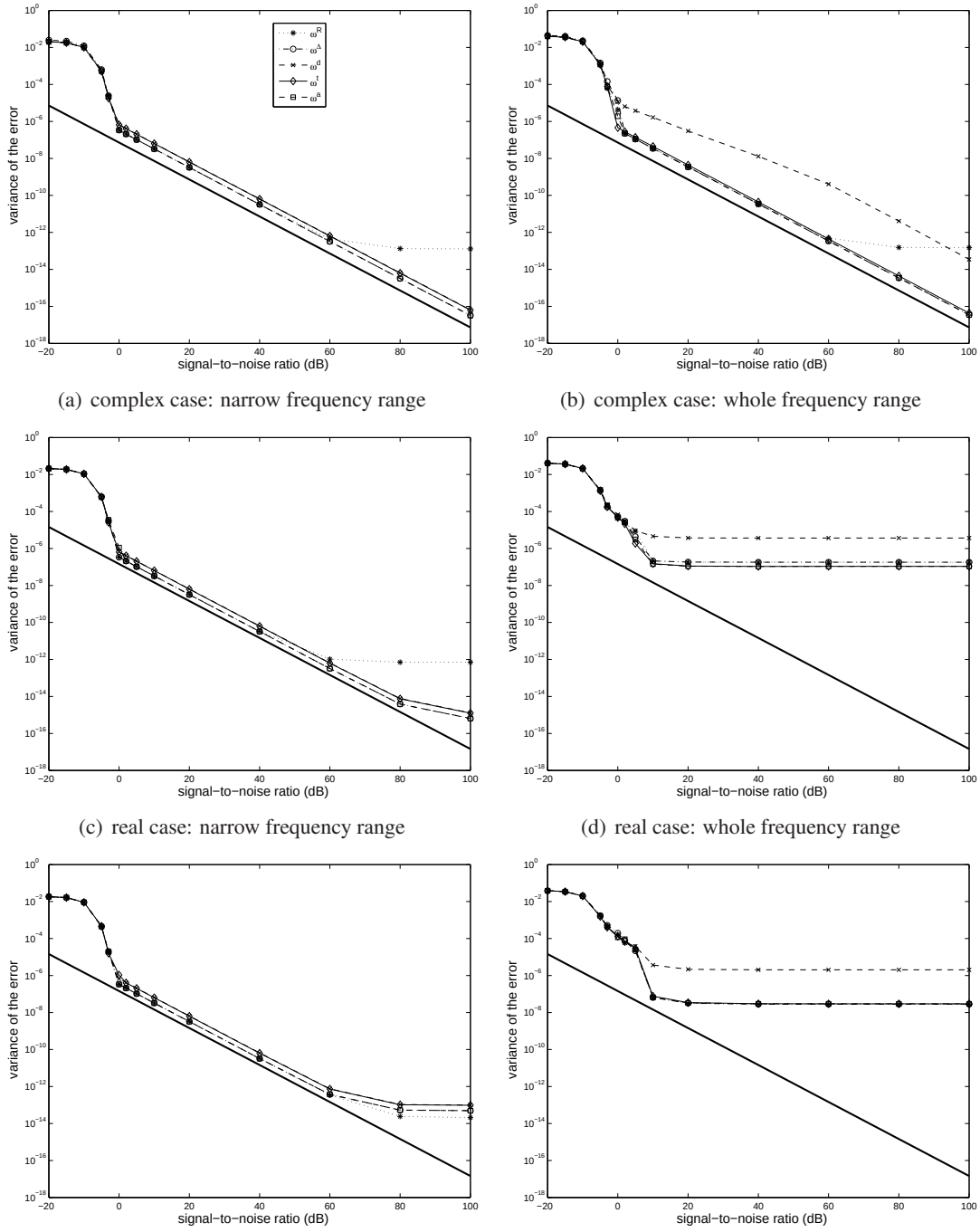
Apart from these remarks, all the estimators are roughly equivalent. We can especially note that the simple difference method $\hat{\omega}^{\Delta}$ performs quite well.

3.2.5.4 Spectral Reassignment and Derivative Algorithm

Together with Depalle [46] we quantitatively evaluate the precision of the reassignment and derivative methods for the estimation of all the model parameters, and compare them to the theoretical lower bounds.

We use here frames of $N = 511$ samples. For each SNR and for each analysis method, we test 99 frequencies (ω) linearly distributed in the $(0, 3F_s/8)$ interval, and 9 phases (ϕ) linearly distributed in the $(-\pi, +\pi)$ interval. The amplitude modulation (μ) is either 0 (no-AM case) or one of 5 values linearly distributed in the $[-100, +100]$ interval (AM case). The frequency modulation (ψ) is either 0 (no-FM case) or one of 5 values linearly distributed in the $[-10000, +10000]$ interval (FM case). We limit the frequency to $3/4$ of the Nyquist frequency because very high frequencies are problematic (cause bias) for the derivative method (see Section 3.2.3.3). This is not really problematic in practice, since with a sampling rate of 44100Hz, this mostly covers the range of audible frequencies. Moreover, nowadays sampling frequencies can be as high as 96kHz, or even 192kHz.

We then compare the reassignment method (R) – see Section 3.2.3.2 – and two variants of the derivative method (D) – see Section 3.2.3.3: the estimated derivative method (ED), where the derivatives s' and s'' are estimated using the differentiator filter of Equation (3.128); and the theoretic derivative method (TD), where the exact derivatives s' and s'' are given by Equations (3.101) and (3.105). As regards the noise, the derivatives are approximated by using the differentiator filter. We consider the TD method because the estimated derivatives can be improved (*e.g.* by increasing the order of the



(a) complex case: narrow frequency range

(b) complex case: whole frequency range

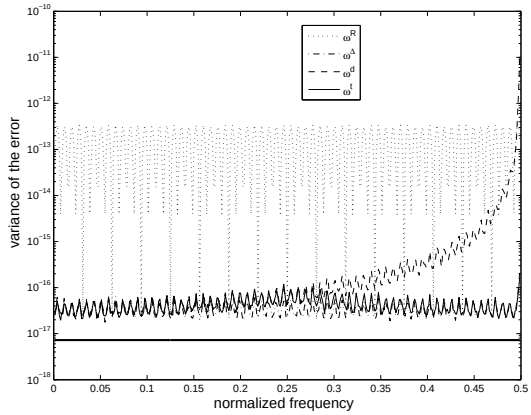
(c) real case: narrow frequency range

(d) real case: whole frequency range

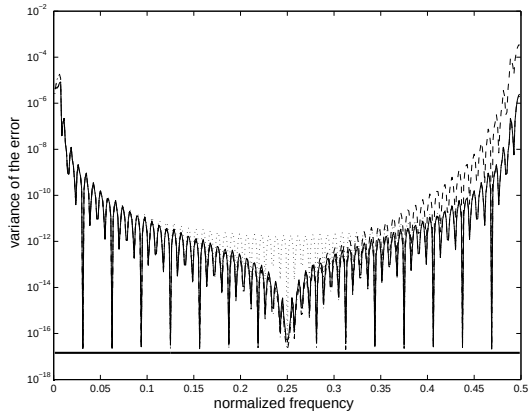
(e) analytic case: narrow frequency range

(f) analytic case: whole frequency range

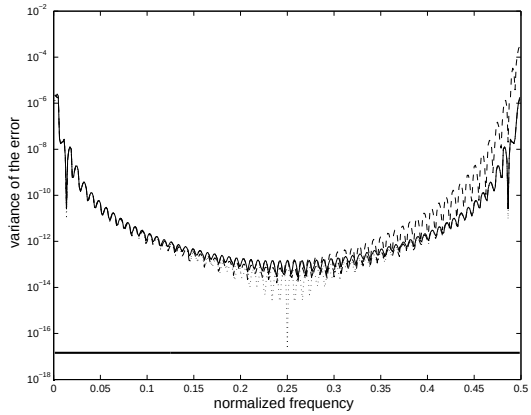
Figure 3.9: Performances of frequency estimators for the analysis of a complex (top), real (middle), or analytic (bottom) sinusoidal signal with frequency lying in the narrow $(0.24, 0.26)$ normalized frequency range (left), and in the whole $(0, 0.5)$ range (right): reassignment $\hat{\omega}^R$ (dotted line with $*$), difference $\hat{\omega}^\Delta$ (dash-dotted line with $\circ$), discrete derivative $\hat{\omega}^d$ (dashed line with $\times$), trigonometric $\hat{\omega}^t$ (solid line with $\diamond$), and arctan $\hat{\omega}^a$ (dashed line with $\square$). $\mathrm{CRB}_{\underline{\omega}}$ is plotted with a bold solid line.



(a) complex case



(b) real case



(c) analytic case

Figure 3.10: Performances of frequency estimators at SNR=100dB versus the frequency of the analyzed complex (top), real (middle), or analytic (bottom) sinusoidal signal: reassignment $\hat{\omega}^R$ (dotted line), difference $\hat{\omega}^\Delta$ (dash-dotted line), discrete derivative $\hat{\omega}^d$ (dashed line), and trigonometric $\hat{\omega}^a$ (solid line). (These are the same graphic conventions as in Figure 3.9, except that the symbols have been removed for the sake of readability.) The performances of the arctan estimator are not plotted here since they are very close to those of the trigonometric estimator. $\mathrm{CRB}_{\underline{\omega}}$ is plotted with a bold solid line.

differentiator filter). Thus the results of the TD method can be regarded as the best performance the D method could achieve, though with a better approximation of the discrete derivatives.

When looking at the results of these experiments (see Figures 3.11–3.13, it turns out that with the proposed differentiator filter, the order of 1023 is sufficient for the method to achieve in practice (ED) performances very close to the theory (TD).

Below -10dB (high-error range), the derivative method appears to make larger errors than the reassignment method (except for phase and frequency, where the two methods behave similarly). However, this area is of poor practical interest since the errors are too important.

Above -10dB, as the SNR increases the reassignment method gets almost always biased, whereas the derivative method achieves performances closer to the CRB in nearly all cases. When frequency modulation is present, the derivative method is better for the estimation of the amplitude, phase (see Figure 3.12), and amplitude modulation (see Figure 3.13); both methods are equivalent for the estimation of the frequency (see Figure 3.11); and the reassignment method is better for the estimation of the frequency modulation (see Figure 3.13), while the error of the derivative method is very low.

Moreover, we have observed the following trends: for small frame sizes, the derivative method outperforms the reassignment method in all cases. When the frame size increases, both methods are quite equivalent without frequency modulation, even if the order of the differentiator might have to be increased to lower the bias for the estimation of the amplitude modulation and frequency.

3.3 Synthesis

Once we have a precise analysis method, the problem is yet to be able to generate the signal from the model parameters in a very efficient way (and reasonable quality), since we are interested in real-time synthesis.

3.3.1 Spectral Method

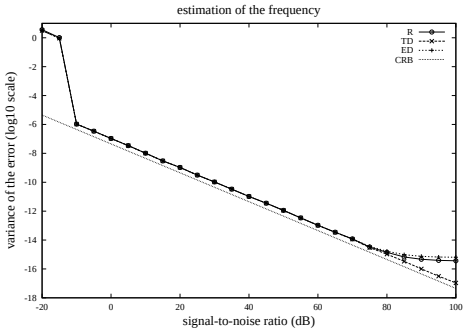
In order to efficiently synthesize many sinusoids simultaneously, Freed, Rodet, and Depalle [FRD92] propose to use the inverse Fourier transform, in the stationary case though. The idea is to reconstruct the short-term spectrum of the sound at time t, by adding the band-limited contribution of each partial (using the window spectrum, see Section 3.2), then to apply the inverse fast Fourier transform (IFFT) in order to obtain the temporal representation of the sound, and finally to repeat the same computation further in time, thus performing a kind of inverse STFT together with the overlap-add technique. Laroche [Lar00] proposes a technique without overlap. In a recent paper, Kutil [Kut08] proposes a way to choose the coefficients and the window in an optimal way, uses an optimized truncated IFFT, and studies the complexity and the practical performance on nowadays computers.

This method requires, for each frame of N samples, first to reconstruct (an approximation of) the short-term spectrum, then to switch to the temporal domain by an IFFT. The overall complexities for these two steps are, respectively

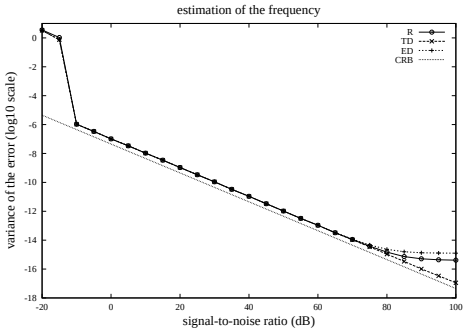
$$\begin{aligned} C_1 &\propto PS/N, \\ C_2 &\propto N\log(N)S/N = \log(N)S \end{aligned}$$

where P and S are, respectively, the number of partials and samples to be synthesized. Thus, the overall complexity per sample and per partial for this method is of the form

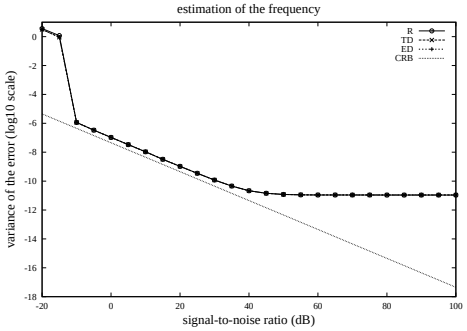
$$\underline{C}_{\text{IFFT}} = O(1/N + \gamma\log(N)/P) \tag{3.171}$$



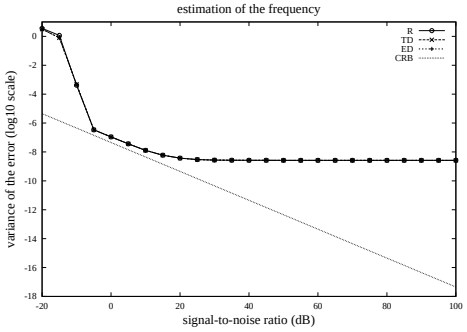
(a) stationary case ($\mu = 0, \psi = 0$)



(b) AM-only case ($|\mu| \leq 100, \psi = 0$)

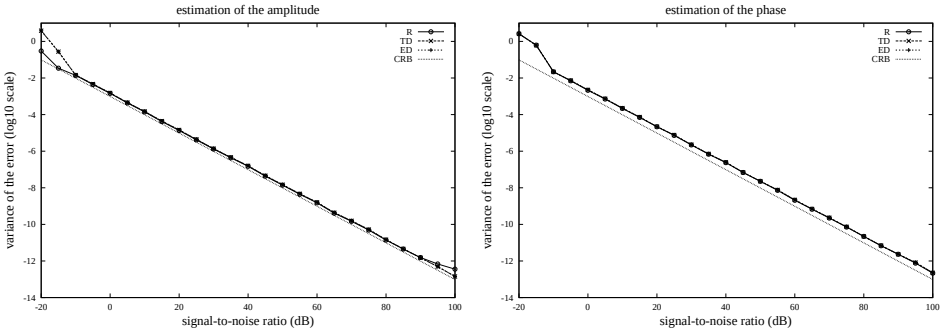


(c) FM-only case ($\mu = 0, |\psi| \leq 10000$)

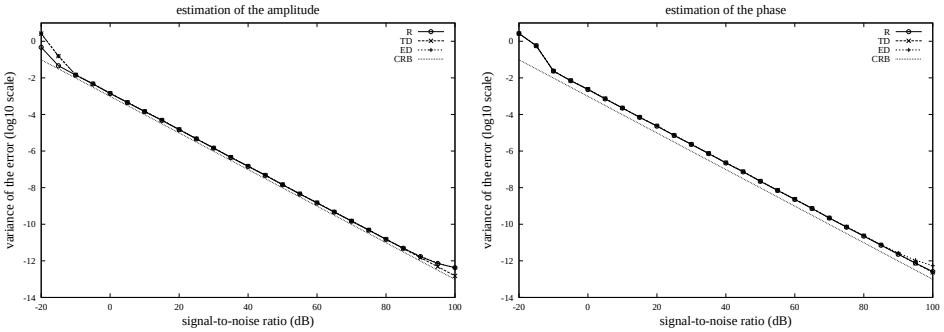


(d) AM/FM case ($|\mu| \leq 100, |\psi| \leq 10000$)

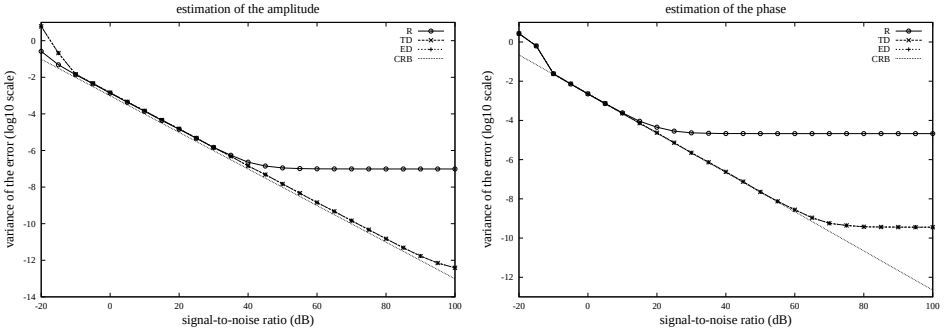
Figure 3.11: Frequency estimation error as a function of the SNR (stationary, AM-only, FM-only, and AM/FM cases) for the R, ED, and TD methods, and comparison to $\mathrm{CRB}_{\underline{\omega}}$.



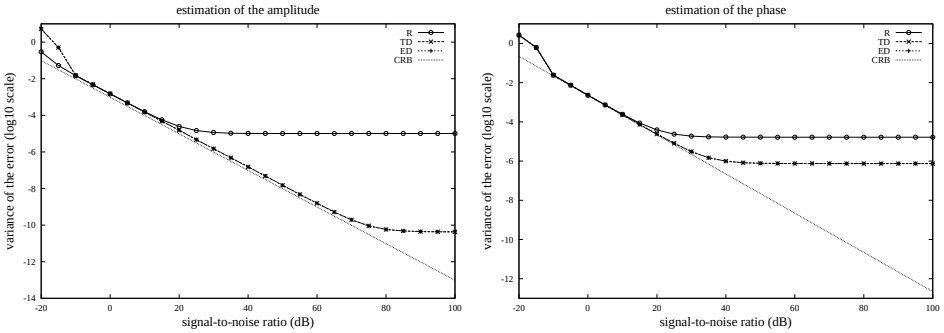
(a) stationary case ($\mu = 0, \psi = 0$)



(b) AM-only case ($|\mu| \leq 100, \psi = 0$)

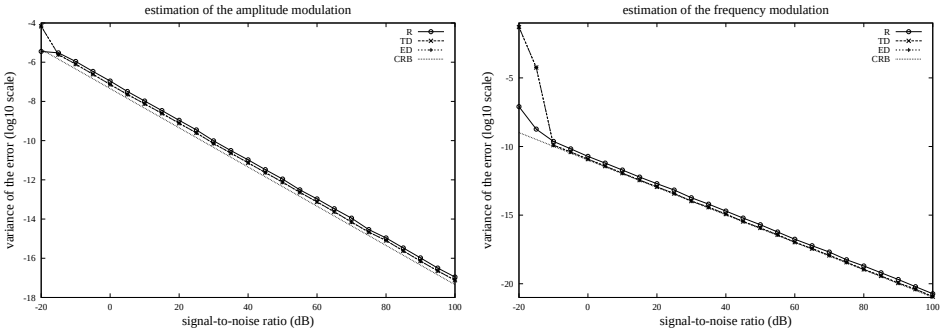


(c) FM-only case ($\mu = 0, |\psi| \leq 10000$)

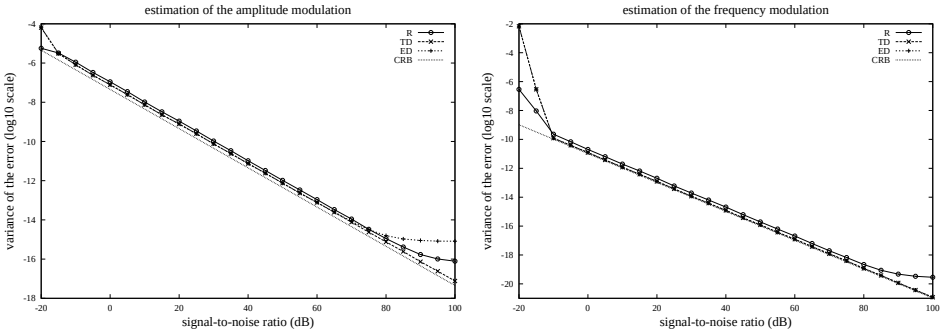


(d) AM/FM case ($|\mu| \leq 100, |\psi| \leq 10000$)

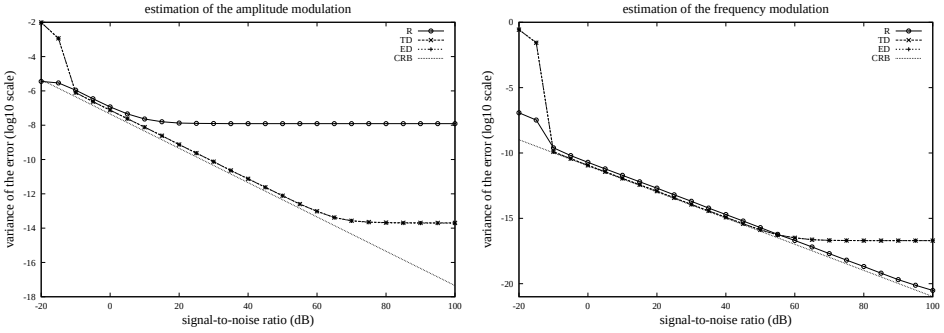
Figure 3.12: Amplitude (left) and phase (right) estimation errors as functions of the SNR (stationary, AM-only, FM-only, and AM/FM cases) for the R, ED, and TD methods, and comparison to CRB_a and CRB_ϕ, respectively.



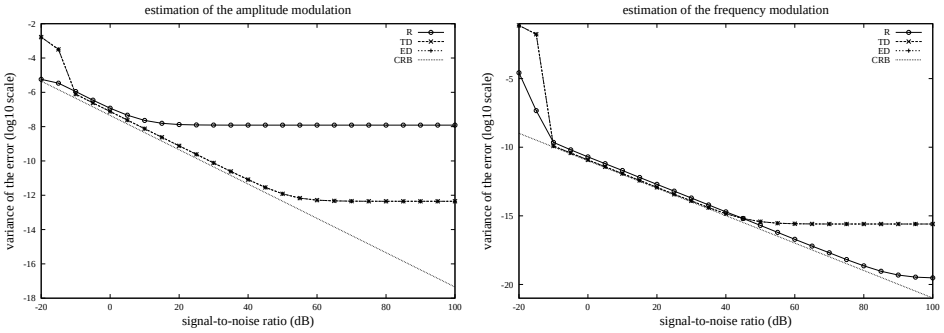
(a) stationary case ($\mu = 0, \psi = 0$)



(b) AM-only case ($|\mu| \leq 100, \psi = 0$)



(c) FM-only case ($\mu = 0, |\psi| \leq 10000$)



(d) AM/FM case ($|\mu| \leq 100, |\psi| \leq 10000$)

Figure 3.13: Amplitude modulation (left) and frequency modulation (right) estimation errors as functions of the SNR (stationary, AM-only, FM-only, and AM/FM cases) for the R, ED, and TD methods, and comparison to $\mathrm{CRB}_{\underline{\mu}}$ and $\mathrm{CRB}_{\underline{\psi}}$, respectively.

where γ is an architecture-dependent coefficient.

Thus, the gain in complexity is when the number of oscillators P is large in comparison to the number of samples N to be computed at each frame. This approach is very interesting, because its overall complexity for a fixed synthesis time is no more the product of the number of partials and the sampling rate as with the software oscillators (Section 3.3.2.1).

However, the control of the parameters is more complex. Changing them at frame boundaries might be insufficient, especially if the frame size N is large. After Laroche [Lar00], Kutil [Kut08] shows that it is possible to adapt the synthesis algorithm to in-frame linear amplitude modulation, at the expense of an increased complexity though. Also, the short-term spectra are only approximated, thus the performance is at the expense of some degradation in the synthesis quality.

3.3.2 Temporal Methods

The temporal methods have more suppleness, in a sense that the parameters can be changed at almost any time. It is then just a matter of applying Equation (3.4) as fast as possible. The direct computation of the (co)sine function in this equation should be avoided. Early attempts were using wave tables to approximate the (co)sine function, which was suitable when memory accesses were faster than CPU instructions. Nowadays, this is not the case anymore.

3.3.2.1 Software Oscillators

A first solution is to evaluate the (co)sine function incrementally. For this purpose, many "software oscillators" algorithms have been proposed. Their overall complexity is always of the form

$$C_{\text{osc}} = O(PS) \quad i.e. \quad \underline{C}_{\text{osc}} = O(1) \tag{3.172}$$

meaning that their complexity per sample and per partial is a constant. The problem is then to minimize this constant.

Coupled Form

As shown in Figure 3.7, in the stationary case, from one sample to the next, the complex signal value is rotated by a phase increment Δ_ϕ corresponding to the reduced frequency $\underline{\omega}_0$. This rotation in the complex plane can be done simply by a multiplication among complex numbers. Thus, the coupled form algorithm is defined by

$$\begin{cases} C & = & e^{j\underline{\omega}_0} \\ s[0] & = & a_0 e^{j\phi_0} \\ s[n+1] & = & C \cdot s[n] \quad \forall n \geq 0 \end{cases} \tag{3.173}$$

and this algorithm requires only one complex multiplication per sample, that is 4 real multiplications and 2 real additions, and achieves a perfect synthesis quality.

This algorithm can easily be generalized to the non-stationary case. For the exponential amplitude (linear log-amplitude) modulation, it can be done without affecting the complexity: it is just a matter of multiplying the constant C by $\exp(\underline{\mu}_0)$ in Equation (3.173). For the linear frequency modulation,

the generalization is also possible this way:

$$\begin{cases} D & = & e^{j\underline{\psi}_0/2} \\ C_0 & = & e^{\underline{\mu}_0+j\underline{\omega}_0} \\ s[0] & = & e^{\lambda_0+j\phi_0} \\ C_{n+1} & = & D\cdot C_n \quad \forall n\geq 0 \\ s[n+1] & = & C_n\cdot s[n] \quad \forall n\geq 0 \end{cases} \tag{3.174}$$

but at the expense of doubling the complexity.

Digital Resonator (DR)

In Section 4.3, we will show that although non-stationary synthesis increases the synthesis quality in theory, in practice natural sounds do not exactly follow the model and this gain in quality is not clear anymore. Thus, we might prefer to favor computation speed and thus revert to the simplest case: linear phase (constant frequency) and constant amplitude, at least over a small time frame.

Gordon and Smith [GS85] use the digital resonator (DR) to perform the incremental computation of the sine function using two previous values to compute the new one with only one multiplication and one addition. This algorithm is optimal in terms of complexity. It is indeed impossible to achieve better results, since an addition without multiplication or a multiplication without addition will produce, respectively, only arithmetic or geometric progressions, far from to the expected sinusoids.

More precisely, since

$$s[n] = a_0\cos(\phi_0+\underline{\omega}_0 n) \tag{3.175}$$

we have

$$\begin{aligned} s[n+1] &= a_0\cos(\phi_0+\underline{\omega}_0(n+1)) \\ &= a_0\cos(\phi_0+\underline{\omega}_0 n+\underline{\omega}_0) \\ &= a_0\cos(\phi_0+\underline{\omega}_0 n)\cos(\underline{\omega}_0)-a_0\sin(\phi_0+\underline{\omega}_0 n)\sin(\underline{\omega}_0) \end{aligned}$$

and

$$\begin{aligned} s[n-1] &= a_0\cos(\phi_0+\underline{\omega}_0(n-1)) \\ &= a_0\cos(\phi_0+\underline{\omega}_0 n-\underline{\omega}_0) \\ &= a_0\cos(\phi_0+\underline{\omega}_0 n)\cos(\underline{\omega}_0)+a_0\sin(\phi_0+\underline{\omega}_0 n)\sin(\underline{\omega}_0) \end{aligned}$$

thus

$$\begin{aligned} s[n+1]+s[n-1] &= 2a_0\cos(\phi_0+\underline{\omega}_0 n)\cos(\underline{\omega}_0) \\ &= 2\cos(\underline{\omega}_0)s[n] \\ &= C\cdot s[n] \\ &\text{where} \quad C=2\cos(\underline{\omega}_0). \end{aligned}$$

Finally, the DR algorithm can be summarized by the following system:

$$\begin{cases} C & = & 2\cos(\underline{\omega}_0) \\ s[0] & = & a_0\cos(\phi_0) \\ s[1] & = & a_0\cos(\phi_0+\underline{\omega}_0) \\ s[n+1] & = & C\cdot s[n]-s[n-1] \quad \forall n\geq 0. \end{cases} \tag{3.176}$$

However, there might be problems with numeric stability [MP02, GS85], because the DR represents an IIR filter with two poles on the unit circle at $e^{\pm j\omega_0}$.

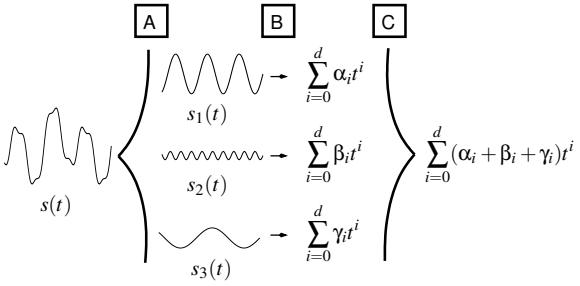


Figure 3.14: The PASS method. *Step A:* a periodic signal can be divided into sinusoidal components. *Step B:* computing polynomial coefficients to approximate the signal for each partial. *Step C:* a polynomial generator is obtained by summing the coefficients from the polynomials of the partials. The values computed by the generator will be the samples of the sound signal.

3.3.2.2 Polynomial Generator

The problem of the preceding software oscillators is that their complexity is proportional to PS, that is to the PF_s product for a fixed synthesis time, and both P and F_s are often very large. In these conditions, with an appropriate frame length the spectral approach (Section 3.3.1) can perform best [MP02, Kut08].

Together with Robine and Strandh [38], we propose to use polynomials to approximate the sine function. Our polynomial additive sound synthesis (PASS) method consists in first calculating a set of polynomial coefficients for each partial. Polynomial values from polynomials computed with these coefficients approximate the signal of the partial on a part of its period. The classic approach would evaluate the polynomial associated to each oscillator, and then sum up the results, which is quite inefficient. The idea is yet to sum the coefficients in a polynomial generator, then to evaluate the resulting polynomial only once. Indeed, summing polynomials leads to another polynomial of the same degree. The sound samples can be computed from this single resulting polynomial, with a fairly low degree – independent of the number of partials to synthesize. The general process is illustrated by Figure 3.14.

Partial Approximation

In the stationary case, the time-domain signal generated for each partial is defined by a sine function. We propose to approximate this function by polynomials. We have to choose a part of the period where we will do the approximation. Thus, we uniformly divide the period of the sine function in D parts.

Measuring the performance of the approximation of a signal $u(t)$ by a polynomial $U(t)$ on a validity period $T = 1/D$ can be done using the SNR over this period (see Equation (3.155)), and more precisely by maximizing $\mathrm{SNR}(u, u-U)$, which is equivalent to minimizing its denominator. Moreover, the polynomial coefficients have to respect other constraints to maintain a piecewise continuity. For example, with a polynomial of degree $d = 2$, a period division of $D = 2$, to impose a $\mathcal{C}^1$ continuity, it is sufficient that $U(0) = U(1/2) = 0$, and the coefficients c_i that minimize

$$\int_0^{\frac{1}{2}} (\sin(x) - (c_0 + c_1 x + c_2 x^2))^2 dx$$

D	d	$\mathcal{C}^0$ SNR (dB)	$\mathcal{C}^1$ SNR (dB)
4	2	36	28
4	3	57	28
4	4	79	59
4	5	102	59
2	2	28	28
2	3	28	28
2	4	59	59
2	5	59	59
1	4	17	17
1	5	42	42

Table 3.1: Error of polynomial approximation of a partial. For two different continuity requirements ($\mathcal{C}^0$ and $\mathcal{C}^1$), the signal-to-noise ratio (SNR) obtained with the approximation error $u-U$ compared to the target signal u are shown as functions of the period division D and the polynomial degree d.

are

$$\begin{cases} c_0 &= 0 \\ c_1 &= 120/\pi^4 \\ c_2 &= -120/\pi^5. \end{cases}$$

To approximate the sine function, we can use in this case alternately $U_1(x) = c_1 x + c_2 x^2$ for the first half period and $U_2 = -U_1$ for the second.

The choice of the polynomial degree d and period division D influences the performance of the approximation, as shown in Table 3.1. A high polynomial degree can lead to numerical instability, and a short validity period increases the computation time. We can note that using a period division $D = 2$ with a polynomial degree $d = 2$ is particularly suited for a very fast synthesis (or $d = 4$ for a better quality and still an interesting speed).

The coefficients we compute define a unit polynomial $U(t)$ by validity period. When the unit polynomial is found, every partial can be approximated from it. In the general case of a partial p with amplitude a_p, frequency $\omega_p = 2\pi f_p$, and initial phase ϕ_p, the approximating polynomial Π_p is then given by

$$\Pi_p(t) = a_p U(\phi_p + \omega_p t)$$

Notice that the amplitude, frequency, or phase parameters do not modify the approximation error given in Table 3.1. In addition to the sinusoid, the polynomial approximation generates a noise consisting of harmonics of this sinusoid. The magnitudes of these harmonics are small, and depend on the period division D and the polynomial degree d.

Since for now we consider only constant parameters for the partials, the generated functions are periodic. It is thus possible to compute the polynomial coefficients for only one period of any partial.

Each set of polynomial coefficients is valid for a part of the period of the sine function. For example, if we choose to approximate sine functions using a half of their period, we need to compute two sets of coefficients by partial. As long as the amplitude and the frequency of a partial are constant, we can continue with the same pre-calculated sets. During the sound synthesis, the coefficients that approximate the partials must be updated regularly (the rate depending on the frequency of each partial), and must also be changed if the parameters have changed.

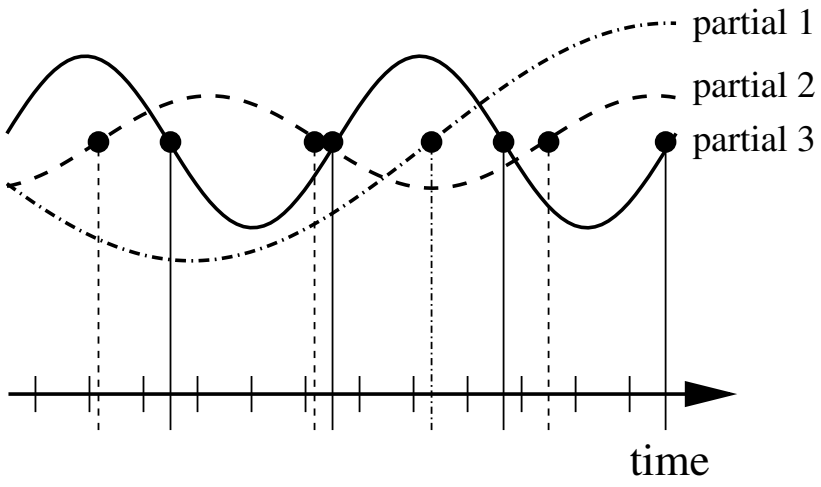


Figure 3.15: Update events for 3 partials (dash-dotted, dashed, and solid lines). For each partial, they occur here at each $1/D$ period (here $D = 2$) of the sinusoid of the partial, possibly between two output samples.

Incremental Calculation of Polynomials

To avoid the problem of computing a polynomial with large time values, leading to numerical imprecision, we propose to use the Taylor's theorem to compute it. The polynomial can be evaluated at every instant $t_0 + \Delta_t$ by using its value and the values of its derivatives at a preceding instant t_0:

$$\Pi^k(t_0 + \Delta_t) = \Pi^k(t_0) + \sum_{i=k+1}^{d} \frac{\Delta_t^{i-k}}{(i-k)!} \Pi^i(t_0) \tag{3.177}$$

where Π^k is the k-th derivative of the polynomial function (Π^0 being the polynomial itself). The number of necessary values depends on the degree of the polynomial (*e.g.* three values with a degree-2 polynomial).

With the polynomial coefficients corresponding to the partials, we compute the first value of the polynomial and of its derivatives. To compute each of the following values, we use Equation (3.177) with a step Δ_t corresponding to the time between two time events. A time event is either the time of a sound sample or of a scheduled update of the coefficients. When time reaches or exceeds the period division we have chosen, the coefficients are updated, and the incremental algorithm goes on with the new coefficients.

Global Polynomial Generator

Using this incremental method for each individual partial would be very expensive in terms of computation time. For that reason, we propose a technique in which we sum the coefficients to compute only one global polynomial, the generator. During the synthesis, the generator is computed incrementally. When a partial reaches the end of its period division, the different values of the generator (value and values of the derivatives) are summed with the new values from the partial. When a sound sample must be produced, the generator is computed to get the sound sample value.

As the generator is computed incrementally, we have to care about numerical precision: using preceding values to compute new ones accumulates floating-point precision errors in the result. Thus,

there is a validity limit for updating the generator. According to the polynomial degree used, to the number of partials in the sound, and to the floating-point precision we can use, we need to re-initialize the generator coefficients regularly with the authentic sine function.

Efficient Data Structure: an Optimized Heap as a Priority Queue

The complexity of our method is dominated by the management of the update events from individual partials. To optimize this process, we propose the use of an efficient data structure: a priority queue implemented as a binary heap.

A priority queue is an abstract data type supporting two operations: *insert* adds an element to the queue with an associated priority; *delete-max* removes the element from the queue that has the highest priority, and returns it. We use a priority queue to manage update events from partials of the sound. During synthesis, update events are regularly inserted, or processed and removed.

The standard implementation of priority queues is based on binary heaps (see [AHU83]). With this implementation, queue operations have $O(\log(P))$ complexity, with P being the number of elements in the queue. A binary heap is a binary tree satisfying two constraints:

1. the tree is either a perfect binary tree, or, if the last level of the tree is not complete, the nodes are filled from left to right;

2. each node is greater (in priority) than or equal to each of its children. The top of the binary heap is always the next event to process.

Most of the computation time of the PASS method is due to the management of the heap. Consequently, heap primitives need to be highly optimized. Our first approach was to *delete-max* the priority queue, to process the update event and to *insert* a new one in the queue. With this approach, the heap is substantially reorganized twice for each insert / delete pair of operations.

To improve the performance, we replaced the insert / delete primitives with *top* and *replace*. The *top* method returns the element of the queue that has the highest priority, without removing it. It is possible to process an update event without removing it from the queue. The *replace* method replaces the element from the queue that has the highest priority with an other element by initially putting it on top of the heap, and then letting it trickle down according to normal heap-reorganization primitives. In the worst case, the *replace* method needs $O(\log(P))$ operations, P being the number of elements in the heap. The heap is reorganized only once per update. Partials with high frequencies must be updated more often than the others, because their validity period is smaller. With the *replace* method, the update events concerning high frequencies stay near to the top of the heap. Using fewer operations for the most frequent updates improves the complexity of our method. Figures 3.16 and 3.17 give examples of heap management, where the *delete-max* then *insert* methods use 4 elements swaps, whereas the *replace* method takes only 1 swap.

Complexity and Results

The complexity of the PASS method depends on the use of the priority queue. For one second of synthesis, the priority queue will be used $f_p D$ times per partial, where D is the period division chosen and f_p is the frequency of the partial (in Hz). For every partial, the queue operations are called X times per second, where

$$X = D \sum_{p=1}^{P} f_p = DP\bar{f} \tag{3.178}$$

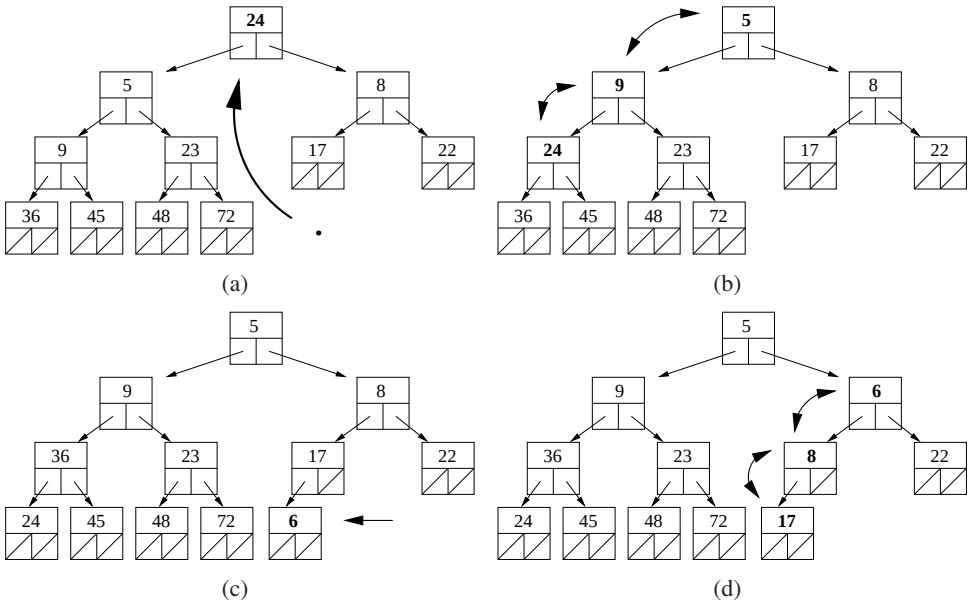


Figure 3.16: *delete-max* then *insert* priority queue methods within a heap: (a) *delete-max* of the element with highest priority; (b) heap reorganization; (c) *insert* of the new element; (d) heap reorganization.

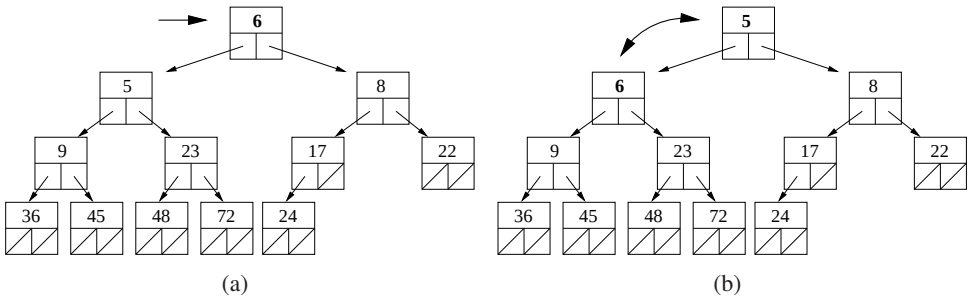


Figure 3.17: *replace* priority queue method with a heap: (a) *replace* of the element with highest priority by the new one; (b) heap reorganization.

with P being the number of partials in the sound, D the period division, f_p the frequency of the partial p, and we denote by $\bar{f}$ the mean frequency of the P partials.

If we consider that each priority queue operation is done with $O(\log(P))$ complexity for an update event, with P being the number of elements simultaneously in the queue (*i.e.* the number of partials in the sound), the complexity C_1 due to the priority heap managing is given by

$$C_1 \propto P\log(P)\frac{\bar{f}}{F_s}DS \tag{3.179}$$

where S is the number of samples to synthesize. We can note that this complexity is very dependent from the mean frequency of the partials. This explains why the method is particularly efficient for low-frequency signals. If the period division D is halved (from a fourth to a half of period for example), the performance of the PASS method is doubled. The lower is D and the better is the complexity of the method. But if we want to decrease D, we need to use a higher polynomial degree to approximate the sine function. And it leads to numerical instability. We have to find a trade-off between complexity and stability.

In addition to C_1 is the C_2 complexity, to produce the samples of the sound:

$$C_2 \propto dS. \tag{3.180}$$

Thus, the general complexity of PASS is of the form

$$\underline{C}_{\text{PASS}} = O\left(\log(P)\frac{\bar{f}}{F_s}D + \gamma d/P\right) \tag{3.181}$$

where γ is some constant which is architecture-dependent (in practice, the synthesis methods were implemented in C language, compiled using the GNU C compiler (gcc) versions 4.0 and 4.1, and executed on PowerPC G4 1.25-GHz and Intel Pentium 4 1.8-GHz processors). This global complexity is a function of the sum of the frequencies of the partials, and we notice that it does not strongly depend on the sampling rate anymore. Indeed, increasing the sampling rate F_s from 44.1kHz to 96kHz does not really affect the computation time, as illustrated by Table 3.2.

One might reasonably ask how our method compares to other methods for additive synthesis. In the Ph.D. manuscript ([Mar00], Section 4.1.3) we showed that the DR method was a little better than the IFFT method. But later, Meine and Purnhagen [MP02] reached a different conclusion. Recently, Kutil [Kut08] published a detailed comparison for various values of P and N. In fact, this highly depends on the implementation details and architecture used, as well as the number of partials and sampling rate. However, the DR method easily allows the fine control of each partial of the sound, which is not really the case the IFFT method. The PASS method allows the same fine control, and thus we have compared the performance of our method with that of the DR method. The difference of complexity between the DR and PASS methods is illustrated by Table 3.2.

As shown in Table 3.3 or in Figure 3.18, PASS is clearly better than DR for low frequencies. Using a degree-2 polynomial to approximate a half of a period of each partial, PASS is better for 2500 partials when the mean frequency of the partials is under 300Hz, and even 500Hz for a 96-kHz sampling rate. Real-time synthesis can be achieved with 5000 partials with a frequency of 150Hz for example.

3.3.2.3 Hybrid Method

In the near future, we plan to tune the trade-off between complexity and stability for our method on the fly, by combining the advantages of the PASS and DR methods, since these methods both manipulate

P	$\bar{f}$ (Hz)	F_s (Hz)	DR (s)	PASS (s)
4000	300	22050	**3.2**	6.6
4000	300	44100	**6.3**	6.6
4000	300	96000	13.7	**6.6**

Table 3.2: Comparison of the computation time of the DR and PASS methods using different sampling rates, for 5 seconds of sound synthesis, implemented in C language, compiled using the GNU C compiler (gcc) version 4.1, and executed on an Intel Pentium 4 1.8-GHz processor. The PASS method is used with degree-2 polynomials and a period division $D = 2$. P is the number of partials, $\bar{f}$ is the mean frequency of the partials, and F_s is the sampling rate.

P	$\bar{f}$ (Hz)	DR (s)	PASS (s)
2500	200	3.9	**2.0**
2500	300	3.9	**3.0**
2500	400	**3.9**	4.0
2500	500	**3.9**	5.0
5000	200	7.9	**7.3**
5000	300	**7.9**	10.6
5000	400	**7.9**	14.4

Table 3.3: Comparison of the computation time of the DR and PASS methods, for 5 seconds of sound synthesis with a sampling rate of 44100Hz, implemented in C language, compiled using the GNU C compiler (gcc) version 4.1, and executed on an Intel Pentium 4 1.8-GHz processor. The PASS method is used with degree-2 polynomials and a period division $D = 2$. P is the number of partials, $\bar{f}$ is the mean frequency of the partials.

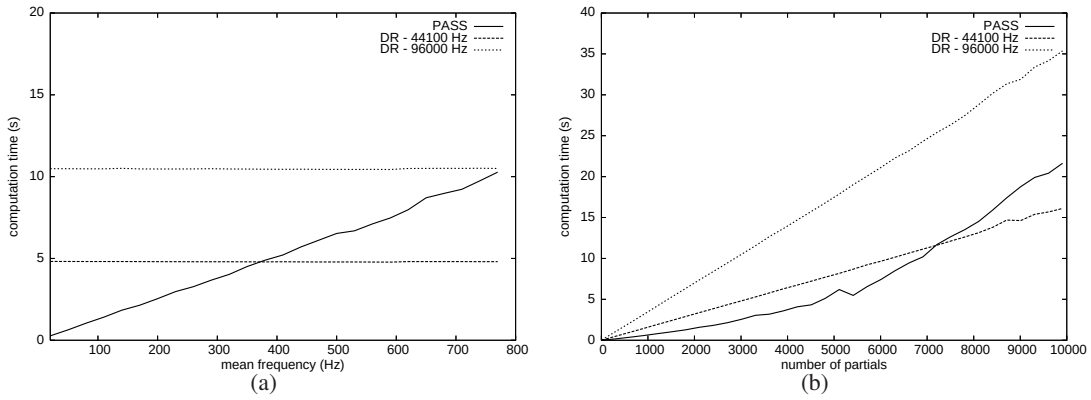


Figure 3.18: Comparison of the computation times of the DR and PASS methods, for 5 seconds of sound synthesis. (a) Computation times are functions of the mean frequency $\bar{f}$, with a fixed number of partials $P = 3000$. (b) Computation times are functions of the number of partials P, with a fixed mean frequency $\bar{f} = 200$Hz. The PASS method is used with degree-2 polynomials and a period division $D = 2$. For this comparison, both methods were implemented in C language, compiled using the GNU C compiler (gcc) version 4.0, and executed on a PowerPC G4 1.25-GHz processor.

oscillators. For this hybrid method, the idea is, for a given partial, to use either PASS or DR depending on the frequency of the partial. For low frequencies, PASS will be preferred. Also, the DR method will take advantage of the priority queue to schedule its optimal update times (see Section 4.3.3). At each update time, for the concerned oscillator the decision of switching from PASS to DR or from DR to PASS could be done. And since at this update time, the amplitude, frequency, and also phase of the oscillator is known, the switch of method is really straightforward.

3.3.3 Taking Advantage of Psychoacoustics

When we cannot reduce the complexity of the solution, we can try to reduce the size of the problem. Thus, before the classic additive synthesis algorithm itself, together with Lagrange [20] we propose to add another algorithm in order to reduce the number of partials to be synthesized by filtering out inaudible partials.

3.3.3.1 Threshold of Hearing

This algorithm first decides whether a partial can or cannot be detected by a listener in a noiseless environment. More precisely, for each partial we first compare its amplitude a_p to the threshold of hearing $\mathcal{T}_a$ at the corresponding frequency f_p (see Section 1.5). If it turns out that this partial could be heard, then we check if it is not masked by some other stronger partial.

3.3.3.2 Frequency Masking

More precisely, we propose an algorithm to decide whether a partial is masked or not, while computing the global mask M incrementally (see Section 1.5 and Figure 3.19). First, the partials are sorted by decreasing amplitudes. Then, for each partial p:

- if $M(f_p)+\Delta < V(a_p)$, then p is a **masking** partial and M must be updated with its contribution;
- if $M(f_p) < V(a_p) \leq M(f_p)+\Delta$, then p is neither masking nor masked;
- if $V(a_p) \leq M(f_p)$, then p is simply **masked**.

where $V(a)$ is the volume in dB corresponding to the amplitude a (see Equation (1.54)).

We use a list in order to store the contributions of the masking partials to the global mask M. Since this list is ordered by increasing frequencies, only the left and right neighbors are to be considered when inserting the contribution of a new masking partial. The new partial cannot mask them, since its amplitude is lower than theirs (remember that the partials have been previously sorted by decreasing amplitudes), but it can shorten the frequency area where they contribute to M.

The contributions of the masking partials to the global mask M are stored in a double-linked list, sorted by increasing frequencies. In order to decide if a new partial p is masking, neither masking nor masked, or simply masked, we need to search the list for the two contributions surrounding its frequency f_p. If it turns out that p is a masking partial, then its contribution must be inserted into the list at the right position – in order to maintain the order of the list – and the contributions of its neighbors are to be updated.

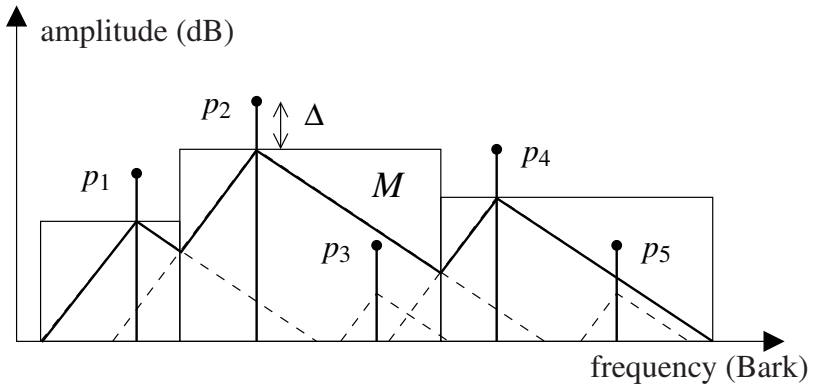


Figure 3.19: Five partials and the associated mask M (bold polyline). p_1, p_2, and p_4 are masking partials and contribute to M. The frequency areas of their contributions are represented by rectangles. p_5 is neither masking nor masked, and p_3 is masked (by p_2).

Efficient Data Structure: Skip List

As a consequence, we need a data structure ordered in frequency, with the notion of left and right neighbors, and where searching and inserting are as fast as possible. Thus, we choose a tree-like structure, the skip list, introduced by Pugh [Pug90] as an alternative to balanced trees. Both data structures show a logarithmic complexity for searching.

Let us first explain the mechanism of the skip list on the example of Figure 3.20. To find out the node with a key value equal to 6 in this skip list, we begin at the top of the head pointer array (leftmost node). The node pointed is `NIL` (rightmost node), which contains a key value greater than those of all the nodes in the list. So, we have to go down in the array of pointers. The pointed node has now the key value 2, which is lower than 6. We can safely skip to node 2 (without going through node 1) and start again the same process on this new node. Doing so, we jump directly to node 6. With this mechanism, we can search very efficiently a list for a value. At the end of a search, if the current node does not contain the desired key value, then the searched value was simply not present in the list. Concerning insertion, balanced trees algorithms explicitly balance the data structure during insertions – which is time consuming – whereas skip lists algorithms balance it probabilistically, by choosing the size of the array of pointers of the new node randomly. Thus, insertion is a simple process, consisting mainly in searching for the first node with a key value greater than the one of the new node, followed by pointer updates. The cost of pointer updates is constant, that is why insertion and searching are both $O(\log(Q))$ in terms of complexity – Q being the number of elements in the list – which is optimal.

Let us then explain how we generate the mask using this data structure. We first allocate enough memory to be able to store at least the contributions of all the partials and initialize the head and `NIL` nodes. Then, for each partial p which has an amplitude a_p greater than the threshold of hearing, we try to insert its contribution into the global mask M.

We first search M for the lowest node with a frequency greater than f_p – the right neighbor. During this search, we store all the addresses of the nodes whose frequency is greater than f_p. We obtain the left neighbor with the backward link of the right neighbor. With this local information (left and right neighbors), we check whether the contribution of the partial p has to be inserted or not.

In case of an insertion, we choose randomly the size of the array of pointers of the new node

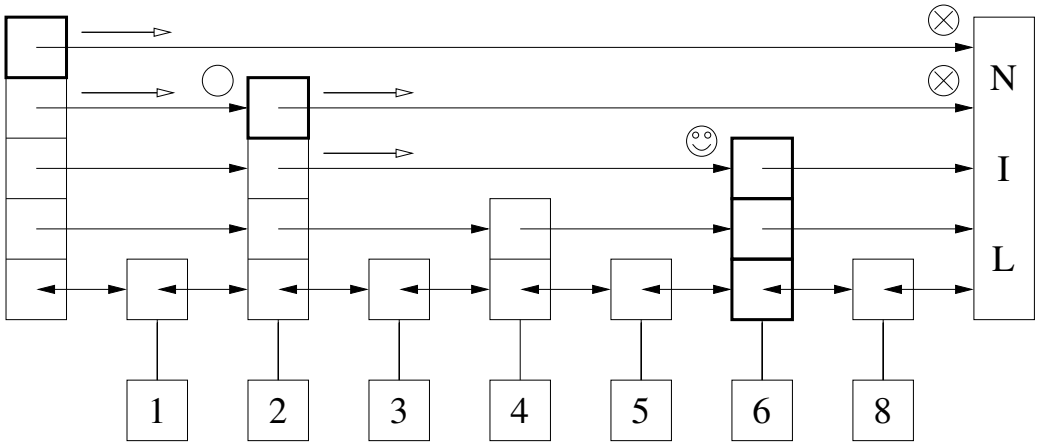


Figure 3.20: Example of a skip list. Bold rectangles indicate addresses that would be updated during the insertion of value 7 in the skip list.

– lower than the size of the head array though. We copy in this array all the addresses previously stored. Finally, we update the pointers structures with the address of the new element. At the end of our algorithm, we know which partials have to be synthesized and we can send them to our efficient algorithm for additive synthesis.

3.3.4 About Non-Linear Techniques

A function g is linear if and only if $g(\alpha x + \beta y) = \alpha g(x) + \beta g(y)$ for all x, y, and constants α and β. The summation in Equation (3.4) is an example of such a linear function. With this linear approach for the synthesis, the computation time is likely to be proportional to the number of partials P, which can be problematic for large values of P. On the contrary, non-linear functions can generate many frequencies, that are not in the original signal. This is often regarded as a drawback. However, Risset [Ris69], Arfib [Arf79], Le Brun [Bru79], or Chowning [Cho73] propose non-linear synthesis techniques, for musical creation purposes.

During the Master's Thesis of Robine, we have investigated the use of non-linear techniques for additive synthesis, possibly with a restriction to the synthesis of harmonic sounds. More precisely, we wanted to generate a large number of partials from a reduced number of oscillators, while still being able to control the parameters of these partials to some extent. The results are summarized in his Ph.D. manuscript ([Rob06], Chapter 7). However, since these results are written in French, we will give here a very short overview. We have identified 3 interesting approaches, namely spectral enrichment, formant-based synthesis, and a source / filter scheme.

3.3.4.1 Spectral Enrichment

The first possibility is to use a linear technique to precisely generate some signal, then use a non-linear technique for the enrichment of its spectrum. This is used for example in MPEG (de)coders.

Using a Non-Linear Function

Any non-linear function admitting a Taylor expansion in the $[-1,+1]$ interval can be used to generate a large (possibly infinite) number of partials from only one sinusoid. Indeed, if g is such a non-linear

function, approximated by a polynomial (of infinite degree)

$$g(x) = \sum_{p=0}^{+\infty} g_p x^p \tag{3.182}$$

then for one input sinusoid $x(t) = \cos(\omega t)$, the function g generates many partials with frequencies which are multiples of ω, because (from Euler's formula and binomial expansion)

$$\cos^{2h}(\omega t) = \frac{1}{2^{2h}}\left[\binom{2h}{h} + 2\sum_{k=0}^{h-1}\binom{2h}{k}\cos\left(2(h-k)\omega t\right)\right] \tag{3.183}$$

$$\cos^{2h+1}(\omega t) = \frac{1}{2^{2h}}\sum_{k=0}^{h}\binom{2h+1}{k}\cos\left((2(h-k)+1)\omega t\right) \tag{3.184}$$

$$\forall h \geq 0, \quad \text{where} \quad \binom{n}{k} = \frac{n!}{k!(n-k)!} \tag{3.185}$$

and because the sum of sinusoids at the same frequency is a sinusoid of this frequency, since

$$\sum_{k=1}^{K} a_k \exp(j(\phi_k + \omega_0 t)) = \left(\sum_{k=1}^{K} a_k \exp(j\phi_k)\right) \cdot \exp(\omega_0 t). \tag{3.186}$$

In general the resulting signal is not zero-centered. To avoid clipping issues, one can pre-compute and subtract the corresponding constant to each output sample. Spectral aliasing can also be a problem, if the g_p coefficients do not fade out enough rapidly as p grows (thus the function g has to be carefully chosen). Also, the coefficients of the output partials depend on the amplitude of the input sinusoid. A last problem is the inter-modulation phenomenon that occurs when the input signal x is not a pure tone anymore, but a complex sound: linear combinations of frequencies present in the input are found in the output. This last drawback can be turned into an advantage...

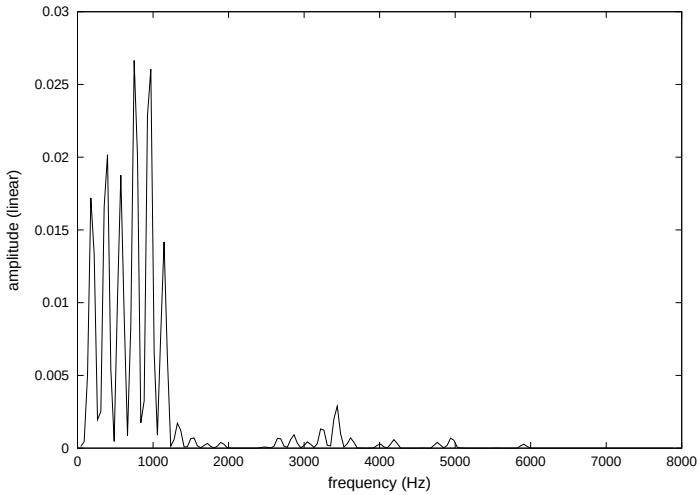
For example, Collen [Col02] uses the abs function ($x \rightarrow |x|$) because

$$|x| \approx 4.77x^2 - 18.8x^4 + 40.43x^6 - 40x^8 + 14.6x^{10} \quad (-1 \leq x \leq 1) \tag{3.187}$$

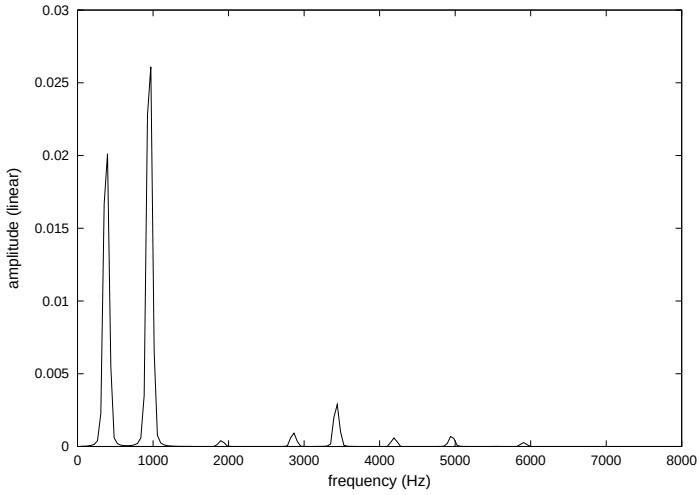
and when the abs function is applied to the first half (lower frequencies) of the spectrum, it produces a second half with a convincing (realistic) high-frequency content. This is part of the "spectral band replication" technique used in MPEG decoders. This technique gives convincing results (sounding close to the originals) because the spectral structure is conserved (*e.g.* harmonic series are continuated, stochastic signals are extended), and because our ears are not very sensitive to these high frequencies. Moreover, the abs function can be implemented very efficiently: with IEEE floating-point units, it is just a matter of setting one bit (the one corresponding to the sign).

Instead of a band-limited version of the original sound, we can also consider a simplified broad-band version, consisting of one sinusoid per formant of the original sound. More precisely, for each formant we keep the partial with the greatest amplitude (these partials roughly correspond to the local maxima in the sequence $\{a_k\}_k$, k being the rank of the harmonic). When the abs function is applied to the simplified signal (synthesized using some linear technique), then the resulting spectrum looks like if the right half of each formant had been partly reconstructed (see Figure 3.21).

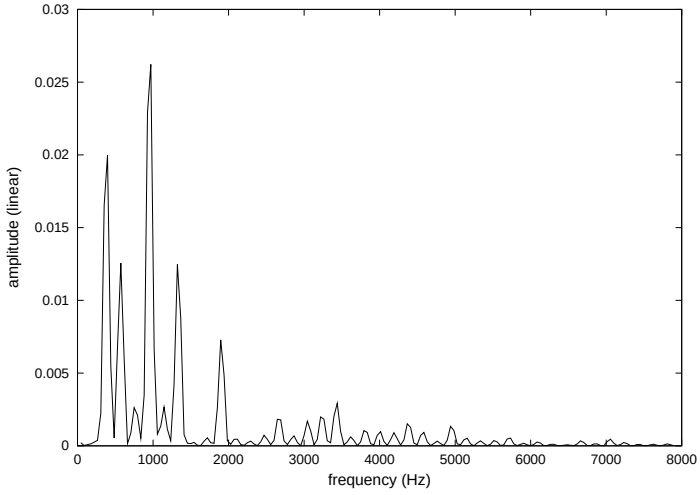
Indeed, when applied to a single sinusoid $x = a\cos(\omega t)$, the abs function generates many even multiples of this frequency (*i.e.* frequencies $\omega_k = 2k\omega$, $k > 0$), whose amplitudes a_k are almost always decreasing with k (at least when $a < 1$), as shown in Figure 3.22.



(a) original spectrum



(b) simplified spectrum (one partial per formant)



(c) synthesized spectrum

Figure 3.21: Magnitude spectra of the original (a), simplified (b), and abs-synthesized (c) versions of a male voice singing the French “a” vowel. The synthesized version has a more complex harmonic structure, although very different from the original.

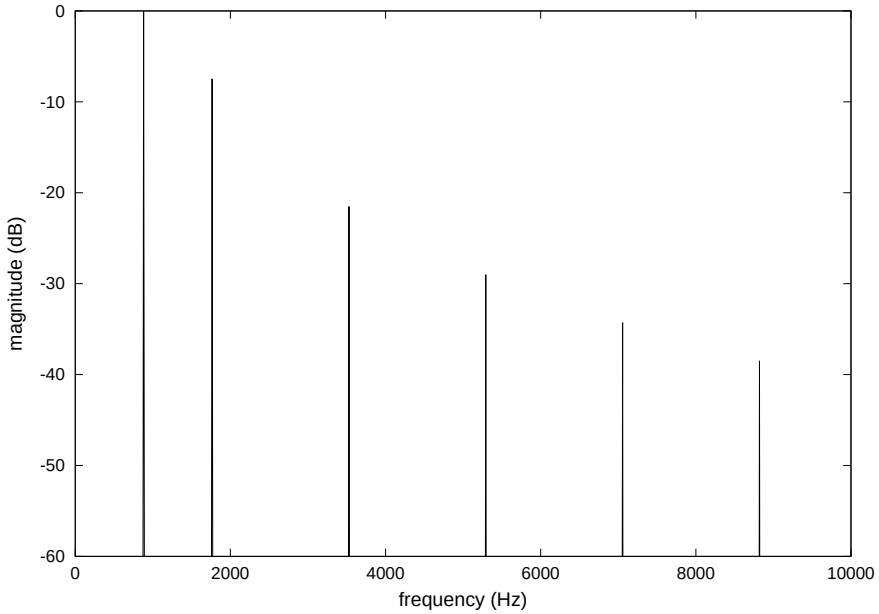


Figure 3.22: Magnitude spectrum of $x+|x|$ (zero-centered) for x being a sinusoid of amplitude 1 (0 dB) and frequency 884Hz ($20F_s/N$, with $F_s = 44100$Hz, FFT size $N = 1000$, using a rectangular window). The abs function has generated even multiples of the frequency, with decreasing magnitudes.

Amplitude Modulation (AM)

Another simple way to turn each input partial into an output formant of known shape is to use amplitude modulation (AM). P oscillators (of carrier frequencies ω_p) are multiplied by another oscillator (of modulation frequency ω_m) in order to generate a more complex timbre, consisting of $3P$ partials:

$$\underbrace{\left(\sum_{p=1}^{P} a_p \cos(\omega_p t)\right)}_{\#n} \underbrace{(1+\cos(\omega_m t))}_{\#1} = \underbrace{\left(\sum_{p=1}^{P} a_p \cos(\omega_p t)\right)}_{\#P} + \underbrace{\frac{1}{2}\left[\sum_{p=1}^{P} a_p \cos((\omega_p-\omega_m)t) + \sum_{p=1}^{P} a_p \cos((\omega_p+\omega_m)t)\right]}_{\#2P}. \tag{3.188}$$

This way, when each input partial corresponds to a formant (see above), each output formant consists of 3 partials of amplitudes $[a_p/2, a_p, a_p/2]$, thus with a symmetric and triangular shape in the linear scale. Figure 3.23 shows the magnitude spectrum of an amplitude-modulated sinusoid ($P = 1$ and $\omega_1 = \omega_c$ in Equation (3.188)).

3.3.4.2 Formant-Based Synthesis

In order to perform formant-based synthesis using non-linear techniques, one would better consider frequency modulation or discrete summation formulas.

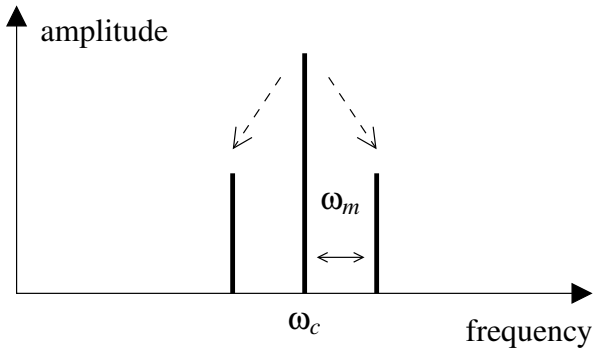


Figure 3.23: Magnitude spectrum of an amplitude-modulated sinusoid of carrier frequency ω_c by a modulation frequency ω_m. The carrier frequency generates two frequencies $\omega_c \pm \omega_m$, the corresponding amplitudes being the half of the amplitude of the original signal.

Frequency Modulation (FM)

Frequency modulation (FM) consists in using the output of one oscillator to modulate the frequency (or more precisely the phase) of another oscillator:

$$s(t) = a\sin(\omega_c t + I\sin(\omega_m t)) \tag{3.189}$$

where a is the amplitude, ω_c is the carrier frequency, ω_m is the modulation frequency, and I is called the modulation index. De Poli [Pol83] shows that Equation (3.189) is equivalent to

$$s(t) = a\sum_{n=-\infty}^{+\infty} J_n(I)\sin((\omega_c + n\omega_m)t) \tag{3.190}$$

where $J_n(I)$ it the Bessel function of first kind and order n evaluated at I. Thus the FM synthesis can be regarded as a spectral modeling technique. With only 2 oscillators, it generates a – potentially infinite – set of partials, centered on the carrier frequency ω_c and spaced by the modulation frequency ω_m. This way, we can model a formant, though the amplitudes of the corresponding partials are controlled by the Bessel functions. The Bessel functions of the first kind and order n are the solutions of the Bessel's differential equation

$$x^2y'' + xy' + (x^2 - n^2)y = 0, \quad n \geq 0 \tag{3.191}$$

and are given by, for n being a positive integer:

$$J_n(x) = \sum_{k=0}^{+\infty} \frac{(-1)^k (x/2)^{n+2k}}{k!(n+k)!} \tag{3.192}$$

and can also be obtained using the following scheme:

$$J_{n+1}(x) = \frac{2n}{x}J_n(x) - J_{n-1}(x) \quad (n > 0) \tag{3.193}$$

$$\text{with } \begin{cases} J_0(x) &= 1 - \frac{x^2}{2^2} + \frac{x^4}{2^2\cdot 4^2} - \frac{x^6}{2^2\cdot 4^2\cdot 6^2} + \cdots \\ J_1(x) &= \frac{x}{2} - \frac{x^3}{2^2\cdot 4} + \frac{x^5}{2^2\cdot 4^2\cdot 6} - \frac{x^7}{2^2\cdot 4^2\cdot 6^2\cdot 8} + \cdots \end{cases} \tag{3.194}$$

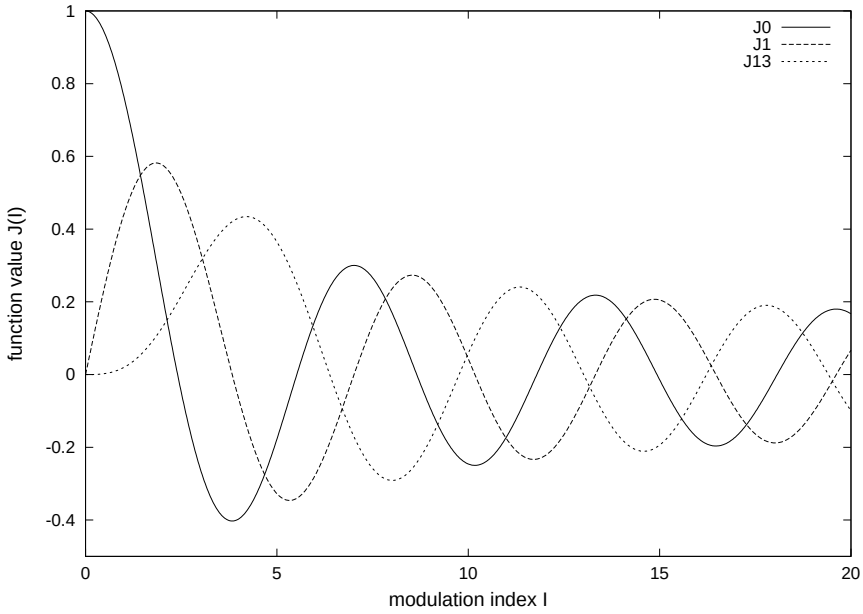


Figure 3.24: Examples of Bessel functions of the first kind.

As shown in Figure 3.24, the Bessel functions look like damped sinusoids. When $I = 0$, only the carrier frequency ω_c is present in the spectrum, with amplitude A (since $J_0(0) = 1$ and $\forall n > 0, J_n(0) = 0$). As I increases, the spectrum gets new partials situated in a symmetric manner each side of the carrier frequency ω_c, and spaced by the modulation frequency ω_m. Thus the single carrier frequency turns into a formant centered at ω_c, as shown in Figure 3.25. As shown by De Poli [Pol83], the number of generated partials is approximatively $2(I+1)+1$. However, as shown by Chowning [Cho73], as I gets larger the amplitude associated to the central frequency is not necessary the greatest one, and thus the formant can split in several formants. A solution to this problem has been proposed by Moorer [Moo77], involving modified Bessel functions.

Using a Discrete Summation Formula (DSF)

Discrete summation formulas [Moo76] can be interesting for formant-by-formant synthesis. For example

$$\begin{aligned}\sum_{p=0}^{P} a^p \cos((\omega_c + p\omega_m)t) &= \big(\cos(\omega_c t) - a\cos((\omega_c - \omega_m)t) + a^{P+2}\cos((\omega_c + P\omega_m)t) \\ &- a^{P+1}\cos((\omega_c + (P+1)\omega_m)t)\big) \big/ \big(1 - 2a\cos(\omega_m t) + a^2\big) \quad (3.195)\end{aligned}$$

generates $P+1$ partials using only 5 oscillators. Thus we can generate the right part of a formant centered at frequency ω_c and corresponding to an harmonic source of fundamental frequency ω_m. To generate the left part, we can use the opposite modulation frequency $-\omega_m$. However, the shape of the resulting formant will be a triangle in the dB scale (since $a \leq 1$, $\{a^p\}_p$ is a decreasing geometric sequence), as shown in Figure 3.26.

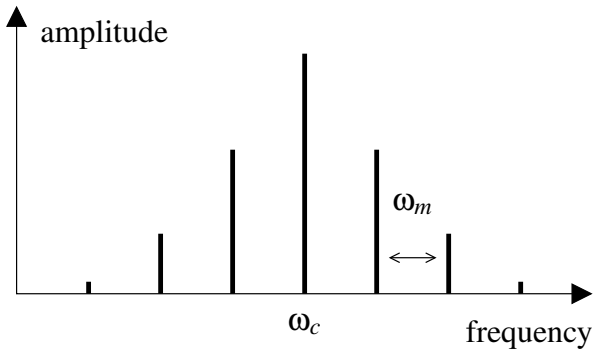


Figure 3.25: Magnitude spectrum of a frequency-modulated sinusoid of carrier frequency ω_c by a modulation frequency ω_m. Approximatively $2(I+1)+1$ partials are generated, were I is the modulation index (here $I=2$). The series of partials is centered on the carrier frequency ω_c and spaced by the modulation frequency ω_m. The amplitudes of the partials follow the Bessel functions.

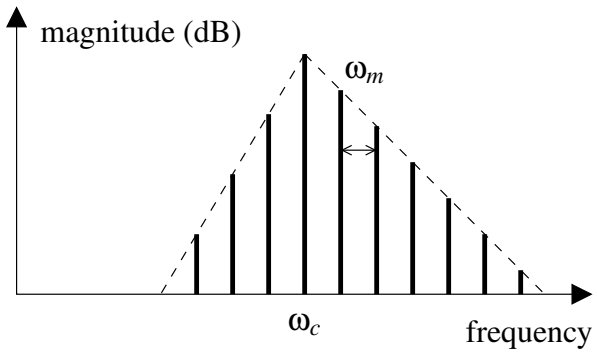


Figure 3.26: Magnitude spectrum of a formant generated using two DSF formulas (for the left and right parts). The formant is centered on the carrier frequency ω_c, and its shape is triangular in the dB scale. The left and right half-formants have been generated using the modulation frequencies $-\omega_m$ and $+\omega_m$, respectively.

3.3.4.3 Source / Filter Approach

With the spectral enrichment techniques (see above), some source signal is generated using a linear technique and then a non-linear technique adds new partials in the spectrum. Another – probably more interesting – approach is to use a non-linear technique to generate the source, and then a linear technique (filter) to shape its spectral envelope.

For example, to generate a band-limited harmonic comb we can use the simplified version of the discrete summation formula of Equation (3.195), with $a = 1$ and $\omega_c = 0$:

$$\sum_{p=1}^{P} \cos(p\omega t) = \frac{\cos((P+1)\frac{\omega}{2}t)\sin(P\frac{\omega}{2}t)}{\sin(\frac{\omega}{2}t)} \tag{3.196}$$

which generates a rich band-limited spectrum of P partials with only 3 oscillators.

For the filter, we can use the coefficients coming from linear prediction coding (LPC) techniques (see Section 4.2.3). This way, the exact spectral structure of the P partials can be respected using an order of $2P$, since two poles correspond to each real sinusoid. However, in order to respect the structure of F formants, an order $2F$ is sufficient. This will give a smooth approximation of the spectral envelope with a very small filter while reducing the complexity from proportional to the number of partials to proportional to the number of formants (much lower). With this approach, it would thus be possible to tune the trade-off of quality versus speed.

Chapter 4

Long-Term Sinusoidal Modeling

In this chapter, we still decompose sounds as sums of sinusoids, but the parameters of these sinusoids vary with time (*e.g.* from frame to frame). This is a long-term approach. For each partial, we consider the trajectories in time of its parameters (see Figure 2.2).

4.1 Model

Formally, the long-term sinusoidal (LTS) representation of a sound corresponds to a set of P partials

$$\mathcal{L} = \bigcup_{p=1}^{P} \{\mathcal{P}_p\} \tag{4.1}$$

where the partial number p, of length l_p, and that appeared (was born) at frame index b_p, is noted

$$\mathcal{P}_p = (\mathcal{P}_p[n])_{n\in[b_p,\cdots,b_p+l_p-1]} \tag{4.2}$$

where the peak of this partial at time index n is, in the stationary or non-stationary cases, respectively

$$\mathcal{P}_p[n] = (n, \phi_p[n], a_p[n], \omega_p[n]) \tag{4.3}$$

$$\text{or} \quad \mathcal{P}_p[n] = (n, \phi_p[n], \lambda_p[n], \omega_p[n], \mu_p[n], \psi_p[n])\,. \tag{4.4}$$

Following these notations and the ones of Chapter 3, a peak that does not belong to any partial is noted $\mathcal{P}_{i,n}$ and noted $\mathcal{P}_p[n]$ otherwise. For example, $\mathcal{P}_p[n] = \mathcal{P}_{i,n}$ means that the i-th peak of the frame n has been assigned to the partial number p. Establishing the correspondence between peaks and partials is the long-term analysis, also called "partial tracking".

4.2 Partial Tracking

4.2.1 Basic Partial-Tracking Algorithm

The first partial-tracking algorithm was introduced by McAulay and Quatieri [MQ86] in the field of the sinusoidal modeling of the voice. A similar algorithm, with some improvements, was proposed by Smith and Serra [Ser89] in the field of the sinusoidal modeling of musical sounds.

This algorithm follows a short-term analysis, where short-term spectra are extracted from the sound at different time frames. The spectral peaks are then tracked from frame to frame to form the

trajectories of the partials. In fact, the partial-tracking algorithm manages a set of partials and tries to extend their trajectories with peaks of the next frame.

This involves the scheduling of the partials, to know in which order they should look for a continuation in the next frame. In the original McAulay and Quatieri (MAQ) algorithm, the partials are processed by increasing frequency. However, since the partials of great amplitude are more perceptively important, it was proposed to process them first to avoid discontinuities leading to noisy "clicks" that can be heard at the synthesis stage.

Also, there is an underlying prediction mechanism: for each partial tracked until frame k, the algorithm predicts the trajectory of some parameter (typically the frequency) and a continuation is searched among the peaks of the next frame $k+1$ with parameters close to the prediction. The MAQ algorithm is based on the assumption that the partials composing the signal have stationary frequency evolutions (thus the frequency stays constant).

Finally, there is also a notion of distance between the last inserted peak and the peak selected for continuation (which should be "close" to each other). In the MAQ algorithm, the frequency difference is considered. A maximal frequency difference threshold Δ_ω between successive peaks of a partial is set, resulting in the condition

$$|\omega_p[k+1] - \omega_p[k]| < \Delta_\omega \tag{4.5}$$

where $\omega_p[k]$ is the frequency of the p-th partial at frame k.

Then, the algorithm is processed iteratively frame by frame and partial by partial. For each partial, a continuation is searched among the unlinked peaks of the next frame whose parameters are the closest to the predicted ones. If no continuation is possible, the partial is labeled "dead". After all the active partials have been processed, the unlinked peaks of the next frame give rise to new (born) partials.

For various reasons such as decreasing amplitude, strong modulations or STFT bin corruption, the STS representation may lack some peaks. To overcome this analysis drawback, it is proposed in [Ser89] to add a "zombie" state to the partials, so that if a partial cannot link to any peak in a frame, it can still look for its next peak in a limited number of frames. If a peak can be found eventually, the missing parameters of "zombie" peaks are then interpolated.

The number of samples necessary for the computation of the STFT is constrained by the spectral structure of the analyzed sound. Since the frequencies of the sinusoidal components of an harmonic monophonic sound are separated by the fundamental frequency, the size of the DFT can be adapted according to a pitch estimate as proposed in [MQ86]. The frequency resolution is then sufficient to separate harmonics and the time resolution is close to optimal. On the contrary, the frequency distribution of the sinusoidal components of polyphonic sounds is not known in advance. The size of the window should then be set arbitrarily. If the frequency resolution is too small, two sinusoidal components may lay in the same frequency bin and may be mis-detected. On the other hand, the loss of temporal resolution and short-term stationarity may lead to poor – averaged – estimates. In order to reduce interpretation problems due to the bad temporal resolution, the hop size can be set significantly shorter than the window length. This method is far from perfect though because even if the spectrum is estimated at a given rate, the number of samples used to estimate the spectrum is significantly larger. This leads to a temporal "smearing" of the representation as shown in Figure 4.1.

During our experiments, the STFT of a signal sampled at $F_s = 44100$Hz is done with a window length of $N = 2048$ samples since a shorter window length is not convenient for the analysis of polyphonic sounds and a larger one gives very poor estimates in case of modulations. The parameters of the underlying partials are evolving with time and the analysis of these evolutions will be a key fact for the enhancements proposed in this chapter. The hop size is set to $I = 512$ samples to provide a

sufficiently dense sampling of these modulations ($F_s/I \approx 86$Hz).

4.2.2 Parameters as Time Signals

In the basic algorithm for partial tracking, the parameters (frequency and amplitude) of the partials were considered as simple successions of numerical values. However, as shown by Equations (4.3) or (4.4), every parameter can be regarded as a (discrete) time signal, with specific properties.

4.2.3 Using Linear Prediction

Our first contribution to partial tracking with Lagrange (during his Ph.D. [Lag04], co-supervised by Rault) and Raspaud (during his Master) was to consider the parameters of the partials as deterministic and introduce the use of linear prediction to extrapolate their trajectories in the future frames from the observations in the past frames [24, 27, 7].

4.2.3.1 Deterministic Evolutions

Indeed, although the parameters of the partials are generally not stationary (*e.g.* see Figure 2.2), we propose to consider them as deterministic: their future evolutions can (often) be predicted from past observations. This seems justified physically, by looking at the trajectories of the partials. There is also a psycho-physiological reason: the fact that the neurons are silent most of the time at the cortical level when we listen to sounds is most probably an indication that the brain is performing some extrapolation (and then the information is transferred only when the observations do not match the predictions).

In the McAulay-Quatieri (MAQ) algorithm [MQ86], a constant predictor is implicitly used, meaning that the predicted frequency is the last observed one (long-term stationarity hypothesis):

$$\hat{\omega}_C[n+d] = \omega[n] \tag{4.6}$$

where d is the distance between the prediction and the last observation (done at time index n). Assuming that the best evolution for frequency is the constant one may not be harmful for tracking if the STS representation is of high clarity. However, in degraded STS representations as in Figure 4.1, a better prediction is crucial to identify the correct continuation of a partial among several noisy peaks.

An improved predictor is used in the hidden Markov model (HMM) algorithm [DGR93], by considering (a rough estimate of) the slope between the two last inserted peaks:

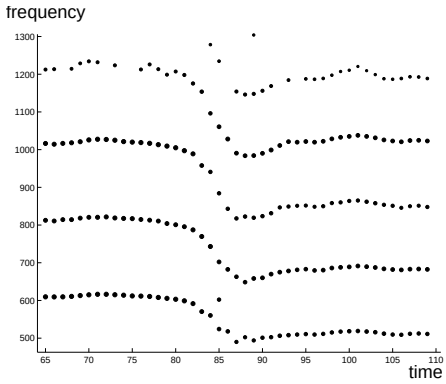
$$\hat{\omega}_L[n+d] = \omega[n] + d \cdot (\omega[n] - \omega[n-1]) \tag{4.7}$$

that is a linear predictor, or the reflection method presented in Section 1.1.3:

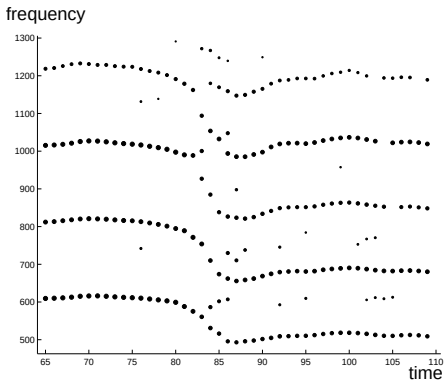
$$\hat{\omega}_R[n+d] = 2\,\omega[n] - \omega[n-d] \tag{4.8}$$

both methods being equivalent when $d = 1$. The problem with the HMM approach is that transition matrices have to be learned from data. Using pre-extracted statistical information, Sterian and Wakefield [SW98] propose to model the evolutions of the partials of instrumental sounds of the brass family by means of Kalman filtering.

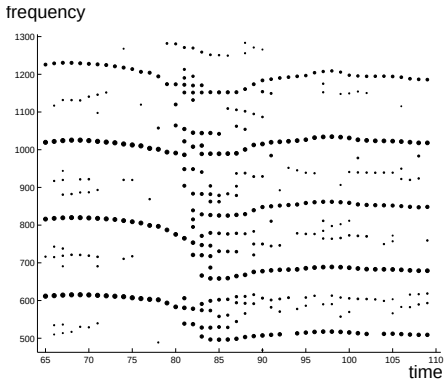
However, these statistical approaches highly depend on the database used. We prefer to gain generality, with a very limited number of parameters fixed from both practical and theoretic considerations.



(a) $N = 1024$



(b) $N = 2048$



(c) $N = 4096$

Figure 4.1: Spectral peaks (local maxima in the short-term magnitude spectra) of a singing voice sampled at $F_s = 44100$Hz, analyzed with a hop size of $I = 512$ and a window size of $N = 1024$ (a), 2048 (b), or 4096 (c). When the window size is too small, peaks are missing (because of bin contamination – too low frequency resolution). When the window size is too large, spurious peaks appear for modulated partials (especially around frame 85, because of insufficient time resolution – too high frequency resolution).

4.2.3.2 Linear Prediction

We propose to further improve the prediction capability by considering a more complex predictor suitable for the modeling of a wide variety of natural modulations. The evolution of partials in the time / frequency and time / amplitude planes can be constant, exponentially increasing or decreasing (portamento in the time / frequency plane and fade in / out in the time / amplitude plane) or sinusoidal (vibrato in the time / frequency plane and tremolo in the time / amplitude plane).

These evolutions can be modeled by an auto-regressive (AR) model. The linear prediction (LP) is then used to predict the evolutions of the parameters of partials in future frames. The current sample $x[n]$ is approximated by a linear combination of past samples of the input signal (FIR filtering)

$$\hat{x}[n] = \sum_{k=1}^{K} a[k]\,x[n-k]. \tag{4.9}$$

The challenge is to find an efficient method to estimate the vector $a[k]$ of coefficients, K being the order of the LP predictor, given N successive past samples considered as observations. This method should minimize the power of the estimation error $\hat{x} - x$.

Choice of the Method

The autocorrelation method (see [Mak92, KAZ00]) is very stable – maybe too much, see Figure 4.2 – but it minimizes the forward prediction error power on an infinite support. However, since in practice the signal is finite, samples of the $x[n]$ process that are not observed are then set to zero and observed samples are zero-centered and windowed in order to minimize the discontinuity at the boundaries. Thus, this method requires $N >> K$ to be effective. However, in practice only few samples are available since the parameters of the partials are sampled at a low rate.

On the contrary, the covariance method (see [Mak92]) assumes a finite support for the minimization of the forward prediction error power. Since no zeroing of the data is necessary, this method is a good candidate for coefficients estimation of process having few observed samples. Unfortunately this method can lead to unstable filters (the estimated poles are not guaranteed to lie within the unit circle, *i.e.* not all $a[k] < 1$), and thus should be avoided for data extrapolation (see Figure 4.2).

The Burg method (see [Bur75, KAZ00]) only requires $N > 2K$ and combines advantages of both previous methods (see Figure 4.2). As the autocorrelation method, it produces stable filters ($\forall k, a[k] < 1$), because the expression of r_k is of the form $r_k = 2ab/(|a|^2 + |b|^2)$, where a and b are vectors (see below), and using the Schwarz inequality it is verified that r_k has magnitude lower than one (and the estimated poles are then guaranteed to lie within the unit circle). Moreover, as the covariance method, the Burg method estimates the $a[k]$ on a finite support.

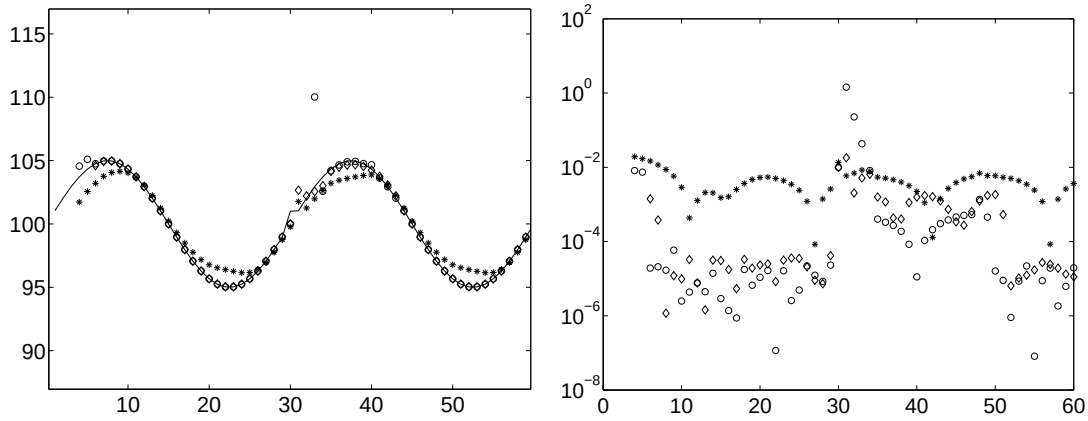
Let $e_k^f[n]$ and $e_k^b[n]$ respectively denote the forward and backward prediction errors for a given order k:

$$e_k^f[n] = x[n] + \sum_{i=1}^{k} a_k[i]\,x[n-i], \tag{4.10}$$

$$e_k^b[n] = x[n-k] + \sum_{i=1}^{k} a_k[i]\,x[n-k+i]. \tag{4.11}$$

The Burg method minimizes the average of the forward and backward errors power on a finite support in a recursive manner. More precisely, to obtain a_k we minimize

$$\varepsilon_k = \frac{1}{2}(\varepsilon_k^f + \varepsilon_k^p) \tag{4.12}$$



(a) predicted trajectories (b) prediction errors

Figure 4.2: On the left, the extrapolation at different times of one value in the future with a maximum of 12 past samples and a order-4 predictor using three linear prediction methods: correlation ($*$), covariance ($\circ$), and Burg ($\diamond$). The original (solid line) is a periodic signal with one sample – in the middle – displaced by a multiplication by a factor 1.01 (for clarity sake, some diverging covariance samples are not plotted). On the right, the evolutions of the associated prediction errors are displayed.

with

$$\varepsilon_k^f = \frac{1}{N-k}\sum_{n=k}^{N-1}|e_k^f[n]|^2, \tag{4.13}$$

$$\varepsilon_k^b = \frac{1}{N-k}\sum_{n=0}^{N-1-k}|e_k^b[n]|^2 \tag{4.14}$$

(where N is the number of samples used for the prediction) and

$$a_k[i] = \begin{cases} a_{k-1}[i] + r_k\, a_{k-1}[k-i] & \text{for } i = (1,\cdots,k-1) \\ r_k & \text{for } i = k \end{cases} \tag{4.15}$$

where r_k is called the reflection coefficient. By substituting Equation (4.15) in Equations (4.13) and (4.14), we find a recursive expression for the forward and backward errors:

$$e_k^f[n] = e_{k-1}^f[n] + r_k\, e_{k-1}^b[n-1], \tag{4.16}$$

$$e_k^b[n] = e_{k-1}^b[n-1] + r_k\, e_{k-1}^f[n] \tag{4.17}$$

where

$$e_0^f[n] = e_0^b[n] = x[n]. \tag{4.18}$$

To find r_k, we differentiate the k-th prediction error power with respect to r_k and by setting the derivative to zero, we obtain

$$r_k = \frac{-2\sum_{n=k}^{N-1} e_{k-1}^f[n]\, e_{k-1}^b[n-1]}{\sum_{n=k}^{N-1}|e_{k-1}^f[n]|^2 + |e_{k-1}^b[n-1]|^2}. \tag{4.19}$$

Finally, the following algorithm computes the vector a of linear prediction coefficients using the Burg method, at the order K:

$e^f \leftarrow (x[n+1-N], \cdots, x[n])$
$e^b \leftarrow (x[n+1-N], \cdots, x[n])$
$a \leftarrow 1$
for k from 1 to K do
  $\tilde{e}^f \leftarrow (e^f[1], \cdots, e^f[N])$
  $\tilde{e}^b \leftarrow (e^b[0], \cdots, e^b[N-1])$
  $r_k \leftarrow -2\tilde{e}^b \cdot \tilde{e}^f / (\tilde{e}^b \cdot \tilde{e}^b + \tilde{e}^f \cdot \tilde{e}^f)$
  $e^f \leftarrow \tilde{e}^f + r_k \tilde{e}^b$
  $e^b \leftarrow \tilde{e}^b + r_k \tilde{e}^f$
  $a \leftarrow (a[0], a[1], \cdots, a[k-1], 0) + r_k (0, a[k-1], a[k-2], \cdots, a[0])$
end for.

Tuning of the Method

The performance of the LP predictor depends on the choice of a relevant number of observations N and model order K. We choose parameter range values by both theoretic and experimental considerations. In the latter case, experimental tests were processed on the known evolutions of frequencies of already-tracked partials of different kinds, of pseudo-stationary monophonic signals such as a saxophone, a guitar (all from the Iowa database [iow]), and different singing voices. We considered the mean prediction error and the maximal error.

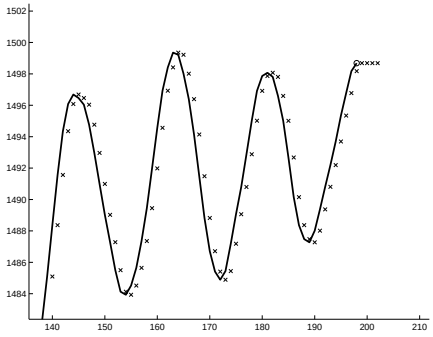
The number of observations should be large enough to extract the signal periodicity, and short enough not to be too constrained by the past evolution. Our short-term analysis module uses a STFT with a hop size of $I = 512$ samples on sound signals sampled at CD quality ($F_s = 44100$Hz). This means that the frequency and amplitude trajectories are sampled at ≈ 86Hz. Since we want to handle natural vibrato with frequencies down to ≈ 4Hz, we need at least 20 samples to get the period of the modulation. Practical experiments (such as the one illustrated in Figure 4.3) show that for most cases a number of samples in the $[4, 32]$ range is convenient (see Table 4.1, giving a summary of the results of these experiments for an alto saxophone), and confirm that a maximum of $N = 20$ observations – if available – is a good choice. Otherwise, all observations and a model order of half the number of observations are considered.

For frequency evolutions, since we want to model exponentially increasing or decreasing evolutions (portamento) and sinusoidal evolutions (vibrato), the model order should not be below 2 (2 poles being required to model a real sinusoid). In practice, the order should be set at a higher value because observations suffer from imprecision of the estimation of the spectral parameters as shown by the experimental results summarized in Table 4.1. Practical experiments show that a model order in the $[2, 8]$ range is convenient. When possible, we use the highest value $K = 8$.

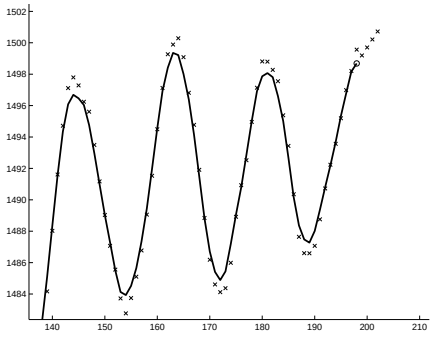
Using these values ($N = 20$ and $K = 8$), the LP predictor reduces the mean error by a factor of 2 and the maximal error by a factor of 1.3 over the constant predictor.

4.2.3.3 Application: Sound Restoration

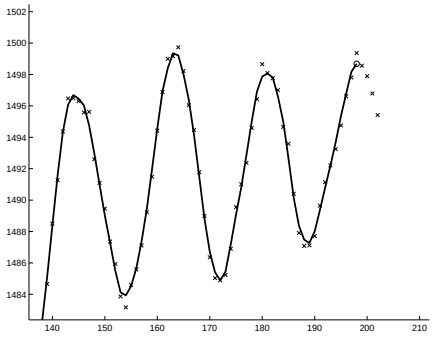
Because of erroneous network transmission (*e.g.* lost packets during streaming) or damaged data storage (*e.g.* scratches on a CD), portions of samples can be missing. Sound restoration consists in recovering (an approximation of) the missing audio data during these gaps.



(a) constant predictor



(b) linear predictor



(c) LP predictor

Figure 4.3: Evolutions of different predictors: constant (a), linear (b), LP (c) for one partial of an alto saxophone with vibrato. The order of the LP predictor is $K = 6$ and the number of past observations used to compute the coefficients is $N = 20$. The frequency evolutions of the partials are drawn with lines and the end of each partial is represented by a circle. To show the ability of the predictors to extrapolate at a longer term (useful in case of missing peaks), the predicted values are plotted even after the death of the partial.

d	constant	linear	reflection	LP order K: 2	4	6	8
	0.35 (0.9)	0.16 (0.8)	0.16 (0.8)	0.20 (0.8)	– (–)	– (–)	– (–)
1	– (–)	– (–)	– (–)	0.17 (0.6)	0.17 (0.6)	0.18 (0.7)	– (–)
	– (–)	– (–)	– (–)	0.16 (0.6)	0.14 (0.6)	0.14 (0.6)	0.14 (0.6)
	– (–)	– (–)	– (–)	0.16 (0.7)	0.13 (0.6)	0.13 (0.7)	0.12 (0.6)
	0.69 (1.8)	0.42 (1.4)	0.51 (1.6)	0.48 (1.4)	– (–)	– (–)	– (–)
2	– (–)	– (–)	– (–)	0.43 (1.3)	0.42 (1.3)	0.45 (1.6)	– (–)
	– (–)	– (–)	– (–)	0.41 (1.3)	0.35 (1.3)	0.34 (1.4)	0.34 (1.3)
	– (–)	– (–)	– (–)	0.41 (1.3)	0.32 (1.2)	0.29 (1.1)	0.28 (1.1)
	1.01 (2.5)	0.76 (2.4)	1.00 (3.1)	0.85 (2.3)	– (–)	– (–)	– (–)
3	– (–)	– (–)	– (–)	0.75 (2.3)	0.77 (2.3)	0.80 (2.7)	– (–)
	– (–)	– (–)	– (–)	0.72 (2.2)	0.62 (2.0)	0.57 (2.2)	0.57 (2.1)
	– (–)	– (–)	– (–)	0.71 (2.1)	0.52 (1.7)	0.46 (1.7)	0.43 (1.4)
	1.31 (3.1)	1.18 (3.7)	1.63 (4.6)	1.29 (3.5)	– (–)	– (–)	– (–)
4	– (–)	– (–)	– (–)	1.13 (3.5)	1.18 (3.6)	1.20 (4.1)	– (–)
	– (–)	– (–)	– (–)	1.09 (3.5)	0.92 (3.3)	0.83 (3.1)	0.81 (3.0)
	– (–)	– (–)	– (–)	1.06 (3.3)	0.77 (2.5)	0.65 (2.3)	0.58 (1.9)

Table 4.1: Mean and (maximal) prediction errors in Hz of different predictors on the frequency evolution of the partials of an alto saxophone with vibrato. The mean error is obtained by considering the whole frequency trajectory of all the partials of the saxophone tone. The error obtained for each partial is normalized by the harmonic rank of the considered partial prior to summation. The maximal error is also considered because it indicates the robustness of the predictor. The prediction errors are computed for several simple predictors (left part) and the LP predictor (right part) for different values of d (the distance in frame indices between the last observation and the predicted value). Concerning the part dedicated to the LP predictor, the model order K grows from left to right, and for each values of d the number of samples considered N is in $[4, 8, 16, 32]$ (from top to bottom). The prediction errors of the LP predictor are lower than those of the best simple predictor, and the improvement is getting more and more significant when d is increasing.

For the practical experiments described here, we created artificial gaps of several durations on audio signals, and performed the restoration using different methods. Figure 4.4 shows the result of the restoration of a trombone sound with a method directly based on our LP predictor. Figure 4.5 shows the results of the subjective evaluation – using listening tests – of all the methods for a violin tone with vibrato, a piano tone, an orchestra piece, a gong tone, and the recording of two female soprano voices. The gap could be at a sustained or at a transitional segment of the sound.

As shown by Kauppinen *et al.* [KKS01], linear prediction can be successfully used in the temporal domain to reconstruct up to thousands of samples at CD quality without audible distortion. This is done at the expense of a large number of observed samples ($N \geq 2000$) and a large AR model order ($K \geq 1000$). The interpolation quality of longer gaps depends on the characteristics of the signal. An attenuation problem occurs especially when the parameters of the partials are modulated (*e.g.* with a vibrato), which explains the marks ranging from 30 to 50 in Figure 4.5. Sinusoidal modeling can be used to cope with this attenuation problem.

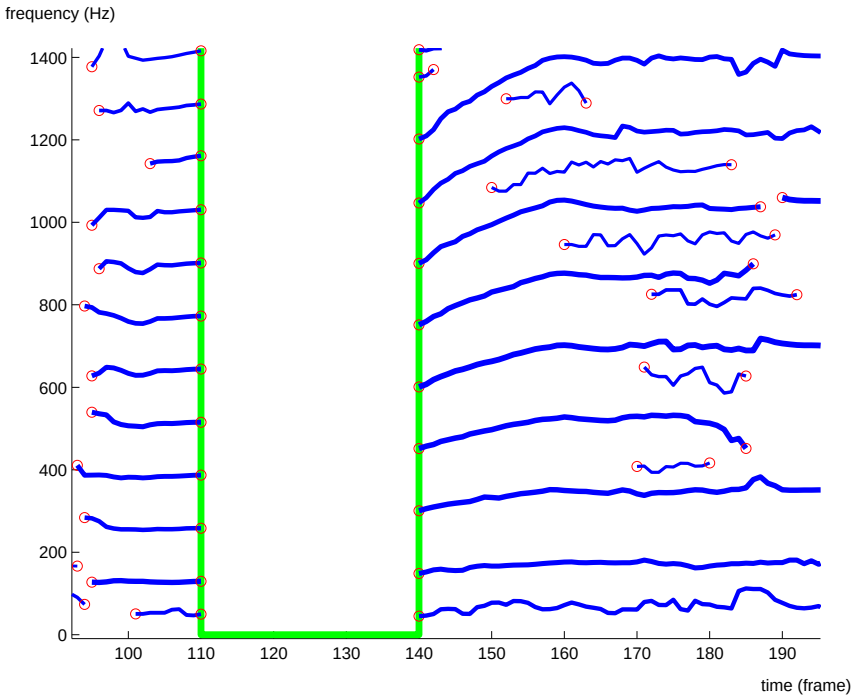
Thus, another approach is to use the polynomial interpolation scheme proposed by McAulay and Quatieri [MQ86] (see Section 4.3), with larger time interval between frames. The missing phases and frequencies are computed using the maximally-smooth cubic phase polynomial while the amplitude is linearly interpolated. The sinusoidal interpolation scheme based on polynomial interpolation outperforms the temporal method for gap sizes up to 320ms, see Figure 4.5. In counterpart, all kinds of modulations disappear. This effect is perceived by the listeners as a "freezing" of the sound during the interpolated region. For larger gaps, the linear interpolation sounds artificial, and is badly marked by the listeners. The mark can be even worse than the one obtained by the temporal method. This case appears for the interpolation of a 820ms gap of the violin tone, see Figure 4.5.

Together with Lagrange and Rault [4], we show the efficiency of our LP predictor (see above), that is linear prediction of the spectral parameters, even for larger gaps. Our method keeps the benefit from the two previous methods. The use of sinusoidal modeling avoids the problem of attenuation so that long gap interpolation can be achieved. Additionally, the AR modeling of the parameters of the partials is useful to preserve the modulations important to perception. The gong tone and the two soprano voices have partials with small range modulations. The violin tone with vibrato has a larger range of frequency modulation. For all these sounds, the marks go from 90 to 70 in a regular decay for gap sizes from 320 to 820ms. The soprano voices can even be interpolated during 1.6 second with a *good* mark. The partials extracted from the orchestra piece have complex modulations because they represent harmonics of several notes and noise. The prediction capability is then lower than in the previous cases, but a *fair* quality can be achieved for gap sizes up to 450ms.

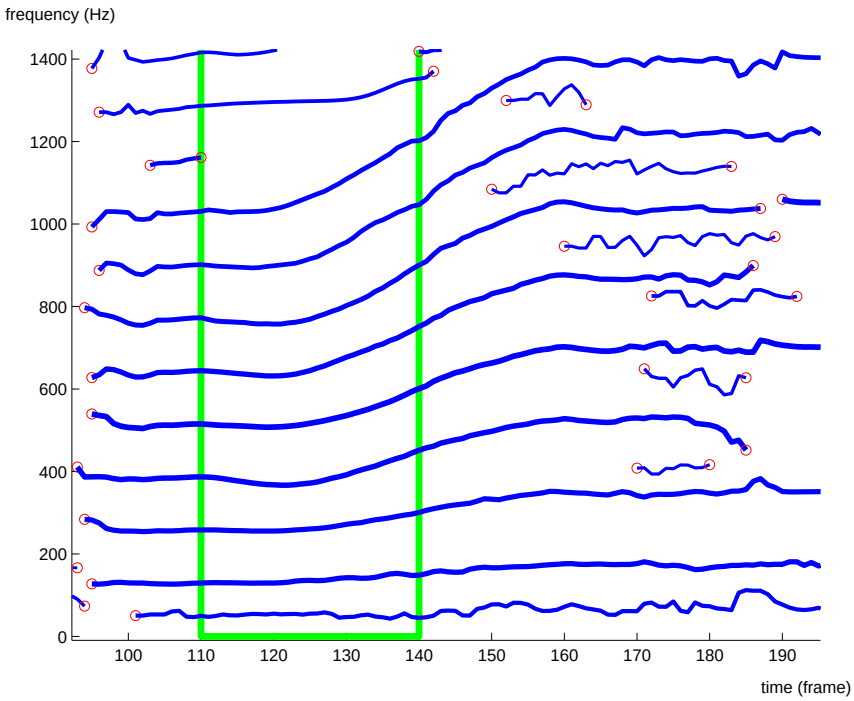
The implementation of the method and the listening tests have been done at the France Télécom R&D company during Lagrange's Ph.D. The tuning of the LP predictor is roughly the same as proposed before (though slightly adapted to a different hop size $I = 360$ used by the sinusoidal coder of the company). However, in order to obtain these results with our method, two practical improvements were proposed by Lagrange and Rault and patented by France Télécom (PCT/FR 04/00619, 2004/20/01 and PCT/FR 04/01415, 2004/06/08). For the details, see the patents and [4].

The first improvement is to find an efficient criterion for the matching of the partials (one partial p_1 ending at the beginning of the gap should be connected to one partial p_2 starting at the end of the gap). The Euclidean distance is used between the predictions associated to p_1 and p_2, normalized by the length of the gap, and divided by the sum of the standard deviations of the two predictions. Two partials match if both their frequency and amplitude predictions fall under some thresholds.

The second improvement is to reconstruct the parameters within the gap as a combination of the forward and backward predictions, using an asymmetric window (derived from the Hann window) in order to favor the prediction done with the largest data set (see Figure 4.6). Also note that, before the



(a) gap with missing data



(b) data reconstructed with the LP predictor

Figure 4.4: Reconstruction of the frequency trajectories of the partials of a trombone sound: (a) the original sound with an artificial gap, (b) the reconstructed sound using the LP predictor.

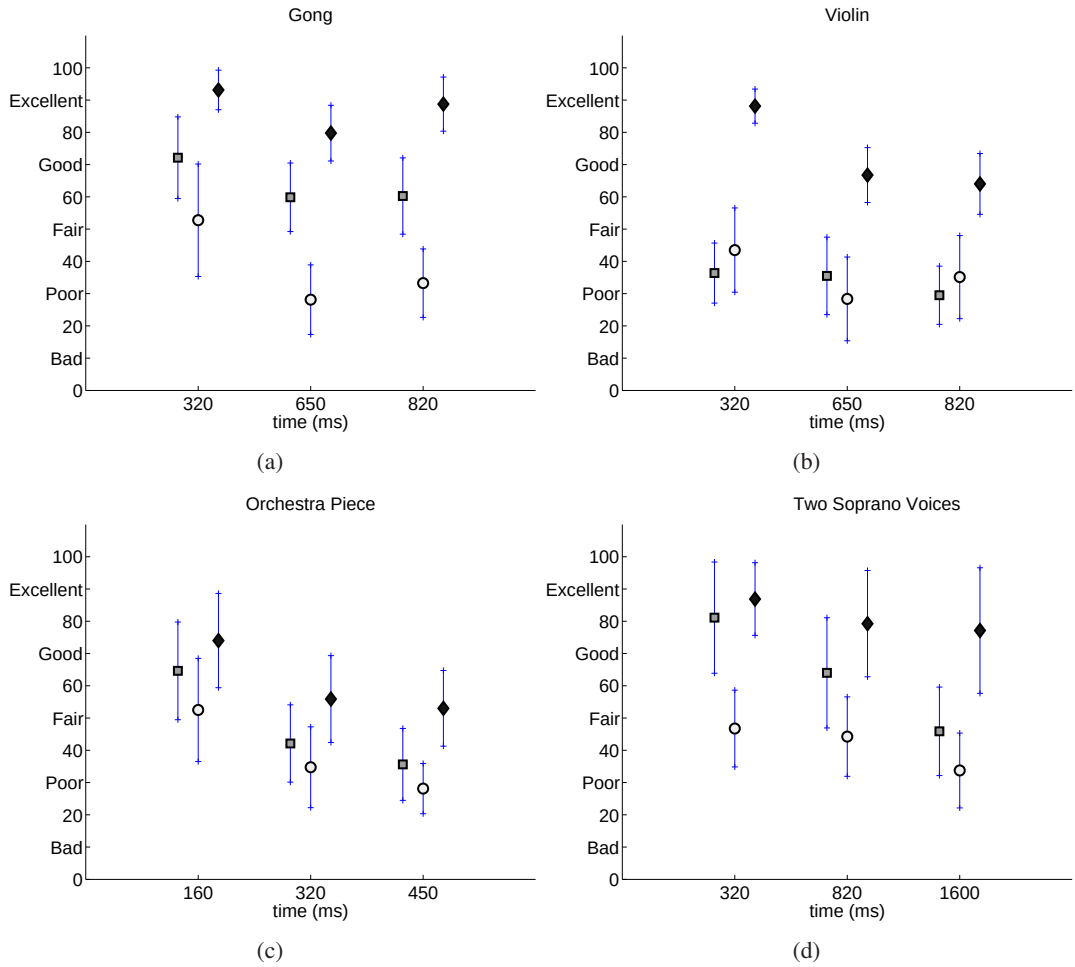


Figure 4.5: Results of the listening tests comparing the temporal method (○), the polynomial method (□), and our spectral LP method (◊), for three gap sizes. Symbols are the mean of votes and lines are confidence intervals for each method.

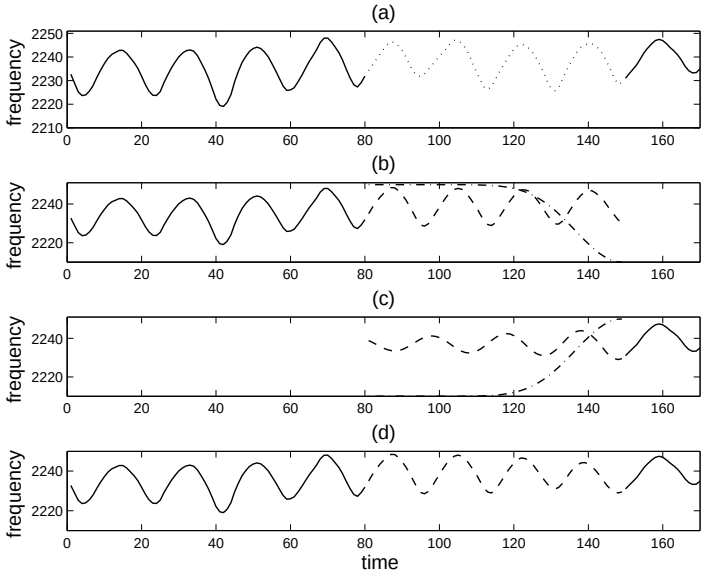


Figure 4.6: Interpolating the frequencies of a partial of an alto saxophone with vibrato using linear prediction (LP): (a) some frequencies (dots) are unavailable; (b) the left part is forward predicted and (c) the right part is backward predicted with the LP formalism (dashed lines); finally, the two predictions are cross-faded (d) using an asymmetric window (dashed dot lines) favoring the prediction computed with more samples (the left part in this case).

cross-fade, the predicted amplitudes have to be adjusted (linearly over the prediction time interval) so that the mean amplitude of the predicted parts and known parts match at the boundaries.

4.2.4 High-Frequency Content (HFC)

We have shown that the continuation of modulated partials in the next frames can then be identified more precisely using the LP predictor by selecting peaks with parameters close to the predicted ones. The next problem to address is to determine which of the trajectories that go through these peaks will give the best LTS representation.

As we proposed in previous work [27], one can consider the peak whose frequency is the closest to the predicted one and effectively prolongate the partial with this peak if the absolute difference between its frequency and the predicted one is below a given threshold. This simple smoothness criterion was not found satisfactory because the higher frequency partials are more modulated than the lowest ones. The threshold should therefore be adaptive which cannot be safely done without any assumption about relationships between partials of the same source such as harmonicity.

Our second contribution to partial tracking with Lagrange (again during his Ph.D. [Lag04], co-supervised by Rault) was to consider the slow time-varying parameters of the partials as (control) signals that are band-limited in frequency, and to derive a quality measure for the trajectories.

4.2.4.1 Slow Time-Varying Evolutions

The definition of the LTS model given in Section 4.1 states that the frequency and the amplitude of a partial must evolve slowly with time. From a perceptual point of view, we can consider that these parameters evolve slowly with time if they do not show noticeable energy level in frequency

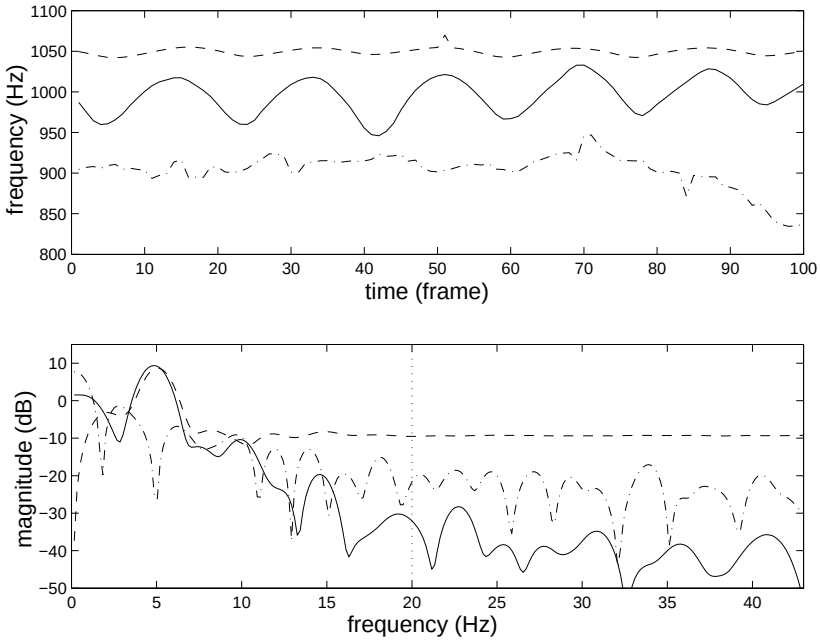


Figure 4.7: Three evolutions of the frequency of partials tracked with the MAQ algorithm and their corresponding magnitude spectra computed with a Hann-windowed DFT. From top to bottom, an harmonic of a saxophone tone with a local burst (tracking error) around frame 50, a well-tracked harmonic with vibrato, and a partial tracked by error from a white noise signal. Only the well-tracked partial has a low HFC.

bands upper than 20Hz. Otherwise, the fast parameters variations turn into frequency or amplitude modulations and question the perceptive coherence.

We then propose to study spectral properties of possible continuations of partials to detect if they satisfy the constraints of the LTS model. In a first attempt, slow time-varying evolutions can be discriminated from the others by considering the power of a Hann-windowed DFT spectrum of the evolutions of the frequency of the partials. As shown in Figure 4.7, only the well-tracked partials have a high-frequency content (HFC) around -30dB. Noisy evolutions, local burst, or change of harmonic rank induce a higher HFC.

Such a spectral analysis can only be used at a post processing stage because the number of samples required to compute the DFT with a sufficient frequency resolution is too consequent. Furthermore, the removal of wrong partials after the tracking process may lead to an incomplete sinusoidal representation because the partials with a local discontinuity will be removed erroneously.

4.2.4.2 HFC Metric

Thus, the HFC estimation must be integrated within the tracking process itself to determine whether a given continuation is relevant or not. The HFC estimation method should then be as responsive as possible. We use low-delay elliptic IIR high-pass filters to estimate the HFC. The high-pass filtered version of the frequencies of partial $\tilde{\omega}$ is

$$\tilde{\omega}[n] = \sum_{l=0}^{L} d[l]\,x[l] - \sum_{l=1}^{L} c[l]\,y[l] \tag{4.20}$$

where $x[l]$ and $y[l]$ are the memories of the filter and $c[l]$ and $d[l]$ are the coefficients respectively corresponding to the poles and the zeros of the IIR filter. These coefficients are mainly determined by the desired cutting frequency and the order of the filter which depend on the frame rate. For frequency and amplitude parameters sampled at ≈ 86 Hz, order-4 filters with a normalized cutting frequency of 0.25 are convenient. Thus, the following coefficients are used in the experiments ($L = 4$):

$$c[0,\cdots,L] = (1, 0.7358, 1.0762, 0.5540, 0.2346), \tag{4.21}$$

$$d[0,\cdots,L] = (0.06, -0.2274, 0.335, -0.2274, 0.06). \tag{4.22}$$

At the beginning of a partial, two filters are dedicated to the estimation of the HFC in the evolutions of frequency and amplitude respectively. The memories of these filters $x[l]$ and $y[l]$ are first set to 0 and updated as follows (here in the case of the frequency ω):

$$x[l] = \omega_m[n-l] - \omega_m[n_b], \tag{4.23}$$

$$y[l] = \tilde{\omega}_m[n-l] \tag{4.24}$$

each time a peak $\mathcal{P}_{i,n}$ is inserted. As can be seen on Figure 4.8, the output of the proposed high-pass filter is quite responsive. The insertion of a peak with parameters inducing noticeable HFC in the evolutions of the parameters can be detected very rapidly, with a response delay around 2 to 3 samples.

The problem to address now is the definition of a metric that considers the HFC both in the frequency and the amplitude evolutions to determine the best continuation. Considering that a partial is correctly tracked, its frequency and amplitude are slow time-varying, so as the trajectory composed of predicted peaks plotted with stars in Figure 4.9(a). Moreover, the frequency or the amplitude of a trajectory made up with peaks of STS frames will have more HFC than the predicted trajectory mostly made of extrapolated – thus virtual – peaks.

A small HFC difference between these two types of trajectories may be due to some measurement imprecision of the STS representation or a smooth change of dynamic. In this case, the non-extrapolated trajectory should be used for continuation. On the contrary, a larger HFC difference indicates that the non-extrapolated trajectory contains spurious peaks or peaks of another partial and should therefore be avoided.

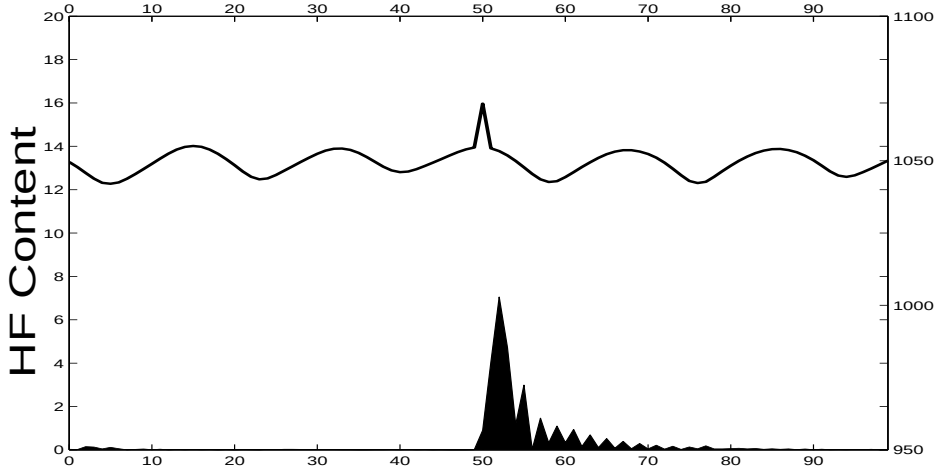
The chosen trajectory should then contain the highest number of peaks of STS frames possible while maintaining a small HFC both in frequency and amplitude. A cost function is associated to each trajectory that considers the HFC both in frequency and amplitude and is divided by a factor $\Gamma \in]0,1]$ each time an extrapolated peak is used:

$$\Pi_T = \left(\frac{1}{\Gamma}\right)^{N_T} \cdot \frac{\sum_{l=1}^{L_T} |\tilde{a}_T[l]|^2}{K_a} \cdot \frac{\sum_{l=1}^{L_T} |\tilde{f}_T[l]|^2}{K_f} \tag{4.25}$$

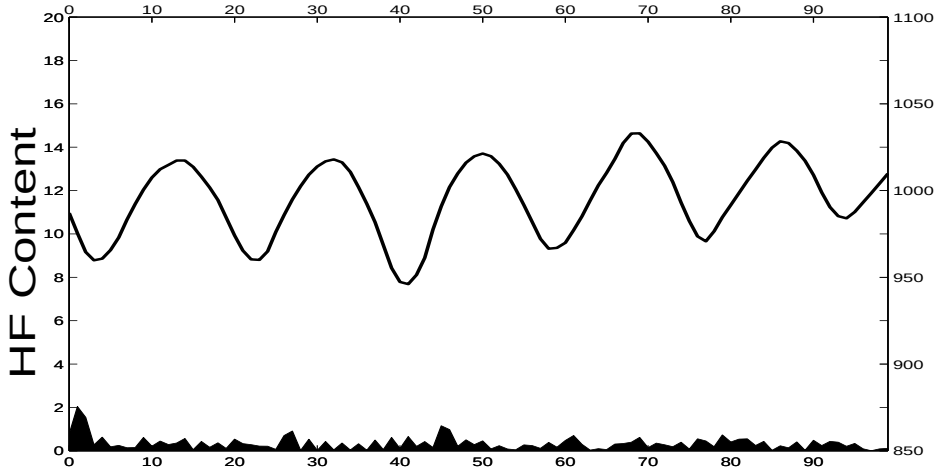
where $\tilde{a}_T[l]$ and $\tilde{f}_T(l)$ are, respectively, the high-frequency filtered amplitude and frequency of the l-th peak of trajectory T of length L_T. This filtering is done using memories of the filters associated to the current partial. N_T is the number of extrapolated peaks in the trajectory, K_a and K_f are normalizing constants. The choice of the best trajectory leads to constraints on the relative order between costs and not on their absolute values, so that K_a and K_f can be safely set to 1. In our experiments, we set $\Gamma = 0.9$ and $L_T = 6$.

4.2.4.3 Application: Note Onset Detection

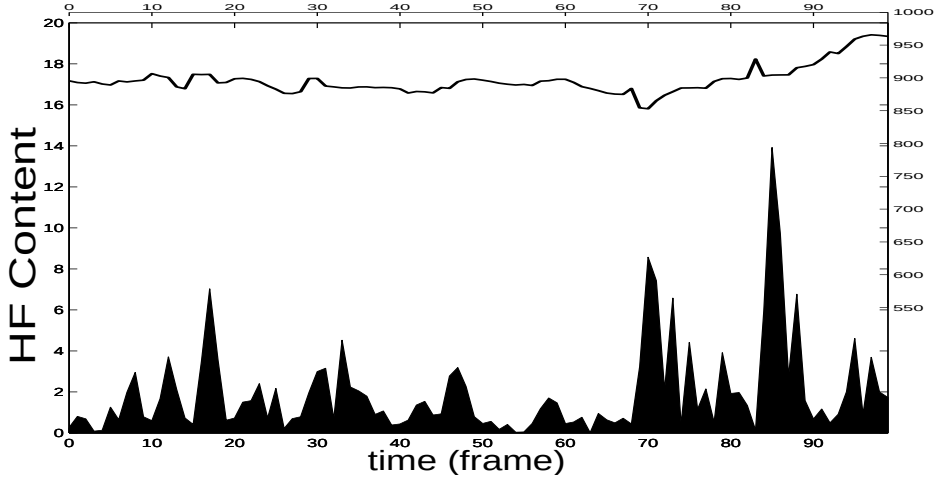
An interesting property of the HFC partial-tracking algorithm is that it betters identifies the note onsets / offsets (see Figure 4.10). This property will be exploited in Section 8.1.



(a) sinusoid with local error

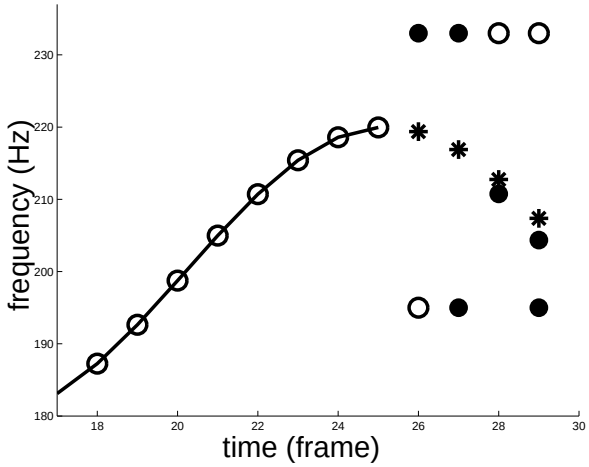


(b) saxophone with vibrato

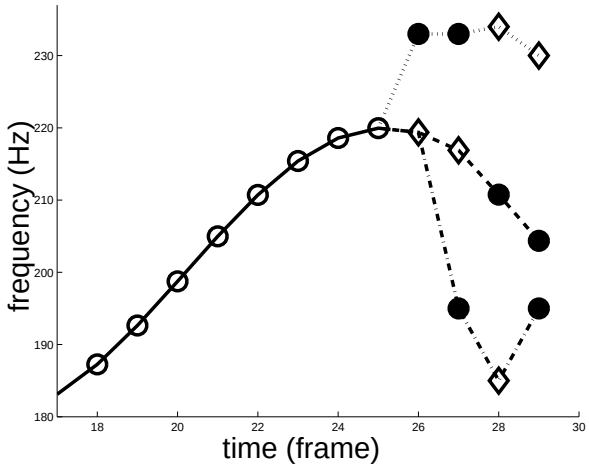


(c) noise (tracked by error)

Figure 4.8: Output of the high-pass filter (plain) given three different evolutions of the frequency parameter of the partials (line). The discontinuities are noticeable with the output of the high-pass filter with a small delay.

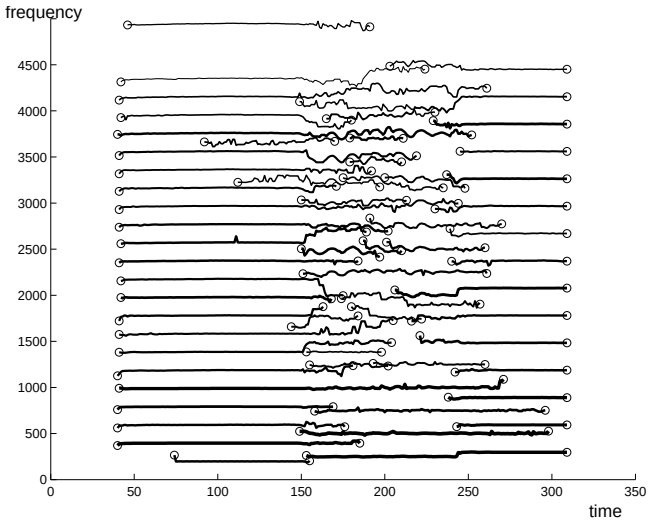


(a)

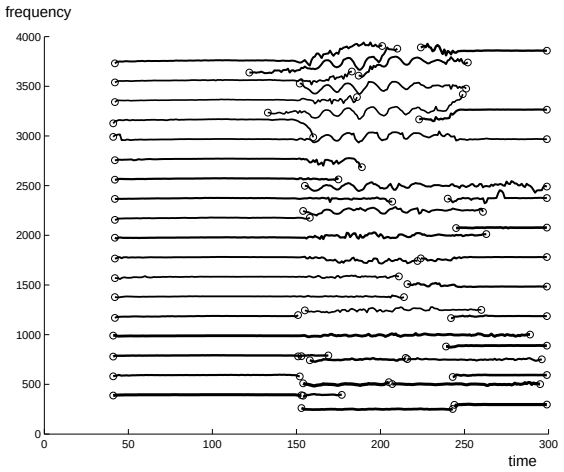


(b)

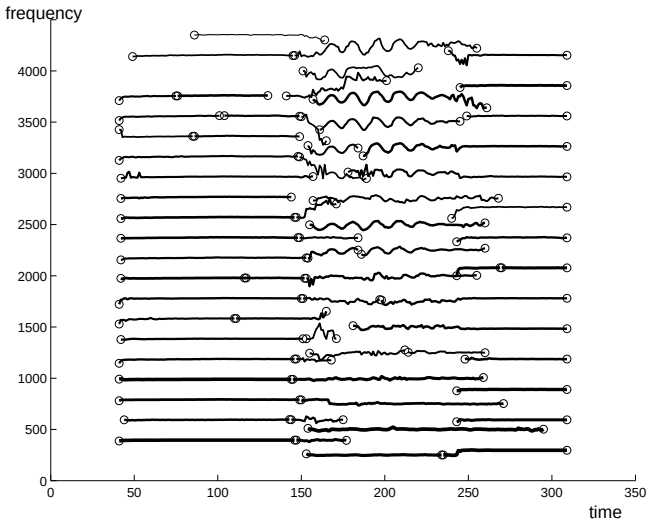
Figure 4.9: Selecting candidate peaks in the future frames and exploring possible trajectories. On top (a), the predicted frequencies are plotted with stars. Some STS peaks are chosen so that the frequency difference between the frequency of the peak and the predicted one is below Δ_ω. At bottom (b), possible trajectories that go through these selected measured peaks ($\circ$) and extrapolated ones ($\diamond$) are tested. The first peak of the chosen trajectory is added to the partial.



(a) MAQ



(b) LP



(c) HFC

Figure 4.10: Partials extracted from three successive violin tones by the MAQ (a), LP (b), and HFC (c) algorithms. The partials are represented by solid lines, starting and ending with circles matching the birth and the death of the partials.

4.2.5 Evaluation of Partial-Tracking Algorithms

Although other partial-tracking algorithms have been proposed, to our knowledge, no methodology has yet been proposed in the literature to evaluate or compare the capability of these algorithms to extract the structure of the pseudo-periodic part of the analyzed sound.

4.2.5.1 Perceptual Criteria

Representation Readability. In applications such as indexing or source separation of stationary pseudo-periodic sounds, a good partial representation should provide a higher level of description, useful to detect robustly high-level information such as note onset / offset, pitch detection, and source identification. In order to easily detect the note onset / offset, one would like to have a good precision, meaning that a partial should belong to only one source (see below). And in order to detect the pitch and to identify the sources, the partials should show clear time / frequency and time / amplitude evolutions in order to be able to cluster partials using common variation cues (see Section 8.1.3). For the experiment illustrated in Figure 4.10, we have chosen as input a three-tone violin sequence. The hissing of the bow leads to many noisy peaks and since the three tones are played *legato*, the transitions are not easy to identify. The LP algorithm better identifies the vibrato than the MAQ method does, but the representation is not precise (many partials belong to more than one source). The HFC algorithm better identifies the vibrato of the second tone as well as the onset / offset of each partial.

Listening Tests. The sinusoidal analysis system described in this document was implemented at France Télécom and tested in a low bit-rate audio coding framework where no more than 60 sinusoids are synthesized at one time using the synthesis strategy proposed in [MQ86]. In case of polyphonic recordings (trials "sc02" and "sc03" of the MPEG Audio listening test samples), the MAQ partial-tracking algorithm leads to two problems. The transitions between tones appear to be "smeared" and an unnatural "bubbling" effect appears. Informal listening tests were performed at France Télécom R&D and the listeners all notice a significant improvement of the perceived quality when considering the proposed partial-tracking algorithm. The bubbling effect due to wrong links in the high frequencies is almost removed and the transitions are synthesized with a higher fidelity.

4.2.5.2 Signal-to-Noise Ratios

However, the above graphical or auditive criteria are too subjective. We need more objective criteria, for now evaluated *via* the temporal signals corresponding to these sinusoidal representations, with the following methodology. A signal s is synthesized from a reference LTS source $\mathcal{L}$ and a perturbation p is added which is either a Gaussian white noise or a signal synthesized from an artificial LTS source. A STS representation $\hat{\mathcal{S}}$ is extracted from $s+p$ using the method described in Section 3.2. Then the partial-tracking algorithm under investigation is used to estimate $\hat{\mathcal{L}}$ from $\hat{\mathcal{S}}$. This LTS representation is then synthesized to obtain the signal $\hat{s}$. The closeness of $\mathcal{L}$ and $\hat{\mathcal{L}}$ is evaluated according to the reconstruction versus degradation SNRs (see Equation (3.155)) defined as, respectively

$$\text{D-SNR} = \text{SNR}(s,p), \tag{4.26}$$

$$\text{R-SNR} = \text{SNR}(s,\hat{s}-s). \tag{4.27}$$

Using these metrics, we compare the three partial-tracking algorithms of this chapter:

- the MAQ algorithm (Section 4.2.1) with $\Delta_\omega = 80$Hz;

- the LP algorithm (Section 4.2.3) with $\Delta_\omega = 40$Hz;
- the HFC algorithm (Section 4.2.4) with $\Gamma = 0.9$ and $L_T = 6$.

All the partials shorter than 100ms are discarded.

In the following experiments, audio inputs of increasing complexity are considered, each modulated by a vibrato since this kind of modulation is a worst-case scenario as far as tracking is concerned.

Resistance to Noise

Efficiency and discriminating capabilities of the three algorithms are evaluated using a synthetic constant-amplitude vibrato tone of 2-kHz base frequency, with a vibrato depth and rate of respectively 50 and 4Hz, mixed with a white noise of increasing level.

In a first experiment, to evaluate the efficiency, only the partial having the highest mean amplitude was synthesized to compute the R-SNR. At D-SNR below -7dB, the MAQ algorithm produces partials that are combinations of noisy peaks and tonal peaks so that the tones are split into several partials. With improved prediction capabilities, the LP and HFC methods are both able to track correctly the tone with vibrato even at high D-SNR, and perform similarly, as shown on Figure 4.11(a).

In the second experiment, to evaluate the discriminating capability of the two algorithms, all retained partials that lay in the $[1900, 2100]$Hz band are synthesized to compute the R-SNR. As shown on Figure 4.11(b), the LP method provides a significant improvement over the MAQ method. And compared to the LP method, the HFC method achieves an additional improvement of the same magnitude, thanks to its original selection criterion.

Management of Polyphony

The time / frequency analysis of polyphonic sounds requires a high frequency resolution, but the trade-off between time and frequency in a musical context leads to the use of analysis windows of reasonable lengths. Pitch relation between harmonic tones leads to DFT bin contamination and closely-spaced sinusoids in most natural cases. To evaluate the management of the closely-spaced sinusoids, a natural saxophone tone with vibrato is mixed with a set of synthetic constant-frequency and constant-amplitude sinusoids harmonically related, beginning 20 frames later. The fundamental frequency of this synthetic set is the same than the one of the saxophone tone, but all the frequencies within this set have been shifted by 70Hz towards the low frequencies in order to obtain the same DFT bin contamination for all the harmonics of the original source. Only the extracted partials starting before frame 20 were synthesized to compute the R-SNR. When the synthetic tone begins, the spectral information is blurred and some noisy peaks are present between the two close harmonics. The LP algorithm is unable to avoid bad links and performs as the MAQ algorithm does. Thanks to its original selection criterion computer over several frames, the HFC algorithm better tracks closely-spaced partials and thus performs quite well even at high D-SNR levels, as shown on Figure 4.11(c).

The problem of crossing partials arises when dealing with a mixture of non-stationary sounds. The tracking algorithm has to be able to identify the evolutions of the partials and to extrapolate missing spectral data. In order to test the management of crossing, a natural A-440Hz saxophone tone is corrupted by a synthetic constant-amplitude sinusoid beginning 500ms later and whose frequency is increasing linearly from 200Hz to 4kHz. Only the extracted partials starting before 500ms were synthesized to compute the R-SNR. Having a model of the evolutions of the parameters leads to an easier management of crossing partials, by being more selective and by having better prediction

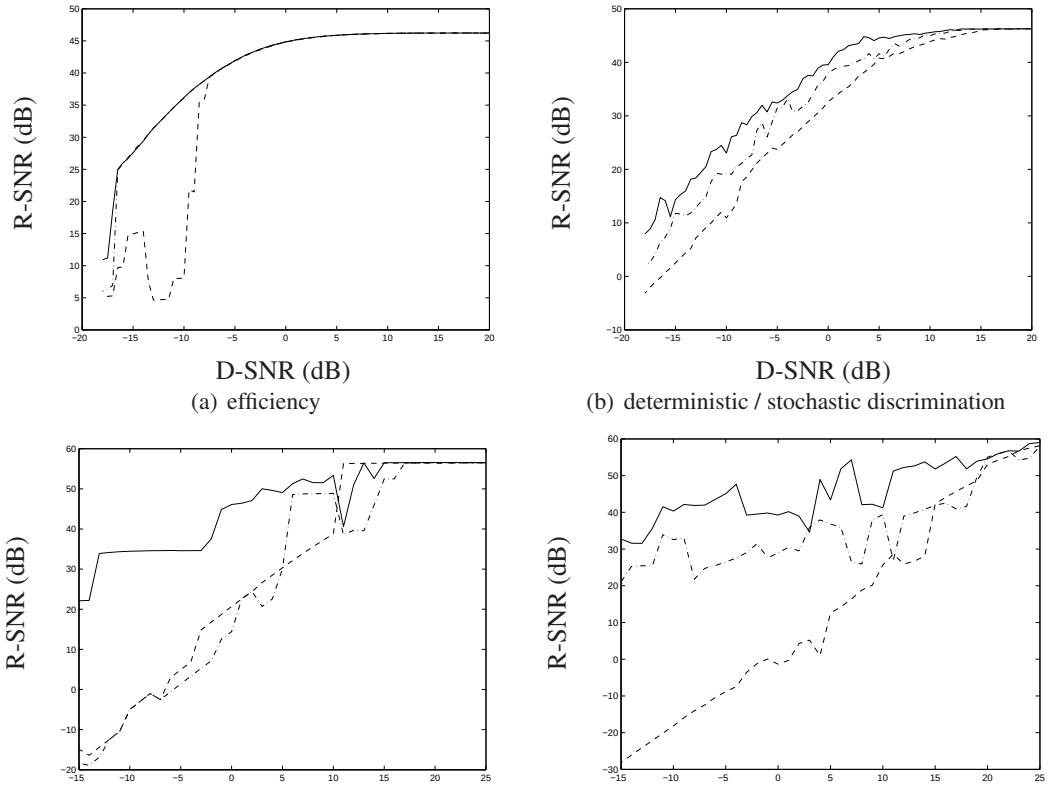


Figure 4.11: Evaluation of the resistance to noise (top) and management of polyphony (bottom). The efficiency (a), capability of deterministic / stochastic discrimination (b), resolution for closely-spaced sinusoids (c), and management of crossing partials (d) are evaluated for the three partial-tracking methods: MAQ (dashed line), LP (dotted line), and HFC (solid line).

and extrapolation capabilities. Furthermore, the LP and HFC algorithms sort the partials in decreasing amplitude, so that the partial with the lower degradation is processed first. This reduces the probability of handling the crossing incorrectly, leading to better results, as shown on Figure 4.11(d).

4.2.5.3 Efficiency and Precision

Although the above criteria are objective, they depend on the temporal signals. We need a way to compare LTS representations without going back to the temporal domain. A first attempt was proposed together with Lagrange [Lag04, 41].

Let $\mathcal{L}$ be a reference LTS representation and $\hat{\mathcal{L}}$ be the LTS representation computed with the tested partial-tracking algorithm from a synthesized version of $\mathcal{L}$. In theory, a perfect partial tracker should establish a bijection between $\mathcal{L}$ and $\hat{\mathcal{L}}$. In practice, together with Lagrange we propose 2 criteria.

First, $\hat{\mathcal{L}}$ has to be *efficient*, meaning that a partial of $\mathcal{L}$ should be represented with only one partial of $\hat{\mathcal{L}}$. More precisely, this efficiency property is given by

$$\forall k, \exists l | \mathcal{P}_k \subseteq \hat{\mathcal{P}}_l. \tag{4.28}$$

Second, $\hat{\mathcal{L}}$ should also be *precise*, meaning that a partial of $\hat{\mathcal{L}}$ represents only one partial of $\mathcal{L}$. More precisely, this precision property is given by

$$\forall k, \exists l | \hat{\mathcal{P}}_k \subseteq \mathcal{P}_l. \tag{4.29}$$

Thus, in theory, a perfect partial tracker is both (perfectly) efficient and (perfectly) precise.

In practice, we must be able to quantify these criteria in a non-boolean way. A representation $\hat{\mathcal{L}}$ is precise when it has very few partials whose peaks belong to different partials in the reference representation $\mathcal{L}$. Thus a quantitative precision criterion could be

$$C_p = 1 - \frac{\text{Card}\left\{\hat{\mathcal{P}}_i \mid \exists k,l,m,n, \hat{\mathcal{P}}_i[n] \in \mathcal{P}_k \wedge \hat{\mathcal{P}}_i[m] \in \mathcal{P}_l \wedge k \neq l\right\}}{\text{Card}\,\hat{\mathcal{L}}}. \tag{4.30}$$

A representation $\hat{\mathcal{L}}$ is efficient when very few partials of $\mathcal{L}$ have peaks allocated to several partials of $\hat{\mathcal{L}}$. Thus a quantitative efficiency criterion could be

$$C_e = 1 - \frac{\text{Card}\left\{\mathcal{P}_i \mid \exists k,l,m,n, \mathcal{P}_i[n] \in \hat{\mathcal{P}}_k \wedge \mathcal{P}_i[m] \in \hat{\mathcal{P}}_l \wedge k \neq l\right\}}{\text{Card}\,\mathcal{L}}. \tag{4.31}$$

The criteria above are still too strict, because they try to compare peaks without using a distance between them (in order to weight the error). In fact, we would need a distance between partials, and eventually between LTS representations. This quest for a distance between LTS representations has only begun, and is one of our main research topics for the future, since defining such a distance with the associated objective criteria will help designing efficient partial-tracking algorithms.

For now, together with Lagrange [41, 7] we propose to use rough approximations of these objective criteria and simulate degradations on the STS representation $\mathcal{S}$ associated to the sound s that are very likely to occur with real sounds. Typical perturbations of the STS representation due to the addition of noise or other sources in the polyphonic case are, respectively, the addition of noisy peaks, the degradation of the precision of the parameters of the peaks, or even the removal of relevant peaks.

From a given STS representation $\mathcal{S}$ of only one partial, such degradations are simulated by, respectively, adding peaks with random parameters, randomizing the parameters of randomly selected peaks of $\mathcal{S}$, or removing peaks from $\mathcal{S}$. The strength of the degradation is expressed as the ratio between added, modified, or removed peaks versus the size of $\mathcal{S}$. The randomized parameters are set to be in the same range as those of peaks of $\mathcal{S}$, *i.e.* the frequency is randomly chosen between the minimal and maximal values of the frequencies in the $\mathcal{S}$ set, respectively noted $\omega_{\min}$ and $\omega_{\max}$. The amplitude is chosen similarly. From this degraded STS representation, a set of partials $\hat{\mathcal{L}}$ is extracted using a partial-tracking algorithm.

The efficiency is evaluated as the inverse of the number of partials $\hat{\mathcal{L}}$ in or 0 if $\hat{\mathcal{L}}$ is empty. The precision is evaluated by means of the frequency and amplitude errors defined as

$$\text{FP} = (\omega_{\max} - \omega_{\min}) N_T \Big/ \sum_{n=0}^{N_T-1} \sum_{p=1}^{\text{Card}\hat{\mathcal{L}}} |\omega[n] - \hat{\omega}_p[n]|\, \varepsilon_p[n] \tag{4.32}$$

where N_T is the number of frames, $\omega[n]$ is the frequency of the original partial at frame n, and $\varepsilon_p[n]$ is equal to 1 if the partial k exists at frame n and 0 otherwise. The amplitude precision (AP) is defined similarly.

These criteria were evaluated using a large set of partials extracted using the MAQ algorithm from monophonic individual tones of every musical instruments of the Iowa database [iow]. The

(a) addition of peaks

Degradation	Efficiency			Precision F (and A)		
	MAQ	LP	HFC	MAQ	LP	HFC
20%	56	98	99	98 (98)	99 (99)	99 (99)
50%	32	51	63	82 (87)	90 (93)	92 (96)
80%	15	34	35	61 (72)	62 (78)	70 (82)

(b) randomization of the frequency parameter

Degradation	Efficiency			Precision F (and A)		
	MAQ	LP	HFC	MAQ	LP	HFC
20%	61	63	98	98 (91)	99 (91)	99 (99)
50%	47	53	51	90 (82)	91 (82)	98 (85)
80%	28	48	47	88 (79)	89 (80)	92 (83)

(c) removal of peaks

Degradation	Efficiency			Precision F (and A)
	MAQ	LP	HFC	all
10%	99	99	99	100 (100)
30%	53	97	88	100 (100)
40%	32	82	61	100 (100)

Table 4.2: Performances versus increasing degradation of the partial-tracking algorithms.

mean results expressed in percentages are presented in Table 4.2. The results show that the use of the LP method provides a significant improvement over the MAQ method. Compared to the LP method, the HFC method achieves most of the time an additional improvement of the same magnitude in terms of precision by successfully discarding partials with noisy evolutions.

4.3 Partial Synthesis

The parameters of the partials are estimated at the centers of the analysis frames, separated by I samples (see Figure 4.3). In general, $I > 1$ and a model is needed at the synthesis stage to reconstruct the $I-1$ missing values of the parameters of the partials.

4.3.1 Piecewise Polynomial Models

The classic approach is to use a model based on polynomials. More precisely, for each frame all the parameters are represented by polynomials. The constraints at the synthesis frame boundaries – that is the analysis frame centers, where the parameters are known – give the coefficients of the polynomials. For simplicity sake, let us consider the case of no overlap ($I = N$). We consider also only one partial ($P = 1$). Then, we denote by a_k, $\underline{b}_k$, ϕ_k, $\underline{\omega}_k$, and $\underline{\psi}_k$, respectively, the amplitude, (normalized) amplitude derivative, (wrapped) phase, (normalized) frequency, and (normalized) frequency derivative measured at frame number k. We consider only the synthesis frame delimited by the centers of the analysis frames k and $k+1$, and choose the local time origin $n = 0$ at the beginning of this synthesis frame.

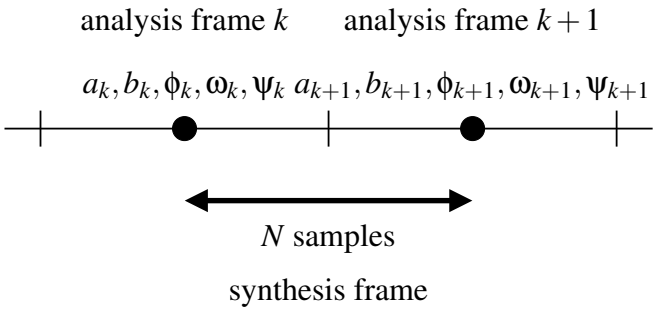


Figure 4.12: Analysis and synthesis frames (here in the case of no overlap, that is $I = N$). The parameters are estimated at the centers of the analysis frames. Synthesis frames are intervals between consecutive analysis frame centers.

4.3.1.1 Polynomial Phase Models

Regarding phase, we consider three polynomial models.

Order 1

A simple linear phase model (order-1 polynomial) is

$$\phi[n] = \phi_k + \frac{\phi_{k+1} - \phi_k + 2\pi M}{N} n \tag{4.33}$$

where M is the phase unwrapping factor given by Equation (4.38). Indeed, the measured phases are known modulo 2π (wrapped phases). But the true phase $\phi[n]$ is not wrapped. So the (measured) phase difference has to be unwrapped.

Order 3

The McAulay-Quatieri cubic phase model [MQ86] consists of an order-3 polynomial

$$\phi[n] = \phi_k + \underline{\omega}_k n + \alpha n^2 + \beta n^3 \tag{4.34}$$

because the continuity constraints at the left ($n = 0$) and right ($n = N$) frame boundaries (where estimations of the phase and frequency – phase derivative – parameters are known) give a system of 4 equations

$$\left\{ \begin{array}{lll} \phi[0] & = & \phi_k \\ \underline{\omega}[0] & = \phi'[0] = & \underline{\omega}_k \\ \phi[N] & = \phi_k + \underline{\omega}_k N + \alpha N^2 + \beta N^3 = & \phi_{k+1} + 2\pi M \\ \underline{\omega}[N] & = \phi'[N] = \underline{\omega}_k + 2\alpha N + 3\beta N^2 = & \underline{\omega}_{k+1} \end{array} \right. \tag{4.35}$$

from which we can derive the solutions, the α and β coefficients

$$\left[\begin{array}{c} \alpha \\ \beta \end{array} \right] = \left[\begin{array}{cc} 3/N^2 & -1/N \\ -2/N^3 & 1/N^2 \end{array} \right] \cdot \left[\begin{array}{c} \phi_{k+1} - \phi_k - \underline{\omega}_k N + 2\pi M \\ \underline{\omega}_{k+1} - \underline{\omega}_k \end{array} \right] \tag{4.36}$$

where the phase unwrapping factor M is calculated by using the "maximally smooth" criterion [MQ86], that is the maximal smoothing of the energy of the second derivative of the phase ($\underline{\psi} = \phi''$), and more

precisely by choosing the M that minimizes the function

$$f : M \mapsto \int_0^N \left(\underline{\Psi}(n)\right)^2 dn \tag{4.37}$$

which leads to

$$M = e\left[\frac{1}{2\pi}\left((\phi_k - \phi_{k+1}) + (\underline{\omega}_k + \underline{\omega}_{k+1})\frac{N}{2}\right)\right] \tag{4.38}$$

where $e[x]$ denotes the nearest integer from x.

Although this order-3 model ensures the continuity of the phases and frequencies at the frame junctions, it does not ensure the continuity of the frequency derivatives.

Order 5

Together with Girin *et al.* [25] we generalized the McAulay-Quatieri approach to the non-stationary case. Now, we suppose that we can also estimate the frequency derivatives at frame boundaries (non-stationary analysis, see Section 3.2). The resulting phase model is now an order-5 polynomial

$$\phi[n] = \phi_k + \underline{\omega}_k n + \frac{\underline{\Psi}_k}{2} n^2 + \alpha n^3 + \beta n^4 + \gamma n^5 \tag{4.39}$$

with the corresponding system of 6 equations resulting from the constraints at the frame boundaries

$$\begin{cases} \phi[0] & = & \phi_k \\ \underline{\omega}[0] & = \phi'[0] = & \underline{\omega}_k \\ \underline{\Psi}[0] & = \phi''[0] = & \underline{\Psi}_k \\ \phi[N] & = \phi_k + \underline{\omega}_k N + \frac{\underline{\Psi}_k}{2} N^2 + \alpha N^3 + \beta N^4 + \gamma N^5 = & \phi_{k+1} + 2\pi M \\ \underline{\omega}[N] & = \phi'[N] = \underline{\omega}_k + \underline{\Psi}_k N + 3\alpha N^2 + 4\beta N^3 + 5\gamma N^4 = & \underline{\omega}_{k+1} \\ \underline{\Psi}[N] & = \phi''[N] = \underline{\Psi}_k + 6\alpha N + 12\beta N^2 + 20\gamma N^3 = & \underline{\Psi}_{k+1} \end{cases} \tag{4.40}$$

from which we can derive the solutions, the α, β, and γ coefficients

$$\begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix} = \begin{bmatrix} 10/N^3 & -4/N^2 & 1/(2N) \\ -15/N^4 & 7/N^3 & -1/N^2 \\ 6/N^5 & -3/N^4 & 1/(2N^3) \end{bmatrix} \cdot \begin{bmatrix} \phi_{k+1} - \phi_k - \underline{\omega}_k N - \frac{\underline{\Psi}_k}{2} N^2 + 2\pi M \\ \underline{\omega}_{k+1} - \underline{\omega}_k - \underline{\Psi}_k N \\ \underline{\Psi}_{k+1} - \underline{\Psi}_k \end{bmatrix}. \tag{4.41}$$

where the unwrapping factor M is derived from the same maximally-smooth criterion of Equation (4.37) – although its derivation is more complicated in this case

$$M = e\left[\frac{1}{2\pi}\left((\phi_k - \phi_{k+1}) + (\underline{\omega}_k + \underline{\omega}_{k+1})\frac{N}{2} + (\underline{\Psi}_k - \underline{\Psi}_{k+1})\frac{N^2}{40}\right)\right]. \tag{4.42}$$

Examples of the behavior of the frequency within a synthesis frame for the three phase models are illustrated in Figure 4.13.

4.3.1.2 Polynomial Amplitude Models

Regarding amplitude, we also consider three polynomial models. They are much simpler, since the is no wrapping problem for the amplitude parameter.

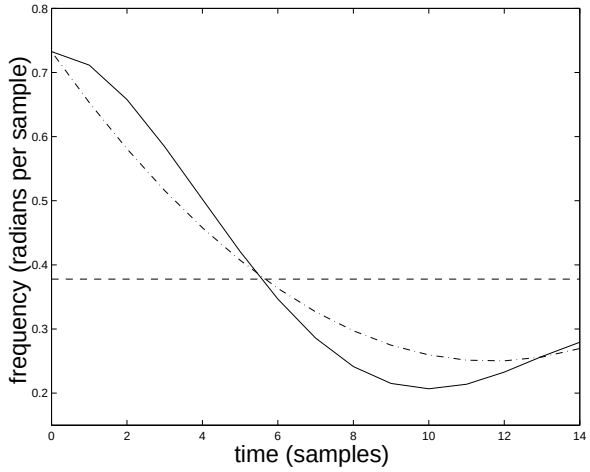


Figure 4.13: Examples of frequency trajectories within one synthesis frame. The order 1 phase model leads to a constant frequency (dashed line); the order 3 model makes the frequency appear as a parabolic segment (dash-dotted line), while the order 5 model is more flexible (solid line).

Order 0

The simplest model consists in considering the amplitude as constant:

$$a[n] = a_k. \tag{4.43}$$

Order 1

However, avoiding discontinuities can be done with a very simple order-1 polynomial:

$$a[n] = a_k + \frac{a_{k+1} - a_k}{N} n. \tag{4.44}$$

used by McAulay and Quatieri [MQ86].

Order 3

Together with Girin *et al.* [25], again we generalized the McAulay-Quatieri approach to the non-stationary case. Now, we suppose that we can also estimate the amplitude derivatives at frame boundaries (non-stationary analysis, see Section 3.2). Thus, we obtain a cubic model for the amplitude:

$$a[n] = a_k + \underline{b}_k n + \alpha n^2 + \beta n^3 \tag{4.45}$$

where $\underline{b}$ is the derivative of the amplitude a. The continuity constraints at the frame boundaries are

$$\begin{cases} a[0] & = & a_k \\ \underline{b}[0] & = a'[0] = & \underline{b}_k \\ a[N] & = a_k + \underline{b}_k N + \alpha N^2 + \beta N^3 = & a_{k+1} \\ \underline{b}[N] & = a'[N] = \underline{b}_k + 2\alpha N + 3\beta N^2 = & \underline{b}_{k+1} \end{cases} \tag{4.46}$$

and thus the α and β coefficients are

$$\alpha = \frac{1}{N^2}\left(3(a_{k+1}-a_k)-(2\underline{b}_k+\underline{b}_{k+1})N\right), \tag{4.47}$$

$$\beta = \frac{1}{N^3}\left(2(a_k-a_{k+1})+(\underline{b}_k+\underline{b}_{k+1})N\right). \tag{4.48}$$

4.3.1.3 Practical Evaluation

We conducted series of tests for the three phase models on both synthetic and natural sound signals.

Synthetic Sound Examples

The synthetic sound examples are tools for the investigation of the theoretic limits of the different models in case of ideal analysis, since all the parameters are known analytically and thus there is no short-term analysis imprecision to consider.

For all the synthetic sound examples, the sampling rate is $F_s = 44100$Hz, the length of the synthesis frames is $N = 64$, and the total length of the sound in samples is $L = 1000N$. All these examples are harmonic sounds, made of $P = 20$ partials. The signals are also quantified using 16-bit precision, thus resulting in CD-quality sound examples.

The first example is a perfectly stationary sound, where the fundamental frequency is $F_0 = 440$Hz and the amplitude a_p and (normalized) frequency $\underline{\omega}_p$ of the p-th partial are, respectively:

$$a_p = 1/P \quad \text{and} \quad \underline{\omega}_p = p2\pi T_s F_0. \tag{4.49}$$

The second example contains only linear variations. The amplitude is fading out while the fundamental frequency is raising (portamento from $F_0 = 440$Hz to $2F_0$):

$$a_p[n] = (1-n/L)/P \quad \text{and} \quad \underline{\omega}_p = p2\pi T_s F_0 \cdot (1+n/L). \tag{4.50}$$

It is clear that – unlike the previous examples – sinusoidal evolutions cannot be perfectly approximated by polynomials of finite degrees. The third example shows sinusoidal evolutions for the frequencies (vibrato), where the mean fundamental frequency is again $F_0 = 440$Hz, and the vibrato depth and rate are respectively $F_1 = F_0/2$ and $F_v = 8$Hz:

$$\underline{\omega}_p[n] = p2\pi T_s\left(F_0 + F_1 \sin(2\pi F_v T_s n)\right). \tag{4.51}$$

The vibrato (sinusoidal variation of the fundamental frequency) was tested with and without tremolo (sinusoidal variation of the amplitude). Without tremolo, the expression of a_p is the same as in the constant case, see Equation (4.49). In the presence of tremolo, with a mean amplitude of $A = 0.5$, and a tremolo depth and rate respectively set to $A_t = A_0/2$ and $F_t = 8$Hz, we have:

$$a_p[n] = \left(A + A_t \sin(2\pi F_t T_s n)\right)/P. \tag{4.52}$$

The results obtained on these synthetic – but related to musical evolutions of the frequency and the amplitude, thus significant – examples can be found in Table 4.3. In this table, the reconstruction signal-to-noise ratio (R-SNR, see Equation (4.27)) measures the energy ratio between the original signal s and the residual part (noise) obtained by subtracting the re-synthesis $\hat{s}$ from the original. The infinite symbol means that the two signals – original and re-synthesis – are identical (we reached the limit of the precision of 16-bit quantization for every sample). However, it is important to recall that

(a) order-1 amplitude

synthetic examples	1	3	5
constant	∞	∞	∞
linear	47.19	∞	∞
vibrato only	18.93	88.84	∞
vibrato+tremolo	19.20	74.80	74.82

(b) order-3 amplitude

synthetic examples	1	3	5
constant	∞	∞	∞
linear	47.19	∞	∞
vibrato only	18.93	88.84	∞
vibrato+tremolo	19.20	**88.34**	**116.93**

Table 4.3: Reconstruction SNRs in dB obtained on synthetic examples for the three phase models (polynomials of order 1, 3, or 5), with linear (a) or cubic (b) amplitude models. All the signals are quantized on 16 bits. The symbol ∞ means that – with this quantization – there is no difference between the original and the synthesized versions.

the R-SNR is not a perceptual metric. The case of constant parameters is perfectly handled by the three phase models, whereas linear evolutions (phase of order 2) require a polynomial order strictly greater than 1. The order 5 phase model better handles the sinusoidal evolutions of the frequency, and it is clear that the synthesis quality increases significantly with the phase model order in all cases except for the vibrato+tremolo case where the order of the amplitude model has to be increased too.

Natural Sound Examples

We conducted experiments on a variety of quasi-periodic signals such as voiced speech and music (pieces of guitar and bass) sampled at $F_s = 44100$Hz. To avoid partial-tracking problems (long-term analysis errors), only harmonic sound were used, with a fundamental frequency extracted semi-automatically (see [25] for details). Indeed, in the harmonic case, harmonics of the same rank are simply connected to each other across the frames. Of course, the results are still sensitive to the short-term analysis method used for the extraction of the model parameters. For this reason, we repeated our comparisons on two different short-term analysis methods: a pitch-synchronous (PS) method and a reassignment-based method (non PS).

The corresponding results of Tables 4.4 and 4.5 indicate that the performances are only slightly increasing with the order of the polynomial phase model, which is quite disappointing in comparison to the observations done one synthetic cases (see above). An explanation could be that the frame size N was too small for the improvement to be significant. The fact that the order-5 phase model does not significantly increase the SNR might also be – similar to the case of the synthetic sound with amplitude modulation – due to the fact that the performance is limited by the order-1 model we used for the amplitude[1]. Also, all the SNR values were surprisingly low, although the synthesized signals were perceptually quite close to their corresponding originals. This could be due to the imprecision of the short-term analysis methods, since the precise methods of Section 3.2 were not available in 2003. Indeed, the lower SNR of the reassignment-based analysis method compared to the PS method is an indication that this former method – now very precise – was still in its infancy. Thus these experiments might have to be reproduced with the new analysis methods. Anyway, when the frame size N is small, the gain in quality does not seem to be significant when the model order increases, which is the reason why we reverted to simpler (low-order) models for fast synthesis.

[1] At the time of these experiments, the estimation of the amplitude derivative was not available.

natural samples (PS)	1	3	5
speech (male)	21.15	23.55	23.87
speech (female)	25.75	27.52	27.83
singing voice	20.14	20.39	20.42
bass (short)	10.83	12.04	13.91
bass (long)	12.98	14.79	15.47
cello	18.00	19.24	19.67
electric guitar	10.83	12.04	14.39

Table 4.4: Reconstruction SNRs in dB obtained for different signals and the three tested phase model polynomial orders (PS analysis method).

natural samples (non PS)	1	3	5
bass (short, small window)	8.39	9.25	9.44
bass (short, large window)	8.71	9.56	9.76
cello	16.35	16.92	17.02
violin	17.68	17.91	17.94

Table 4.5: Reconstruction SNRs in dB obtained for different signals and the three tested phase model polynomial orders (non PS analysis method).

4.3.2 Resampling the Parameters

It is indeed possible to assume a very low model order on very short synthesis frames [12, Kut08]. Typically, for $F_s = 44100$Hz our frame size is $N = 64$. During each short-time synthesis frame, we consider the amplitude and frequency parameters as constant.

However, during the analysis stage, with a hop size of $I = 512$, the parameters of the partial were sampled at a much lower rate. In our case, it is necessary to upsample them by a factor 8 (see Figure 4.14) to match the rate of the synthesis frames. We do not assume a linear variation of the parameters. We intend to do even better to obtain a high quality with a low rate for the parameter flow.

4.3.2.1 Synthesis via Complete Resampling

When real-time synthesis is not an issue, for a high-quality synthesis we propose to use the resampling method of Section 1.1.3 to upsample the parameters by a factor I (the hop size of the analysis, that is the sampling period of the partials), which can seen as considering frames of $N = 1$ sample. Then, the synthesis is done by the direct application of (the discrete-time version of) the model Equation (3.4).

4.3.2.2 Upsampling the Flow of Parameters

When real-time synthesis is a priority, we have to revert to the larger frame size $N = 64$. However, the parameters of the partials were estimated each $I = 512$ samples. Thus, an upsampling (here of factor 8) is required.

With the method of Section 1.1.3, the upsampling would be done using a convolution with a reconstruction filter (a windowed sinc function), that would severely degrade the performance of our synthesis algorithm. In the Ph.D. manuscript ([Mar00], Section 4.4.2) we propose to use an approximation of the ideal reconstruction, and more precisely to use cubic cardinal splines.

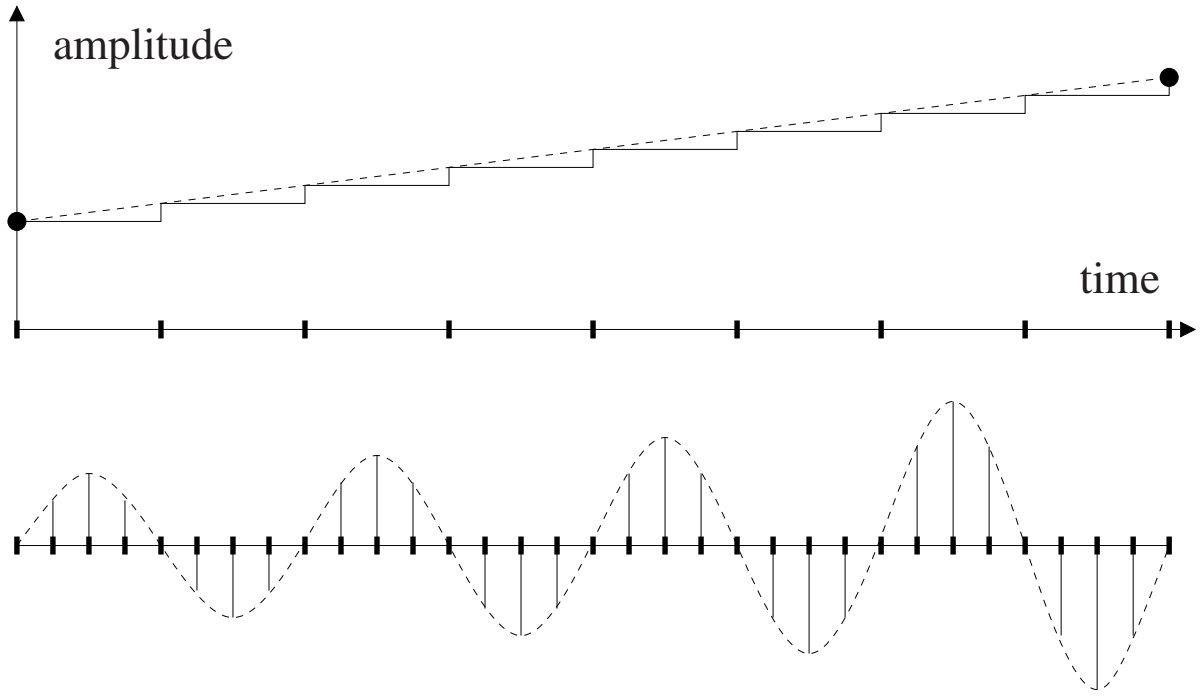


Figure 4.14: Variation of the amplitude of an oscillator and the resulting audio signal. Between two parameter changes, some interpolated values are computed (8 on the top of this figure). And between two interpolated values, many samples are computed (only 4 at the bottom of this figure, but 64 in our ReSpect library).

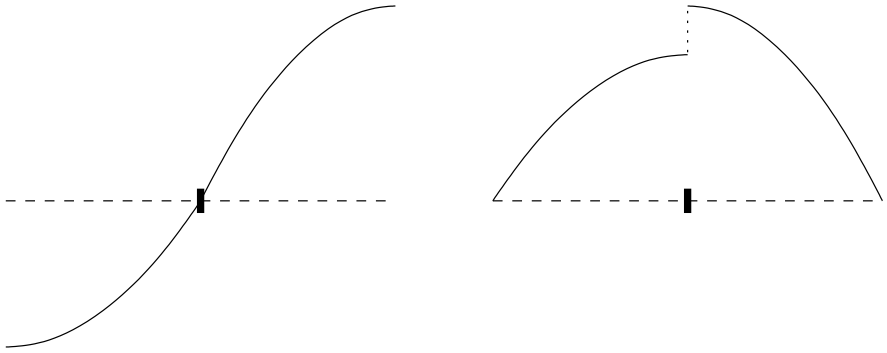


Figure 4.15: Changing the amplitude either when the signal is minimal (left) or maximal (right). It appears that the left case is much better, since it avoids amplitude discontinuities ("clicks").

The cardinal splines are interpolating uniform splines widely used in computer graphics [FvDFH95]. The s signal can be locally reconstructed from its samples using

$$s(kT_s + t) = \sum_{i=0}^{3} c_i(t)\ s[k-1+i] \quad (0 \leq t < 1) \tag{4.53}$$

where T_s is the sampling period of the (control) signal – indicating times where the signal values are know – and the c_i functions are, for cubic splines, the following Hermite blending functions:

$$c_0(t) = \frac{1}{2}(-t + 2t^2 - t^3), \tag{4.54}$$

$$c_1(t) = \frac{1}{2}(2 - 5t^2 + 3t^3), \tag{4.55}$$

$$c_2(t) = \frac{1}{2}(t + 4t^2 - 3t^3), \tag{4.56}$$

$$c_3(t) = \frac{1}{2}(-t^2 + t^3). \tag{4.57}$$

The use of the cubic cardinal splines in conjunction to the well-known Horner method for the evaluation of the polynomials result in a very efficient resampling technique with a sufficient precision. This reconstruction turns out to be of very good quality. The reason for this quality is that these Hermite functions constitute indeed a piecewise-polynomial approximation of the impulse response of the reconstruction filter (see [Mar00]).

4.3.3 Changing the Parameters

Also, for real-time synthesis, one would better reconstruct the parameter values only at strategic times. Together with Strandh [12] we indicate the best times in a period to (abruptly) change the parameters of a partial, as illustrated by Figure 4.15 and 4.16. The best moment to change the amplitude is when the signal is minimal, to preserve the continuity of the signal. And the best moment to change the frequency is when the amplitude of the signal is maximal, to preserve the continuity of the signal derivative. This way, the parameters are updated at the right moment to avoid clicks in the sound. This moment is different for every partial, depending on its frequency. Our PASS method presented in Section 3.3.2.2 is very flexible and has an event manager, which uses an optimized priority queue and allows the oscillators to take specific actions at scheduled times. Between two events, no parameter

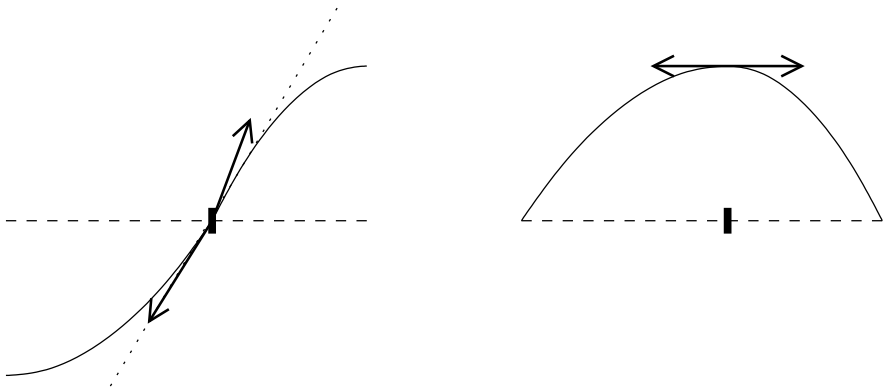


Figure 4.16: Changing the frequency either when the signal is minimal (left) or maximal (right). It appears that the right case is better, since it avoids derivative discontinuities (“clicks”).

change can occur, and the synthesis algorithm can generate the output samples in a very efficient way. Events are attached to oscillators, and there are events of different kinds:

- change of synthesis parameter (see Figure 4.14), possibly at optimal times (see Figures 4.15 and 4.16);
- update of the polynomial generator (D times per period, see Figure 3.15 and Section 3.3.2.2);
- reset of the digital resonator or incremental polynomial generator (depending on the synthesis method, see Section 3.3), to avoid numerical imprecision and drifts.

With our PASS method, when the period division is $D = 4$ changing the parameters of the partials at the best moments consists in changing the polynomial coefficients of an oscillator when this oscillator must be normally updated, because of the end of its validity period. Thus, this change does not require more computation time than without changing the parameters. With another period division (*e.g.* $D = 2$), update events must be added in the priority queue, to indicate the time to change the parameters of the partials.

Part III

Advances in Musical Sound Modeling

The third part of this manuscript describes ongoing research subjects (PhDs in progress) and research perspectives (subjects for future PhDs and projects).

Chapter 5 presents our very-long term modeling approach together with Girin and Firouzmand [29, 32, 5] and the **polynomials+sinusoids model** (Section 5.1.3) together with Raspaud [35] leading to **hierarchic sinusoidal modeling** (Section 5.2, [30, Ras07]) of great interest for time scaling while preserving the modulations (vibrato, tremolo). This research was done during the Ph.D. of Raspaud [Ras07]. Further work is still required: considering non-uniform time scaling to preserve the transients, finding a way to extend the hierarchic towards the macroscopic (musical) level – at least for repetitive (quasi-periodic) music, and proposing a better analysis method for the polynomials+sinusoids model for the second level of the hierarchic model – taking advantage of high-resolution methods (ongoing collaboration with Badeau) and an adapted version of our enhanced partial-tracking algorithm.

Until 2004, I have considered only sounds with a low noise level (and with deterministic parameters). Chapter 6 describes our work in progress on stochastic modeling together with Hanna and Meurisse (Master and ongoing Ph.D.). The aim of Meurisse's Ph.D. is to propose an unified model where deterministic and stochastic signals can be analyzed and synthesized in a same framework – with a model based on probability density functions. We have proposed an **analysis method based on the Rice probability density function** (Section 6.2, [39]), particularly efficient for medium signal-to-noise ratios (SNRs) – typically encountered in musical sounds – but still to be enhanced for low SNRs as well as in the non-stationary case. We still have to propose the corresponding synthesis method.

Also in 2004, I started to consider binaural signals (most people now listen to music using headphones, and a special effort is made in recording studios on spatial effects for the stereophonic – yet binaural – signals stored on standard compact discs). Chapter 7 describes our work in progress on spatial sound together with Mouba (ongoing Ph.D.). The aim of Mouba's Ph.D. is to propose a binaural to multi-diffusion ("upmixing") technique: the sound entities are extracted from the binaural mix and send to loudspeakers, possibly with intermediate manipulations (volume change, sound relocation, *etc.*). First, we propose a **simplified binaural model** (Section 7.3, [45]) based on spatial cues as functions of the azimuth of the source. Second, we also consider the distance of the source, with a localization based on the brightness. We then propose a **source separation algorithm** (Section 7.5, [40]) and **binaural and multi-diffusion synthesis techniques** (Sections 7.4.1 and 7.6, [45]).

Finally, Chapter 8 lists our main research directions for the future. The first one is the **decomposition of polyphonic music in sound entities** (see Section 8.1), according to the perceptive criteria by Bregman[2]. A major research direction is to find a way to get a unique criterion and replace partial tracking by entity structuring. For now, the structuring is done in a sub-optimal way after the long-term analysis stage, and by using each criterion separately: the common onsets (Ph.D. of Lagrange [Lag04]), the harmonic structures (Master's thesis of Cartier [Car03], Ph.D. of Lagrange [Lag04]), the correlated evolutions (Ph.D. of Lagrange [Lag04], Ph.D. of Raspaud [Ras07]), and the spatial location (ongoing Ph.D. of Mouba).

Since extracting the sound entities is an extremely difficult task, a solution to ease this extraction could be to take advantage of extra (inaudible) information stored (watermarked) in the sound (Section 8.2). Together with Girin [28] we have shown the suitability of spectral modeling for audio **watermarking**. Including such inaudible information in each sound entity prior to the mixing process should ease the extraction of the entities from the mix. We initiated a collaboration with Girin on this subject, called "informed separation".

[2]This will be our contribution to the French ANR SIMBALS project – see *Curriculum Vitæ* for details.

Our ultimate goal has a social aspect: with **“active listening”** (Section 8.3, [44]), we bet that a musician lies within any listener, and that we should change the way people can listen to music. We aim at providing the listener with the freedom to interact with the sound in real-time during its diffusion, instead of listening to this sound in the usual – passive – way.

Chapter 5

Advanced Sinusoidal Modeling

5.1 Modeling the Partials

5.1.1 Polynomial Model

We have seen in Section 4.3 that the parameters of the partials can be modeled using piecewise polynomials. With this approach, each partial parameter is modeled with several low-order polynomials, one for each frame. Another approach together with Girin and Firouzmand [29, 32] was to use only one polynomial, of higher degree K

$$x[n] = \sum_{k=0}^{K} c_k n^k \tag{5.1}$$

where x denotes the parameter of the partial to be modeled, and c_k are coefficients (the parameters of the model). This *very* long term modeling turned out to be efficient for compression purposes.

However, polynomials of finite degree can hardly handle sinusoidal modulations such as vibrato or tremolo.

5.1.2 Sinusoidal Model

Then, again with Girin and Firouzmand, we also considered a (co)sinusoidal model

$$x[n] = \sum_{k=0}^{K} c_k \cos\left(k\pi\frac{n}{N}\right) \tag{5.2}$$

which turned out to be also very efficient for compression purposes, especially when the order is optimally found during the weighted minimum mean square error (WMMSE) process used for the estimation of the coefficients [5]. Indeed, we use perceptual constraints: the amplitude and frequency errors have to stay under the masking threshold or frequency modulation threshold (see Section 1.5), respectively.

However, these very long term models are mainly for compression purposes, since the modeling of the trajectory of each parameter is done globally, thus losing the time dimension. This is against our time-varying perception of sound, and is not suitable for real-time sound transformations.

Moreover, for the phase parameter a linear term $c_{K+1}n$ had to be added in Equation (5.2), to handle the linear trend of the phase (a quasi-stationary frequency can be regarded locally as a constant, and the phase is the integral of the frequency).

5.1.3 Polynomials+Sinusoids (PolySin) Model

More generally, any partial parameter can be decomposed as a slow time-varying envelope plus quasi-periodic modulations. Thus, together with Raspaud and Girin [35], we propose the PolySin (polynomials + sinusoids) model

$$x(t) = \Pi(t) + \sum_{p=1}^{P} a_p(t)\cos(\phi_p(t)) \quad \text{with} \quad \phi_p(t) = \phi_p(0) + \int_0^t \omega_p(u)\,du. \tag{5.3}$$

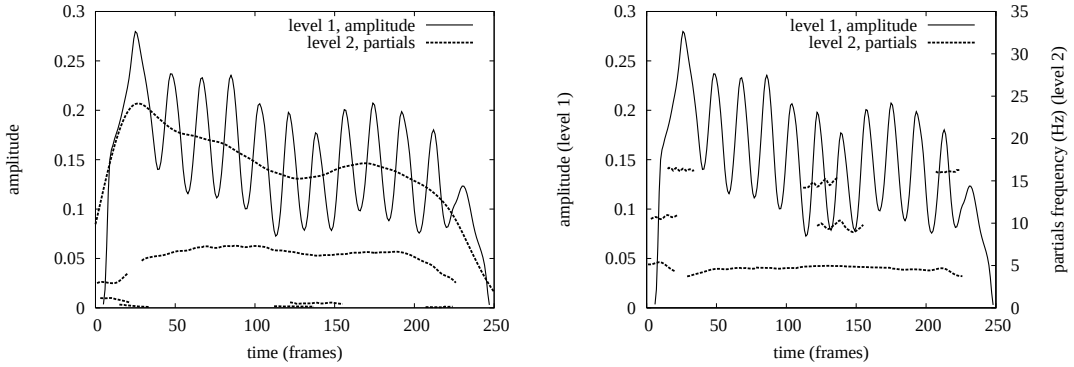
The low-order polynomials Π are well-suited for the slow time-varying envelope (*e.g.* to model glissando), whereas the sinusoidal part is perfect for the quasi-periodic modulations (*e.g.* vibrato). Here, we revert to the classic long-term approach: the parameters Π, a_p, ϕ_p, ω_p are functions of time that vary very slowly.

With all these models, the parameter signal x is decomposed on some basis (polynomials or/and sinusoids). However, although the previous models used an orthogonal basis, the PolySin model has to perform a decomposition on an over-complete "basis". Indeed, polynomials of high order can approximate sinusoids, and sums of many sinusoids can approximate polynomials. We want the polynomials to model only the envelope, and the sinusoids to model only the modulations. Fortunately, there is an orthogonality in the frequency domain: the envelope is very band-limited in frequency, to frequencies below 3Hz, whereas the modulations (vibrato, tremolo) are above 3Hz.

Regarding the analysis, we consider frames a small as possible, but large enough to contain at least one period of the modulations. We also chose a small degree (3) for the polynomials, to prevent them from capturing the oscillations corresponding to the modulations. Then, the PolySin analysis is twofold. First, we extract Π using a classic least-square error minimization scheme (see [35] for details). Second, we subtract the polynomial part and perform a sinusoidal analysis on the residual. Without this sinusoidal part, the polynomial would have been better estimated. Indeed, by subtracting the resynthesis of the estimated sinusoidal part, and going back to the first part of the twofold analysis, we can improve the estimation. As mentioned by Raspaud [Ras07], repeating the twofold analysis 5 times seems to lead to the best results.

This PolySin analysis is still in its infancy. For example, for the sinusoidal analysis we are currently using a classic DFT that requires a (too) large frame size. A better-suited solution could be using high-resolution (HR) methods. Indeed, the HR method used by Badeau [Bad05] finds a given number of complex exponentials that can either express sine functions or polynomials. This comes in very handy in our case, since we decide that the exponentials under 3Hz belong to polynomials, while the others belong to sinusoids. In theory it would then be possible to simultaneously compute the coefficients of both the sines and the polynomial, moreover with a small frame size. Although with Raspaud [Ras07] we noticed some problems in practice, we are currently investigating this research direction together with Badeau.

Also, the sparseness of the spectrum of a partial parameter – used with Girin for compression purposes – should not occasion much surprise. Indeed, on Figure 4.7 we see that the energy is concentrated at the frequencies corresponding to the modulations (at $\approx$ 5Hz on the figure).



(a) amplitudes of the amplitude of a partial (b) frequencies of the amplitude of a partial

Figure 5.1: Amplitudes (a) and frequencies (b) of the amplitude of a partial of an alto saxophone sound (with quasi-sinusoidal tremolo of frequency $\approx$ 5Hz).

5.2 Hierarchic Modeling

So, the quasi-periodic modulations present in the parameters of the partials generate peaks in their spectra. And since we track these peaks in time[1], with the PolySin model we do in fact a sinusoidal modeling of the parameters of the sinusoidal modeling described in Part II. The result is a 2-level sinusoidal model.

More precisely, starting from the temporal model (level 0), the PolySin model can be used to obtain the partials (of the classic sinusoidal model), which form the level-1 sinusoidal representation. Indeed, for zero-mean audio signals, we should have $\Pi \approx 0$, and the frequencies ω_p should have values in the audible range corresponding to $[20\text{Hz}, 20\text{kHz}]$. As seen in Section 4.2.4, ω_p and a_p are "inaudible" control signals (band-limited to 20Hz).

Then, from this level-1 representation, by using again the PolySin model, we shall obtain polynomials Π for the amplitude or frequency envelopes (band-limited to 3Hz), ω_p and a_p capturing the control of the modulations, thus ω_p should have values in the modulation range $[3, 20]$Hz. Moreover, ω_p and a_p are signals that vary very slowly in time.

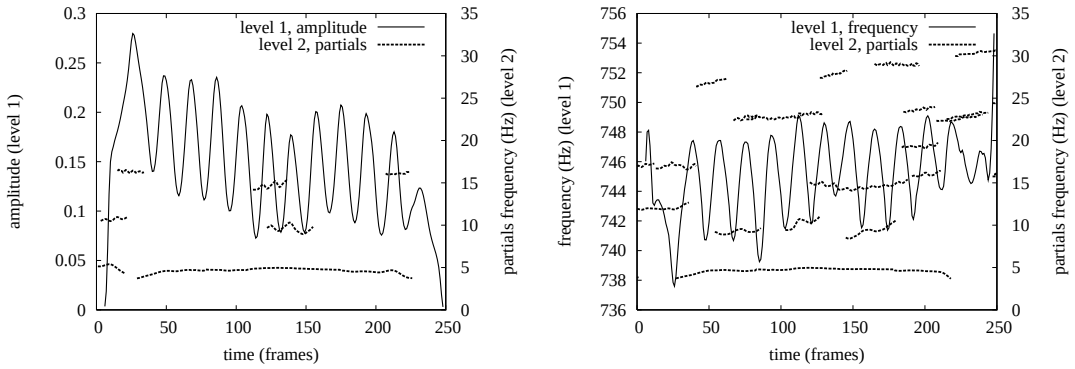
5.2.1 Partials of Partials

The level 2 of the hierarchic model consists of "partials of partials". More precisely, every parameter of every partial of the first level is represented by a set of these level-2 partials.

Figure 5.1 shows the amplitudes and frequencies of the level-2 partials for the amplitude of a partial of an alto saxophone sound with a tremolo of $\approx$ 5Hz. Only two components have a significant amplitude (Figure 5.1(a)): the envelope and one level-2 partial, of frequency $\approx$ 5Hz (Figure 5.1(b)). There are also level-2 harmonics, but of very low amplitude, because the tremolo is quasi-sinusoidal.

Figure 5.2 shows the frequencies of the level-2 partials for the amplitude (Figure 5.2(a)) and the frequency (Figure 5.2(b)) of the same (level-1) partial. We clearly see that the two modulations – vibrato on the frequency and tremolo on the amplitude – are of the same frequency $\approx$ 5Hz.

[1] For now, we have not adapted the tuning of the long-term analysis method (see Section 4.2) yet, and we are using the classic MAQ algorithm.



(a) frequencies of the amplitude of a partial

(b) frequencies of the frequency of a partial

Figure 5.2: Frequencies of the amplitude (a) and frequency (b) of a partial of an alto saxophone sound (showing that the tremolo and the vibrato are of the same rate).

5.2.2 Application to Time Scaling

At any level of the hierarchy, the sinusoidal modeling parameters are time signals band-limited in frequency (but often not zero-centered). In Section 1.1.3, we introduced an efficient technique[2] for reconstructing such signals at any time, and not necessarily at their original sampling periods. We propose now to first scale the time axis, then reconstruct the parameters according to this new scale, which is roughly equivalent to resampling. The time evolutions of the sinusoidal parameters are then scaled, but the values of these parameters are preserved. Note that for the phase parameter the scaled version also has to be multiplied by the scaling ratio in order to be consistent, because of the relation between frequency and phase given in Equation (5.3).

This time-scaling technique can be applied at different levels of the hierarchy (see Figure 5.3), though with different artifacts.

5.2.2.1 Level 0

Level 0 is the temporal domain, where the sound is described with its amplitude as a function of time. Applying the time-scaling technique to this sound signal is like playing a tape at a different speed, with typical artifacts. Indeed, at level 0 the change of duration is done at the expense of the shift of the pitch. For voices, it turns human beings into Disney's characters...

5.2.2.2 Level 1

When the time-scaling technique is applied at the level 1 of the hierarchy, that is on the sinusoidal modeling parameters, the pitch is then correct, but the modulations (vibrato, tremolo) rates are altered.

5.2.2.3 Level 2

When the time-scaling technique is applied at the level 2 of the hierarchy, that is on the parameters of "partials of partials", the modulations rate and depth are preserved, since they are encoded respectively in the frequency and amplitude parameters of the level-2 partials.

[2]Note that this technique is based on the PolySin (polynomials+sinusoids) approach.

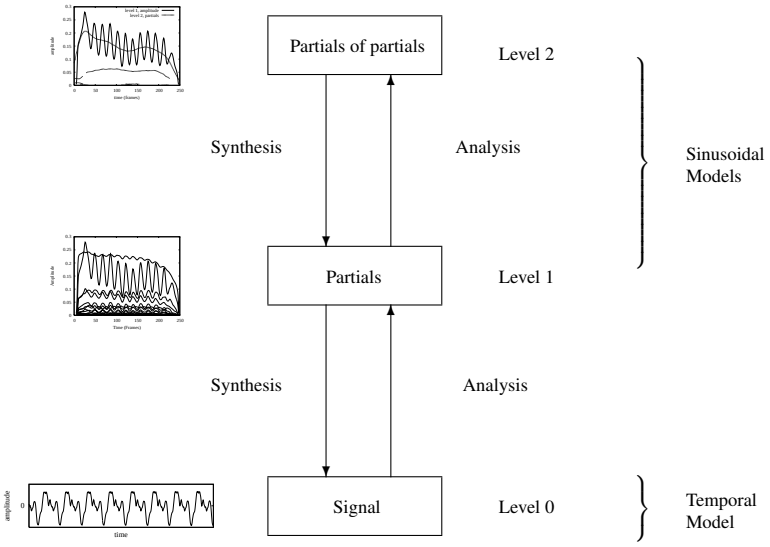


Figure 5.3: Hierarchy with two levels of sinusoidal modeling: level 0 is the temporal domain, level 1 corresponds to the classic sinusoidal representation described in Part II, and level 2 is the sinusoidal modeling of the parameters of the level-1 representation. Sinusoidal analysis and synthesis methods allow to go from one level to an other.

5.2.2.4 Transients

Then, the problem is that each level of the hierarchy involves a sinusoidal modeling that smoothes the attacks. The solution should be to implement a non-uniform time scaling: stationary parts could be scaled with huge factors, whereas transients prevent any scaling and would rather be reproduced without modification.

Identifying and modeling transients at each level of the hierarchy is a huge work. Even at the level 0, this is still an open research subject, perhaps even controversial. Indeed, from a psycho-acoustical point of view, transients have a rich spectral content but very localized in time. Thus, temporal masking phenomena (see Section 1.5) are likely to make all of them sound very similar. And yet, they seem extremely important to perception, and more precisely for the recognition of musical instruments. In fact, from an acoustical point of view, most transients are the result of the convolution of an impulse with some filter, dependent on the body of the instrument. Thus, the transients should embed the color – that is the timbre – of the instrumental sound.

5.2.2.5 Noise

Another issue is that the evolutions of the parameters are not exactly deterministic, and thus contain a stochastic part. Fortunately, many stochastic processes are stationary, and controlled by deterministic evolutions. These deterministic evolutions can be resampled. If the stochastic synthesis respects the statistical properties (*e.g.* distributions) of the original, then the resulting time-scaled sound is realistic. We have started our investigation on stochastic modeling with level 0, see Chapter 6.

5.2.3 Towards Multi-Scale Musical Sound

The idea behind Raspaud's Ph.D. [Ras07] was to continue deeper in the levels of the hierarchy, to reach the macroscopic form of the music [11]. Indeed, each level exhibits slower parameter evolutions: levels 0, 1, and 2 are respectively band-limited to 20000, 20, and 3Hz. The next level should then be below 120 cycles per minute, which is a typical value for the *beat* of a music. At this level, the envelope of the frequency parameter should show the pitches of the notes.

The problem is that the sinusoidal basis does not appear to suit the music level – discrete by nature – and for now we have not found a satisfactory basis. However, we are convinced that the principles of quasi-periodicity and prediction used in Part II are still valid at the musical level. Indeed, most popular musics are repetitive, though with variations. Also, music is often a balance between repetition (quasi-periodicity) and surprise (transients). When quasi-periodic patterns get repeated (typical of dance musics), prediction can be performed.

The ultimate time-scaling algorithm would then act on the level of the hierarchy adapted to the scaling factor. Indeed, for factors close to 1, the scaling can be done on the samples of the level-0 representation (microscopic level) without audible artifacts. But when the factor is close to 2, it is probably better to repeat (or predict, as in Section 4.2.3.3) a measure of the musical score (macroscopic level). In general, the scaling factor would have to be distributed among all the levels of the hierarchy.

Chapter 6

Stochastic Modeling

For now, we have only considered signals with deterministic parameters, and noise was regarded as an undesirable perturbation to these signals. However, there should be some stochastic component in the evolutions of these parameters. Also, there is an important noisy part in many musical sounds (*e.g.* flutes, voices, *etc.*). Moreover, some sounds are stochastic by nature, such as the sounds of the wind or oceanic waves. Then, the stochastic part of the sound is very important, and requires some model, associated to analysis and synthesis methods.

In this chapter, we describe our work on stochastic modeling together with Meurisse in the context of his ongoing Ph.D., co-supervised by Hanna.

6.1 Model

Regarding the model, again we consider short-term spectra. More precisely, for each bin of the discrete spectrum, together with Meurisse and Hanna [39] we consider the distribution of the amplitudes observed in this frequency bin as time goes by. Due to the stochastic nature of noise, a single short-term amplitude spectrum does not contain enough information to retrieve its statistical properties. That seems to be in agreement with perception: more time is needed to identify spectral content from noise than sinusoidal sounds. Our approach consists in considering a long-term analysis of the amplitude spectrum. Several short-term spectra are computed from several consecutive frames. The estimation thus relies on the study of the variations of the short-term amplitude spectra. The observation of successive short-term amplitude spectra shows significant differences between noise and sinusoidal spectral contributions. Amplitude spectra appear to be nearly constant in the case of sinusoidal sounds provided that frequencies and amplitudes do not highly vary over time (tremolo, vibrato). At the opposite, amplitude spectra of noisy sounds vary very rapidly with time (see Figure 6.1).

High variations seem to indicate the presence of noise whereas stationarity seems to characterize sinusoidal components. We propose here to revert to statistical considerations. The way these variations occur – and more precisely the study of their statistical distributions – leads to a new analysis method for the stochastic part. We present in this section the theoretical distribution of the amplitude spectrum of noises.

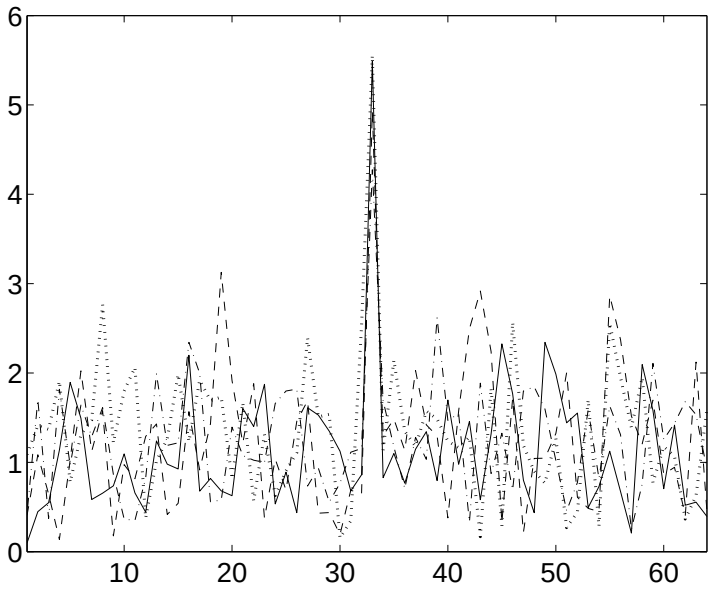


Figure 6.1: Several FFTs computed on a signal composed of a sinusoid and a white noise.

6.1.1 Sinusoid

If a stationary sinusoid with amplitude a is present in this bin, the distribution of the observed amplitudes is a normal (Gaussian) distribution with mean a and standard deviation 0 (a peak). The (wrapped) phase has an uniform distribution over $[-\pi, +\pi)$, although the derivative of the phase – the frequency – has a normal distribution, like the amplitude.

When two sinusoids are present at the same frequency, then the resulting signal is also a sinusoid of the same frequency (straightforward when considering the spectral domain). More precisely, the complex amplitudes of the sinusoids are summing up together, resulting in a complex amplitude of

$$A = a_1 e^{j\phi_1} + a_2 e^{j\phi_2} \tag{6.1}$$

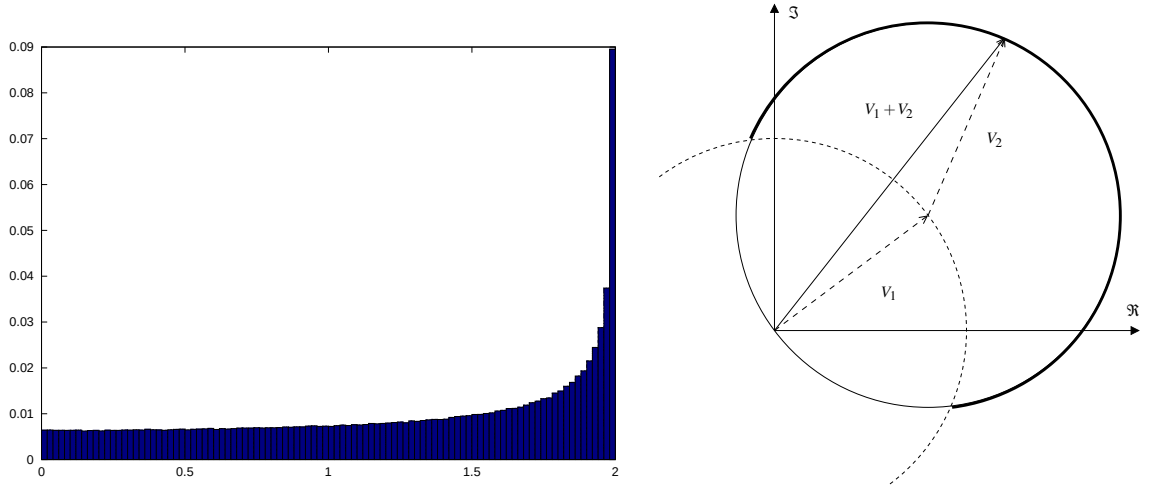
where a_p and ϕ_p are the amplitude and phase of the p-th sinusoid ($p \in \{1,2\}$), and thus the corresponding magnitude is (*via* the Cartesian representation of complex numbers)

$$a = |A| = \sqrt{(a_1\cos(\phi_1) + a_2\cos(\phi_2))^2 + (a_1\sin(\phi_1) + a_2\sin(\phi_2))^2}. \tag{6.2}$$

Physically, the addition of two sinusoidal signals of the same amplitude a_0 ($a_1 = a_2 = a_0$) is ruled by a nonlinear addition law which gives a maximum of $2a_0$ (thus $\approx$ 6dB above the volume of a_0) when the sinusoids are in-phase ($\phi_1 = \phi_2$) and a minimum of 0 when they are opposite-phase ($\phi_1 = \phi_2 + \pi$). One might intuitively think that all the cases in the $[0, 2a_0]$ interval are equiprobable. Not at all! In fact, $a \approx a_1 + a_2$ is the most probable, as shown in Figure 6.2. Anyway, the interference of several sinusoids at the same frequency results in one sinusoid. Thus, in the remainder of this chapter, we can only consider (at most) one sinusoid per bin.

6.1.2 Noise

We model stochastic signals as filtered white noise. Nowadays computers run deterministic processes. Even the `rand()` function is deterministic. However, it outputs a sequence of numbers that are statistically (nearly) indistinguishable from the realization of a stochastic process.



(a) distribution (b) intuition

Figure 6.2: Interference of two sinusoids of the same frequency and amplitude 1. The signal resulting from the addition of the these two sinusoids is also a sinusoid with the same frequency, but its amplitude depends on the phases of the initial sinusoids. (a) shows an histogram illustrating the probability density function of this amplitude (for uniform distributions of the phases). The sum of the amplitudes (2) is the most probable. (b) gives an intuitive justification. It shows that, considering the sum of the two vectors corresponding to the complex amplitudes of the sinusoids in the complex plane (see Equation (6.2)), its norm is more likely to be greater than 1 (bold line).

6.1.2.1 White Noise

A noise is white if its power spectral density is flat. Practically, this means that the average power spectrum (the square of the amplitude spectrum) is a constant. This is the base of the Welch method [Wel67] (see below). In this chapter we consider two types of white noise:

- **Uniform White Noise**:
  This is the one obtained using the classic `rand()` function. In this case, the values are uniformly distributed in the $[-1,+1]$ interval, with an expectation (mean) of 0.
- **Gaussian White Noise:**
  With this second type of white noise, the values are normally distributed, with a Gaussian function centered on the mean 0 and with a standard deviation of $\sigma = 1$.

In both cases, the phase is still uniformly distributed over the $[-\pi,+\pi)$ interval (as for sinusoids).

Because of the famous Central Limit Theorem, we assume that real world's noises are Gaussian. In the case of a Gaussian noise x, at any bin m the complex amplitude $X[m]$ in the discrete short-term spectrum is a complex random variable of time whose real and imaginary parts, denoted X_r and X_i, follow a Gaussian probability density function (PDF) with a standard deviation σ. The probability of the magnitude $M = |X| = \sqrt{X_r^2 + X_i^2}$ is given by the Rayleigh PDF defined by

$$p(M) = \frac{M}{\sigma^2} \exp\left(\frac{-M^2}{2\sigma^2}\right) \tag{6.3}$$

and σ is then the most probable value.

The total variance of the noise is $N\sigma^2$, where N is the size of the DFT used for the computation of the short-term spectra. In fact, when two (independent) noises are added, their variances are summing up together, thus resulting in a noise of variance

$$\sigma^2 = {\sigma_1}^2 + {\sigma_2}^2 \tag{6.4}$$

where ${\sigma_1}^2$ and ${\sigma_2}^2$ are the variances of the two initial noises.

6.1.2.2 Colored Noise

In most practical situations, the noise is not white. The square root of the mean of the short-term power spectra has a color $C[m] = \sigma[m]$ which is not constant over frequency.

However, we will consider only one frequency bin of the short-term spectra. At this bin, the noise can be considered as "locally white", meaning that σ is constant around the considered frequency.

6.1.3 Sinusoid and Noise

Sinusoids+noise models [Ser89, SS90] decompose natural sounds into two independent parts: the deterministic part and the stochastic part. The deterministic part is a sum of sinusoids evolving slowly, whereas the stochastic part corresponds to the noisy part of the original sound. Both parts are assumed quasi stationary. These hybrid models considerably improve the quality of the synthesized sounds. However, considering the two parts as orthogonal, they miss the correlations that often occurs between partials and noise in musical sounds. For this reason, we propose an unified approach, representing all signals in the same framework.

If we add a complex Gaussian noise X with standard deviation σ to a complex value $A_r + jA_i$ (the complex amplitude of some sinusoid) of magnitude $A = \sqrt{A_r^2 + A_i^2}$, the resulting magnitude $M = \sqrt{(X_r + A_r)^2 + (X_i + A_i)^2}$ is a random variable distributed according to the Rice PDF [Pap91] defined by

$$p_{A,\sigma}(M) = \frac{M}{\sigma^2} \exp\left(\frac{-\left(M^2 + A^2\right)}{2\sigma^2}\right) I_0\left(\frac{AM}{\sigma^2}\right) \tag{6.5}$$

where I_0 is the modified Bessel function of the first kind of order 0, these modified Bessel functions being given by

$$I_n(x) = j^{-n} J_n(jx) \tag{6.6}$$

where J_n is the Bessel function of the first kind of order n (see Section 3.3). Figure 6.3 shows the Rice PDF with $\sigma = 1$ and various values of A. The A value represents a fixed amplitude value due to the presence of a sinusoid. That is the reason why if A is zero, the Rice PDF turns into the Rayleigh PDF. At the opposite, if A is much greater than σ, the amplitudes are distributed according to a normal (Gaussian) distribution with standard deviation σ and mean A.

6.2 Analysis

6.2.1 Sinusoids

The problem of the estimation of the parameters of sinusoids in presence of Gaussian white noise has been extensively studied in Section 3.2. Here, we will focus on the estimation of the noise.

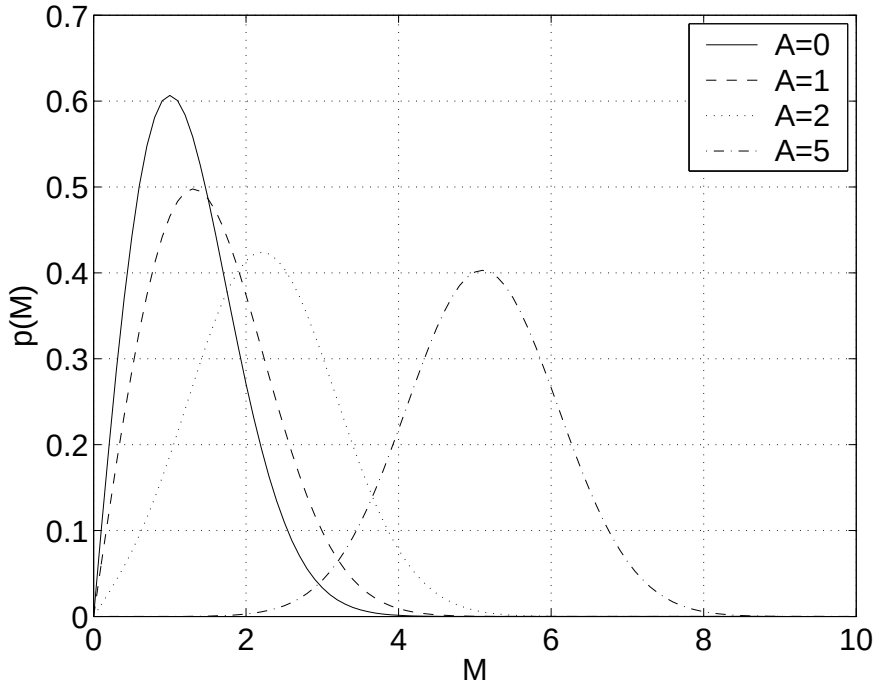


Figure 6.3: The Rice probability density function (PDF): if the parameter A of the function is zero, the Rice PDF turns into the Rayleigh PDF, whereas if A is much greater than σ, the Rice PDF turns into the normal (Gaussian) distribution.

6.2.2 Noise

In the absence of sinusoid, the Welch method [Wel67] estimates σ by computing the mean power of the short-term spectra over L consecutive (non-overlapping) frames of N samples, then taking its square root, that is

$$\hat{\sigma} = |X| = \sqrt{\frac{1}{L} \sum_{l=-(L-1)/2}^{l=+(L-1)/2} |X_l|^2} \tag{6.7}$$

here L being odd. X_l is the l-th spectral frame, the estimation being done at the time corresponding to the center of frame number 0.

6.2.3 Sinusoids and Noise

The previous method is severely biased in the presence of sinusoids.

The classic approach with hybrid sinusoids+noise models [Ser89] is to first estimate the parameters of the sinusoids, then subtract the sinusoidal part from the original, and finally model the residual – considered as noise. Therefore, the estimation of the stochastic part is dependent on the estimation of the deterministic part. However, this approach works only in the case of sounds with low noise levels. Indeed, the presence of high-level noise considerably degrades the quality of the results of the sinusoidal analysis methods: as seen in Section 3.2, the precision of the estimation is limited according to the Cramér-Rao lower bounds (CRBs). This precision decreases when the signal-to-noise ratio (SNR) decreases. Thus, in the presence of noise errors cannot be avoided, and these inevitable

errors in the estimation of the parameters of the deterministic part result in errors – often worse – in the estimation of the stochastic part.

We propose a method to analyze the stochastic part without any prior knowledge of the deterministic part. This method relies on a long-term analysis of the variation of the magnitude spectrum, and more precisely on the study of the distribution of the amplitude values in successive short-term spectra. As explained in Section 6.1.3, whatever the noise level is, the amplitude of each DFT bin is theoretically distributed according to the Rice law. Here, it is important to note that the probability that the amplitude of a bin reaches a very high or a very low value is not null.

Empirical methods based on the observations of Figure 6.1 have already been proposed [OKK04] with some success. A first possibility is to consider for each DFT bin the minimum of the amplitude spectra. In the case of noisy bins, this minimum may take values near zero whereas in the case of sinusoidal bins, this minimum approximates the amplitude of the sinusoid (slightly lower in presence of noise). Another similar idea is to consider the maximum of the amplitude spectra. Again, this maximum approximates the amplitude of the sinusoid (slightly higher in presence of noise). But if the energy of the analyzed bin is due to the presence of noise, the maximum of the amplitude spectra may take very high values. The last empirical method is to consider the average of the amplitude spectra, which also leads to errors in the case of noisy bins.

Some works propose to improve previous estimators using statistical properties of the magnitude spectrum. Those properties can be used to compensate the bias induced by a method based on minima tracking in the spectrum [Mar06] or to correct the estimation obtained by smoothing from the spectrum of the residual [YR06].

We propose a method that relies on a study of variations in the magnitude spectrum along the time axis and does not uses statistics in a corrective way. We keep with the sinusoid+noise model of Section 6.1.3, and propose two methods to estimate the σ parameter of each bin, using the Rice PDF. (We do not consider the estimation of A, since it has been extensively studied in Section 3.2).

We make a long-term stationarity hypothesis: the noise power density and the frequencies and the amplitudes of the sinusoids are assumed to be constant during L consecutive frames of size N. The discrete Fourier transform is computed for each frame. For each frequency bin m the magnitude is computed for each frame in order to obtain a data set $M[m]$ of L realizations per bin. The following methods are estimators for $\sigma[m]$ from a single set of realizations. The variance $\sigma^2[m]$ for the all bins leads to an estimation of the noise power density.

6.2.3.1 Maximum Likelihood Method

The amplitude A and the standard deviation σ can be computed by maximizing the log-likelihood function [SdDDR98]

$$\{\hat{A},\hat{\sigma}\} = \underset{A,\sigma}{\arg\max}\ \log(\mathcal{L}) \tag{6.8}$$

where the likelihood function is given by

$$\mathcal{L} = \prod_{i=1}^{L} p_{A,\sigma}(M_i) \tag{6.9}$$

for a probability density function $p_{A,\sigma}(x)$ and a set M of L realizations, M_i being the i-th element of M. In the case of the Rice PDF (see Equation (6.5)), the log-likelihood function is

$$\log(\mathcal{L}) = \sum_{i=1}^{N} \frac{M_i}{\sigma^2} I_0\left(\frac{AM_i}{2\sigma^2}\right) - \frac{NA^2}{2\sigma^2} - \sum_{i=1}^{N} \frac{M_i^2}{2\sigma^2}. \tag{6.10}$$

The maximization of the log-likelihood function on each data set gives us the parameters A and σ of the associated bin. The maximization of 2-dimensional functions can be time consuming. Fortunately, the maximization problem can be here reduced to a 1-dimension problem by normalizing the data set M by the square root of the second moment (the second moment of the data set being an unbiased estimator of the Rice second moment), see [39] for details. However, it is still time consuming and another – much faster – approach leads to similar results.

6.2.3.2 Moments Method

The standard deviation σ can also be estimated by using any pair of moments and by finding the values that match these moments, the first and second moments being the best suited [TAGK01].

If the random variable M follows a Rice PDF, its first moment is

$$\mu_1 = \left[(1+\gamma)I_{e_0}\left(\frac{\gamma}{2}\right)+\gamma I_{e_1}\left(\frac{\gamma}{2}\right)\right]\cdot\sigma\sqrt{\frac{\pi}{2}} \tag{6.11}$$

where γ denotes the signal-to-noise ratio (SNR, see Section 3.2.5.2)

$$\gamma = \frac{A^2}{2\sigma^2} \tag{6.12}$$

and I_{e_0} and I_{e_1} are the scaled modified Bessel functions defined by

$$I_{e_n}(x) = e^{-x}I_n(x) \tag{6.13}$$

with I_0 and I_1 being the modified Bessel functions of the first kind of orders 0 and 1, respectively, see Equation (6.6). Indeed, for high SNRs the regular (not scaled) modified Bessel functions might have exceeded the maximum value for double-precision floating-point arithmetic.

The second moment is much simpler:

$$\mu_2 = A^2+2\sigma^2 = 2\sigma^2(\gamma+1). \tag{6.14}$$

The normalized mean μ' is then defined as the mean computed on the data set normalized by the square root of the second moment, that is

$$\mu' = \frac{\mu_1}{\sqrt{\mu_2}} \tag{6.15}$$

and can be easily expressed as a function of the SNR [TL91]:

$$\mu'(\gamma) = \frac{\sqrt{\pi}}{2\sqrt{1+\gamma}}\left[(1+\gamma)I_{e_0}\left(\frac{\gamma}{2}\right)+\gamma I_{e_1}\left(\frac{\gamma}{2}\right)\right]. \tag{6.16}$$

In practice, the moments are estimated using the expectation E:

$$\hat{\mu}_1 = E[M] = \frac{1}{L}\sum_{i=1}^{L} M_i \quad \text{and} \quad \hat{\mu}_2 = E[M^2] = \frac{1}{L}\sum_{i=1}^{L}(M_i)^2. \tag{6.17}$$

Then, from the estimated normalized mean

$$\hat{\mu}' = \frac{\hat{\mu}_1}{\sqrt{\hat{\mu}_2}} \tag{6.18}$$

we can compute the value of γ that makes the theoretical normalized mean match this estimated value, by solving

$$\mu'(\gamma) = \hat{\mu}'. \tag{6.19}$$

In practice, the inverse of the function of Equation (6.16) can be pre-computed in a table (though with the SNR preferably expressed in dB for arithmetic issues). Finally, from the estimated $\hat{\gamma}$, inverting Equation (6.14) leads to an estimation of σ:

$$\hat{\sigma} = \sqrt{\frac{\hat{\mu}_2}{2(\hat{\gamma}+1)}}. \tag{6.20}$$

The smoothing in time and/or frequency of these raw estimates can have a positive effect on estimation precision, though at the expense of lowered time and/or frequency resolutions.

6.2.4 Experimental Results

The two estimation methods described previously have been compared with various SNRs and number of realizations. Both methods give similar results. The moments method being much faster, it has been selected for the practical implementation.

The data sets are computed using a sound sampled at $F_s = 44100$Hz, composed of a white noise of standard deviation $\sigma\sqrt{N}$ and a sine wave of frequency $F_s/4$ whose amplitude is $\sqrt{2\gamma}$. FFTs are computed on $N = 1024$ samples. The data sets are computed on the central bin $N/4$. This way, the data set values obtained when the frames are not overlapping are realizations of a Rice PDF with standard deviation σ and amplitude $\sqrt{\gamma\sigma^2}$. The estimated σ (respectively standard deviation of the estimator) is the mean (respectively the standard deviation) computed over 1000 data sets from distinct synthetic sounds.

6.2.4.1 Bias at Low SNRs

For low SNRs, it would be better to revert to the Welch method. Indeed, experimentations show that the estimation is biased at low SNRs. Figure 6.4 shows the estimated σ as a function of γ with different numbers of realizations. The distributions are computed with $\sigma = 1$, and A varies according to γ. For 20 samples, the estimator is biased for γ lower than 1. When $\gamma = 0$, the estimation is biased by 20%. For 1000 samples, the estimation is biased for γ lower than 0.5 and the bias is very low (5%). Therefore, increasing the number of realizations reduces the bias. For low SNRs, the Rice PDF changes only slightly. More observations are thus needed to fit closely the Rice PDF. So, if the number of frames is not sufficient, errors in the estimation of γ are likely to occur. If the SNR is low, more observation are needed to avoid bias. If the SNR on a bin is *a priori* known to be $-\infty$ ($\gamma = 0$), the distribution follows a Rayleigh PDF and the computation of the first moment gives an unbiased estimator: this is exactly the Welch method.

6.2.4.2 Number of Observations

Experimentations show that increasing the number of realizations reduces both error and bias. When using more than 1000 realizations, the spectral envelope obtained is smooth. However, using 1000 frames is not acceptable in practice. Indeed, considering 1000 observations imposes a sound duration of more than 23 seconds when the sampling frequency is 44100Hz and the DFT size is 1024. The signal is likely to change in a significant way during such a long period. Since the estimator is unbiased from 20 realizations (see Figure 6.5), it can be a good choice to compute distributions on 20 frames.

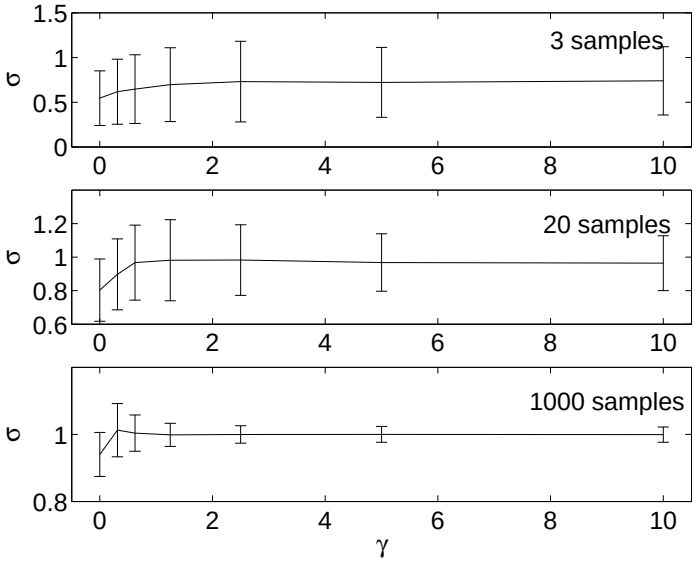


Figure 6.4: Estimated σ on Rice-distributed sets of size 3, 20, or 1000 (theoretic values $\sigma = 1$ and γ varies from 0 to 10). Vertical bars indicate the standard deviation of the estimation.

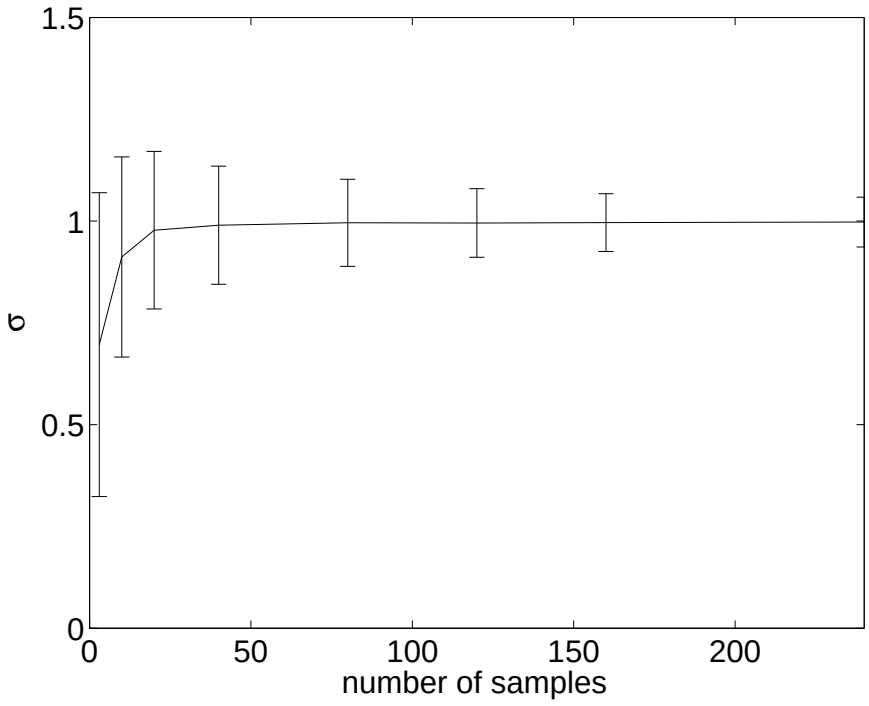


Figure 6.5: Estimated σ of a Rice-distributed set (theoretic values $\sigma = 1$ and $\gamma = 2$) for several numbers of realizations. Vertical bars indicate the standard deviation of the estimation.

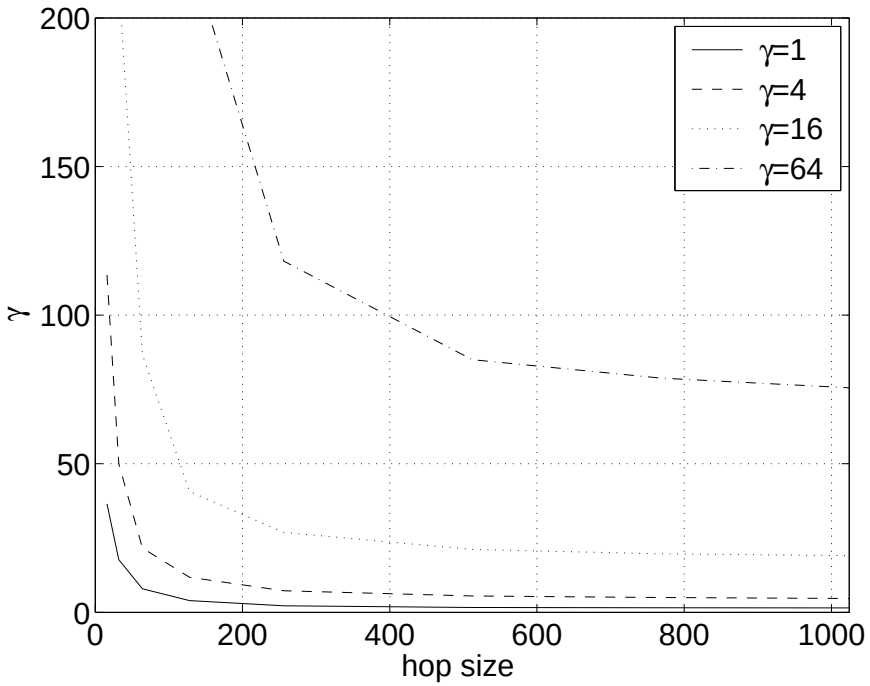


Figure 6.6: SNR γ estimated for several Rice-distributed DFT bins of a FFT of size $N = 1024$ and with various hop sizes I. The distributions are computed with $L = 20$ frames. Thus, for $I = N = 1024$ there is no overlap. The overlap factor increases when the hop size I decreases.

6.2.4.3 Effect of the Overlap

Computing σ on overlapping frames will improve the precision in time, since it will reduce the duration of the observed sound. But our estimation method assumes that the L realizations are statistically independent. Overlapping frames breaks this assumption and induces correlation in the data set. More precisely, A may be overestimated while σ is underestimated. However, overlapping frames by 50% seems to have a small impact on the results (see Figure 6.6). Moreover, the bias and mean square error on the estimation of γ have been calculated using L non-overlapping frames on the one hand, and $2L-1$ overlapping frames (50% overlap factor) on the same duration on the other hand. And perhaps surprisingly, it appeared that a 50% overlap reduces these bias and mean square error for the estimation of γ (see [39] for details).

To conclude, from our tests, it appears that using $L = 21$ overlapped frames ($I = N/2$) gives the best results. Using less frames strongly degrades the performances whereas increasing the number of frames improves the results only slightly. However, due to the underestimation of the method at low SNRs, in this case it may be suitable to increase the number of frames L if the bias is not acceptable.

6.2.4.4 Effect of the Non-Stationarity

Since our method relies on statistical variations of the amplitude, any variation of the sinusoids is interpreted as noise. Thus amplitude modulation leads to overestimation of the noise power density. Also, if the frequency of the sinusoid varies (frequency modulation), then the peak moves inside the DFT bin and because of the analysis window shape the amplitude observed on the bin varies (see the Γ_w function of Equation (3.31), Section 3.2). This leads again to an overestimation of the noise level

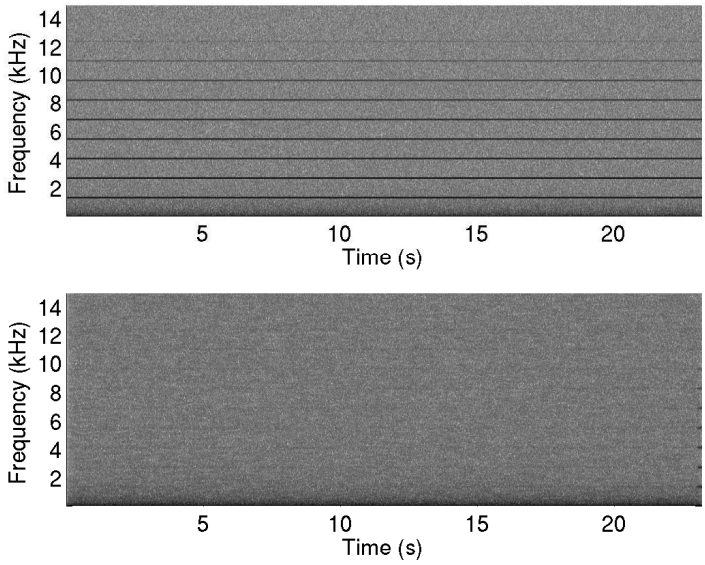


Figure 6.7: Spectrogram of a synthetic sound (23s) composed of 9 stationary sinusoids (fundamental frequency 1378Hz) with a colored noise (up) and its analyzed stochastic component (bottom).

on this bin. The bias is maximal when the modulated sines moves inside the whole bin.

6.2.4.5 Sound Examples

Synthetic Sound. Figure 6.7 shows the spectrogram of a synthetic sound and its stochastic component. This sound is composed of a pink noise and several sinusoids with various amplitudes. Due to the under-estimation of σ at low SNR, horizontal lines appear on the spectrogram. These lines are located on the frequency bins inhabited by the sines. However this error is hardly audible (and hardly visible on the figure).

Natural Sound. The method has then been tested on a sound composed of a saxophone tone and wind noise (see Figure 6.8). Due to the length of the analysis frame, variations in the color of the noise are stretched in time while the attack and the release from the saxophone disturb the magnitude distribution. When the sound is nearly stationary during the analysis frame, the sinusoids are correctly removed. Due to the under-estimation at low SNRs and the small amplitude modulation of the harmonics of the saxophone sound, some estimation errors appear for the frequency bins inhabited by the sinusoids. This error is not disturbing, when the amplitude modulations are limited.

6.3 Synthesis

There are mainly two methods for the synthesis of Gaussian colored noise.

6.3.1 Temporal Method

The temporal approach consists in generating a Gaussian white noise of variance 1 in the temporal domain, then in filtering it for example using the FFT to perform the convolution with a filter corre-

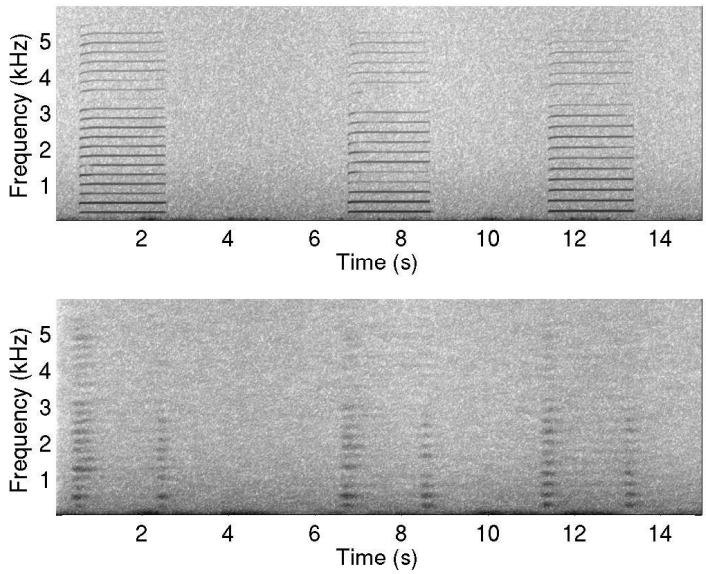


Figure 6.8: Spectrograms of a natural sound (15s) composed of 3 notes (`A#4`, `C#4`, and `D4`) of saxophone with a background wind noise (up) and its analyzed stochastic component (bottom).

sponding to the color $C = \sigma$.

It is interesting to note that when the source is replaced by an uniform white noise (obtained from the `rand()` function), the difference can hardly be heard[1].

6.3.2 Spectral Method

The spectral approach consists in generating each short-term spectrum, then in switching to the temporal domain using an FFT. For a given spectrum $X[m]$, the amplitude $|X[m]|$ is $\sigma[m]$ and the phases $\angle X[m]$ are randomly chosen and uniformly distributed. Then it is better to use the overlap-add technique, for example using the cosine window (the square root of the Hann window), since it preserves the power (see Section 3.2.2.2).

In the future, for the general sinusoid+noise case, we plan to use a similar approach with the model of Section 6.1.3 to distribute the parameters (amplitude and phase). Thus, we would be able to synthesize both the deterministic and the stochastic parts in the same way, which could be useful for musical sounds where the these two parts may show some correlations. However, we have to study first these correlations. Moreover, although we have a model for the amplitude, we are currently missing an unified model for the phase. Indeed, the distribution is uniform in both deterministic and stochastic cases. We have to consider the sequences of the values...

[1] A reason might be the following: the hearing system is very complex, and going through the different mechanisms of the ear, the uniform distribution might turn into a normal (Gaussian) one in the end, thanks to the Central Limit Theorem...

Chapter 7

Spatial Sound and Hearing

In this chapter, we describe our work with Mouba in the context of his ongoing Ph.D. We use the observations of Section 1.4, and use the fact that human beings have in fact 2 ears to localize sounds...

7.1 Source Position

As mentioned in Section 1.4, we consider a punctual and omni-directional sound source in the horizontal plane, located by its (ρ, θ) coordinates, where ρ is the distance of the source to the head center and θ is the azimuth angle. Moreover, we consider that we are in outdoor conditions. We consider neither any room nor any obstacle. In this free-field case, only the torso, the head, and the outer-ear geometry modify the sound activity content by reflections and shadowing effect.

7.2 Head-Related Transfer Functions

The source s will reach the left (L) and right (R) ears through different acoustic paths, characterizable with a pair of head-related impulse responses (HRIRs). The effects of the distance ρ have been described in Section 1.4. Let us here consider this distance as fixed and consider in turn the effects of the azimuth. For a source s located at the azimuth θ, the left (x_L) and right (x_R) signals are given by

$$x_L = s * \mathrm{HRIR_L}(\theta), \tag{7.1}$$
$$x_R = s * \mathrm{HRIR_R}(\theta) \tag{7.2}$$

where $*$ is the convolution among temporal signals (see Chapter 1).

In fact, each HRIR can be regarded as a filter, defined in the temporal domain by its impulse response for each azimuth θ. The HRIRs are functions of the location of the source, but also depend on the morphology of the head of the listener and might be slightly different for each ear. The CIPIC database [ADTA01] contains this information for several listeners and different directions of arrival. However, in practice, when someone listens to a binaural recording done in the ear canals of someone else, the spatial image is preserved. Thus, to be subject-independent, we can deal with representative (average) HRIRs, such as the one of the KEMAR manikin [GM94].

Equations (7.1) and (7.2) can be greatly simplified by switching to the spectral domain, since then the convolutions are replaced by simple multiplications. The spectral versions of the HRIRs are called head-related transfer functions (HRTFs).

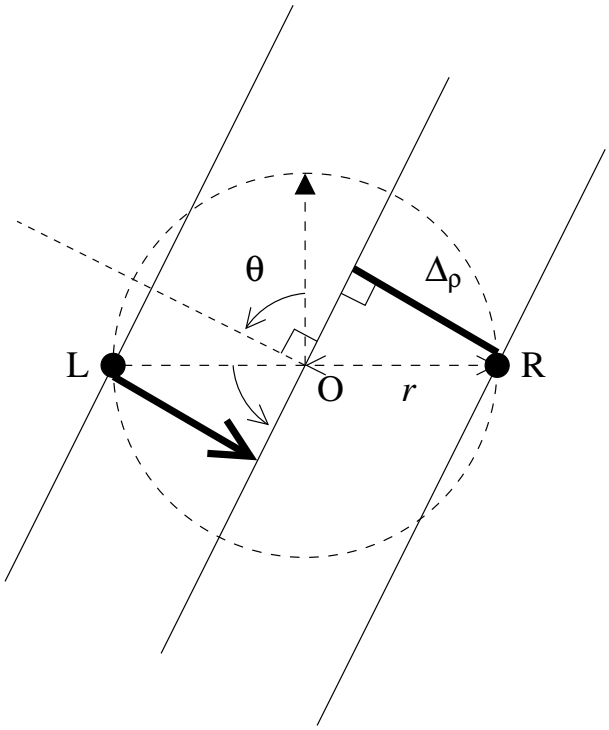


Figure 7.1: Free-field case, where the acoustic waves reach the left (L) and right (R) ears without encountering any obstacle. O is the center of the head; r is its radius.

7.3 Binaural Cues

In the case of s being a pure sinusoid of frequency f, the x_L and x_R signals are also sinusoids of the same frequency f, but with different phases and amplitudes. In fact, the head-torso system impacts the delay and level to each ear. A sound source positioned to the left will reach the left ear sooner than the right one, in the same manner the right level should be lower due to wave propagation and head shadowing (see Figure 7.1). More precisely, between the two ears, the phase difference is approximatively Δ_ϕ and the amplitude ratio is approximately 10^{Δ_a}, with

$$\Delta_\phi(\theta, f) = \mathrm{ITD}(\theta, f) \cdot 2\pi f, \tag{7.3}$$

$$\Delta_a(\theta, f) = \mathrm{ILD}(\theta, f)/20 \tag{7.4}$$

where the difference in amplitude or interaural level difference (ILD, expressed in decibels) and difference in arrival time or interaural time difference (ITD, expressed in seconds) are the main spatial cues for the human auditory system [Bla97].

Since each point of the time-frequency plane can be regarded as the contribution of a single sinusoid, we have the following relation between the spectra measured at the left and right ears:

$$X_L(t, f) = X_R(t, f) \cdot 10^{\Delta_a(\theta, f)} e^{j\Delta_\phi(\theta, f)}. \tag{7.5}$$

Lord Rayleigh mentioned in his Duplex Theory [Ray07] that the ILDs are more prominent at high frequencies whereas the ITDs are crucial at low frequencies. In fact, the human auditory system is well adapted to the natural environment. Indeed, high frequencies are more sensitive to frequency-selective amplitude attenuation (by the air or the head shadowing, see Section 1.4), but the associated

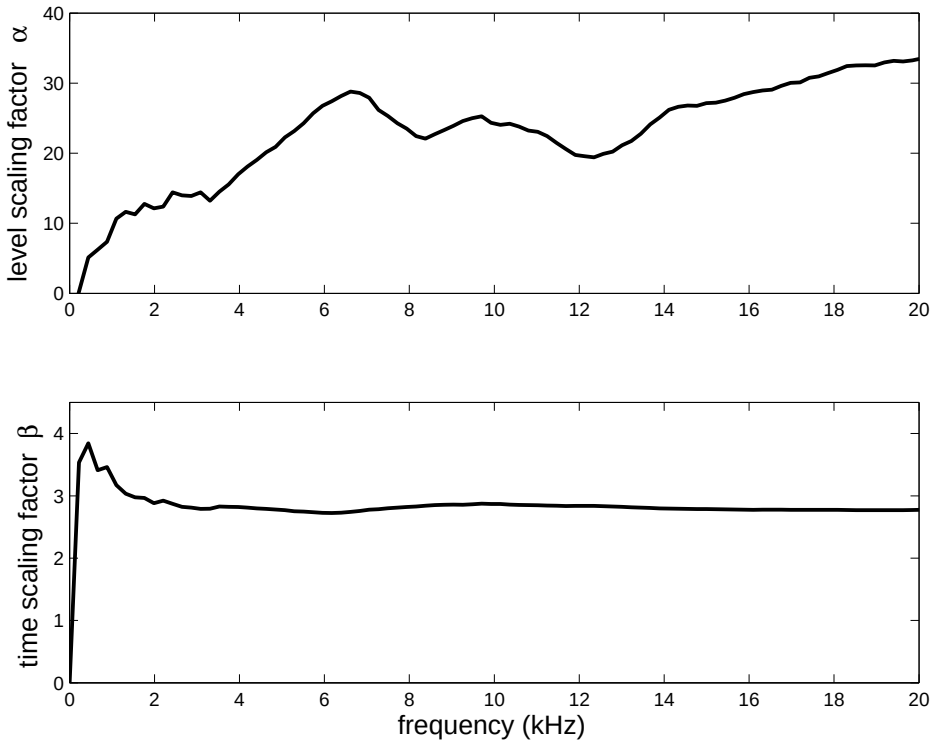


Figure 7.2: Frequency-dependent scaling factors: α (top) and β (bottom).

signals exhibit phase ambiguities (see below). In contrast, low frequencies are not ambiguous, but are less sensitive to amplitude attenuation.

In the remainder, α and β are frequency-dependent scaling factors that encapsulate the head / ears morphology (for example, the β coefficient takes into account the fact that the head is not perfectly spherical). In our experiments, we use the mean of individual scaling factors over the 45 subjects of the CIPIC database. For each subject, we measure the interaural cues from the HRIRs and derive the individual scaling factors that best match the model – in the least-square sense – for all azimuths.

7.3.1 Interaural Time Differences (ITDs)

7.3.1.1 Theory

Let us first ignore the influence of the head (free-field case). We also consider that $\rho >> r$ (the head radius). In this case, the acoustic waves can be regarded as planes (spheres of infinite radius) when they reach the ears. By simple geometric considerations on Figure 7.1, and specifically relationships within right-angled triangles, we deduce

$$\begin{aligned} \rho_L &= \rho + \Delta_\rho, \\ \rho_R &= \rho - \Delta_\rho, \\ \Delta_\rho &= r\sin(\theta) \end{aligned}$$

thus

$$\mathrm{ITD}(\theta) = \frac{\rho_L - \rho_R}{c} = \frac{2\Delta_\rho}{c}$$

and finally

$$\mathrm{ITD}(\theta) = \frac{2r\sin(\theta)}{c} \tag{7.6}$$

where r is the head radius and c the sound celerity. This is similar to the "sine law" by von Hornbostel and Wertheimer [vHW20]. In fact, the head is an obstacle to acoustic waves propagation. In the case of head shadowing, the waves may propagate around the head (on Figure 7.1, although the acoustic waves are still propagating directly to the left (L) ear, they have to propagate around the head for θ radians to reach the right (R) ear). This leads to the $\sin(\theta)+\theta$ model by Woodworth and Schlosberg [WS54], generalized by Viste [Vis04]. However, according to Kuhn [Kuh77], from the theory of the diffraction of an harmonic plane wave by a sphere (the head), the ITDs should be proportional to $\sin(\theta)$, the proportionality coefficient $2r/c$ for low frequencies and $3r/c$ for high frequencies giving good results.

7.3.1.2 Practice

We propose to generalize the Kuhn's model by taking into account the inter-subject variation and the full-frequency band. The ITD model is then expressed as

$$\mathrm{ITD}(\theta, f) = \beta(f) r \sin(\theta)/c \tag{7.7}$$

where β is the average scaling factor that best suits our model, in the least-square sense, for each listener of the CIPIC database (see Figure 7.2), r denotes the head radius, and c is the sound celerity. The measured ITDs come from the phases of the HRTFs. Thanks to the observations by Kuhn, we know that the ITDs do no really depend on the frequency f thus, from Equation (7.3), the phase difference Δ_ϕ should be proportionnal to the frequency, and should appear in phase spectra as lines. However, as shown in Figure 7.3, in these spectra the measured phase is wrapped, and it is important to use the unwrapped version to find β. The overall error of this model over the CIPIC database is 0.052ms (thus comparable to the 0.045ms error of the model by Viste). The average model error and inter-subject variance are depicted in Figure 7.4.

Practically, our model is easily invertible, which is suitable for sound localization, contrary to the $\sin(\theta)+\theta$ model by Viste which introduced mathematical errors at the extreme azimuths (see [40]).

7.3.2 Interaural Level Differences (ILDs)

7.3.2.1 Theory

As shown in Section 1.4, for sound waves propagating in the air in the free-field case, the sound intensity I is inversely proportional to ρ^2 (inverse square law). As a consequence, we have

$$\mathrm{ILD} = 20\log_{10}\left(\frac{A_L}{A_R}\right) = 10\log_{10}\left(\frac{I_L}{I_R}\right) = 10\log_{10}\left(\frac{{\rho_R}^2}{{\rho_L}^2}\right) = \frac{20}{\log(10)}\log\left(\frac{\rho_R}{\rho_L}\right)$$

that is, with $C = 20/\log(10)$,

$$\mathrm{ILD} = C\log\left(\frac{\rho_R}{\rho_L}\right) = C\log\left(\frac{\rho-\Delta_\rho}{\rho+\Delta_\rho}\right) = C\sum_{n=0}^{\infty}\frac{-2}{2n+1}\left(\frac{\Delta_\rho}{\rho}\right)^{2n+1}. \tag{7.8}$$

For $\Delta_\rho << \rho$, order 1 is a good approximation, thus the ILD (in dB) is proportional to the ITD, and with Equations (7.7) and (7.8) (with $n=0$), we have

$$\mathrm{ILD} = K\cdot\mathrm{ITD} \quad \text{with} \quad K = \frac{-20c}{\log(10)\rho} \tag{7.9}$$

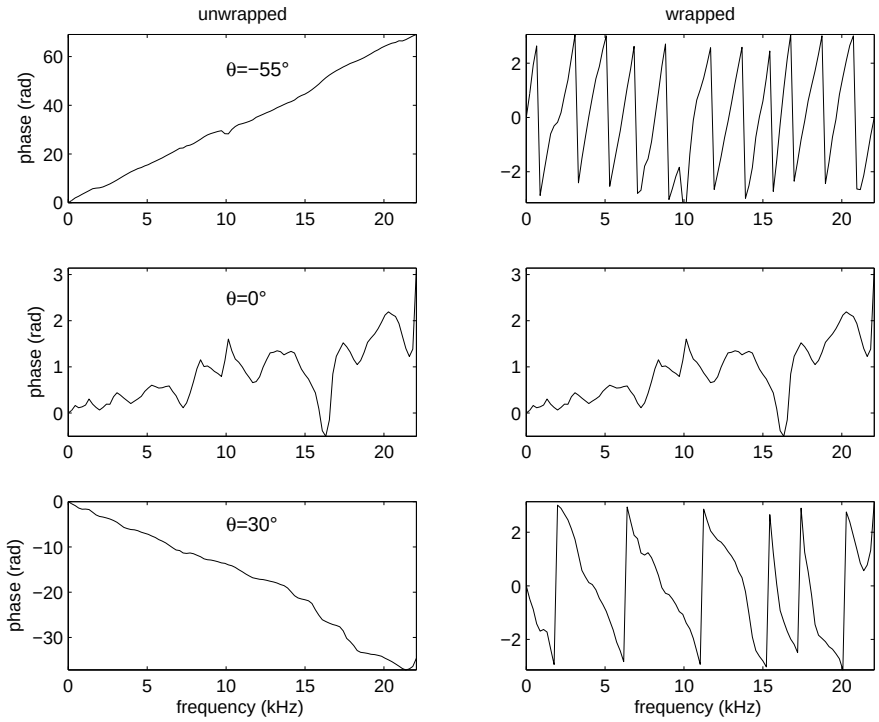


Figure 7.3: Unwrapped (left) and wrapped (right) phase spectra measured from the HRTFs at different azimuths.

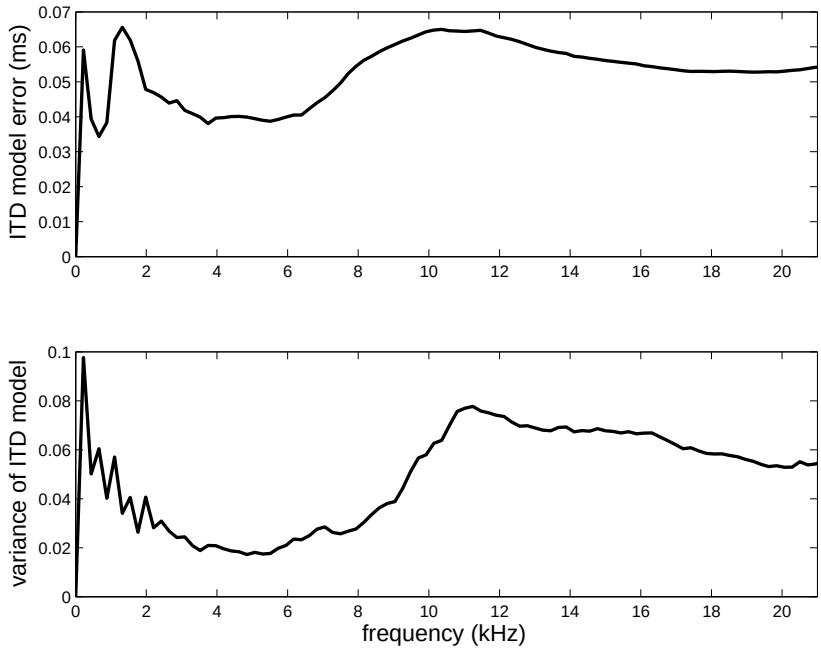


Figure 7.4: Average ITD model error (top) and inter-subject variance (bottom) over the CIPIC database.

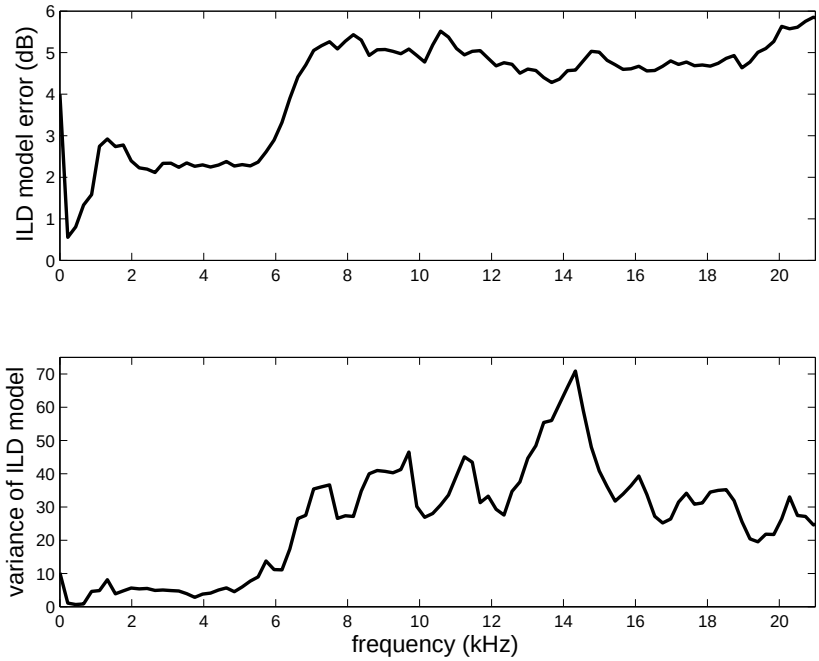


Figure 7.5: Average ILD model error (top) and inter-subject variance (bottom) over the CIPIC database.

where K depends neither on the azimuth θ nor on the frequency f.

7.3.2.2 Practice

In practice, ρ is fixed and the constant K can be calculated. However, in the real (non free field) case, the characteristics of the head / ears make it depend on the source frequency f. After Viste [Vis04], the ILDs can be expressed as functions of $\sin(\theta)$, thus leading to another sinusoidal model

$$\mathrm{ILD}(\theta, f) = \alpha(f)\sin(\theta) \tag{7.10}$$

where α is the average scaling factor that best suits our model, in the least-square sense, for each listener of the CIPIC database (see Figure 7.2). The overall error of this model over the CIPIC database for all subjects, azimuths, and frequencies is of 4.29dB. The average model error and inter-subject variance are depicted in Figure 7.5.

7.4 Single Source

Let us consider first the case of a single source.

7.4.1 Spatialization

We first study the effects of the distance ρ, then we consider it as fixed and focus on the azimuth θ.

7.4.1.1 Distance

As seen in Section 1.4, the sound coming from a (fixed) source situated at distance ρ will reach the listener with a delay $\Delta_t = \rho/c$ and with an attenuation resulting from two phenomena. First, because of the inverse square law the amplitude spectrum is inversely proportional to ρ. Second, because of the air attenuation, for a frequency f the magnitude spectrum has to be lowered by $\mathcal{A}(\rho, f)$dB (see Equation (1.46)). We simulate the distance by changing the magnitude spectrum of the source S to

$$X = S \cdot \text{HRTF}(0) \cdot \frac{\rho_0}{\rho} \cdot 10^{-\mathcal{A}(\rho-\rho_0,f)/20} \tag{7.11}$$

where ρ_0 is some reference distance where the HRTFs have been measured (typically, $\rho_0 = 1$m). The delay has been omitted since it can efficiently be simulated by delaying the short-term spectra in time.

7.4.1.2 Azimuth

There are two possibilities for spatializing the source to some azimuth θ.

HRIR-Based Method. The first possibility is to use some HRIRs functions (*e.g.* the HRIRs of the KEMAR manikin provided – only for some azimuths – together with the CIPIC database), and simply to apply Equations (7.1) and (7.2) to obtain the left x_L and right x_R binaural signals from the source signal s. We use this technique only to generate reference signals.

Cue-Based Method. The second possibility is again spectral. From the spectrum X given by Equation (7.11), we compute the pair of left and right spectra

$$X_L(t,f) = X(t,f) \cdot 10^{+\Delta_a(f)/2} e^{+j\Delta_\phi(f)/2}, \tag{7.12}$$

$$X_R(t,f) = X(t,f) \cdot 10^{-\Delta_a(f)/2} e^{-j\Delta_\phi(f)/2} \tag{7.13}$$

where the spatial cues are given for any azimuth by Equations (7.3), (7.4), (7.7), (7.10), and are divided equally because of the symmetric role of the two ears. Finally, the signals x_L and x_R are computed from their short-term spectra X_L and X_R. In practice, we use a Hann window of size $N = 2048$ with an overlap factor of 50%. We reach a remarkable spatialization realism through informal listening tests.

7.4.2 Localization

7.4.2.1 Distance

The frequency-independent inverse square law has no effect on the sound timbre. But when a source moves far from the listener, the high frequencies are more attenuated by the air than the low frequencies. Thus the sound spectrum changes with the distance. More precisely, the spectral centroid moves towards the low frequencies as the source moves away from the observer. The related perceptive brightness is an important distance cue.

As a reference signal for distance estimation, we use a Gaussian white noise spatialized at azimuth zero. The distance estimation relies on the quantification of the spectral changes during the sound propagation in the air. To estimate the amplitude spectrum, we first estimate the power spectral density of the noise using the Welch method (see Section 6.2). In our experiments, we consider $L = 21$ frames of $N = 2048$ samples, with an overlap factor of 50% (and with a CD-quality sampling rate of 44.1kHz,

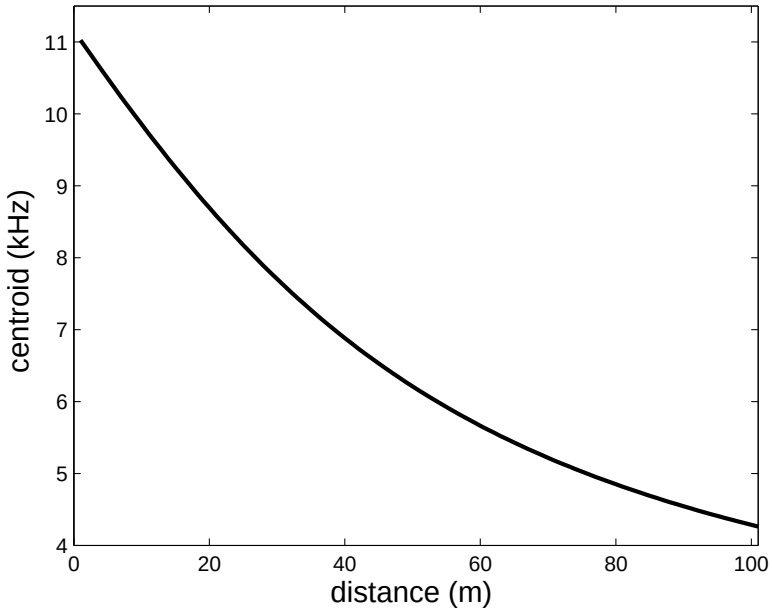


Figure 7.6: Spectral centroid (related to perceptive brightness) as a function of distance at $T_c = 20^\circ$ Celsius temperature, $H_r = 50\%$ relative humidity, and $P_S = 1$ atm atmospheric pressure (for white noise played at CD quality, $F_s = 44100$Hz).

thus the corresponding sound segment has a length < 0.5s). Then we use the resulting amplitude spectrum to compute the spectral centroid (see Equation (2.6)).

We know the reference distance since the CIPIC speakers were positioned on a $\rho_0 = 1$m radius hoop around the listener. By inverting the logarithm of the function of Figure 7.6, obtained thanks to Equations (1.46), (1.47), and (2.6), we can propose a function to estimate the distance from a given spectral centroid

$$\rho(\log(\mathcal{C})) = -38.89044\mathcal{C}^3 + 1070.33889\mathcal{C}^2 - 9898.69339\mathcal{C} + 30766.67908 \tag{7.14}$$

given for the air at $T_c = 20^\circ$ Celsius temperature, $H_r = 50\%$ relative humidity, and $P_s = 1$ atm pressure. Up to 25m, the maximum distance error is theoretically less than 4mm, if the noise power spectral density is known. However, if the amplitude spectrum has to be estimated using Equation (6.7), then the error is greater, though very reasonable until 50m. Figure 7.7 shows the results of our simulations for Gaussian white noise spatialized at various distances in the $[0, 100]$m range.

7.4.2.2 Azimuth

Given the short-term spectra of the left (X_L) and right (X_R) channels, we can measure the ILD and ITD for each time-frequency bin with

$$\text{ILD}(t,f) = 20\log_{10}\left|\frac{X_L(t,f)}{X_R(t,f)}\right|, \tag{7.15}$$

$$\text{ITD}_p(t,f) = \frac{1}{2\pi f}\left(\angle\frac{X_L(t,f)}{X_R(t,f)} + 2\pi p\right). \tag{7.16}$$

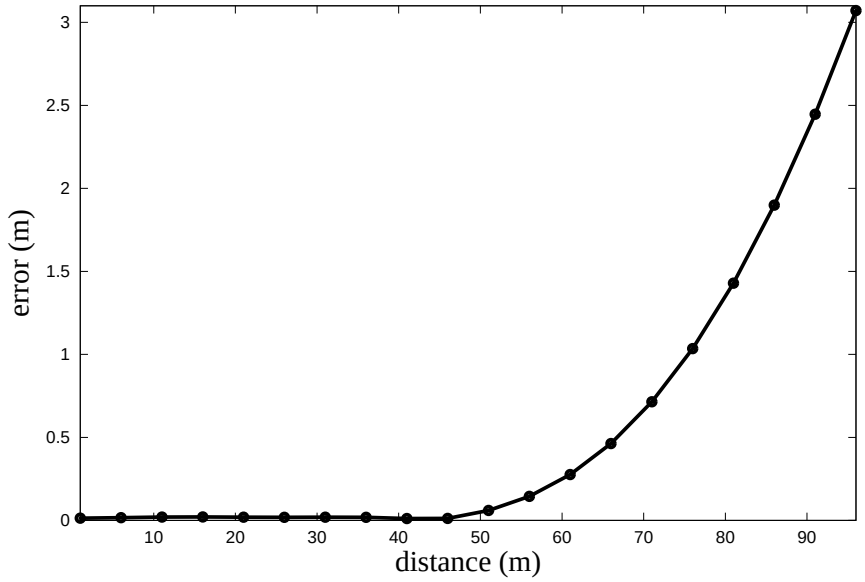


Figure 7.7: Absolute error of the localization of the distance from Gaussian white noise spatialized at various distances.

The coefficient p outlooks that the phase is determined up to a modulo 2π factor. In fact, the phase becomes ambiguous above 1500Hz, where the wavelength is shorter than the diameter of the head.

These two binaural cues are used in the DUET method [ÖYR04] to build a two-dimensional histogram prior to source separation in the "degenerated case" (see Section 7.5). However, these two cues are not independent. They are both functions of the azimuth θ.

Obtaining an estimation of the azimuth based on the ILD information (see Equation (7.15)) is just a matter of inverting Equation (7.10):

$$\theta_L(t,f) = \arcsin\left(\frac{\mathrm{ILD}(t,f)}{\alpha(f)}\right). \tag{7.17}$$

Similarly, using the ITD information (see Equation (7.16)), to obtain an estimation of the azimuth candidate for each p, we invert Equation (7.7):

$$\theta_{T,p}(t,f) = \arcsin\left(\frac{c\cdot \mathrm{ITD}_p(t,f)}{r\cdot\beta(f)}\right). \tag{7.18}$$

The $\theta_L(t,f)$ estimates are more dispersed, but not ambiguous at any frequency, so they are exploited to find the right modulo coefficient p that unwraps the phase. Then the $\theta_{T,p}$ that is nearest to θ_L is validated as the final θ estimation for the considered frequency bin, since it exhibits a smaller deviation:

$$\hat{\theta}(t,f) = \theta_{T,m}(t,f) \quad \text{with} \quad m = \arg\min_p \left|\theta_L(t,f) - \theta_{T,p}(t,f)\right|. \tag{7.19}$$

Practically, the choice of p can be limited among two values ($\lceil p_r \rceil$, $\lfloor p_r \rfloor$), where

$$p_r = \left(f\cdot \mathrm{ITD}(\theta_L,f) - \frac{1}{2\pi}\angle\frac{X_L(t,f)}{X_R(t,f)}\right). \tag{7.20}$$

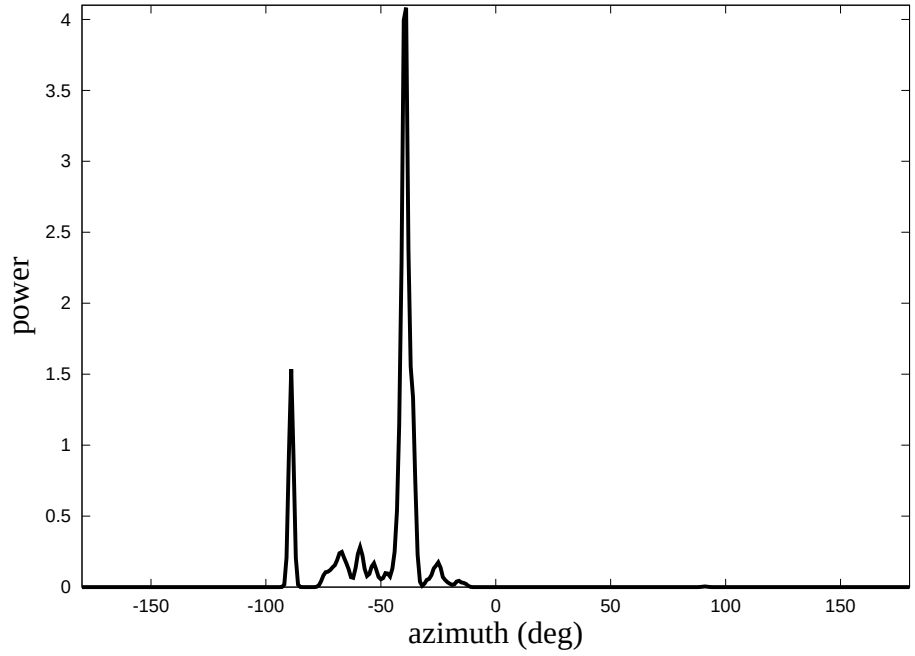


Figure 7.8: Histogram obtained with a source at azimuth -45°. One can clearly see two important local maxima (peaks): one around azimuth -45°, the other at azimuth -90°. The first (and largest) one corresponds to the sound source; the second one is a spurious peak resulting from extreme ILDs.

An estimate of the azimuth of the source can be obtained as the peak in an energy-weighted histogram (see [40]). More precisely, for each frequency bin of each discrete spectrum, an azimuth $\hat{\theta}$ is estimated and the power corresponding to this bin is accumulated in the histogram at this azimuth (possibly using some smoothing Gaussian distribution around $\hat{\theta}$). For the corresponding bin frequency f, the power $|X(f)|^2$ is estimated by

$$|X|^2 \approx |X_L \cdot X_R| \tag{7.21}$$

because of Equations (7.12) and (7.13). This should make the histogram unbiased: rotating the source is equivalent to a shift of the θ axis of the histogram, but then its energy remains unchanged. However, in practice a bias was observed and will require further research.

For specific applications such as source separation, in addition to the magnitude we must also estimate the phase. For this purpose, since we know (an estimate of) θ and thus can derive Δ_ϕ and Δ_a from Equations (7.3) and (7.4), we propose to consider the loudest – supposedly most reliable – left or right channel and invert, respectively, either Equation (7.12) or (7.13) to obtain (an estimate of) X, the spectrum corresponding to the source played at azimuth zero.

Thus, we obtain a power histogram as shown in Figure 7.8, where the histogram is the result of the localization of a Gaussian white noise of 0.5s spatialized at azimuth -45°. On this figure, we can clearly see two important local maxima (peaks), one around azimuth -45°, the other at azimuth -90°. The first (and largest) one corresponds to the sound source; the second one is a spurious peak resulting from extreme ILDs (a problem we have to solve in our future research). For our localization tests, we spatialized a Gaussian white noise using convolutions with the HRIRs of the KEMAR manikin, since they were not part of the database used for the learning of our model coefficients and thus should give results closer to those expected with a real – human – listener. Indeed, in our first experiments with real listeners (see Figure 7.9), the same trends as in Figure 7.8 were observed: a rather broader histogram

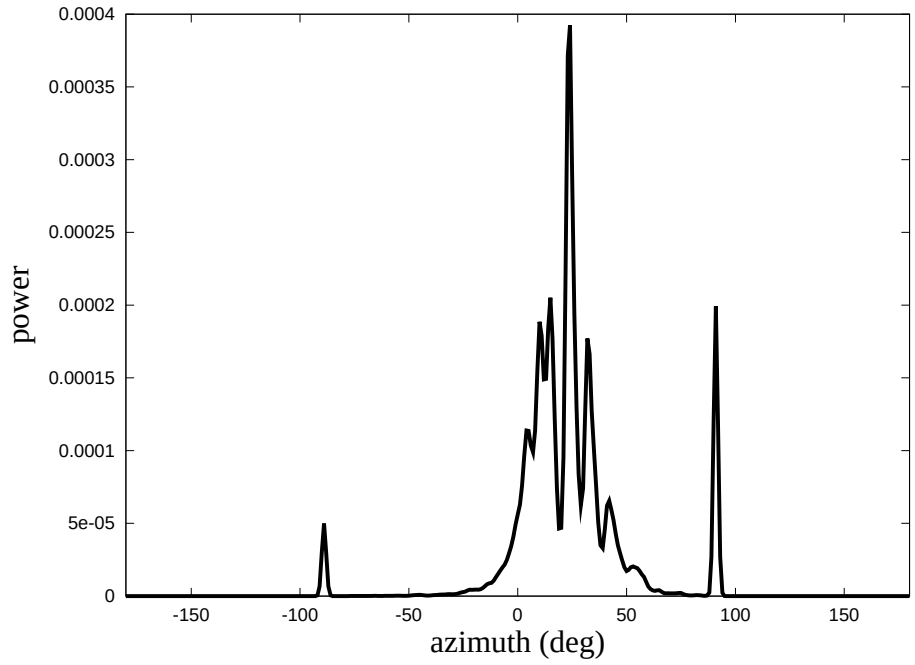


Figure 7.9: Histogram obtained with a real source positioned at azimuth 30° in a real room, with binaural signals recorded at the ears of the musician.

but still with a local maximum close to the azimuth of the sound source, plus spurious maxima at extreme azimuths $\pm 90^\circ$. To verify the precision of the estimation of the azimuth, we spatialized several noise sources at different azimuths in the horizontal plane, between -80° and $+80^\circ$, and we localized them using the proposed method. The results are shown in Figure 7.10. We observe that the absolute azimuth error is less than 5° in the $[-65, +65]^\circ$ range.

7.5 Source Separation

Musical sounds are most often polyphonic, that is with several sources possibly at different azimuths. With (at least) as many microphones as the number of sources, the source separation turns into a very classic mathematical problem. However, we are interested here in the case of many sources but only 2 recordings: one for each ear. This is the "degenerated" case (mathematically speaking).

To achieve the degenerated separation of a arbitrary number of sources given binaural mixtures, we consider any pair of sources $(s_k(t), s_l(t))$ as window-disjoint orthogonal (WDO). This means that their short-term spectra do not superpose, more precisely

$$\forall k \neq l, \quad S_k(t,f) \cdot S_l(t,f) = 0 \qquad (k,l = 1, \cdots, K) \tag{7.22}$$

where K is the number of sources in the mix. Experiments carried out for the DUET algorithm [ÖYR04] verify that speech signals are approximatively WDO. This seems confirmed in more complex (reverberant) conditions [SH08]. For musical signals however, this condition is likely to be more rarely satisfied.

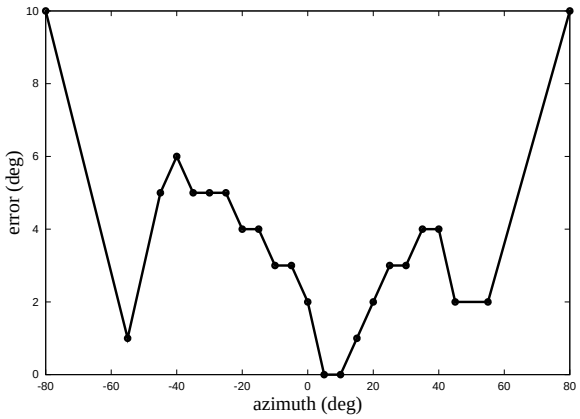


Figure 7.10: Absolute error of the localization of the azimuth from Gaussian white noise spatialized at different azimuths using convolutions with the HRIRs of the KEMAR manikin.

7.5.1 Gaussian Mixture Model

In theory, in the case of a single source all frequencies should give the same azimuth, exactly corresponding to the source position θ. However, in practice, the violation of the WDO assumption, the presence of noise and estimation errors make things a little more complicated. In fact, as a first approximation, we consider that the energy of the source is spread in the power histogram following a Gaussian distribution centered at the theoretical value θ. The Gaussian nature of the distribution is comforted by the well-known Central Limit Theorem as well as practical experiments. In this context, the ideal case is a Gaussian of mean θ and variance 0.

In the case of K sources, we then introduce a model of K Gaussians (K-GMM, order-K Gaussian mixture model)

$$P_K(\theta|\Gamma) = \sum_{k=1}^{K} \pi_k \, \phi_k(\theta|\mu_k,\sigma_k) \text{ with } \pi_k \geq 0 \text{ and } \sum_{k=1}^{K} \pi_k = 1 \tag{7.23}$$

where Γ is a multiset of K triples (π_k,μ_k,σ_k^2) that denotes all the parameters of the model; π_k, μ_k, and σ_k^2 indicate respectively the weight, the mean, and the variance of the k-th Gaussian component described mathematically by

$$\phi_k(\theta|\mu_k,\sigma_k) = \frac{1}{\sqrt{2\pi\sigma_k^2}} \exp\left(-\frac{(\theta-\mu_k)^2}{2\sigma_k^2}\right). \tag{7.24}$$

We are interested in estimating the architecture of the K-GMM, that is the number of sources K and the set of parameters Γ, to be able to setup the separation filtering.

7.5.2 Unmixing Algorithm

In the histogram $h(\theta)$, we observe local maxima which number provides an estimation of the number of sources in the mixture. The abscissa of the k-th local maximum reveals the location θ_k of the k-th source. However, in practice, to avoid spurious peaks, we must deal with a smooth version of the histogram and consider only significant local maxima – above the noise level. Informal experiments show that the estimated source number and location are rather good. This gives the model order K

and a first estimation of the means of the Gaussians (μ_k in Γ). This estimation can be refined and completed – with the variances σ_k^2 and the weights π_k – for example by the EM algorithm.

Expectation Maximization (EM) is a popular approach to estimate parameters in mixture densities given a data set x. The idea is to complete the observed data x with an unobserved variable y to form the complete data (x,y), where y indicates the index of the Gaussian component from which x has been drawn. Here, the role of x is played by the azimuth θ, taking values in the set of all discrete azimuths covered by the histogram. We associate θ with its intensity function $h(\theta)$ (the histogram). The role of y is played by $k \in \{1,\cdots,K\}$, the index of the Gaussian component θ should belong to.

The EM algorithm proceeds iteratively, at each iteration the optimal parameters that increase locally the log-likelihood of the mixture are computed. In other words, we increase the difference in log-likelihood between the current with parameters Γ and the next with parameters Γ'. This log-form difference, noted $Q(\Gamma',\Gamma)$, can be expressed as

$$\begin{aligned} Q(\Gamma',\Gamma) &= \sum_\theta h(\theta)\left(\mathcal{L}(\theta|\Gamma') - \mathcal{L}(\theta|\Gamma)\right) \quad \text{with} \\ \mathcal{L}(\theta|\Gamma) &= \log\left(P_K(\theta|\Gamma)\right). \end{aligned} \tag{7.25}$$

We can then reformulate $\mathcal{L}(\theta|\Gamma)$ like this:

$$\begin{aligned} \mathcal{L}(\theta|\Gamma) &= \log\left(\sum_k P_K(\theta,k|\Gamma)\right) \quad \text{with} \\ P_K(\theta,k|\Gamma) &= \pi_k\, \phi_k(\theta|\mu_k,\sigma_k). \end{aligned} \tag{7.26}$$

The concavity of the log function allows to lower bound the $Q(\Gamma',\Gamma)$ function using the Jensen's inequality. We can then write

$$Q(\Gamma',\Gamma) \geq \sum_\theta \sum_k h(\theta) P_K(k|\theta,\Gamma) \log\left(\frac{P_K(\theta,k|\Gamma')}{P_K(\theta,k|\Gamma)}\right) \tag{7.27}$$

where $P_K(k|\theta,\Gamma)$ is the posterior probability, the degree to which we trust that the data was generated by the Gaussian component k given the data; it is estimable with the Bayes rule

$$P_K(k|\theta,\Gamma) = \frac{P_K(\theta,k|\Gamma)}{P_K(\theta|\Gamma)}. \tag{7.28}$$

The new parameters are then estimated by maximizing the lower bound with respect to Γ:

$$\Gamma' = \arg\max_\gamma \sum_\theta \sum_k h(\theta) P_K(k|\theta,\Gamma) \log\left(P_K(\theta,k|\gamma)\right). \tag{7.29}$$

Increasing this lower bound results automatically in an increase of the log-likelihood, and is mathematically easier. Finally, the maximization of Equation (7.29) provides the following update relations (to be applied in sequence, because they modify – update – the current value with side-effects, thus the updated value must be considered in the subsequent relations):

$$\pi_k \leftarrow \frac{\sum_\theta h(\theta)\, P_K(k|\theta,\Gamma)}{\sum_\theta h(\theta)}, \tag{7.30}$$

$$\mu_k \leftarrow \frac{\sum_\theta h(\theta)\, \theta\, P_K(k|\theta,\Gamma)}{\sum_\theta h(\theta)\, P_K(k|\theta,\Gamma)}, \tag{7.31}$$

$$\sigma_k^2 \leftarrow \frac{\sum_\theta h(\theta)\, (\theta-\mu_k)^2\, P_K(k|\theta,\Gamma)}{\sum_\theta h(\theta)\, P_K(k|\theta,\Gamma)}. \tag{7.32}$$

The performance of the EM depends of the initial parameters. The first estimation parameter should help to get around likelihood local maxima trap. Our EM procedure operates as follows:

1. Initialization step
   - initialize K with the order of the first estimation
   - initialize the weights equally, the means according to the first estimation, and the variances with the data variance (for the initial Gaussians to cover the whole set of data):
   $$\pi_k = 1/K, \quad \mu_k = \theta_k, \text{ and } \sigma_k^2 = \text{var}(\theta)$$
   - set a convergence threshold ε
2. Expectation step
   - compute $P_K(k|\theta,\Gamma)$ with Equation (7.28)
3. Maximization step
   - compute Γ' from Γ with Equations (7.30), (7.31), and (7.32)
   - if $P_K(\theta|\Gamma') - P_K(\theta|\Gamma) > \varepsilon$ then $\Gamma \leftarrow \Gamma'$ and go back to the Expectation step else stop (the EM algorithm has converged).

Finally, to separate the sources, a spatial filtering identifies and clusters bins attached to the same source. Many methods, like DUET, separate the signals by assigning each of the time-frequency bins to one of the sources exclusively. We assume that several sources can share the power of a bin, and we attribute the energy according to a membership ratio – a posterior probability. The histogram learning with EM provides a set of parameters for the Gaussian distribution that characterizes each source. These parameters are then used to parameterize automatically a set of spatial Gaussian filters. In order to recover each source k, we select and regroup the time-frequency bins belonging to the same azimuth θ. We use the parameters issued from the EM-component number k, and the energy of the mixture channels is allocated to the (left and right) source channels according to the posterior probability. More precisely, we define the following mask for each source:

$$M_k(t,f) = P_K(k|\theta(t,f),\Gamma) \tag{7.33}$$

if $10\log_{10}|\phi_k(\theta(t,f)|\mu_k,\sigma_k)| > L_{\text{dB}}$, and 0 otherwise. This mask limits the fact that the tail of a Gaussian distribution stretches out to infinity. Below the threshold L_{dB} (expressed in dB, and set to -20 in our experiments), we assume that a source of interest does not contribute anymore. For each source k, the pair of short-term spectra can be reconstructed according to

$$S_L(t,f) = M_k(t,f) \cdot X_L(t,f), \tag{7.34}$$
$$S_R(t,f) = M_k(t,f) \cdot X_R(t,f). \tag{7.35}$$

7.5.3 Experimental Results

First, we synthesized binaural signals using the HRIR-based technique (see above), then the signals created individually were mixed into a single binaural signal.

A result of demixing is depicted in Figure 7.11 for a two-instrument mixture: xylophone at -55° and horn at -20°; their original spectrograms are shown in Figure 7.12. In the time domain, the

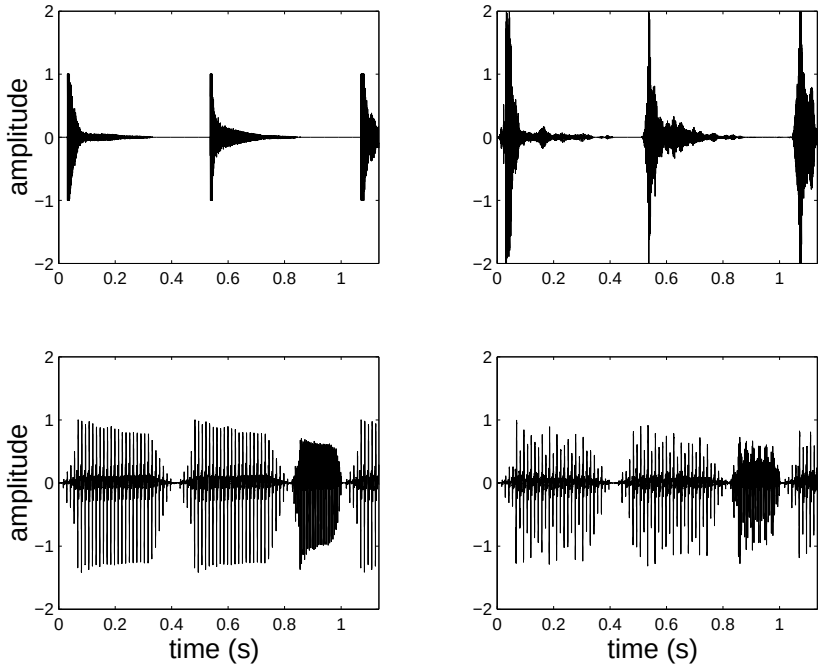


Figure 7.11: Waveforms of the demixtures (on the right, originals being on the left): xylophone ($-55°$) (top) and horn ($30°$) (bottom).

xylophone rhythm is respected, its signal looks amplified and its shape is preserved. Perceptively, the demixed xylophone is very similar to the original one. Also, for the horn, we must tolerate some interference effects, and the spectrograms are partly damaged. A portion of energy was absorbed by an unwanted source generated from interferences. We also conducted tests on speech samples. The reconstruction quality was good, much better than for long-distance telephone lines. Figure 7.13 shows the power histogram for the localization of four instruments in a binaural mixture. This histogram, here of size 65, was built using FFTs of $N = 2048$ samples with an overlap of 50%. Figure 7.14 shows the (normalized) Gaussian mixture model associated to this histogram. In the near future we plan to enhance these results by applying the EM algorithm on the raw azimuth estimates $\hat{\theta}(t,f)$ instead of the data stored in the histogram.

7.6 Multi-Channel Diffusion

In binaural listening conditions using headphones, the sound from each earphone speaker is heard only by one ear. But when the sound is diffused using loudspeakers, the encoded spatial cues may be affected by cross-talk signals (see C_{LR} and C_{RL} in Figure 7.15). In a setup with many speakers we use the classic pair-wise paradigm [Cho71], consisting in choosing for a given source only the two speakers closest to it (in azimuth): one at the left of the source, the other at its right.

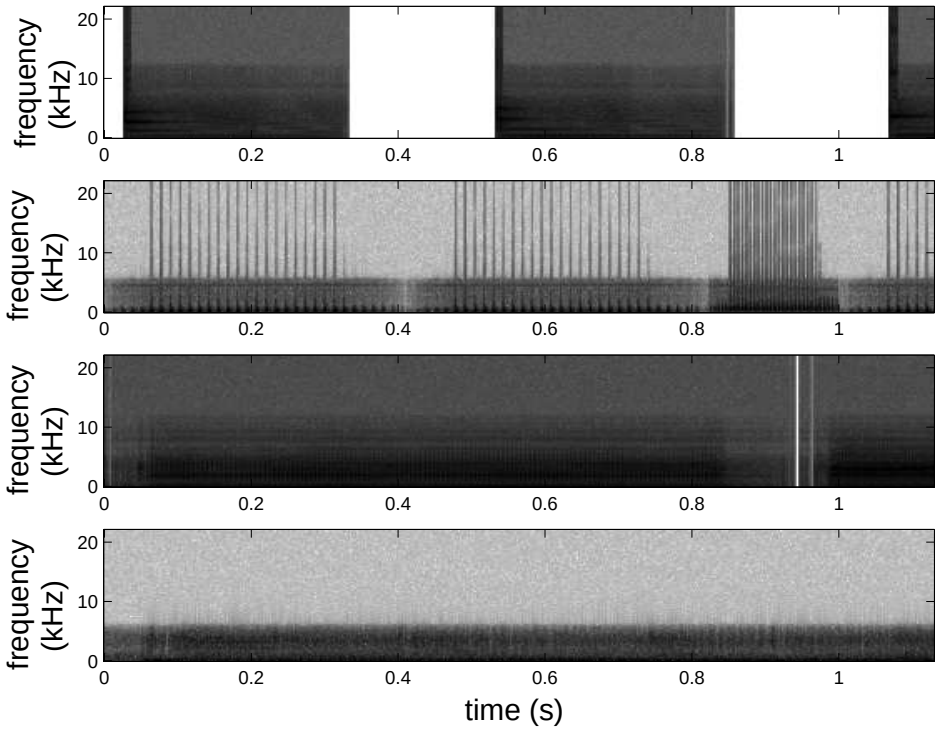


Figure 7.12: Spectrograms of the four sources, from top to bottom: xylophone, horn, kazoo, and electric guitar.

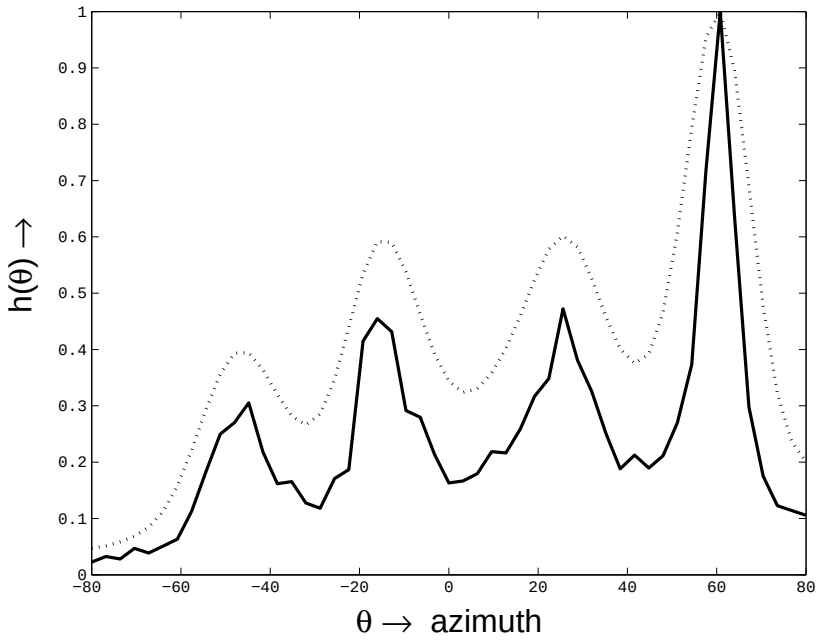


Figure 7.13: Histogram (solid line) and smoother version (dashed line) of the 4-source mix: xylophone at $-55°$, horn at $-20°$, kazoo at $30°$, and electric guitar at $65°$.

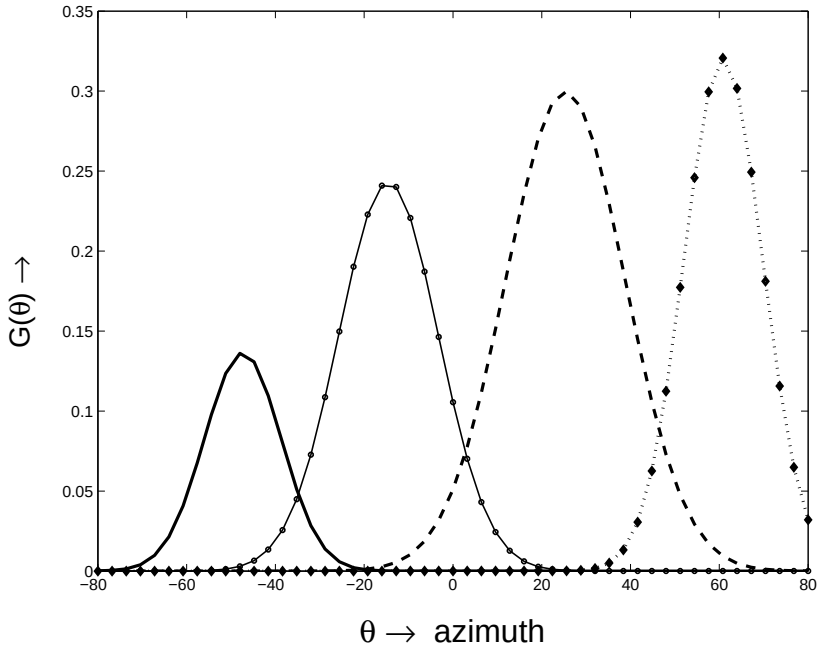


Figure 7.14: GMM (normalized) for the histogram of the 4-source mix.

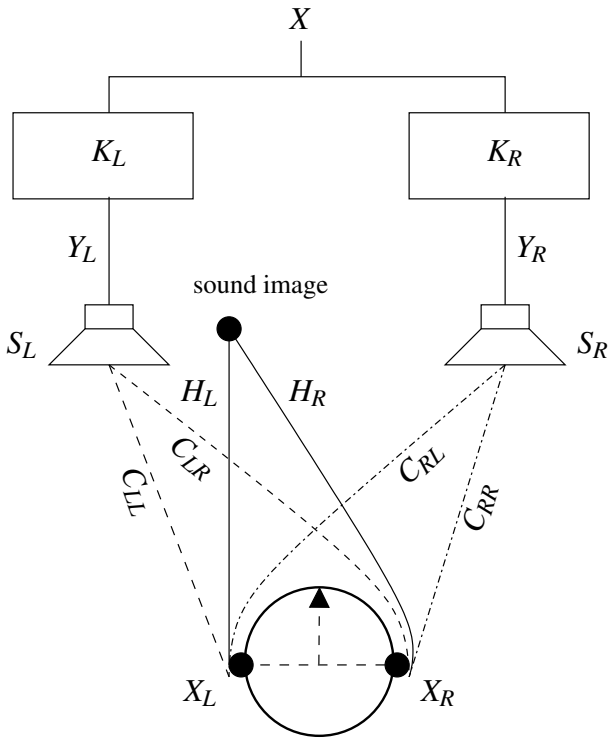


Figure 7.15: Stereophonic loudspeaker display.

7.6.1 Vector Base Amplitude Panning (VBAP)

The classic vector base amplitude panning (VBAP) method [Pul97] is the use a geometric approach. The mixing coefficients for the left (K_L) and right (K_R) speakers are given by

$$\begin{pmatrix} K_L \\ K_R \end{pmatrix} = \begin{pmatrix} L_x & R_x \\ L_y & R_y \end{pmatrix}^{-1} \cdot \begin{pmatrix} p_x \\ p_y \end{pmatrix} \tag{7.36}$$

where p, L, and R are vectors representing the positions of the source, left speaker, and right speaker, respectively. The mixing coefficients have to be normalized afterwards by $\sqrt{K_L^2 + K_R^2}$ in order to have a constant power, that is

$$K_L^2 + K_R^2 = 1. \tag{7.37}$$

During the diffusion, the short-term spectra (Y_L, Y_R) of the signals to feed the left and right speakers speakers are obtained by multiplying the short-term spectrum X with K_L and K_R, respectively

$$Y_L(t,f) = K_L \cdot X(t,f), \tag{7.38}$$
$$Y_R(t,f) = K_R \cdot X(t,f). \tag{7.39}$$

7.6.2 Spectral Diffusion

With the pair-wise paradigm (see above), we propose to adopt a transaural approach using our binaural model. In a stereophonic display, the sound from each loudspeaker is heard by both ears. Thus, the stereo sound is filtered by a matrix of four transfer functions between the two (left and right) loudspeakers and the two (left and right) ears (see Figure 7.15)

$$C = \begin{pmatrix} C_{LL} & C_{LR} \\ C_{RL} & C_{RR} \end{pmatrix}. \tag{7.40}$$

The best panning coefficients for the pair of speakers to match the binaural signals at the ears (see Equations (7.12) and (7.13)) are then given by

$$K_L(t,f) = -(H_R \cdot C_{RL} - H_L \cdot C_{RR})/D, \tag{7.41}$$
$$K_R(t,f) = -(H_L \cdot C_{LR} - H_R \cdot C_{LL})/D \tag{7.42}$$

with the determinant of the mixing matrix C being computed as

$$D = C_{LL} \cdot C_{RR} - C_{LR} \cdot C_{RL}. \tag{7.43}$$

All the acoustic paths (C_{LL}, C_{LR}, C_{RL}, C_{RR}, H_L, and H_R) are artificially given by our binaural model (see Section 7.3), and all the multiplications and divisions are element-wise. In extreme cases where D is close to zero, the mixing matrix is ill-conditioned and the solution becomes unstable. Fortunately, in practice it happens only at frequency $f = 0$ (where we can safely set $D(0) = 1$), or when the two speakers are at the same position (which is impossible).

7.6.3 Comparison of the Panning Coefficients

We used the speaker pair $(-30^\circ, +30^\circ)$ to compute the panning coefficients at any position (between the speakers) with the two techniques: VBAP and our approach. VBAP was elaborated under the

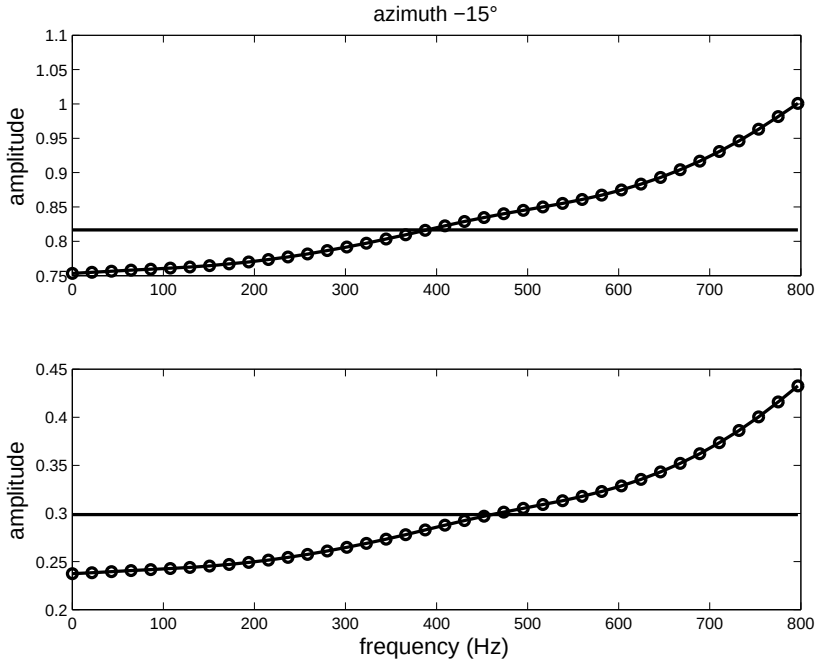


Figure 7.16: Amplitude of the panning coefficients from VBAP (plain) and our approach (dotted), for the left (top) and right (bottom) channels of the panning pair for $-15°$, in the $[0,800]$Hz band.

assumption that the incoming sound is different only in amplitude, which holds for frequencies up to 600Hz. We restrict our comparisons to the $[0,800]$Hz frequency band.

The panning coefficients of the two approaches are very similar until 600Hz (see Figure 7.16), and can differ significantly above. In fact, our coefficients are complex values, and their imaginary parts can contribute in a significant way (see Figure 7.17). Generally, inter-channel differences (*e.g.* ILD, ITD) are perceptually more relevant than absolute values. Given the left and right panning coefficients, K_L and K_R, we compute the *panning level difference* (PLD)

$$\mathrm{PLD} = 20\log_{10}\left|\frac{K_L}{K_R}\right|. \tag{7.44}$$

We computed the absolute difference between the PLDs of both VBAP and our approach. The maximal PLD difference (in the considered frequency band) has a linear trend, and its maximum does not exceed 3dB. Thus, the two approaches seem to be consistent in the $[0,800]$Hz band (see Figure 7.18). For higher frequencies, the control of both amplitude and phase of the new approach should yield better results, as confirmed perceptively in our preliminary and informal listening tests.

7.7 RetroSpat: Retroactive Spatializer

Composers of acousmatic music conduct different stages through the composition process, from sound recording (generally stereophonic) to diffusion (multiphonic). Acousmatic music is then played on an *acousmonium*: an orchestra of loudspeakers, controlled by the interpreter from a special (un)mixing console. The originality of such a device is to map the two stereo channels at the entrance to 8, 16, or even hundreds of channels of projection. Each channel is controlled individually by knobs and

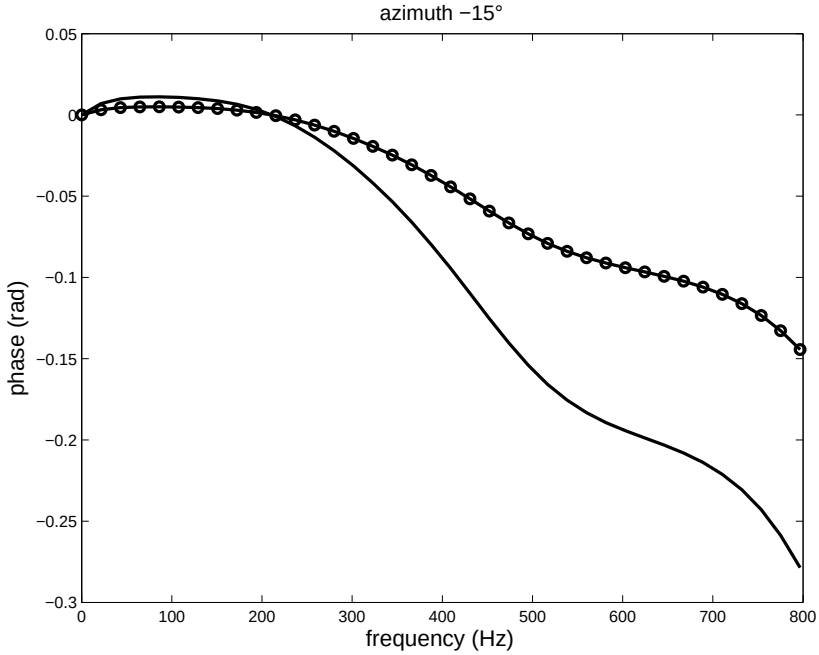


Figure 7.17: Phase of the panning coefficients from our approach, for the left (dotted) and right (plain) channels of the panning pair for -15°, in the $[0, 800]$Hz band.

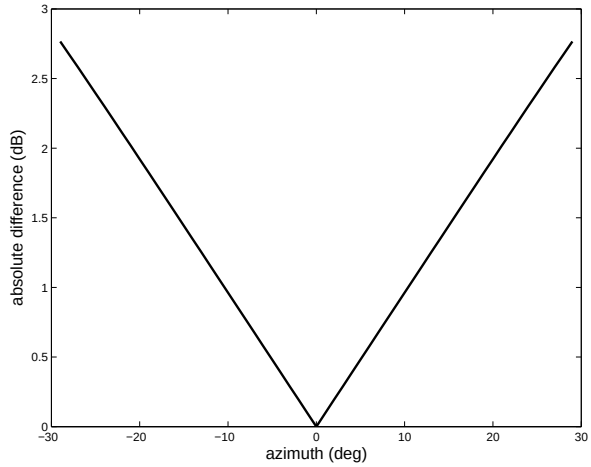


Figure 7.18: Maximum difference per azimuth between PLDs of VBAP and the proposed method in the $[0, 800]$Hz band.

equalization systems. The channel is assigned to one or more loudspeakers positioned according to the acoustical environment and the artistic strategy. During live interpretation, the interpreter wants to interfere decisively on the spatialization of pre-recorded sonorities. With two hands, this becomes hardly tractable with many sources or speakers. Thus, together with Mouba, Mansencal, and Rivet, we propose RetroSpat [45], a system for the semi-automatic diffusion of acousmatic music, with a feedback from the miniature microphones encased in headphones carried by the interpreter, to

- automatically perform the calibration of the room (estimating the location – azimuth and distance – of every loudspeaker), by "listening to the room" (see Section 7.4.2);
- diffuse sound sources on the detected loudspeaker array using our multi-diffusion method (see Section 7.6.2).

In the near future, the system should be able to separate the source present in the binaural input (see Section 7.5) in order to manipulate them individually ("active listening", see Section 8.3). It should also "listen to the result" recorded at the ears of the interpreter. Thus, the system could adapt the diffusion matrix to optimally reconstruct the target binaural signal at the listener's ears from the array of loudspeakers.

The RetroSpat system is being implemented as a real-time musical software under the GNU General Public License (GPL). The actual implementation is based on C++, Qt4[1], JACK[2], FFTW[3] and works on the Linux and MacOS X operating systems. Currently, RetroSpat implements the described methods (*i.e.* localization and spatialization) in two different modules: *RetroSpat Localizer* for speaker setup detection and *RetroSpat Spatializer* for the spatialization process. We hope to merge the two functionalities in one unique software soon.

RetroSpat Localizer. The automatic detection of the positions (azimuth and distance) of the speakers connected to the soundcard is of great importance to adapt to new speaker setups. Indeed, it will be one of the first actions of the interpreter in a new environment. For room calibration, the interpreter carries headphones with miniature microphones encased in earpieces (Sennheiser KE4-211-2 microphones have been inserted in standard headphones). The interpreter orients the head towards the desired zero azimuth. Then, each speaker plays in turn a Gaussian white noise. The binaural signals recorded from the ears of the musician are transferred to the computer running RetroSpat Localizer. Each speaker is then localized in azimuth and distance. The suggested configuration can be adjusted or modified by the interpreter according to the rooms characteristics (see Figure 7.19).

RetroSpat Spatializer. For sound spatialization, mono sources are loaded in RetroSpat, together with the loudspeaker-array configuration. The snapshot of Figure 7.20 depicts a 7-source mix of instruments and voices (note icons), in a 6-speaker front-facing configuration (loudspeaker icons), obtained from RetroSpat Localizer. During the diffusion, the musician can interact individually with each source of the piece, change its parameters (azimuth and distance), or even remove / insert a source from / into the scene. In this early version, the interaction with RetroSpat is provided by a mouse controller. In the future, we might evolve to source control through gesture. Thanks to an efficient implementation using the JACK sound server, RetroSpat can diffuse properly simultaneous sources even within the same speaker pair (see Figure 7.20, three sources in speaker pair $(2,3)$). All

[1] see URL: `http://trolltech.com/products/qt`
[2] see URL: `http://jackaudio.org`
[3] see URL: `http://www.fftw.org`

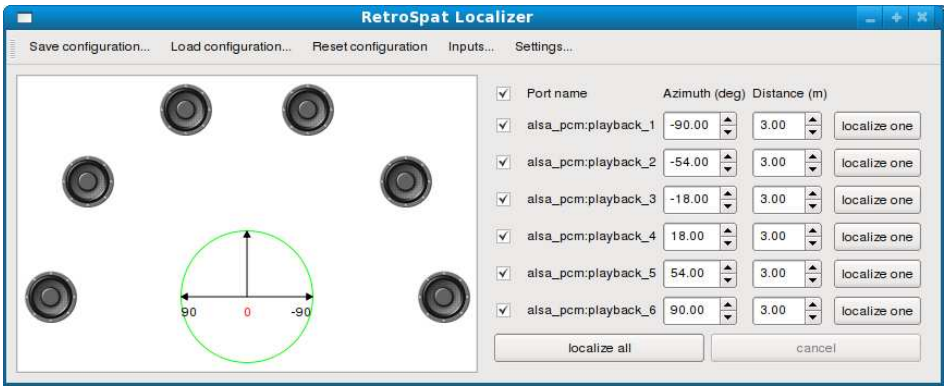


Figure 7.19: RetroSpat Localizer graphical user interface with a 6-speaker configuration.

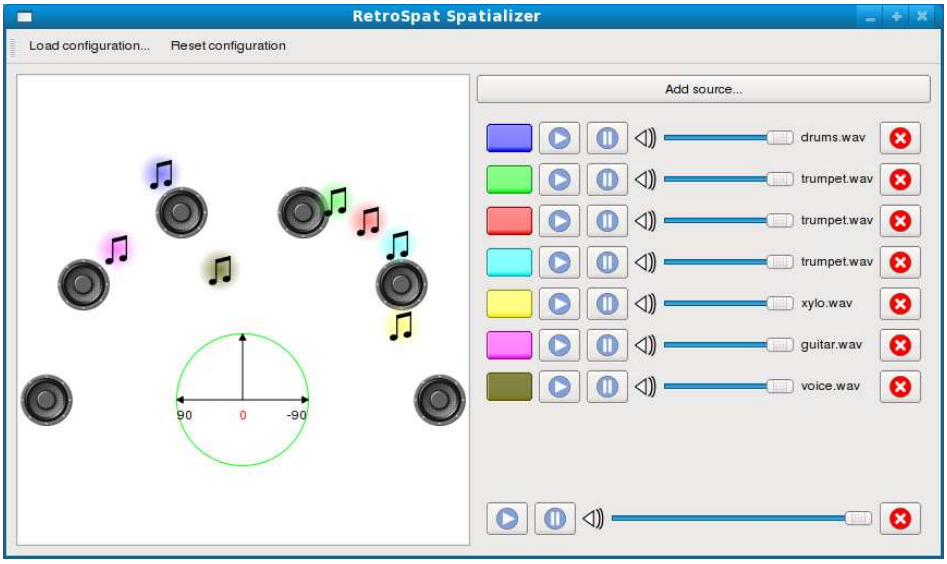


Figure 7.20: RetroSpat Spatializer graphical user interface, with 7 sources spatialized on the speaker setup presented on Figure 7.19.

the speaker pairs have to stay in synchrony. To avoid sound perturbation, the Qt-based user interface runs in a separate thread with less priority than the core signal processing process. We tested RetroSpat on a MacBook Pro, connected to 8 speakers, through a MOTU 828 MKLL soundcard, and were able to play several sources without problems. However, further testing is needed to assess scalability limits.

Chapter 8

Research Perspectives

In this last chapter, we describe important research perspectives: the extraction of sound entities from polyphonic music, the possible use of watermarking techniques to ease this difficult task, and the ultimate goal: "active listening".

Each of these research perspectives could be the subject of a Ph.D. thesis. In the remainder of this chapter we describe them and show promising early results, although we are conscious that much work is still necessary, as well as an updated bibliography.

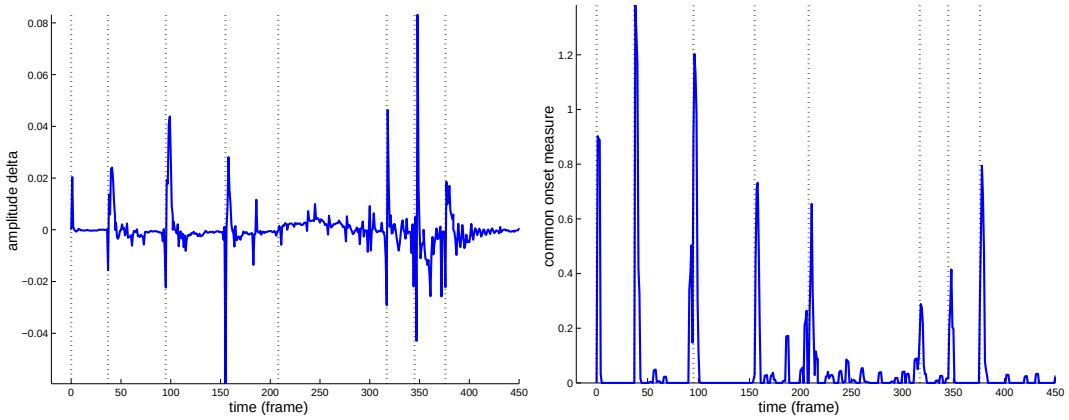
8.1 Sound Entities and Music Transcription

From a perceptual point of view, some partials belong to the same *sound entity* if they are perceived by the human auditory system as a unique sound when played together. There are several criteria that lead to this perceptual fusion. After Bregman [Bre90], we consider

- the common onsets/offsets of the spectral components (Ph.D. of Lagrange [Lag04]);
- the spectral structure of the sound, taking advantage of harmonic relations (Master's thesis of Cartier [Car03], Ph.D. of Lagrange [Lag04]);
- the correlated variations of the time evolutions of these spectral parameters (Ph.D. of Lagrange [Lag04], Ph.D. of Raspaud [Ras07]);
- and the spatial location, estimated by a localization algorithm (ongoing Ph.D. of Mouba, see Chapter 7).

All these criteria allow us to classify the spectral components. Since this is done according to the perception, the classes we obtain should be the sound entities of the auditory scene. And the organization of these sound entities in time should give the musical structure. Music transcription is then possible by extracting musical parameters from these sound entities. But an advantage over standard music information retrieval (MIR) approaches is that here the sound entities are still available for transformation and resynthesis of a modified version of the music.

The use of each criterion gives a different way to classify the partials. One major problem is to be able to fuse these heterogeneous criteria, to obtain a unique classification method. Another problem is to incorporate this structuring within the analysis chain, to obtain a partial tracking algorithm with multiple criteria that would track classes of partials (entities) instead of individual partials, and thus should be more robust. Finding solutions to these problems are major research directions. For now, we



(a) using the amplitude difference (b) using the D measure of Equation (8.1)

Figure 8.1: Examples of note onset detection using two different measures: (a) the difference in amplitude between consecutive frames and (b) the D measure of Equation (8.1). The true onsets – annotated manually – are displayed as vertical bars. Our method manages to identify correctly the note onsets, even if the volume of the note fades in slowly (see around frame 200).

have started to study the perceptive criteria separately. Together with Lagrange [Lag04], we obtained promising results in year 2004 (see below). Although we are conscious that the state of the art of this very active research area has evolved in 4 years, the following results are still good starting points.

8.1.1 Common Onsets

As noted by Hartmann [Har88], the common onset (birth) of partials plays a preponderant role in our perception of sound entities. From the experiences of Bregman and Pinker [BP78] and Gordon [Gor84], the partials should appear within a short time window of around 30ms (corresponding to a number of γ consecutive frames, see below), else they are likely to be heard separately. Many onset detection methods are based on the variation in time of the amplitude or phase of the signal (see [BDA+05] for a survey). Our HFC partial tracker (see Section 4.2) better identifies the partial onsets. From the resulting LTS representation, Lagrange [Lag04] proposes an algorithm based on the D measure defined as

$$D[n] = \frac{B[n]}{C[n]} \tag{8.1}$$

$$\text{with} \quad B[n] = \sum_{p=1}^{P} \varepsilon_p[n]\bar{a}_p \quad \text{and} \quad C[n] = \frac{1}{2\gamma+1} \sum_{p=1}^{P} \sum_{k=-\gamma}^{+\gamma} a_p[n+k] \tag{8.2}$$

where a_p is the amplitude of the partial p, $\bar{a}_p$ is its mean value, and $\varepsilon_p[n]$ is 1 if the partial is born in the $[n-\gamma,n+\gamma]$ interval and 0 otherwise. Thanks to this measure, it seems that we can identify the onsets of notes even if their volume fades in slowly (see Figure 8.1), leading to a better structuring of the sound entities (see Figure 8.2).

8.1.2 Harmonic Relation

The earliest attempts at acoustical entity identification and separation consider harmonicity as the sole cue for group formation. Some rely on a prior detection of the fundamental frequency [Gro96]

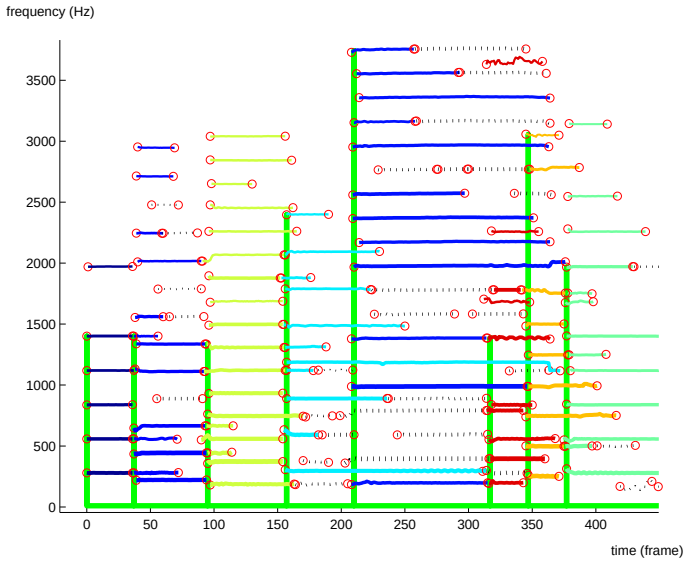


Figure 8.2: Result of the structuring of the partials in sound entities using the common onset criterion, on the example of Figure 8.1.

and others consider only the harmonic relation of the frequencies [Kla03]. A classic approach is to perform a correlation of the short-term spectrum with some template, which should be a periodic function, of period F – the fundamental frequency under investigation. In [21], we proposed to use a Fourier transform of the magnitude spectrum, to detect the F-periodicity in the spectral structure. The underlying template was then a simple sine function (basis of the Fourier transform). Another template can be built using the expression of the Hann window

$$g_{1,F}(f) = \frac{1}{2}\left(1+\cos\left(\frac{2\pi f}{F}\right)\right) \tag{8.3}$$

where f is the frequency and F is the fundamental. However, the problem of these templates is that the width of the template peaks depend on F (see Figure 8.3(a)). For a constant peak selectivity, we propose with Cartier [Car03] another function

$$g_{2,F}(f) = g_{1,F}(f)^{-s/\log(g_{1,F}(1))} \tag{8.4}$$

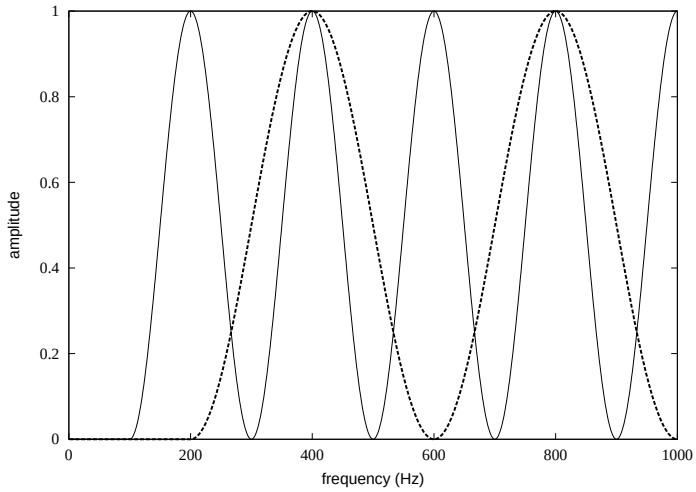
where $s \in (0,1]$ allows us to tune this selectivity (see Figure 8.3(b)). But this template is still too far from the typical structure of musical sounds, whose spectral envelope (in dB) is often a linear function of the frequency (in Hz), see [Kla03]. Thus, together with Lagrange [Lag04] we propose

$$g_{3,F}(f) = g_{2,F}(f) \cdot 10^{-d(f-F)} \tag{8.5}$$

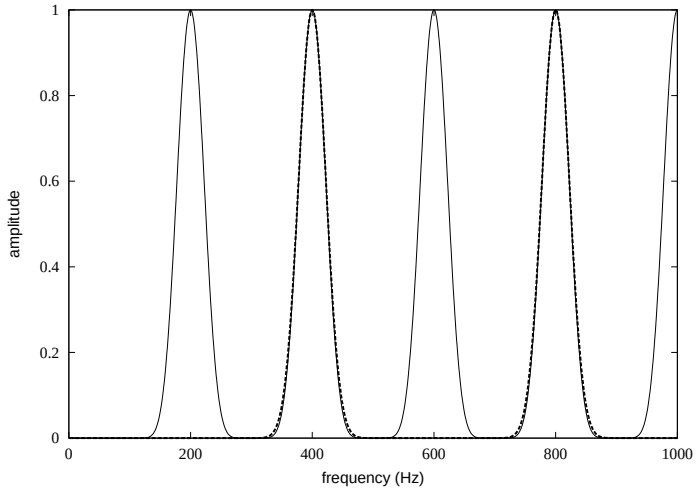
where d is the slope of the spectral envelope, in dB per Hz (see Figure 8.3(c)). Multi-pitch estimation is still an active research area, and yet our simple approach gives very good results, especially when the s and d parameters can be learned from a database. However, many musical instruments are not perfectly harmonic, and the template should ideally also depend on β (see Equation (2.2)).

8.1.3 Similar Evolutions

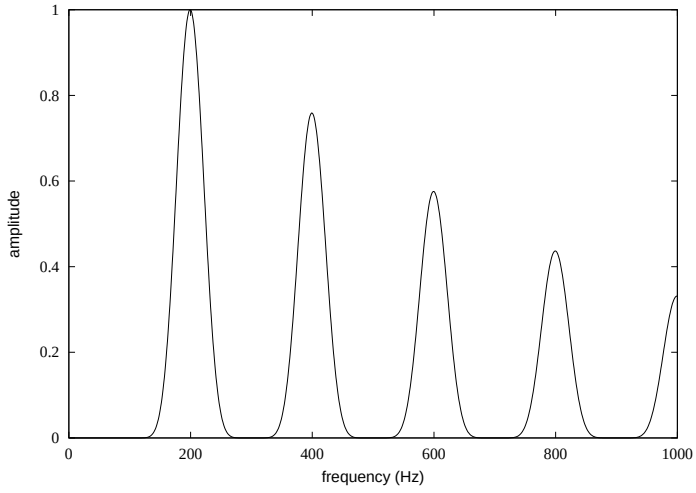
According to the work of McAdams [McA89], a group of partials is perceived as a unique sound entity only if the variations of these partials are correlated, whether the sound is harmonic or not.



(a) $g_{1,200}$ and $g_{1,400}$



(b) $g_{2,200}$ and $g_{2,400}$ ($s = 0.001$)



(c) $g_{3,200}$ ($d = 0.012$dB/Hz)

Figure 8.3: Three kinds of templates for the extraction of harmonic structures: (a) simple periodic function, (b) modified version for a constant peak selectivity, and (c) modified version for a more realistic spectral envelope.

These correlations among the parameters of the partials can be observed in the level-2 representation of our hierarchic sinusoidal model (see Figure 5.2, Section 5.2).

Together with Lagrange [Lag04, Lag05], we searched for a relevant dissimilarity between two elements (the partials), that is a dissimilarity which is low for elements of the same class (sound entity) and high for elements that do not belong to the same class. We have shown in Section 4.2.3 that the auto-regressive (AR) modeling of the parameters of the partials can improve the tracking of the partials. Here, we show that AR modeling is a good candidate for the design of a robust dissimilarity metric.

Let ω_p be the frequency vector of the partial p. According to the AR model (see Section 4.2.3), the sample $\omega_p[n]$ can be approximated as a linear combination of past samples

$$\omega_p[n] = \sum_{k=1}^{K} c_p[k]\omega_p[n-k] + e_p[n] \tag{8.6}$$

where $e_p[n]$ is the prediction error. The coefficients $c_p[k]$ model the predictable part of the signal and it can be shown that these coefficients are scale invariant. On contrary, the non-predictable part $e_p[n]$ is not scale invariant. For each frequency vector ω_p, we compute a vector $c_p[k]$ of 4 AR coefficients with the Burg method. Although the direct comparison of the AR coefficients computed from the two vectors ω_p and ω_q is generally not relevant, the spectrum of these coefficients may be compared. The Itakura distortion measure [Ita75], issued from the speech recognition community can be considered:

$$d_{\mathrm{AR}}(\omega_p, \omega_q) = \frac{1}{2\pi} \log \int_{-pi}^{+pi} \left| \frac{C_p(\omega)}{C_q(\omega)} \right| d\omega \tag{8.7}$$

where

$$C_p(\omega) = 1 + \sum_{k=1}^{K} c_p[k] e^{-jk\omega}. \tag{8.8}$$

Another approach may be considered. Indeed, the amount of error done by modeling the vector ω_p by the coefficients computed from vector ω_q may indicate the dissimilarity of these two vectors. Let us introduce a new notation e_p^q, the cross prediction error defined as the residual signal of the filtering of the vector ω_p with c_q

$$e_p^q[n] = \omega_p[n] - \sum_{k=1}^{K} c_q[k]\omega_p[n-k]. \tag{8.9}$$

The principle of the dissimilarity d_σ is to combine the two dissimilarities $|e_p^q|$ and $|e_q^p|$ to obtain a symmetrical one:

$$d_\sigma(\omega_p, \omega_q) = \frac{1}{2}\left(|e_p^q| + |e_q^p|\right). \tag{8.10}$$

Given two vectors ω_p and ω_q to be compared, the coefficients c_p and c_q are computed to minimize the power of the respective prediction errors e_p and e_q. If the two vectors ω_p and ω_q are similar, the power of the cross prediction errors e_p^q and e_q^p will be as weak as those of e_p and e_q. We can consider an other dissimilarity d'_σ defined as the ratio between the sum of the crossed prediction errors and the sum of the direct prediction errors:

$$d'_\sigma(\omega_p, \omega_q) = \frac{|e_p^q| + |e_q^p|}{1 + |e_p| + |e_q|}. \tag{8.11}$$

Lagrange [Lag05] shows that the metrics based on AR modeling perform quite well.

8.1.4 Spatial Location

The common spatial location is the last – but not least – criterion we will use for the classification of the partials in sound entities. More precisely, we will use the promising results of Chapter 7 to localize the partials in azimuth. A first attempt has been recently done by Raspaud and Evangelista [RE08].

8.2 Audio Watermarking for Informed Separation

Since extracting the sound entities is an extremely difficult task, a solution to ease this extraction could be to take advantage of extra (inaudible) information stored in the sound. We have shown the suitability of spectral modeling for audio watermarking (see below), that is for hiding (inaudible) information within sounds, which is extremely useful for example for audio copyright management. Part of our future research directions is to study how to include such inaudible information in each sound entity prior to the mixing process, to ease the extraction of the entities from the mix.

8.2.1 Using Amplitude Thresholds

A very classic way to embed data in sound in an inaudible way is to generate a signal that remains under the masking threshold (see Sections 1.5 and 3.3.3). With a group of students, in 2003 we developed the MAGIC – Musical Audio / Graphics Interleaved Coding – software program, in order to embed a gray-scale picture in a sound. The principles of MAGIC are the following:

- the picture is split in vertical bands of fixed width ($K = 8$ pixels): the image is then browsed for groups of K pixels, band-by-band from left to right, each band being then browsed from top to bottom;
- for each group of K pixels, a new short-term spectrum of the original sound signal is computed, together with the associated masking threshold M: the audio spectrum is then split in K frequency bands, and for each frequency band the amplitude is set proportionally to M with a coefficient corresponding to the pixel intensity (0 for black, 1 for white).

This way, the embedded data stays below the masking threshold M. Moreover, the way the groups of K pixels are produced ensures a continuity in time (y axis) and frequency (x axis) of the watermark – inherited from the inherent smoothness of natural images. Moreover, to avoid discontinuities at the picture boundaries, the borders of the picture are extrapolated by a progressive fade-out/in to/from black (zero). With this very simple watermarking technique, we obtained interesting results (see Figure 8.4), although the watermark is not resistant to MP3 encoding.

8.2.2 Using Frequency Modulation

A more original watermarking was developed together with Girin [28], after our conclusion when looking at the evolutions of the polynomial phase models illustrated in Figure 4.13: with many models, within a short frame, the frequency trajectory is modulated, although we cannot hear any difference. Thus, we can take advantage of this phenomenon by controlling the frequency modulations so that we can embed useful data in an inaudible way. Starting from smooth frequency trajectories, we add watermarking patterns consisting of Hann functions multiplied by ± 1 to encode 1 or 0, respectively. In theory, the frequency error should stay below the frequency modulation threshold (see Section 1.5).

(a) original (b) decoded

Figure 8.4: MAGIC used on the (upper part) of the famous Len(n)a picture [Sod72] (resized to 128 × 128 pixels): (a) original and (b) decoded from "You can leave your hat on" by Joe Cocker (256s).

However, in practice, we were able to go far beyond this threshold, probably thanks to the harmonic context since we were watermarking only one partial of a complex harmonic sound.

8.3 Application: Active Listening

Nowadays, the listener is considered as a receptor who passively listens to the audio signal stored on various medias (CD, DVD audio, *etc.*). The only modification which is easy to perform is basically to change the volume. Although new formats such as MPEG Audio Layer 3 (MP3) have changed the way people access to music, the interaction with this music is still very limited. However, our feeling is that people are eager to interact with the original media, while the sound is playing. This can be seen for example with the karaoke, where the listener can replace the voice of the original singer. But more freedom and creativity are also possible.

From our research on spectral sound modeling, we can now propose new ways for the identification, separation, and manipulation of the several sound entities (sources) which are perceived by the listener as independent components within the binaural (stereophonic) mix that reaches his/her ears. More precisely, we can find out the sound entities by considering their localization (spatial hearing) and the correlations of their spectral parameters (common onsets, harmonic relations, similar time evolutions). On the other hand, we are interested in musical transformations based on spectral sound models with parameters close to the human perception (psychoacoustics), suitable for any listener, but also for musicians.

We thus propose to enable any listener to have an active listening behavior, by giving him/her the freedom to interact with the sound in real time during its diffusion, instead of listening to this sound in the usual – passive – way, with only very limited controls such as volume changes. We propose to offer the listener the possibility to also change the spatial locations of the individual sound sources, their respective volumes, pitches, and even timbres, as well as their durations, and the rhythm of the music which is stored on the media (CD or MP3 for examples). We will focus on the following manipulations:

- **Enhanced Volume Control:**
  The listener will be offered the possibility to change the volume of each separate sound entity. For example, he/she can attenuate the voice singing, thus obtaining a karaoke effect.

- **Spatial Relocation:**
  We will also give the listener the freedom to change the spatial location of the sound entities.

For example, while listening to music, the listener could decide where each of the instruments he/she hears should be. These positions can be changed in real time, thus *e.g.* it is possible to make a saxophone sound turn around the listener, whereas it was at a fixed position in the original recording.

- **Time Scaling:**
  We can change the duration of the sound entities, without altering the other perceptive parameters (pitch, timbre, spatial location). However, this effect will be very limited (in time), because of the real-time constraints. Indeed, slowing down the music imposes the use of a buffer whose size keeps increasing during the stretching. Moreover, since we analyze the sound in real time, we cannot speed up the music since it is impossible to guess its future evolutions (even though this is possible on a short range, using prediction techniques).

We have started an industrial collaboration with the iKlax Media company [44]. Although active listening has already been investigated by industrial laboratories such as SONY CSL Paris, all started with unmixed sound sources, and none of them managed to extract them directly from the recorded music. In fact, to our knowledge only Creative Labs investigated the use of the spatial localization to separate and manipulate the sources. Our project opens up new horizons in computer music, by focusing on a very challenging problem: the real-time extraction of sound entities from recorded music.

Apart from this technical aspect, for us it is clear that this approach will change the way people can listen to music. We aim at providing the listener with the freedom to interact with the sound in real time during its diffusion – as composers of electroacoustic music do, instead of listening to this sound in the usual – passive – way.

Conclusions and Future Work

After a brief introduction to musical sound in Part I, we presented our research in sinusoidal modeling since the Ph.D. in Part II. We proposed a new short-term analysis method based on the derivatives of the signal, and generalized it to the non-stationary case. We also identified some equivalences between existing analysis methods, and we have now solid evaluation procedures for comparing them in practice. In fact, we are close to optimal precision. The main problem is now with the long-term analysis, and more precisely with partial-tracking algorithms. We proposed a new partial tracker based on the use of linear prediction and the control of the high-frequency content. However, our feeling is that the long-term analysis is still in its infancy, with very empirical approaches, and with a strong need for evaluation procedures. Regarding the synthesis, we proposed a method using software oscillators and yet of low complexity. This method uses low-order amplitude and phase models since our higher-order models coming from the extension to the non-stationary case did not seem to improve the resynthesis quality in practice. We are already close to optimal complexity. Thus we proposed to go further by reducing the size of the problem by taking advantage of psychoacoustics.

In Part III, we presented ongoing research with Ph.D. students. For now, we have investigated hierarchic sinusoidal modeling with the hope that we can perform enhanced time scaling and possibly reach the macroscopic level of repetitive (quasi periodic) musics. We are extending our model to stochasticity and we propose an efficient analysis method based on probability density functions. However, we still need to propose the corresponding synthesis method. We have also extended our model to the space dimension, with a simplified binaural model based on spatial cues. We propose a source separation algorithm from binaural mix, as well as binaural and multi-loudspeaker spatialization methods. We consider the azimuth angle and also take the distance into consideration. Yet, we still have to consider the elevation angle.

We are conscious that beside stochasticity and spatiality, we have to reconsider our modeling approach in terms of perception (*e.g.* with spectral atoms closer to perception [Smi06]) and consider the sparseness (sparse decomposition on over-complete bases, in collaboration with Daudet [Dau00]), since from this recent research our feeling is that sparseness is a key point for the auditory system, and should lead to computer representations that are both more efficient and closer to perception.

However, our main research directions identified for the future are the separation of sound entities from polyphonic music, with or without the help of prior watermarking, and with an application to active listening: instead of listening to the sound in the usual (passive) way, the listener should have the freedom to interact with the sound in real time during its diffusion – as composers of electro-acoustic music do. With as many sound transformations on the sound entities as possible, we want to enable anyone to have an active listening behavior, thus bringing people from passivity to creativity.

Bibliography

[ADTA01] V. Ralph Algazi, Richard O. Duda, Dennis M. Thompson, and Carlos Avendano. The CIPIC HRTF Database. In *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 99–102, New Paltz, New York, USA, October 2001.

[AF95] François Auger and Patrick Flandrin. Improving the Readibility of Time-Frequency and Time-Scale Representations by the Reassignment Method. *IEEE Transactions on Signal Processing*, 43(5):1068–1089, May 1995.

[AHU83] Alfred V. Aho, John E. Hopcroft, and Jeffrey D. Ullman. *Data Structures and Algorithms*, pages 392–407. Series in Computer Science and Information Processing. Addison-Wesley, 1983.

[AKZ99] Rasmus Althoff, Florian Keiler, and Udo Zölzer. Extracting Sinusoids From Harmonic Signals. In *Proceedings of the Digital Audio Effects (DAFx) Conference*, pages 97–100, Trondheim, Norway, December 1999.

[AKZ02] Daniel Arfib, Florian Keiler, and Udo Zölzer. *DAFx – Digital Audio Effects*, chapter 9, pages 299–372. John Wiley & Sons, 2002.

[Arf79] Daniel Arfib. Digital Synthesis of Complex Spectra by Means of Multiplication of Non-Linear Distorted Sine Waves. *Journal of the Audio Engineering Society*, 27(10):757–768, 1979.

[AS05] Mototsugu Abe and Julius O. Smith. AM/FM Rate Estimation for Time-Varying Sinusoidal Modeling. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume III, pages 201–204, Philadelphia, NY, USA, March 2005.

[Bad05] Roland Badeau. *Méthodes à haute résolution pour l'estimation et le suivi de sinusoïdes modulées – Application aux signaux de musique*. PhD thesis, École Nationale Supérieure des Télécommunications, Paris, France, April 2005. In French.

[BCRD06] Michaël Betser, Patrice Collen, Gaël Richard, and Bertrand David. Review and Discussion on Classical STFT-Based Frequency Estimators. In *120th Convention of the Audio Engineering Society*, Paris, May 2006.

[BCRD08] Michaël Betser, Patrice Collen, Gaël Richard, and Bertrand David. Estimation of Frequency for AM/FM Models Using the Phase Vocoder Framework. *IEEE Transactions on Signal Processing*, 56(2):505–517, February 2008.

[BDA+05] Juan Pablo Bello, Laurent Daudet, Samer Abdallah, Chris Duxbury, Mike Davies, and Mark B. Sandler. A Tutorial on Onset Detection in Music Signals. *IEEE Transactions on Speech and Audio Processing*, 13(5):1035–1047, September 2005.

[Bet08] Michaël Betser. *Modélisation sinusoïdale et applications à l'indexation audio*. PhD thesis, École Nationale Supérieure des Télécommunications, Paris, France, April 2008. In French.

[Bla97] Jens Blauert. *Spatial Hearing – The Psychophysics of Human Sound Localization*. MIT Press, Cambridge, Massachusetts, USA, revised edition, 1997.

[BP78] Albert S. Bregman and Steven Pinker. Auditory Streaming and the Building of Timbre. *Canadian Journal of Psychology*, 32(1):19–31, 1978.

[BRD08] Roland Badeau, Gaël Richard, and Betrand David. Performance of ESPRIT for Estimating Mixtures of Complex Exponentials Modulated by Polynomials. *IEEE Transactions on Signal Processing*, 56(2):492–504, February 2008.

[Bre90] Albert S. Bregman. *Auditory Scene Analysis: The Perceptual Organization of Sound*. MIT Press, 1990.

[Bru79] Marc Le Brun. Digital Waveshaping Synthesis. *Journal of the Audio Engineering Society*, 27(4):250–266, 1979.

[BSZ+95] Henry E. Bass, Louis C. Sutherland, Allan J. Zuckerwar, David T. Blackstock, and Daniel M. Hester. Atmospheric Absorption of Sound: Further Developments. *Journal of the Acoustical Society of America*, 97(1):680–683, 1995.

[Bur75] John P. Burg. *Maximum Entropy Spectral Analysis*. PhD thesis, Stanford University, California, USA, 1975.

[Car03] Grégory Cartier. Séparation de sources harmoniques. Master's thesis, LaBRI, University of Bordeaux 1, Talence, France, June 2003. In French.

[Cho71] John M. Chowning. The Simulation of Moving Sound Sources. *Journal of the Audio Engineering Society*, 19(1):2–6, 1971.

[Cho73] John M. Chowning. The Synthesis of Complex Audio Spectra by Means of Frequency Modulation. *Journal of the Audio Engineering Society*, 21(7):526–534, 1973.

[Col02] Patrice Collen. *Techniques d'enrichissement de spectre des signaux audionumériques*. PhD thesis, École Nationale Supérieure des Télécommunications, Paris, France, November 2002. In French.

[CT65] James W. Cooley and John W. Tukey. An Algorithm for the Machine Computation of Complex Fourier Series. *Mathematics of Computation*, 19:297–301, April 1965.

[Dau00] Laurent Daudet. *Représentation structurelle de signaux audiophoniques – Méthodes hybrides pour des applications à la compression*. PhD thesis, Université Aix-Marseille I, France, 2000.

[DGR93] Philippe Depalle, Guillermo Garcia, and Xavier Rodet. Analysis of Sound for Additive Synthesis: Tracking of Partials Using Hidden Markov Models. In *Proceedings of the International Computer Music Conference (ICMC)*, San Francisco, California, USA, 1993. International Computer Music Association (ICMA).

[DK90] Petar M. Djurić and Steven M. Kay. Parameter Estimation of Chirp Signals. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 38(12):2118–2126, December 1990.

[Dol86] Mark Dolson. The Phase Vocoder: A Tutorial. *Computer Music Journal*, 10(4):14–27, 1986.

[FR98] Neville F. Fletcher and Thomas D. Rossing. *The Physics of Musical Instruments*. Springer-Verlag, New York, USA, second edition, 1998.

[FRD92] Adrian Freed, Xavier Rodet, and Philippe Depalle. Synthesis and Control of Hundreds of Sinusoidal Partials on a Desktop Computer without Custom Hardware. In *Proceedings of the ICSPAT'92 Conference*, San José, California, USA, 1992.

[FvDFH95] James D. Foley, Andries van Dam, Steven K. Feiner, and John F. Hughes. *Computer Graphics – Principles and Practice*, chapter 11: Representing Curves and Surfaces, pages 471–531. System Programming Series. Addison-Wesley, second edition, 1995.

[FZ06] Hugo Fastl and Eberhard Zwicker. *Psychoacoustics: Facts and Models*. Springer, third edition, 2006.

[GM94] William G. Gardner and Keith Martain. HRTF Measurements of a KEMAR Dummy-Head Microphone. Technical report, MIT Media Lab, 1994.

[Gor84] John W. Gordon. *Perception of Attack Transients in Musical Tones*. PhD thesis, Department of Music, Stanford University, California, USA, 1984.

[GP99] Guillermo Garcia and Juan Pampin. Data Compression of Sinusoidal Modeling Parameters Based on Psychoacoustic Masking. In *Proceedings of the International Computer Music Conference (ICMC)*, pages 40–43, Beijing, China, October 1999.

[Gre75] John M. Grey. *An Exploration of Musical Timbre*. PhD thesis, Department of Music, Stanford University, 1975.

[Gro96] Stephen Grossberg. *Pitch Based Streaming in Auditory Perception*. MIT Press, 1996.

[GS85] John. W. Gordon and Julius O. Smith. A Sine Generation Algorithm for VLSI Applications. In *Proceedings of the International Computer Music Conference (ICMC)*, pages 165–168, Burnaby, Canada, 1985.

[Hai03] Stephen W. Hainsworth. *Techniques for the Automated Analysis of Musical Audio*. PhD thesis, University of Cambridge, United Kingdom, December 2003.

[Han03] Pierre Hanna. *Modélisation statistique de sons bruités: étude de la densité spectrale, analyse, transformation musicale et synthèse*. PhD thesis, LaBRI, University of Bordeaux 1, Talence, France, December 2003. In French.

[Har78] Fredric J. Harris. On the Use of Windows for Harmonic Analysis with the Discrete Fourier Transform. *Proceedings of the IEEE*, 66(1):51–83, January 1978.

[Har88] William M. Hartmann. *Auditory Function: Neurobiological Bases of Hearing*, chapter Pitch Perception and the Segregation and Integration of Auditory Entities, pages 623–645. Wiley, New York, USA, 1988. Gerald M. Edelman, W. Einar Gall and W. Maxwell Cowan (Eds.).

[Har05] William M. Hartmann. *Signals, Sound, and Sensation*. Springer, 2005.

[IMA88] IMA. *MIDI 1.0 Detailed Specification*. International MIDI Association (IMA), Los Angeles, 1988.

[iow] The Iowa Music Instrument Samples. Online. URL: `http://theremin.music.uiowa.edu`.

[ISO93] ISO (International Organization for Standardization), Geneva, Switzerland. *ISO 9613-1:1993: Acoustics – Attenuation of Sound During Propagation Outdoors – Part 1: Calculation of the Absorption of Sound by the Atmosphere*, 1993.

[Ita75] Fumitada Itakura. Minimum Prediction Residual Principle Applied to Speech Recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 23(1):67–72, 1975.

[Kay93] Steven M. Kay. *Fundamentals of Statistical Signal Processing – Estimation Theory*. Signal Processing Series. Prentice-Hall, 1993.

[KAZ00] Florian Keiler, Daniel Arfib, and Udo Zölzer. Efficient Linear Prediction for Digital Audio Effects. In *Proceedings of the Digital Audio Effects (DAFx) Conference*, pages 19–24, Verona, Italy, December 2000.

[KdVG76] Kunihiko Kodera, Claude de Villedary, and Roger Gendrin. A New Method for the Numerical Analysis of Non-Stationary Signals. *Physics of the Earth and Planetary Interiors*, 12:142–150, 1976.

[KGdV78] Kunihiko Kodera, Roger Gendrin, and Claude de Villedary. Analysis of Time-Varying Signals with Small BT Values. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26(1):64–76, February 1978.

[KKS01] Ismo Kauppinen, Jyrki Kauppinen, and Pekka Saarinen. A Method for Long Extrapolation of Audio Signals. *Journal of the Audio Engineering Society*, 49(12):1167–1180, December 2001.

[Kla03] Anssi P. Klapuri. Multiple Fundamental Frequency Estimation Based on Harmonicity and Spectral Smoothness. *IEEE Transactions on Speech and Audio Processing*, 11(6):804–816, November 2003.

[Kuh77] George F. Kuhn. Model for the Interaural Time Differences in the Azimuthal Plane. *Journal of the Acoustical Society of America*, 62(1):157–167, 1977.

[Kut08] Rade Kutil. Optimized Sinusoid Synthesis via Inverse Truncated Fourier Transform. *IEEE Transactions on Audio, Speech, and Language Processing*, 2008. In Press.

[Lag04] Mathieu Lagrange. *Modélisation sinusoïdale des sons polyphoniques*. PhD thesis, LaBRI, University of Bordeaux 1, Talence, France, December 2004. In French.

[Lag05] Mathieu Lagrange. A New Dissimilarity Metric for the Clustering of Partials Using the Common Variation Cue. In *Proceedings of the International Computer Music Conference (ICMC)*, Barcelona, Spain, September 2005. International Computer Music Association (ICMA).

[Lar00] Jean Laroche. Synthesis of Sinusoids via Non-Overlapping Inverse Fourier Transform. *IEEE Transactions on Speech and Audio Processing*, 8(4):471–477, July 2000.

[MA86] Jorge S. Marques and Luis B. Almeida. A Background for Sinusoid Based Representation of Voiced Speech. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 11, pages 1233–1236, Tokyo, Japan, April 1986.

[Mak92] John Makhoul. Linear Prediction: A Tutorial Review. *Proceedings of the IEEE*, 63(4):561–580, November 1992.

[Mar00] Sylvain Marchand. *Sound Models for Computer Music (analysis, transformation, synthesis)*. PhD thesis, LaBRI, University of Bordeaux 1, Talence, France, December 2000.

[Mar06] Rainer Martin. Bias Compensation Methods for Minimum Statistics Noise Power Spectral Density Estimation. *Signal Processing*, 86(6):1215–1229, 2006.

[MC98] Paul Masri and Nishan Canagarajah. Extracting More Detail From the Spectrum With Phase Distortion Analysis. In *Proceedings of the Digital Audio Effects (DAFx) Conference*, pages 119–122, Barcelona, Spain, November 1998.

[McA89] Stephen McAdams. Segregation of Concurrrents Sounds: Effects of Frequency Modulation Coherence. *Journal of the Acoustical Society of America*, 86(6):2148–2159, 1989.

[Moo76] James A. Moorer. The Synthesis of Complex Audio Spectra by Means of Discrete Summation Formulas. *Journal of the Audio Engineering Society*, 24(9):717–727, November 1976.

[Moo77] James A. Moorer. Signal Processing Aspects of Computer Music: A Survey. *Proceedings of the IEEE*, 65(8):1108–1137, August 1977.

[Moo98] Brian C. J. Moore. *Cochlear Hearing Loss*. Whurr Publishers Ltd, London, United Kingdom, 1998.

[MP02] Nikolaus Meine and Heiko Purnhagen. Fast Sinusoid Synthesis for MPEG-4 HILN Parametric Audio Decoding. In *Proceedings of the Digital Audio Effects (DAFx) Conference*, pages 105–110, Hamburg, Germany, September 2002.

[MQ86] Robert J. McAulay and Thomas F. Quatieri. Speech Analysis/Synthesis Based on a Sinusoidal Representation. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 34(4):744–754, 1986.

[OKK04] Masatsugu Okazaki, Toshifumi Kunimoto, and Takao Kobayashi. Multi-Stage Spectral Subtraction for Enhancement of Audio Signal. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume II, pages 805–808, Montreal, Quebec, Canada, May 2004.

[Pap77] Athanasios Papoulis. *Signal Analysis*. McGraw-Hill, New York, USA, 1977.

[Pap91] Athanasios Papoulis. *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill, third edition, 1991.

[Pie83] John R. Pierce. *The Science of Musical Sound*. Scientific American Books, Inc, New York, USA, 1983.

[Pol83] Giovanni De Poli. A Tutorial on Digital Sound Synthesis Techniques. *Computer Music Journal*, 7(2):76–87, 1983.

[PR99] Geoffroy Peeters and Xavier Rodet. SINOLA: A New Analysis/Synthesis Method using Spectrum Peak Shape Distortion, Phase and Reassigned Spectrum. In *Proceedings of the International Computer Music Conference (ICMC)*, pages 153–156, Beijing, China, October 1999.

[Puc95] Miller Puckette. Phase-locked Vocoder. In *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 222–225, New Paltz, New York, USA, October 1995.

[Pug90] William Pugh. Skip Lists: A Probabilistic Alternative to Balanced Trees. *Communications of the ACM*, 33:668–676, June 1990.

[Pul97] Ville Pulkki. Virtual Sound Source Positioning Using Vector Base Amplitude Panning. *Journal of the Audio Engineering Society*, 45(6):456–466, June 1997.

[Ras07] Martin Raspaud. *Modèle spectral hiérarchique pour les sons et applications*. PhD thesis, LaBRI, University of Bordeaux 1, Talence, France, May 2007.

[Ray07] John W. Strutt (Lord Rayleigh). On Our Perception of Sound Direction. *Philosophical Magazine*, 13:214–302, 1907.

[RE08] Martin Raspaud and Gianpaolo Evangelista. Binaural Partial Tracking. In *Proceedings of the Digital Audio Effects (DAFx) Conference*, pages 123–128, Helsinki, Finland, September 2008.

[Ris69] Jean-Claude Risset. Catalog of Computer Synthesized Sounds. Technical report, Bell Telephone Laboratories, Murray Hill, USA, 1969.

[Rob06] Matthias Robine. *Analyse de la performance musicale et synthèse sonore rapide*. PhD thesis, LaBRI, University of Bordeaux 1, Talence, France, December 2006. In French.

[RZR04] Axel Röbel, Miroslav Zivanovic, and Xavier Rodet. Signal Decomposition by Means of Classification of Spectral Peaks. In *Proceedings of the International Computer Music Conference (ICMC)*, pages 446–449, Miami, Florida, USA, November 2004.

[Röb02] Axel Röbel. Estimating Partial Frequency and Frequency Slope Using Reassignment Operators. In *Proceedings of the International Computer Music Conference (ICMC)*, pages 122–125, Göteborg, Sweden, September 2002.

[Röb03] Axel Röbel. A New Approach to Transient Processing in the Phase Vocoder. In *Proceedings of the Digital Audio Effects (DAFx) Conference*, pages 344–349, London, United Kingdom, September 2003.

[SdDDR98] Jan Sijbers, Arnold J. den Dekker, Dirk Van Dyck, and Erick Raman. Estimation of Signal and Noise from Rician Distributed Data. In *Proceedings of the International Conference of Signal Processing and Communications*, pages 140–142, Gran Canaria, Canary Islands, Spain, February 1998.

[Ser89] Xavier Serra. *A System for Sound Analysis/Transformation/Synthesis Based on a Deterministic plus Stochastic Decomposition*. PhD thesis, CCRMA, Department of Music, Stanford University, California, USA, 1989.

[SG84] Julius O. Smith and Phil Gossett. A Flexible Sampling-Rate Conversion Method. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 2, pages 19.4.1–19.4.2, San Diego, California, USA, March 1984.

[SH08] Sylvia Schulz and Thorsten Herfet. On the Window-Disjoint-Orthogonality of Speech Sources in Reverberant Humanoid Scenarios. In *Proceedings of the Digital Audio Effects (DAFx) Conference*, pages 43–50, Helsinki, Finland, September 2008.

[Smi06] Evan C. Smith. *Efficient Auditory Coding*. PhD thesis, Carnegie Mellon University, California, USA, 2006.

[Smi07] Julius O. Smith. *Spectral Audio Signal Processing*. CCRMA, Stanford University, California, USA, March 2007. (Draft URL:`http://ccrma.stanford.edu/~jos/sasp/`).

[Sod72] Lena Soderberg. (untitled). *Playboy*, page centerfold, November 1972.

[SS87] Julius O. Smith and Xavier Serra. PARSHL: An Analysis/Synthesis Program for Non-Harmonic Sounds based on a Sinusoidal Representation. In *Proceedings of the International Computer Music Conference (ICMC)*, pages 290–297, Champaign-Urbana, USA, 1987.

[SS90] Xavier Serra and Julius O. Smith. Spectral Modeling Synthesis: A Sound Analysis/Synthesis System Based on a Deterministic plus Stochastic Decomposition. *Computer Music Journal*, 14(4):12–24, 1990.

[SW98] Andrew Sterian and Gregory H. Wakefield. A Model-Based Approach to Partial Tracking for Musical Transcription. In *Proceedings of the SPIE Annual Meeting*, San Diego, California, USA, 1998.

[TAGK01] Cihan Tepedelenlioglu, Ali Abdi, Georgios B. Giannakis, and Mostafa Kaveh. Performance Analysis of Moment-Based Estimator for the K Parameter of the Rice Fading Distribution. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 4, pages 2521–2524, Salt Lake City, Utah, USA, May 2001.

[TL91] Kushal K. Talukdar and William D. Lawing. Estimation of the Parameters of the Rice Distribution. *Journal of the Acoustical Society of America*, 89(3):1193–1197, 1991.

[TSB05] Hiroko Terasawa, Malcolm Slaney, and Jonathan Berger. The Thirteen Colors of Timbre. In *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 323–326, New Paltz, New York, USA, October 2005.

[vHW20] Erich M. von Hornbostel and Max Wertheimer. Über die Wahrnehmung der Schallrichtung [On the Perception of the Direction of Sound]. *Sitzungsber. d. Preuss. Akad. Wissensch.*, pages 388–396, 1920.

[Vis04] Harald Viste. *Binaural Localization and Separation Techniques*. PhD thesis, École Polytechnique Fédérale de Lausanne, Switzerland, July 2004.

[Wel67] Peter D. Welch. The Use of Fast Fourier Transform for the Estimation of Power Spectra: A Method Based on Time-Averaging over Short, Modified Periodograms. *IEEE Transactions on Audio and Electroacoustics*, 15(22):70–73, June 1967.

[Wes79] David L. Wessel. Timbre Space as a Musical Control Structure. *Computer Music Journal*, 3(2):45–52, 1979.

[WS54] Robert S. Woodworth and Harold Scholsberg. *Experimental Psychology*. Holt, New York, USA, 1954.

[YR06] Chunghsin Yeh and Axel Röbel. Adaptive Noise Level Estimation. In *Proceedings of the Digital Audio Effects (DAFx) Conference*, pages 145–148, Montreal, Quebec, Canada, September 2006.

[ZF81] Eberhard Zwicker and Richard Feldtkeller. *Psychoacoustique – L'oreille, récepteur d'information*. Masson, Paris, France, 1981. In French.

[ZGS96] Guotong Zhou, Georgios B. Giannakis, and Ananthram Swami. On Polynomial Phase Signal with Time-Varying Amplitudes. *IEEE Transactions on Signal Processing*, 44(4):848–860, April 1996.

[ÖYR04] Özgür Yılmaz and Scott Rickard. Blind Separation of Speech Mixtures via Time-Frequency Masking. *IEEE Transactions on Signal Processing*, 52(7):1830–1847, July 2004.

Annexe A

Curriculum Vitæ

A.1 Cursus

1989–1990	**Baccalauréat scientifique série C**, Lycée Pape Clément (Pessac) Mention : Bien
1990–1993	**Classes préparatoires**, Lycée Michel Montaigne (Bordeaux) (Mathématiques Supérieures (1990–91) et Spéciales (1991–93) section M) Admissibilité au Concours Commun Polytechnique (ex-ENSI) **DEUG A Mathématiques-Informatique**, Université Bordeaux 1
1993–1994	**Licence Informatique**, Université Bordeaux 1 Mention : Très Bien, major de promotion
1994–1995	**Maîtrise Informatique**, Université Bordeaux 1 Mention : Bien, 2ème de promotion Sujet du mémoire : « Conversion entre protocoles d'informatique musicale »
1995–1996	**DEA Informatique**, Université Bordeaux 1 - LIX (École Polytechnique) Mention : Très Bien, major de promotion Spécialité : informatique répartie avec tolérance aux pannes Sujet : « Le problème du consensus en environnement distribué asynchrone » Directeurs : Robert Cori (LaBRI) et Bernadette Charron-Bost (LIX)
1996–1997	Service National effectué du 01/09/1996 au 31/08/1997 en tant que Scientifique du Contingent au Laboratoire d'Informatique de l'École Polytechnique (LIX)
1997–2000	**Doctorat en Informatique**, LaBRI, Université Bordeaux 1 Spécialité : informatique musicale, modélisation sonore Sujet : « Modélisation informatique du son musical » Directrice : Myriam Desainte-Catherine Jury : André Arnold, Myriam Desainte-Catherine, François Pachet (SONY CSL), Julius Smith (CCRMA, Stanford University), Robert Strandh, Horacio Vaggione, Udo Zölzer (University of the Federal Armed Forces, Hambourg) Thèse soutenue le 12 décembre 2000 – Mention : Très Honorable **Monitorat d'initiation à l'enseignement supérieur** (1998–2000) Tuteur : Eric Sopéna (IUT-A, Dépt. Informatique, Université Bordeaux 1)
2000–2001	**ATER** à l'Université Bordeaux 1
2001–···	**Maître de Conférences** en informatique à l'Université Bordeaux 1
09/2007–02/2009	en délégation auprès du CNRS (au LaBRI)

A.2 Présentation générale

A.2.1 Carrière

Après un DEA d'informatique en 1996 dans le domaine de l'algorithmique répartie avec tolérance aux pannes, co-encadré par Robert Cori (LaBRI, Université Bordeaux 1) et Bernadette Charron-Bost (LIX, École Polytechnique), j'ai décidé de changer de domaine de recherche et ma thèse en 1997–2000 fut dans l'informatique musicale, et plus précisément dans la modélisation du son numérique (thèse financée par une allocation MNERT, avec monitorat d'initiation à l'enseignement supérieur, et sous la direction de Myriam Desainte-Catherine). Après une année passée comme ATER, j'ai été nommé Maître de Conférences à l'Université Bordeaux 1 le 01/09/2001 et titularisé le 01/09/2002. Je fais partie de l'équipe « Image et Son » du LaBRI (Université Bordeaux 1, UMR 5800 du CNRS). Je suis actuellement en délégation CNRS (pour 18 mois à compter du 01/09/2007). De janvier à mars 2008, j'ai été professeur invité au CIRMMT, *McGill University*, Montréal, Canada. D'avril à juin 2008, j'ai été chercheur invité au CCRMA, *Stanford University*, Californie, USA.

A.2.2 Recherche

Mes activités de recherche se situent principalement dans le domaine de la modélisation, l'analyse, la transformation et la synthèse du son musical. Le son est alors un mélange d'entités sonores résultant d'une structuration musicale (organisation temporelle de ces entités). Par définition, chaque entité sonore est perçue individuellement par notre système auditif, et peut correspondre par exemple à une note d'un instrument de musique, ou bien à un phonème, un bruit, etc.

Objectifs

À partir d'un signal sonore, par exemple le son fixé sur un support comme le *Compact Disc* (CD) audio, je m'efforce d'analyser les différentes entités sonores perçues à l'intérieur de ce son musical afin de pouvoir les transformer individuellement. Si cette analyse et ces transformations se font en temps réel, il est possible de proposer à l'auditeur une expérience d'**écoute active** : modifier la musique pendant son écoute, par exemple en supprimant la voix chantée (effet karaoke), en modifiant les volumes des différents instruments (par exemple en atténuant la batterie), voire leurs positions (par exemple en déplaçant une guitare de la gauche à la droite). Si ces pratiques sont habituelles pour les compositeurs de musique électro-acoustique avec qui nous collaborons dans le cadre du SCRIME[1], elles sont révolutionnaires pour le grand public, habitué à une écoute plus passive.

La séparation des entités sonores est un défi scientifique majeur. En effet, nous percevons le mélange musical avec nos oreilles, au nombre de 2 (d'où les 2 canaux sur les CD audio), or le nombre d'entités sonores présentes simultanément (polyphonie) dans un son musical est en général bien supérieur à 2. Nous sommes dans un cas mathématiquement "dégénéré", avec moins d'équations que d'inconnues, c'est-à-dire moins de capteurs que de sources. Or le cerveau donne pourtant une solution. En s'inspirant des mécanismes de la perception, nous effectuons une décomposition fréquentielle (spectrale) des signaux et considérons que pour chaque signal il n'y a qu'une entité présente en chaque point (atome) du plan temps / fréquence (hypothèse d'orthogonalité des spectres à court terme). Ensuite, il s'agit de regrouper les atomes spectraux en entités sonores, par exemple sur des critères perceptifs en tirant parti de lois d'acoustique et de psychoacoustique (travaux d'Albert Bregman) : structures spectrales remarquables (sources harmoniques : DEA de Grégory Cartier, thèse de Mathieu

[1] Studio de Création et de Recherche en Informatique et Musique Électro-acoustique

Lagrange), simultanéité et cohérence temporelles (mêmes instants d'apparition, évolutions corrélées dans le temps : thèse de Mathieu Lagrange), coïncidence spatiale (mêmes angles d'arrivée : thèse de Joan Mouba), etc. Afin d'aider cette analyse traditionnelle de type CASA *(Computational Auditory Scene Analysis)*, nous envisageons (en collaboration avec Laurent Girin et Cléo Baras, Grenoble) une approche originale consistant à tirer parti d'informations supplémentaires rajoutées de manière inaudible à l'intérieur des sons (tatouage audio-numérique [28]).

Une fois les entités sonores identifiées, il est possible de les contrôler dans le temps (étirement temporel) et l'espace (spatialisation sonore ou "son 3D"). D'autres effets sonores et transformations musicales sont possibles (transposition – changement de la hauteur perçue, amplification – changement de l'intensité perçue, modification du timbre, etc.). Au final, il s'agit de synthétiser le son musical transformé, et ce le plus efficacement possible pour une écoute en temps réel.

Modélisation

Je m'intéresse aux modèles spectraux, et plus particulièrement aux modèles sinusoïdaux, dérivés des travaux d'Helmholtz et explorés par des pionniers de l'informatique musicale comme Jean-Claude Risset et Max Mathews, suivis plus tard par Robert McAulay et Thomas Quatieri dans le domaine de la parole et par Julius Smith et Xavier Serra dans le domaine de la musique. Ces modèles nécessitent des compétences pluridisciplinaires. Ils reposent sur des bases mathématiques et physiques solides, sont proches de la perception humaine, sont bien adaptés au discours musical, et génèrent un grand nombre de problèmes de nature informatique très intéressants. La structure de base de ces modèles est le partiel, oscillateur quasi sinusoïdal dont les paramètres (fréquence et amplitude) évoluent lentement dans le temps. Nous avons proposé (avec Myriam Desainte-Catherine) le modèle de Synthèse Additive Structurée (SAS) [2] qui contraint ces paramètres pour permettre de modifier indépendamment des grandeurs musicales telles que la hauteur, l'intensité ou la durée, tout en constituant une base solide pour l'étude scientifique du timbre des sons quasi harmoniques. Modéliser les variations de paramètres spectraux également sous forme spectrale nous a récemment permis (avec Martin Raspaud) d'aboutir à des modèles spectraux hiérarchiques, autorisant encore plus de souplesse dans le contrôle du son [30, 35]. La plupart des modèles sonores sont actuellement des modèles hybrides, où la composante bruitée (stochastique) est séparée de la composante sinusoïdale (déterministe). Nous souhaitons maintenant définir un modèle spectral unifié, plus souple, où parties déterministes et stochastiques seraient indissociables. Une collaboration avec Pierre Hanna est en cours, initiée en 2004 sous la forme du co-encadrement de Guillaume Meurisse (Master puis thèse).

En plus du modèle lui-même, il faut également pouvoir disposer d'une méthode d'analyse précise pour obtenir les paramètres du modèle à partir de sons existants, ainsi qu'une méthode de synthèse rapide pour générer un son numérique à partir de sa modélisation, si possible en temps réel.

Analyse

Traditionnellement, l'analyse se fait en deux temps : le signal est observé à court terme sur une petite fenêtre temporelle, puis ces observations sont exploitées à plus long terme pour reconstruire les évolutions des paramètres.

Analyse à court terme. Dans un premier temps, il s'agit d'estimer les paramètres instantanés (fréquence et amplitude) des partiels via des méthodes à la fois précises et efficaces. Parmi les méthodes les plus efficaces, on trouve celles basées sur la transformée de Fourier rapide. Mais il faut alors trouver des méthodes pour améliorer la précision. Bien que de nombreuses méthodes aient déjà été

proposées précédemment, il s'avère que bien peu sont suffisamment précises [23] (étude faite avec Florian Keiler, suite à notre atelier COST commun, voir plus loin). Parmi les plus prometteuses, citons celles exploitant les relations de phase au sein des spectres à court terme obtenus par la transformée de Fourier. Le vocodeur de phase estime de cette manière la fréquence instantanée. La réallocation spectrale est une autre méthode proposée pour améliorer la précision de l'estimation. Nous avons proposé une nouvelle méthode d'analyse basée sur les dérivées du signal sonore [10, 1]. Cette méthode a été utilisée en pratique notamment par France Télécom, Philips et l'Institut Fraunhofer (travaux de Karin Dressler). Récemment, grâce à cette méthode, nous avons montré l'équivalence théorique de méthodes d'analyse spectrale parmi les plus utilisées en informatique musicale [37, 6]. Ces méthodes d'analyse spectrale sont en cours de généralisation au cas non stationnaire où les paramètres peuvent varier à l'intérieur même de la fenêtre d'analyse. La généralisation de notre méthode basée sur les dérivées a été étendue récemment en collaboration avec Philippe Depalle [46] (*McGill University*, Montréal). Cette méthode s'avère être la plus précise dans la plupart des cas.

Analyse à long terme. Le point faible des méthodes d'analyse existantes est le suivi des trajectoires des partiels (en fréquence et en amplitude) dans le temps. Notre contribution (avec Mathieu Lagrange et Martin Raspaud) a été de considérer les trajectoires des partiels comme des signaux déterministes et inaudibles. Premièrement, nous avons montré l'utilité de la **prédiction linéaire** pour effectuer le suivi des partiels [24, 27, 7]. Cette prédiction est si performante qu'elle permet la **restauration** des sons [4], qui consiste à retrouver de l'information manquante dans la structure d'un son altéré. Secondement, la **limitation du contenu fréquentiel** des trajectoires améliore encore le suivi [31, 7]. L'évaluation des méthodes de suivi est un point essentiel, mais difficile et toujours ouvert [41]. En attendant, nos contributions sont reconnues fondamentalement par des spécialistes du domaine comme Xavier Serra (Université Pompeu Fabra, Barcelone) et Julius Smith *(Stanford University)*, avec qui nous poursuivons une collaboration scientifique depuis 1998. L'algorithme résultant est utilisé en pratique par France Télécom pour le codage et la transmission à bas débit.

Synthèse

Les modèles spectraux nécessitent le calcul d'un grand nombre d'oscillateurs sinusoïdaux. Le problème est alors de trouver un algorithme très efficace pour générer la séquence des échantillons de chaque oscillateur avec le moins d'instructions possible. Nous avons développé une méthode de synthèse sonore de complexité quasi optimale, qui repose sur un algorithme optimisé de génération incrémentale de la fonction sinus.

Afin d'accélérer encore le processus de synthèse, nous avons étudié (DEA de Mathieu Lagrange) avec succès la possibilité de réduction à la volée du nombre de partiels à synthétiser en tirant parti de phénomènes psychoacoustiques comme le masquage et de structures de données efficaces comme les *skip-lists* [20]. Cette technique a été utilisée dans le cadre du projet INRIA REVES (travaux de Nicolas Tsingos).

Nous disposons de la technique de synthèse linéaire la plus rapide à ce jour. Le DEA de Matthias Robine a montré le manque de souplesse des techniques mathématiques non linéaires qui, bien qu'extrêmement efficaces, s'avèrent inutilisables. Cependant, nous avons proposé avec Robert Strandh (thèse de Matthias Robine) une technique de synthèse rapide originale basée sur un générateur polynomial couplé à une file de priorité [38].

Nous nous sommes également intéressés à l'augmentation de la qualité de la resynthèse. Nous avons étudié (en collaboration avec le LORIA (Nancy), l'Institut de la Communication Parlée (Grenoble) et l'IRCAM (Paris)) divers modèles de phase polynomiaux [25] pour les oscillateurs.

A.2.3 Encadrement doctoral

Thèses de doctorat

Sauf indication contraire, les thèses suivantes sont co-encadrées par Myriam Desainte-Catherine (Professeur) et moi-même (pas encore habilité à diriger les recherches). Toutefois, je suis souvent le principal encadrant de ces thèses, ce qui se traduit par une participation personnelle d'au moins 90% :

- (2001–2004) Doctorat de **Mathieu Lagrange** – Modélisation sinusoïdale des sons polyphoniques, thèse financée par France Télécom R&D (Rennes), co-encadrée à hauteur de 90% avec Myriam Desainte-Catherine. La soutenance a eu lieu le 16 décembre 2004 (nombreuses publications internationales, dont 6 depuis 2004 : [26, 27, 31, 36, 37, 41]). Après 2 ans passés au Canada (universités de Victoria et McGill), Mathieu Lagrange est actuellement en séjour post-doctoral à TELECOM ParisTech ;
- (2003–2006) Doctorat de **Matthias Robine** – Analyse de la performance musicale et synthèse sonore rapide, thèse financée par une allocation MNERT (et un monitorat d'initiation à l'enseignement supérieur), co-encadrée à hauteur de 30% avec Robert Strandh (Professeur). J'ai suivi entièrement la dernière année de la thèse de Matthias Robine, Robert Strandh étant à l'étranger en 2005–2006. La soutenance a eu lieu le 13 décembre 2006 (1 conférence internationale [38]). Matthias Robine est actuellement à l'Université Bordeaux 1, dans le cadre du projet ANR SIMBALS ;
- (2003–2007) Doctorat de **Martin Raspaud** – Modèle spectral hiérarchique pour les sons et applications, thèse financée par une allocation MNERT, co-encadrée à hauteur de 90% avec Myriam Desainte-Catherine. La soutenance a eu lieu le 24 mai 2007 (3 conférences internationales [30, 35, 43]). Martin Raspaud est actuellement en séjour post-doctoral à l'Université de Linköping, Suède ;
- (2004–) Doctorat de **Joan Mouba** – Manipulations spatiales sur les sons spectraux, thèse financée par une bourse du gouvernement gabonais, co-encadrée à hauteur de 90% avec Myriam Desainte-Catherine. La soutenance est prévue début 2009 (2 conférences internationales [40, 45]) ;
- (2005–) Doctorat de **Guillaume Meurisse** – Vers un modèle spectral unifié pour les sons, thèse financée par une allocation MNERT, co-encadrée à hauteur de 30% avec Pierre Hanna (non habilité, 40%) et Myriam Desainte-Catherine (Professeur, 30%). La soutenance est prévue en 2009 (1 conférence internationale [39]).

Mémoires de DEA / Master Recherche

Sauf indication contraire, j'ai encadré seul les mémoires de DEA / Master Recherche suivants :

- (2000–2001) DEA de **Mathieu Lagrange** – Accélération de la synthèse sonore (1 publication [20]) ;
- (2001–2002) DEA de **Daniel Hollaar** – Spatialisation des sons spectraux ;
- (2002–2003) DEA de **Grégory Cartier** – Extraction de hauteurs et séparation de sources sonores ;
- (2002–2003) DEA de **Martin Raspaud** – Prédiction linéaire et extrapolation de signaux sonores (1 publication [24]) ;
- (2002–2003) DEA de **Matthias Robine** – Synthèse sonore rapide via des transformations non linéaires ;
- (2003–2004) DEA de **Nicolas Sarramagna** – Son 3D : perception spatiale et écoute active ;

– (2003–2004) DEA de **Vincent Goudard** – Modélisation d'une percussion virtuelle. Vincent Goudard suivait le DEA ATIAM à l'IRCAM (Paris). Cet étudiant est venu à Bordeaux au second semestre 2004 pour travailler avec Myriam Desainte-Catherine et moi-même. Ma participation dans l'encadrement de son mémoire était de 50% (1 conférence internationale [33]) ;
– (2004–2005) Master Recherche de **Guillaume Meurisse** – Vers un modèle spectral unifié pour les sons, co-encadré avec Pierre Hanna à hauteur de 50% (1 conférence internationale [39]). Continuation en thèse.

A.2.4 Relations avec le monde industriel

Thèse et DEA en collaboration industrielle

Un sujet de thèse et un sujet de DEA ont donné lieu à l'établissement de conventions avec des entreprises :

– **France Télécom R&D** (Rennes) : contrat pour la thèse de doctorat de Mathieu Lagrange sur l'analyse de sons en vue de l'indexation et de la compression, commencée en 2001 et soutenue fin 2004 ;
– **Philips** (Eindhoven, Pays-Bas) : contrat pour la continuation du DEA de Martin Raspaud (sous la forme de stage en entreprise dans le cadre d'une double formation DEA / DESS) sur l'extraction de mélodie à partir de signaux audio, de juin à septembre 2003. J'ai participé activement à la négociation et à la rédaction de ce contrat.

Transferts technologiques

J'ai participé avec la société **Algory / Lumiscaphe**[2] (Bordeaux) à l'élaboration d'un projet utilisant la technologie SAS introduite lors de ma thèse. Cette technologie a fait l'objet en 2000 d'un transfert technologique entre le LaBRI et cette société.

Plus récemment, une collaboration est en cours avec la société **iKlax Media**[3] (Bidart) dans le cadre du projet d'écoute active (confidentiel, dépôts de brevets en cours).

A.2.5 Animation scientifique

Organisation du colloque JIM 2000 `http://scrime.labri.fr/JIM2000/jim2000-fr.html`

Les Journées d'Informatique Musicale (JIM) sont des conférences qui ont un rôle fédérateur au sein de la communauté française d'informatique musicale. Les chercheurs en informatique musicale en France sont de plus en plus nombreux. Au sein des universités, ils sont particulièrement isolés. Les occasions de rencontres se limitaient aux colloques internationaux et aux colloques organisés par certains centres de recherche (IRCAM, Grame, etc.). Pour cela, ces journées « en terrain neutre » ont rencontré un vif succès. L'année 2000 coïncidait avec la 7$^{\text{ème}}$ édition. La conférence était alors organisée par le SCRIME et le LaBRI à Bordeaux. J'ai assuré diverses fonctions au sein du comité d'organisation de cette conférence. J'ai eu notamment l'entière responsabilité de la préparation et de l'édition des actes de la conférence. Initialement d'audience internationale, cette conférence est devenue nationale en 2004 avec la création de la conférence internationale associée *Sound and Music Computing* (SMC).

[2] URL : `http:www.lumiscaphe.com`
[3] URL : `http:www.iklax.com`

Action Européenne COST G-6

J'ai co-organisé avec Florian Keiler (*University of the Federal Armed Forces*, Hambourg, Allemagne) un atelier COST[4] qui s'est déroulé à Bordeaux début juillet 2001, qui portait sur la détection de la hauteur dans les sons et réunissait un petit groupe de chercheurs européens (France, Allemagne, Angleterre). Cette rencontre a donné lieu à une publication en conférence d'audience internationale [23].

Responsable de thème à la conférence ICMC 2005 `http://mtg.upf.edu/icmc2005`

En 2005, Xavier Serra, professeur à l'Université Pompeu Fabra (Barcelone) et responsable du comité de programme de la conférence internationale ICMC 2005 *(International Computer Music Conference)*, m'a demandé d'être le responsable de la session « analyse / synthèse sonores ». Cette conférence est la plus importante conférence internationale dans le domaine de l'informatique musicale. La responsabilité d'une session au sein du comité de programme consiste en le choix et la coordination d'une équipe de rapporteurs, l'évaluation des articles soumis, et la décision finale pour chaque article. Cette responsabilité se prolonge naturellement avec le rôle de *chairman* pour le pilotage des exposés oraux lors de la conférence.

Pour information, le taux d'acceptation d'ICMC 2005 est d'environ 50%, et plus de 300 articles ont été soumis.

Chairman **d'une session de la conférence DAFx 2006** `http://www.dafx.ca`

En 2006, Philippe Depalle, professeur à l'Université McGill (Montréal), m'a demandé d'animer (en tant que *chairman*) une session de la conférence internationale DAFx 2006 *(Digital Audio Effects)*.

Organisation de la conférence internationale DAFx 2007 `http://www.dafx.u-bordeaux.fr`

L'équipe Image et Son du LaBRI (Laboratoire Bordelais de Recherche en Informatique) et le SCRIME (Studio de Création et de Recherche en Informatique et Musique Électroacoustique) ont organisé, du 10 au 15 septembre 2007, la 10$^{\text{ème}}$ édition de la conférence internationale DAFx *(Digital Audio Effects)*. Membre du comité scientifique de cette conférence depuis 2006, j'étais en 2007 à la tête du comité d'organisation.

Cette conférence rassemble une communauté de chercheurs universitaires et industriels dans les divers domaines de l'audio numérique et de l'informatique musicale : modélisation sonore, analyse / synthèse, effets audionumériques et transformations sonores, perception, codage, fouille de données, spatialisation, séparation de sources, contrôle gestuel, techniques de composition musicale, etc.

Ce cycle de conférences, initié en 1998 grâce à une action européenne COST particulièrement fructueuse, se poursuit maintenant (depuis 2002) uniquement sur les financements récoltés par les organisateurs.

Chaque année est une nouvelle occasion de confirmer la réputation d'excellence des communications, ainsi que l'atmosphère ouverte et amicale propres à cette conférence. C'est dans une certaine mesure le pendant européen de la conférence IEEE WASPAA, tenue uniquement aux États-Unis les années impaires, en octobre, dans l'état de New York. Après Barcelone (1998), Trondheim (1999), Vérone (2000), Limerick (2001), Hambourg (2002), Londres (2003), Naples (2004) et Madrid (2005), DAFx est sortie de l'Europe géographique pour la première fois en 2006 à Montréal. Elle y est donc

[4] *European Cooperation in the Field of Scientific and Technical Research*

revenue en 2007 pour l'édition de Bordeaux, et devrait y rester encore quelques années (Espoo / Helsinki en 2008, Côme / Milan en 2009, Graz en 2010, Paris en 2011, York en 2012, Maynooth en 2013 et Erlangen en 2014).

L'édition 2007 a été un réel succès. Elle a rassemblé 90 participants provenant de 20 pays. Les communications (48 articles, 11 posters) se sont déroulées sur 4 jours, et une journée a été spécialement réservée pour la découverte de la région de Bordeaux. Cette conférence a donné lieu à la publication d'actes en langue anglaise, le comité de lecture étant international, et une édition spéciale du *Computer Music Journal* (MIT Press) a été consacrée à la conférence. Enfin, ce qui est d'autant plus appréciable pour une première expérience d'organisation d'un événement de cette ampleur, le budget est équilibré (43 kEuros).

Chairman d'une session de la conférence DAFx 2008 `http://www.acoustics.hut.fi/dafx08/`

En 2008, Vesa Välimäki, professeur à l'Université de Technologie d'Helsinki (TKK), m'a demandé d'animer (en tant que *chairman*) une session de la conférence internationale DAFx 2008 *(Digital Audio Effects)*.

Participation au projet ANR DESAM `http://www.tsi.enst.fr/~rbadeau/desam`

Je suis le responsable scientifique pour le LaBRI dans le projet ANR Jeunes-Chercheurs DESAM (Décomposition en Éléments Sonores et Applications Musicales), qui associe l'ENST (Paris), le STICS (Toulon), le LAM (Paris) et le LaBRI. J'y participe depuis novembre 2006 et pour 3 ans (à hauteur de 20%). J'apporte à ce projet mes compétences en modélisation sinusoïdale des sons musicaux, analyse spectrale et suivi de partiels.

Participation au projet ANR SIMBALS `http://simbals.labri.fr`

Je participe également au projet ANR Jeunes-Chercheurs SIMBALS *(SIMilarity Between Audio signaLS)* qui implique plusieurs équipes du LaBRI. J'y participe depuis novembre 2007 et pour 3 ans (à hauteur de 20%). J'apporte à ce projet mes compétences en séparation de sources sonores en vue de l'analyse automatique de la musique polyphonique.

Responsable du projet PEPS IBISA `http://dept-info.labri.fr/~sm/Projets/IBISA`

Depuis mai 2007, et pour 2 ans, je suis responsable du projet IBISA, dans le cadre des Projets Exploratoires Pluridisciplinaires (PEPS) du département ST2I du CNRS (participation à hauteur de 50%). C'est un projet qui manipule certes des images (signaux bidimensionnels, et non des sons), mais dans un domaine spectral qui m'est familier, via la transformée de Fourier-Mellin.

IBISA *(Image-Based Identification/Search for Archeology)* est un système logiciel vivement souhaité par une équipe pluridisciplinaire de jeunes chercheurs en informatique et en archéologie, appartenant à 3 UMR du CNRS réparties sur 2 départements (ST2I et SHS) et 2 universités de Bordeaux.

Plus précisément, le projet IBISA doit aboutir à une maquette logicielle (sous licence libre GPL) qui manipulera une base d'images numériques (quelques milliers) d'objets archéologiques (carreaux estampés glaçurés médiévaux et monnaies antiques grecques et romaines). Ces objets ont comme particularités communes d'avoir été produits dans le passé par l'empreinte d'une matrice et d'avoir ensuite subi une usure au cours du temps.

Le logiciel manipulera une base d'images numériques (quelques milliers). Lorsqu'un nouvel objet sera à étudier, son image C (image cible) sera prise à partir d'un scanner à plat ou d'un appareil photo numérique. Seront alors exécutées en séquence les fonctionnalités suivantes.

Les images de la base semblables à celle de l'image cible C cherchée seront automatiquement recherchées et présentées à l'utilisateur par similarité décroissante. Le système déterminera si les objets sont les mêmes, ou proviennent de la même matrice, ou présentent le même motif / style, ou bien sont vraiment différents. L'utilisateur pourra ainsi facilement décider de manière semi-automatique.

Il est important de noter que le système est résistant aux changements de conditions de prise de vue, et peut identifier un même objet à partir d'images prises différemment. La correction des conditions de prise de vue (centrage, orientation, échelle, mais aussi colorimétrie) se fera également de manière automatique. Notamment, si une image S semblable à C a été trouvée dans la base, cela se fera en recalant C sur S (via le recalage d'images). Le détourage de l'image pour isoler la monnaie du fond ou le motif du carreau pourra être effectué semi-automatiquement (via les modèles déformables).

Pour chaque classe d'images provenant de la même matrice, un représentant idéal sera construit automatiquement. Ce représentant serait l'objet le plus complet et dans l'état de conservation optimal, déduit par le système en prenant le meilleur dans chaque objet présent dans la classe. Dans le cas des carreaux, il s'agit de construire le motif complet, obtenu à partir de fragments. Dans le cas des monnaies, il s'agit de reconstruire l'image d'une monnaie sans usure ou presque. Cela revient à retrouver le motif de la matrice originelle (disparue) à partir des objets étudiés (parvenus jusqu'à nous).

A.2.6 Rayonnement scientifique

Prix

- Lauréat du **prix « jeune chercheur » 2000 de la SFIM** (Société Française d'Informatique Musicale) ;
- Accessit au **prix de thèse SPECIF 2001** ;
- Nominé par l'INRIA pour le prix ERCIM Cor Baayen 2002.

Éditeur associé IEEE Transactions

Sollicité par Mari Ostendorf, éditeur en chef, j'ai été nommé par l'IEEE en mars 2007 et pour 3 ans, éditeur associé *IEEE Transactions on Audio, Speech, and Language Processing*. Chaque mois, l'IEEE me confie en moyenne deux articles pour lesquels je dois trouver des rapporteurs et décider au final l'acceptation ou le rejet.

Comités de lecture de revues internationales

J'ai été sollicité par les éditeurs respectifs des revues internationales suivantes, aux années indiquées, pour des rapports sur des articles soumis :

- *Applied Signal Processing, the International Journal of Analog and Digital Signal Processing* (Springer, Londres), en 2000 ;
- *IEEE Transactions on Audio, Speech, and Language Processing*, plusieurs fois chaque année depuis 2003 ;
- *IEEE Transactions on Signal Processing* depuis 2006 ;
- *Perception & Psychophysics* en 2005 ;
- *Computer Speech and Language* (Elsevier) en 2007 ;
- *Computational Statistics and Data Analysis* (Elsevier) en 2007 ;

– *Signal, Image and Video Processing* en 2007 et 2008 ;
– *Computer Music Journal* (MIT Press) en 2007.

Comités de sélection de conférences

J'ai fait partie des comités de lecture des conférences suivantes :
– nationales :
  – CORESA (COmpression et REprésentation des Signaux Audiovisuels) 2005 et 2007,
  – JIM (Journées d'Informatique Musicale) 2007 et 2008 ;
– internationales :
  – ICMC *(International Computer Music Conference)* 2004, 2005 et 2007,
  – SMC *(Sound and Music Computing)* 2004, 2005 et 2006,
  – DAFx *(Digital Audio Effects)* 2006, 2007 et 2008,
  – EUSIPCO *EUropean SIgnal Processing COnference* 2008,
  – MMEDIA *International Conference on Advances in Multimedia* 2009.

Jurys de thèse

J'ai été membre (examinateur) des jurys de thèse suivants :
– **Patrice Collen** (École Nationale Supérieure des Télécommunications (ENST), novembre 2002) :
  « Techniques d'enrichissement de spectre des signaux audionumériques » ;
– **Mathieu Lagrange** (LaBRI, Université Bordeaux 1, décembre 2004) :
  « Modélisation sinusoïdale des sons polyphoniques » ;
– **Matthias Robine** (LaBRI, Université Bordeaux 1, décembre 2006) :
  « Analyse de la performance musicale et synthèse sonore rapide » ;
– **Martin Raspaud** (LaBRI, Université Bordeaux 1, mai 2007) :
  « Modèle spectral hiérarchique pour les sons et applications ».

Conférences et séminaires

J'ai eu l'occasion de présenter mes travaux lors de conférences d'audience internationale (*cf.* Publications). À ces exposés, il faut rajouter des séminaires invités en France (notamment LORIA, Nancy et INPG, Grenoble) ainsi qu'à l'étranger :
– CCRMA, *Stanford University*
  « High Precision Fourier Analysis of Sounds Using Signal Derivatives »
  exposé fait dans le cadre du Groupe de Travail « Digital Signal Processing »
  dirigé à Stanford par le Professeur Julius O. Smith III (juin 1998) ;
– Queen Mary, *University of London*
  « Sinusoidal Sound Modeling – Applications to Time Stretching and Sound Classification »
  exposé fait dans le cadre du Groupe de Travail du C4DM
  (Centre for Digital Music) (mars 2006) ;
– *McGill University*, Montréal
  « Advances in Sinusoidal Sound Modeling »
  exposé fait dans le cadre du séminaire du CIRMMT
  (Centre for Interdisciplinary Research in Music Media and Technology) (mars 2008) ;
– *Stanford University*
  « Advances in Sinusoidal Sound Modeling »

exposé fait dans le cadre du séminaire du CCRMA
(Center for Computer Research in Music and Acoustics) (mai 2008) ;
– *University of California, Santa Barbara*
« Advances in Sinusoidal Sound Modeling »
exposé dans le cadre du séminaire du CREATE
(Center for Research in Electronic Art Technology) (juin 2008).

A.3 Activités d'enseignement

Mes enseignements ont principalement été réalisés au sein de l'Université Bordeaux 1 : à l'IUT-A (Département Informatique), à l'ENSEIRB[5] (filière informatique principalement) et en Master Informatique Multimédia. Les cours de licence ont lieu dans des amphithéâtres d'approximativement 120 étudiants, tandis que les cours de Master ont lieu dans des salles de cours d'environ 40 étudiants.

De 1997 à 2007, j'ai enseigné plus de 1500 heures, avant d'être accueilli en délégation au CNRS pour la première fois en septembre 2007. Outre les nombreux encadrement de projets et de stages en Master Informatique 1ère et 2ème années et ENSEIRB 2ème année (totalisant 214 h eq. TD), mes enseignements se divisent principalement en deux catégories : les enseignements liés à mon domaine de recherche (son, mais aussi image) d'une part, et les enseignements d'informatique générale d'autre part.

A.3.1 Image et son

– **Introduction au son numérique** (**42 h CM + 62 h TD**)
(Licence Informatique 2ème année, Licence Pro Université Bordeaux 3)
Contenu : généralités sur le son (notions d'acoustique – émission, propagation, réception ; notions de psychoacoustique – perception), représentations temporelle et spectrale, formats de fichiers sonores (compressés ou non), utilisation de la carte son, paramètres sonores et musicaux, transformations du son musical et effets audionumériques.
– **Outils et modèles pour l'image et le son** (**56 h CM + 49 h TD**)
(Master Informatique Multimédia 2ème année)
Contenu : physique de la lumière et du son, discrétisation et reconstruction des signaux 1D et 2D, modélisation de la couleur, représentation de la couleur (systèmes XYZ, RGB, CMY, YIQ, HSV, HLS, Lab, Luv, etc.), modèles sonores, paramètres musicaux, notion de timbre, perception et cognition, outils théoriques pour l'image et le son (algèbre linéaire, probabilités, statistiques, domaines spatial / temporel et spectral / fréquentiel, représentations continue et discrète, convolution, transformées).
– **Analyse / synthèse du son musical** (**172 h CM + 144 h TD**)
(Master Informatique Multimédia 2ème année, ENSEIRB 3ème année)
Contenu détaillé :
  – introduction au son numérique :
  représentations temporelle et spectrale, formats de fichiers sonores (formats non compressés – WAV, AIFF, etc. ; formats compressés – codage μ-law, ADPCM, MPEG II niveaux 1 à 3, etc.), utilisation de la carte son ;

[5] École Nationale Supérieure d'Électronique, Informatique et Radiocommunications de Bordeaux

- paramètres sonores et musicaux :
  partiels, sons harmoniques, hauteur, intensité, timbre, couleur – enveloppe spectrale, brillance – centroïde spectrale, etc.
- analyse spectrale :
  analyse par transformée de Fourier (principes, défauts et améliorations), méthodes d'analyse spectrale haute précision (interpolation parabolique, vocodeur de phase, réallocation spectrale, Fourier à l'ordre 1, etc.) ;
- synthèse additive rapide :
  techniques de synthèse par transformée de Fourier inverse et oscillateurs numériques ;
- techniques de synthèse non linéaires :
  modulations de fréquence (FM) et d'amplitude (AM), etc.
- notions de psychoacoustique :
  échelles dB et Bark, seuil d'audibilité, phénomènes de masquage, application au codage MPEG niveau 3 et au tatouage de données ;
- spatialisation et séparation de sources :
  Head-Related Transfer Functions (HRTF), indices acoustiques (ILD, ITD), séparation de sources (technique DUET) ;
- transformations du son musical et effets audionumériques :
  amplification, filtrage, transposition, étirements temporels, hybridations, *morphing*, etc.
- protocole MIDI, langages pour la synthèse sonore.

A.3.2 Informatique généraliste

- **Architecture des ordinateurs** (**40 h CM + 143 h TD**)
  (Licence Informatique, 2ème année)
  Contenu : portes logiques, circuits combinatoires et séquentiels, arithmétique binaire, logique à trois états, mémoires, langages assembleur et machine, instructions, interruptions, cache, etc.
- **Systèmes d'exploitation** (**98 h TD**)
  (IUT, 1ère et 2ème années)
  Contenu : processus, substitution et duplication de processus, mécanismes de communication inter-processus, algorithmes d'ordonnancement, mécanisme d'interruption, *threads*, mémoire, mémoire partagée, accès concurrents, sémaphores, fichiers, gestion des ressources, périphériques, etc.
- **Utilisation des systèmes informatiques** (**30 h TD**)
  (IUP MIAGE, 2ème année)
  Contenu : exercices d'initiation au système d'exploitation UNIX, écriture de scripts.
- **Algorithmique et structures de données** (**128 h TD**)
  (IUT, 1ère année)
  Contenu : structures de contrôle élémentaires et types de base, itération, récursion, tri de tableaux, manipulation de listes, parcours d'arbres, etc.
- **Langages de programmation** (**32 h TD + 58,5 h TP**)
  (IUT, 1ère et 2ème années)
  Contenu : apprentissage du langage de programmation C, bases de la programmation orientée objet (objets, classes, méthodes, héritage, interfaces, etc.) et apprentissage des langages de programmation C++ et Java.
- **Projets de programmation** (**267 h TD + 49,5 h TP**)

(Master Informatique, 1ère année)
Contenu : suivi de projets, initiation au génie logiciel, conception et rédaction du cahier des charges, revue de code, rédaction de rapport, préparation de soutenance.

A.3.3 Master Informatique spécialité Image, Son, Multimédia

De 2005 à 2007, j'ai eu la responsabilité de la spécialité Image, Son, Multimédia du Master Informatique de l'Université Bordeaux 1, avec tout ce que cela implique (participation au recrutement des étudiants, coordination de l'équipe pédagogique, organisation de jurys, etc.). En 2006–2007, j'ai participé activement au renouvellement de l'habilitation de cette formation. Cette spécialité propose désormais 4 parcours :

- Codage, Traitement et Analyse ;
- Synthèse d'Images et Réalité Virtuelle ;
- Imagerie Médicale ;
- Informatique Musicale et Interaction.

A.4 Fonctions d'intérêt collectif

A.4.1 Responsabilités collectives nationales

J'ai été **membre élu du Conseil d'Administration de l'AFIM** (Association Française d'Informatique Musicale) de 2002 à 2003.

A.4.2 Responsabilités collectives locales (Université Bordeaux 1)

- **membre élu de la Commission de Spécialistes, section 27 du CNU**, collège B, depuis 2003 (d'abord suppléant, puis titulaire depuis 2005, reconduit en 2007) ;
- **membre élu du Conseil de Laboratoire du LaBRI** depuis 2007 ;
- **responsable des enseignements « son » du Département Informatique** de 2001 à 2007 ;
- **directeur des projets du SCRIME** en 2004–2005. Je dirigeais la commission chargée coordonner les projets du SCRIME, de définir la politique de leur développement ainsi que le partage des ressources. Le SCRIME est une cellule d'activité rassemblant artistes et scientifiques. Son objectif est de permettre aux premiers de bénéficier d'un transfert de connaissances scientifiques et aux seconds d'une expertise musicale. Le SCRIME résulte d'une convention de coopération entre le Conservatoire National de Région de Bordeaux, l'ENSEIRB, et l'Université Bordeaux 1. Les membres du SCRIME sont des chercheurs en informatique musicale du LaBRI et des compositeurs issus du Conservatoire National de Région de Bordeaux ;
- membre de la Commission de Scolarité du Premier Cycle en 2004 ;
- **responsable des stages du Master 2 Informatique** en 2004–2005 ;
- **responsable du Master 2 Informatique spécialité Image, Son, Multimédia** de 2005 à 2007.

A.5 Publications

Journaux internationaux (avec comité de sélection)

[1] Myriam Desainte-Catherine and Sylvain Marchand. High Precision Fourier Analysis of Sounds Using Signal Derivatives. *Journal of the Audio Engineering Society*, 48(7/8) :654–667, July/August 2000.

[2] Sylvain Marchand. Musical Audio Effects in the SAS Model. *Journal of New Music Research*, 30(3) :259–269, September 2001.

[3] Myriam Desainte-Catherine, György Kurtag, Sylvain Marchand, Catherine Semal, and Pierre Hanna. Playing With Sounds as Playing Video Games. *ACM Journal : Computers in Entertainment*, 2(2) :16, April/June 2004. (22 pages).

[4] Mathieu Lagrange, Sylvain Marchand, and Jean-Bernard Rault. Long Interpolation of Audio Signals Using Linear Prediction in Sinusoidal Modeling. *Journal of the Audio Engineering Society*, 53(10) :891–905, October 2005.

[5] Laurent Girin, Mohammad Firouzmand, and Sylvain Marchand. Perceptual Long-Term Variable-Rate Sinusoidal Modeling of Speech. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(3) :851–861, March 2007.

[6] Mathieu Lagrange and Sylvain Marchand. Estimating the Instantaneous Frequency of Sinusoidal Components Using Phase-Based Methods. *Journal of the Audio Engineering Society*, 55(5) :385–399, May 2007.

[7] Mathieu Lagrange, Sylvain Marchand, and Jean-Bernard Rault. Enhancing the Tracking of Partials for the Sinusoidal Modeling of Polyphonic Sounds. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(5) :1625–1634, July 2007.

Publications dans des ouvrages collectifs (livres)

[8] Sylvain Marchand. *Informatique musicale : du signal au signe musical*, chapter La synthèse additive, pages 67–117. Traité IC2, série Informatique et systèmes d'information. HERMES Sciences, Paris, May 2004. ISBN 2-7462-0825-3 (448 pages).

[9] Sylvain Marchand. *Informatique musicale : du signal au signe musical*, chapter L'analyse de Fourier, pages 365–402. Traité IC2, série Informatique et systèmes d'information. HERMES Sciences, Paris, May 2004. ISBN 2-7462-0825-3 (448 pages).

Conférences internationales (avec comité de programme et actes)

[10] Sylvain Marchand. Improving Spectral Analysis Precision with an Enhanced Phase Vocoder using Signal Derivatives. In *Proceedings of the Digital Audio Effects (DAFx'98) Workshop*, pages 114–118, Barcelona, Spain, November 1998. Audiovisual Institute, Pompeu Fabra University and COST (European Cooperation in the Field of Scientific and Technical Research).

[11] Myriam Desainte-Catherine and Sylvain Marchand. Vers un modèle pour unifier musique et son dans une composition multiéchelle. In *Proceedings of the Journées d'Informatique Musicale (JIM'99)*, pages 59–68, Paris, May 1999. CEMAMu.

[12] Robert Strandh and Sylvain Marchand. Real-Time Generation of Sound from Parameters of Additive Synthesis. In *Proceedings of the Journées d'Informatique Musicale (JIM'99)*, pages 83–88, Paris, May 1999. CEMAMu.

[13] Myriam Desainte-Catherine and Sylvain Marchand. Structured Additive Synthesis : Towards a Model of Sound Timbre and Electroacoustic Music Forms. In *Proceedings of the International Computer Music Conference (ICMC'99)*, pages 260–263, Beijing, China, October 1999. International Computer Music Association (ICMA).

[14] Sylvain Marchand and Robert Strandh. InSpect and ReSpect : Spectral Modeling, Analysis and Real-Time Synthesis Software Tools for Researchers and Composers. In *Proceedings of the International Computer Music Conference (ICMC'99)*, pages 341–344, Beijing, China, October 1999. International Computer Music Association (ICMA).

[15] Sylvain Marchand. Musical Sound Effects in the SAS Model. In *Proceedings of the Digital Audio Effects (DAFx'99) Workshop*, pages 139–142, Trondheim, December 1999. Norwegian University of Science and Technology (NTNU) and COST (European Cooperation in the Field of Scientific and Technical Research).

[16] Sylvain Marchand. ProSpect : une plate-forme logicielle pour l'exploration spectrale des sons et de la musique. In *Proceedings of the Journées d'Informatique Musicale (JIM'2000)*, pages 31–40, Bordeaux, May 2000. SCRIME – LaBRI, Université Bordeaux 1.

[17] Sylvain Marchand. Compression of Sinusoidal Modeling Parameters. In *Proceedings of the Digital Audio Effects (DAFx'2000) Conference*, pages 273–276, Verona, Italy, December 2000. Università degli Studi di Verona and COST (European Cooperation in the Field of Scientific and Technical Research).

[18] Jean-Christophe Gonzato and Sylvain Marchand. Photo-Realistic Simulation and Rendering of Halos. In Vaclav Skala, editor, *Proceedings of WSCG'2001, the 9-th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*, volume Short Communications (ISBN 80-7082-713-0), pages 106–113, Prague, Czech Republic, February 2001. University of West Bohemia.

[19] Jean-Christophe Gonzato and Sylvain Marchand. Efficient Simulation of Halos for Computer Graphics. In *Proceedings of ECSIA'2001, the 8-th European Congress for Stereology and Image Analysis*, page 181, Bordeaux, France, September 2001. LEPT / CDGA, CNAM.

[20] Mathieu Lagrange and Sylvain Marchand. Real-Time Additive Synthesis of Sound by Taking Advantage of Psychoacoustics. In *Proceedings of the Digital Audio Effects (DAFx'01) Conference*, pages 5–9, Limerick, Ireland, December 2001. University of Limerick and COST (European Cooperation in the Field of Scientific and Technical Research).

[21] Sylvain Marchand. An Efficient Pitch-Tracking Algorithm Using a Combination of Fourier Transforms. In *Proceedings of the Digital Audio Effects (DAFx'01) Conference*, pages 170–174, Limerick, Ireland, December 2001. University of Limerick and COST (European Cooperation in the Field of Scientific and Technical Research).

[22] Mathieu Lagrange, Sylvain Marchand, and Jean-Bernard Rault. Sinusoidal Parameter Extraction and Component Selection in a Non Stationary Model. In *Proceedings of the Digital Audio Effects (DAFx'02) Conference*, pages 59–64, Hamburg, Germany, September 2002. University of the Federal Armed Forces.

[23] Florian Keiler and Sylvain Marchand. Survey on Extraction of Sinusoids in Stationary Sounds. In *Proceedings of the Digital Audio Effects (DAFx'02) Conference*, pages 51–58, Hamburg, Germany, September 2002. University of the Federal Armed Forces.

[24] Mathieu Lagrange, Sylvain Marchand, Martin Raspaud, and Jean-Bernard Rault. Enhanced Partial Tracking Using Linear Prediction. In *Proceedings of the Digital Audio Effects*

(DAFx'03) Conference, pages 141–146, London, United Kingdom, September 2003. Queen Mary, University of London.

[25] Laurent Girin, Sylvain Marchand, Joseph di Martino, Axel Röbel, and Geoffroy Peeters. Comparing the Order of a Polynomial Phase Model for the Synthesis of Quasi-Harmonic Audio Signals. In *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA'03)*, pages 193–196, New Paltz, New York, USA, October 2003. Institute of Electrical and Electronics Engineers (IEEE).

[26] Mathieu Lagrange, Sylvain Marchand, and Jean-Bernard Rault. Partial Tracking Based on Future Trajectories Exploration. In *Proceedings of the 116-th Convention of the AES*, Berlin, Germany, May 2004. Audio Engineering Society (AES).

[27] Mathieu Lagrange, Sylvain Marchand, and Jean-Bernard Rault. Using Linear Prediction to Enhance the Tracking of Partials. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'04)*, volume 4, pages 241–244, Montreal, Quebec, Canada, May 2004. Institute of Electrical and Electronics Engineers (IEEE).

[28] Laurent Girin and Sylvain Marchand. Watermarking of Speech Signals Using the Sinusoidal Model and Frequency Modulation of the Partials. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'04)*, volume 1, pages 633–636, Montreal, Quebec, Canada, May 2004. Institute of Electrical and Electronics Engineers (IEEE).

[29] Laurent Girin, Mohammad Firouzmand, and Sylvain Marchand. Long Term Modeling of Phase Trajectories within the Speech Sinusoidal Model Framework. In *Proceedings of the INTERSPEECH – 8th International Conference on Spoken Language Processing (ICSLP'04)*, Jeju Island, Korea, October 2004.

[30] Sylvain Marchand and Martin Raspaud. Enhanced Time-Stretching Using Order-2 Sinusoidal Modeling. In *Proceedings of the Digital Audio Effects (DAFx'04) Conference*, pages 76–82, Naples, Italy, October 2004. Federico II University of Naples. ISBN : 88-901479-0-3.

[31] Mathieu Lagrange, Sylvain Marchand, and Jean-Bernard Rault. Tracking Partials for the Sinusoidal Modeling of Polyphonic Sounds. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'05)*, volume 3, pages 229–232, Philadelphia, New York, USA, March 2005. Institute of Electrical and Electronics Engineers (IEEE).

[32] Mohammad Firouzmand, Laurent Girin, and Sylvain Marchand. Comparing Several Models for Perceptual Long-Term Modeling of Amplitudes and Phase Trajectories of Sinusoidal Speech. In *Proceedings of the INTERSPEECH – EUROSPEECH Conference*, Lisboa, Portugal, September 2005.

[33] Vincent Goudard, Christophe Havel, Sylvain Marchand, and Myriam Desainte-Catherine. Data Anticipation for Gesture Recognition in the Air Percussion. In *Proceedings of the International Computer Music Conference (ICMC'05)*, pages 49–52, Barcelona, Spain, September 2005. International Computer Music Association (ICMA).

[34] Myriam Desainte-Catherine, Pierre Hanna, Christophe Havel, Gyorgy Kurtag, Mathieu Lagrange, Sylvain Marchand, Edgard Nicouleau, Martin Raspaud, Matthias Robine, and Robert Strandh. SCRIME Studio Report. In *Proceedings of the International Computer Music Conference (ICMC'05)*, pages 515–518, Barcelona, Spain, September 2005. International Computer Music Association (ICMA).

[35] Martin Raspaud, Sylvain Marchand, and Laurent Girin. A Generalized Polynomial and Sinusoidal Model for Partial Tracking and Time Stretching. In *Proceedings of the Digital Audio Effects (DAFx'05) Conference*, pages 24–29, Madrid, Spain, September 2005. Universitad Politécnica de Madrid. ISBN : 84-7402-318-1.

[36] Mathieu Lagrange, Sylvain Marchand, and Jean-Bernard Rault. Improving Sinusoidal Frequency Estimation Using a Trigonometric Approach. In *Proceedings of the Digital Audio Effects (DAFx'05) Conference*, pages 110–115, Madrid, Spain, September 2005. Universitad Politécnica de Madrid. ISBN : 84-7402-318-1.

[37] Sylvain Marchand and Mathieu Lagrange. On the Equivalence of Phase-Based Methods for the Estimation of Instantaneous Frequency. In *Proceedings of the 14th European Conference on Signal Processing (EUSIPCO'2006)*, Florence, Italy, September 2006. EURASIP.

[38] Matthias Robine, Robert Strandh, and Sylvain Marchand. Fast Additive Sound Synthesis Using Polynomials. In *Proceedings of the Digital Audio Effects (DAFx'06) Conference*, pages 181–186, Montreal, Quebec, Canada, September 2006. McGill University.

[39] Guillaume Meurisse, Pierre Hanna, and Sylvain Marchand. A New Analysis Method for Sinusoids+Noise Spectral Models. In *Proceedings of the Digital Audio Effects (DAFx'06) Conference*, pages 139–144, Montreal, Quebec, Canada, September 2006. McGill University.

[40] Joan Mouba and Sylvain Marchand. A Source Localization/Separation/Respatialization System Based on Unsupervised Classification of Interaural Cues. In *Proceedings of the Digital Audio Effects (DAFx'06) Conference*, pages 233–238, Montreal, Quebec, Canada, September 2006. McGill University.

[41] Mathieu Lagrange and Sylvain Marchand. Assessing the Quality of the Extraction and Tracking of Sinusoidal Components : Towards an Evaluation Methodology. In *Proceedings of the Digital Audio Effects (DAFx'06) Conference*, pages 239–245, Montreal, Quebec, Canada, September 2006. McGill University.

[42] Myriam Desainte-Catherine, Sylvain Marchand, Pierre Hanna, Mathieu Lagrange, Matthias Robine, Martin Raspaud, Robert Strandh, Antoine Allombert, Guillaume Meurisse, Joan Mouba, Jean-Louis Di Santo, and Gyorgy Kurtag. SCRIME Studio Report. In *Proceedings of the International Computer Music Conference (ICMC'07)*, pages 317–320, Copenhagen, Denmark, August 2007. International Computer Music Association (ICMA).

[43] Martin Raspaud and Sylvain Marchand. Enhanced Resampling for Sinusoidal Modeling Parameters. In *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA'07)*, pages 327–330, New Paltz, New York, USA, October 2007. Institute of Electrical and Electronics Engineers (IEEE).

[44] Fabien Gallot, Owen Lagadec, Myriam Desainte-Catherine, and Sylvain Marchand. iKlax : a New Musical Audio Format for Active Listening. In *Proceedings of the International Computer Music Conference (ICMC'08)*, pages 85–88, Belfast, Ireland, August 2008. International Computer Music Association (ICMA).

[45] Joan Mouba, Sylvain Marchand, Boris Mansencal, and Jean-Michel Rivet. RetroSpat : a Perception-Based System for Semi-Automatic Diffusion of Acousmatic Music. In *Proceedings of the Sound and Music Computing (SMC'08) Conference*, pages 33–40, Berlin, Germany, July/August 2008.

[46] Sylvain Marchand and Philippe Depalle. Generalization of the Derivative Analysis Method to Non-Stationary Sinusoidal Modeling. In *Proceedings of the Digital Audio Effects (DAFx'08)*

Conference, pages 281–288, Espoo, Finland, September 2008. TKK, Helsinki University of Technology.

Conférences nationales (avec comité de programme et actes)

[47] Myriam Desainte-Catherine, Bénédicte Gourdon, György Kurtag, Sylvain Marchand, and Catherine Semal. DOLABIP : un éveil musical avec l'ordinateur. In *Actes du Colloque « Apprendre avec l'ordinateur à l'école »*, Bordeaux, January 2002. Université Bordeaux 2.

[48] Sylvain Marchand. ReSpect : A Free Software Library for Spectral Sound Synthesis. In *Proceedings of the Journées d'Informatique Musicale (JIM'07)*, pages 33–43, Lyon, April 2007. GRAME.

Conférences invitées (internationales)

[49] Anthony Beurivé and Sylvain Marchand. Music Composition with Spectral Sounds. Rencontres Mondiales du Logiciel Libre – Libre Software Meeting, July 2000.

[50] Sylvain Marchand and Myriam Desainte-Catherine. Spectral Modeling, Analysis, and Synthesis of Musical Sounds. *Journal of the Acoustical Society of America*, 112(5, part 2/2) :2237–2238, November 2002. 144th Meeting of the Acoustical Society of America and First Pan-American / Iberian Meeting on Acoustics, Cancun, Mexico. December 2002.

[51] Sylvain Marchand. Sinusoidal Modeling for Speech and Musical Audio Signals. Séminaire France Télécom : Compression et Indexation Audio, June 2003.

[52] Sylvain Marchand. Advances in the Tracking of Partials for the Sinusoidal Modeling of Musical Sounds. *Journal of the Acoustical Society of America*, 123(5) :3803, May 2008. Acoustics'08 (155th Meeting of the Acoustical Society of America, 5th FORUM ACUSTICUM, and 9th Congrès Français d'Acoustique), Paris, France. June/July 2008.

Divers

[53] Sylvain Marchand. *Modélisation informatique du son musical (analyse, transformation, synthèse) / Sound Models for Computer Music (analysis, transformation, and synthesis of musical sound)*. PhD thesis, Université Bordeaux 1, F-33405 Talence cedex, December 2000.

[54] Sylvain Marchand. Le problème du consensus en environnement distribué asynchrone. Master's thesis, Université Bordeaux 1 (LaBRI) / École Polytechnique (LIX), F-33405 Talence cedex, June 1996.