Plan Vraisemblance Temps-Fréquence pour la poursuite des partiels

(dans le cadre de la transcription automatique de partitions)

Vincent Verfaille

ENST
Maurice Charbit
Pierre Duhamel

DEA ATIAM
Université Paris VI
Juillet 2000

Table des matières

Remerciements

Je tiens tout d'abord à remercier Pierre Duhamel, tant pour m'avoir proposé ce sujet de stage que pour son encadrement ainsi que ses conseils avisés.

De même, je remercie vivement Maurice Charbit pour les nombreuses aides théoriques et pratiques apportées tout au long du stage.

J'en profite pour saluer Carine et son travail sur le banc de filtres ainsi que sur les "climbers", qui m'a été d'un grand secours.

Merci aux thésards du bureau C 125 pour l'accueil chaleureux (Pierre, Anahid et Jamal, pour ne pas les citer), aux stagiaires de tous horizons (Emilia, Caroline, Yohann, Alexandre, Thomas, Hichem) pour leur soutien, leurs conseils, leur bonne humeur et leurs préoccupations musicales.

Je pense aussi à ceux sans qui LaTeX n'aurait jamais osé revenir et s'installer sur mon Mac (Gérard, Leslie et bien d'autres).

Sans toutes ces personnes, qui sait ce qu'il en aurait été de cette expérience de la recherche que fut pour moi ce stage de DEA à l'ENST.

Présentation de l'ENST

L'Ecole Nationale Supérieure des Télécommunications appartient au groupe des écoles des télécommunications, établissement public administratif qui comporte également l'Ecole Nationale Supérieure des Télécommunications de Bretagne et l'Institut National des Télécommunications.

Elle appartient également au groupe des écoles d'ingénieurs de Paris (GEI Paris) qui rassemble les huit autres écoles suivantes : Ecole Nationale Supérieure d'Agronomie, Arts et Métiers, Ecole de Chimie de Paris, Ecole des Eaux et Forêts, Ecole du Génie Rural, Ecole des Mines, Ecole de Physique Chimie de Paris, Ecole des Ponts et Techniques Avancées.

Il y quatre départements d'Enseignement et de Recherche :

- communications et électronique,
- économie, gestion, sciences sociales et humaines ;
- traitement du signal et des images ;
- informatique et réseaux.

Le département TSI

Le département TSI a pour missions l'enseignement, la recherche, et la formation par la recherche, dans les domaines du traitement du signal et des images et de leurs applications dans divers contextes de la société de l'information, dont les télécommunications.

Ses missions se déclinent comme suit :

- la formation de base : il s'agit de fournir à tous les élèves de l'ENST la maîtrise de certaines notions de base de mathématiques, de probabilités et de traitement du signal et des images ;
- la formation spécialisée afin que certains élèves appréhendent, à divers niveaux d'expertise, les concepts qui leur permettront de traiter les signaux et les images et de participer activement aux secteurs de l'industrie et de la recherche qui les mettent en œuvre ;
- la formation par la recherche pour laquelle le département TSI est fournisseur à la fois de formation (enseignements de DEA, enseignements doctoraux, séminaires départementaux) et de prestation de recherche (auprès des industriels ou des organismes nationaux et internationaux promoteurs de la recherche). Ces enseignements renforcent l'insertion de TSI dans la communauté universitaire de l'enseignement supérieur ;

- la recherche méthodologique et fondamentale, en relation étroite avec les organismes nationaux et internationaux de coordination de la recherche et en particulier le CNRS ; elle permet au département de contribuer à l'innovation par la découverte de concepts nouveaux ;
- la recherche appliquée, souvent menée en collaboration avec des partenaires industriels français ou étrangers, elle garantit un contact permanent avec les technologies émergentes ainsi qu'avec les nouveaux usages.

Le département TSI participe au rayonnement de l'ENST en la représentant, dans son domaine d'activité, auprès des différentes instances et des organismes nationaux ou internationaux (CNRS, IEEE, RNRT, etc.) et en participant de façon active à la vie scientifique nationale.

L'organisation du département TSI

Le département TSI est organisé en 5 groupes. Certains groupes sont plus orientés vers la recherche académique, d'autres vers la recherche appliquée, d'autres enfin vers l'enseignement. Chacun de ces groupes contribue cependant à l'ensemble des missions du département.

- groupe *Traitement et Interprétation des images* : ce groupe conduit des recherches sur la mise en œuvre de schémas complets de traitement, d'analyse et d'interprétation d'images, en particulier de scènes complexes. Les domaines d'application des méthodes théoriques qui y sont développées concernent en particulier :
  + l'imagerie médicale et notamment l'imagerie cérébrale anatomique et fonctionnelle ;
  + l'imagerie aérienne et satellitale avec un intérêt particulier sur l'imagerie radar et l'imagerie aérienne à très haute résolution des milieux urbains ; et
  + a description des objets complexes tridimensionnels pour leur analyse ou leur représentation fixe ou animée.
- groupe *Traitements statistiques et applications aux communications* : ce groupe travaille dans les domaines suivants :
  + le signal pour les communications ;
  + la séparation de sources ;
  + la modélisation statistique pour le signal et l'image pour laquelle on peut mettre en évidence deux voies de recherche : la reconstruction et la restauration d'images, le télétraffic (analyse et modélisation).
- groupe *Perception, Apprentissage et Modélisation* : ce groupe étudie le rôle des facteurs humains dans l'accès aux divers types d'information :
  + la parole : reconnaissance et identification de locuteurs ;

  + l'image : psychovision (perception du contraste, de la couleur, du relief), et imagerie de très haute qualité ;
  + l'écrit : fax, structuration des documents ;
  + la fusion des modalités perceptives dans l'appréhension de l'environnement ;
  + les interfaces multimodales.

- groupe *Codage* : ce groupe travaille sur des techniques éprouvées de compression de sources ainsi que sur leur adaptation aux applications de l'audiovisuel et du multimédia :

  + la compression de parole et de son ;
  + le codage d'images ;
  + la transmission audiovisuelle sur réseau ;
  + les systèmes temps réel et l'adaptation source-canal ;

- groupe *Audio, Acoustique et Ondes* : ce groupe étudie la physique des ondes dans les deux domaines de l'optique et de l'acoustique.

  + en acoustique : modélisation de la production des sons, perception (psychoacoustique et prothèses auditives) et antennes acoustiques ;
  + en optique : stockage de l'information dans les milieux optiques réinscriptibles.

Chapitre 1

Introduction

On entend par transcription automatique de partition le fait d'extraire d'un signal audio donné (au format AIFF, AU, WAV ou autres) une représentation symbolique contenant les mêmes informations que le signal, en termes de caractéristiques temporelles et fréquentielles. Cette représentation peut être une partition, un fichier MIDI, etc. Dans tout les cas, elle correspond à une représentation temps-événement : des événements musicaux, clairement définis par leurs propriétés spectrales et leurs évolutions temporelles, sont placés sur une grille temporelle.

Une analyse temps-fréquence adaptée permet de connaître les propriétés des événements sonores, à condition d'avoir des connaissances à priori sur le signal. Ainsi, un signal de parole ne se traitera pas de la même façon qu'un signal d'instrument harmonique, de même qu'un mélange de signaux d'instruments harmoniques identiques (par exemple un duo de guitare) ne se traitera pas de la même manière qu'un signal d'instrument percussif.

A l'heure actuelle, aucun système n'est capable de transcrire automatiquement une pièce musicale inconnue (c'est-à-dire sans connaissance à priori sur le signal) contenant plusieurs voix dont les spectres se mélangent. Cependant, un certain nombre de méthodes existent et permettent de réaliser la transcription sous certaines restrictions : signal monophonique à une seule voix, duo entre deux instruments monophoniques aux tessitures séparées, instrument polyphonique (piano), etc. Nous aimerions à terme du projet aboutir à un système pouvant transcrire une pièce polyphonique et stéréophonique.

La première étape de la transcription automatique consiste à réaliser une description temps-fréquence du signal : c'est le cadre de ce stage. Ensuite, l'information recueillie doit être traitée afin d'extraire les paramètres du son, tant sur le plan temporel (début et fin de chaque note) que sur le plan fréquentiel (timbre).

Dans cette étude, nous allons mettre au point un outil d'analyse temps-fréquence adapté à cette problématique, en vue d'une détection de l'information fréquentielle présente dans un signal. Une analyse par octave à l'aide d'un banc de filtre à coefficients constants est proposée, suivie d'une évaluation du maximum de vraisemblance d'une hypothèse harmonique sur le signal. Pour la poursuite des partiels dans le plan vraisemblance temps-fréquence proposé, l'algorithme des "climbers" sera adapté et testé. Enfin, une direction pour la suite de l'étude sera proposée.

Chapitre 2

Positionnement du problème

Préalablement à la description des méthodes actuellement mises en œuvre dans le cadre de la transcription automatique de partition, nous allons présenter les motivations qui nous poussent à résoudre cette problématique, ainsi que l'étendue du travail à accomplir.

2.1 Motivations

La transcription classique d'une œuvre musicale, réalisée par le compositeur lui-même ou par une autre personne chargée de la retranscrire (à l'écoute, d'après les documents originaux ou toute infirmation pertinente laissée par le compositeur), en permet la divulgation, autant de manière géographique (connaissance d'un auteur par ses contemporains, même éloignés géographiquement parlant) que temporellement (conservation d'une trace concernant des œuvres anciennes).

La transcription automatique offre quant à elle de nouveaux horizons : d'une part, c'est un outil pour mener à bien, plus facilement et plus rapidement la tâche de retranscription ; d'autre part, elle permet de nouvelles applications.

2.1.1 Une simplification des applications de transcription classique

Un certain nombre de types de transcriptions gagneraient à être automatisées. On pense notamment aux partitions vendues dans le commerce, à l'usage privé qui est fait des partitions par les musiciens, et à l'usage protecteur qui en est fait par des organismes de respect des droits d'auteur.

Usage commercial

Une partition destinée au public est commercialisée après différentes phases. Tout d'abord, une phase d'écoute du morceau et de saisie de la partition est réalisée. En vue de l'impression, la partition est réécrite à l'aide d'un logiciel spécialisé, qui en crée finalement une image.

La phase de saisie et de réécriture pourrait être supprimée dès lors qu'un système de transcription automatique existera : ne restera qu'une phase de relecture et de corrections, d'annotations et de mise en page finale. En effet, une phase de corrections reste envisageable : à l'instar des logiciels de reconnaissance de caractères à partir d'images scannées, on admet un petit nombre de fautes soient commises. Il est plus aisé de corriger quelques erreurs que de saisir en totalité un texte. Il en sera de même pour une transcription de partition.

Usage privé

Beaucoup de musiciens cherchent à retranscrire de la musique, soit celle d'autres musiciens qu'ils aimeraient ajouter à leur répertoire, soit la leur lorsqu'elle n'a pas encore été écrite, et ceci afin de l'archiver, d'en faire des copies pour la jouer en ensembles. Il peut aussi s'agir de musique improvisée, tel le jazz, dont ils désirent extraire la trame harmonique, ou des parties de solos, etc.

Tout ceci pourrait être rendu plus facile et accessible aux néophytes, mais aussi permettrait un travail plus rapide aux musiciens confirmés, ce qui leur laisserait plus de temps pour se concentrer sur d'autres priorités (travailler leur technique, composer; etc.).

Usage protecteur

Les organismes de protection des droits d'auteur, telle la S.A.C.E.M., utilisent (bien que ceci ait tendance à disparaître avec les musiques électroniques) des fiches descriptives de chaque morceau, indiquant la ligne mélodique selon une représentation classique. Ces fiches sont écrites par des opérateurs humains. Un système automatisé faciliterait donc cette tâche. De plus, un système d'évaluation de proximité entre deux mélodies (calcul d'une distance) permettrait de détecter d'éventuels plagiats.

2.1.2 Transcription de musiques inconnues

Si l'on se place dans le cadre de la transcription complète, c'est-à-dire de tous les instruments présents, l'automatisation permettrait de transcrire des musiques non encore transcrites.

Musiques improvisées

Dans le jazz ou le rock, il serait possible, comme expliqué précédemment, d'extraire un accompagnement ou une ligne mélodique. Ainsi, le déchiffrage de constructions complexes et de solos improvisés serait tout à fait envisageable.

Accompagnement

On peut aussi s'intéresser à un accompagnement automatique, où le morceau est d'abord analysé puis rejoué, dans sa forme originale ou modifiée par l'utilisateur.

D'autres applications dans le domaine de l'apprentisage de la musique et de l'étude de la musique sont imaginables à partir de la transcription automatique.

Aide à la composition

En outre, ce serait une excellente aide à la composition. En effet, on pourrait s'affranchir de la connaissance des systèmes de représentation de la musique si l'on suppose qu'un ordinateur peut le faire.

Par exemple en musique électronique, le GRM propose aux compositeurs des représentations adaptées à leur façon de composer et aux sons utilisés (ne correpondant pas forcément à des instruments traditionnels). Une fois cette représentation connue par le programme de transcription automatique et la phase d'apprentissage réalisée, on s'attend à ce qu'il puisse directement écrire selon ce code d'autres pièces du même compositeur.

On peut aussi envisager toute sorte de moyens d'éditions d'une superposition de mélodies enregistrées par un seul individu, ou même d'une improvisation créative jouée par un ensemble.

Reconnaissance de mélodies

Un autre application existe dans la recherche dans une base de données. Des travaux d'indexation de signaux sonores dans ce but sont réalisés (notamment par S. Rossignol et X. Rodet, dans [**?**]).

On peut imaginer cette recherche se faisant directement par comparaison de lignes mélodiques et non d'index, ou encore appliquée à des sons non instrumentaux. Dans [**?**], cette approche a déjà été envisagée. Dans le système qu'ils proposent, il est possible d'indexer une base de données d'extraits sonores pour l'essentiel constitués de bruitages (bruits concrets, cris d'animaux, sons de la vie quotidienne, ...) par description d'un son ou par reproduction sonore.

Dans le même ordre d'idées, un discriminateur musique / parole permettrait de filtrer les passages parlés (détection des interruptions de programme musical radiophonique par de la publicité, par exemple, bien que ceci ne soit pas commercialement viable). Dans [**?**], des travaux sont en cours sur cette problématique.

Instruments inconnus

En supposant qu'un système complet de transcription voie le jour, il serait intéressant de pouvoir représenter sous forme de partition des morceaux utilisant des instruments inconnus mais qui soient descriptibles sous la forme de partitions généralisées. de la même manière que certains compositeurs indiquent l'emplacement et les déplacements des sources sonores sur leurs partitions, le système de transcription pourrait prendre en compte ces informations (stéréophoniques, quadriphoniques ou selon tout autre format du son) et ajouter une description des timbres des instruments utilisés.

Mais n'éxagérons rien...

La tâche dont nous avons présenté bon nombre d'avantages et d'avancées pour le musicien, le compositeur, l'éditeur de partitions ou le protecteur demeure très difficile à réaliser. Les systèmes existants se heurtent encore à des difficultés nées des différences existant entre les signaux musicaux et ceux de parole ou de l'écriture, pour lesquelles la recherche est nettement plus avancée.

2.2 Transcription de musique - Reconnaissance de la parole

Des applications de la reconnaissance de la parole existent déjà dans le domaine grand public. Par exemple, les téléphones portables peuvent reconnaître un nom ou prénom et composer le numéro correspondant.

Les variations de hauteur pour la parole sont environ d'un octave, alors que pour la musique et la voix chantée, elle s'étendent sur plusieurs octaves. Le domaine de fréquences fondamentales couvert est donc nettement plus étendu en musique qu'en parole, où le problème d'octave est alors extrêmement réduit.

D'autre part, en musique, on s'intéresse autant à l'aspect temporel des événements qu'a l'aspect fréquentiel. Cependant, toute représentation Temps-Fréquence admet une limite de précision similaire à celle d'Heisenberg.

En mécanique quantique, on ne peut indiquer simultanément et avec précision la position et la vitesse d'une particule. En traitement du signal, on ne peut donner simultanément et avec précision la date d'un événement et la fréquence à laquelle il se produit. Cette limite est de la forme $\Delta t \ \Delta f \geq C^{te}$.

Ainsi, une grande résolution en fréquence nous donnera très précisément les notes jouées, mais leur disposition temporelle nous sera mal connue. A l'inverse, dans le cas d'une très bonne quantification temporelle, les indications fréquentielles seront imprécises quant aux notes jouées. L'une des subtilités des méthodes réside en ce compromis à trouver.

2.3 Musique polyphonique - musique monophonique

Contrairement au traitement de la parole, il est très souvent nécessaire de considérer un contexte multivoix, sinon les applications de la transcription se limiteraient à un traitement piste par piste en studio, par exemple, où à des enregistrements de solistes.

Pour traiter la musique monophonique, les algorithmes proposés reconnaissent les fréquences fondamentales avec une bonne précision (moins de 5% d'erreurs dans certains cas). En revanche, beaucoup de choses restent à faire concernant la musique polyphonique.

2.3.1 Détermination du nombre de voix

La première difficulté réside en la détermination du nombre de sources présentes dans le signal. Puisque l'on ne dispose que d'une source sonore (éventuellement stéréophonique) dans laquelle toutes les voix se retrouvent mélangées, les harmoniques des différentes notes sont inévitablement mélangées. En particuler, la fréquence fondamentale d'une note peut facilement se confondre avec une harmonique d'une autre note.

Prenons l'exemple suivant : un $Do4$ et un $Sol5$ sont joués simultanément. Les fréquences des harmoniques pour chacune des notes sont :

	$f_1(Hz)$	$f_2(Hz)$	$f_3(Hz)$
Do 4	261.6	523.2	784.8
Sol 5	784.3	1568.6	2352.9

La troisième harmonique de $Do4$ et la fréquence fondamentale de $Sol5$ ne différent que de $0.5Hz$, ce qui est très difficilement séparable avec une analyse de Fourier classique.

On imagine bien que le nombre d'instruments croissant, les harmoniques présentent dans le signal pourront provenir simultanément de plusieurs instruments. La détermination du nombre de voix sera donc difficile à mettre en œuvre, d'autant plus que certains instruments sont monophoniques tandis que d'autres sont polyphoniques. Plus encore, tout instrument ne joue pas du début à la fin du morceau à transcrire : il faut donc détecter les moments auxquels tel ou tel instrument est présent.

2.3.2 Les différences de niveau de puissance entre voix

Lorsqu'un instrument joue plus fort que les autres (un soliste, par exemple), un phénomène de masquage apparaît : les harmoniques des instruments de niveau d'enregistrement plus faible risque d'être considérées comme du bruit et non comme de l'information utile, par comparaison avec celles de l'instrument de niveau plus fort.

Rappellons que l'oreille et particulièrement sensible aux phénomènes de masquage, à la fois fréquentiel et temporel. Ainsi, l'effet du masquage fréquentiel est largement utilisé dans le cadre de systèmes de compression de signal audio. Le fait de disposer de toute l'information présente dans le signal, mais pas forcément perceptible, est certes un avantage. Cependant, l'oreille humaine et tout le système nerveux concerné par les signaux acoustiques forment un système extrêmement complexe, très efficace pour la transcription. On ne s'attend pas à le surpasser si vite, uniquement avec cette information supplémentaire dont on dispose.

2.3.3 Problème des tessitures mêlées

Lorsque deux instruments jouent dans des zones fréquentielles communes et que l'on arrive à détecter les notes jouées, il faut encore être capable d'attribuer chaque note au bon instrument. On souhaitera donc savoir séparer les différents avant de chercher à savoir quelles notes sont présentes dans le signal. En fait, un grand nombre d'auteurs proposent cette approche dans leurs travaux, en ceci qu'ils procèdent à de drastiques restrictions sur le type de musique et d'instruments dès le début.

2.3.4 Problème d'octave

Lorsqu'on connaît une suite de partiels et qu'on veut déterminer la note jouée, c'est-à-dire la fréquence fondamentale correspondante, on peut comparer cette répartition des harmoniques à un gabarit et déterminer la note jouée. De là à donner précisément l'octave… Pour certains instruments, la fréquence fondamentale est d'amplitude négligeable par rapport aux partiels suivants, elle n'est donc pas forcément présente (ou mesurable) dans le spectre : on parle alors du phénomène de fondamentale cachée.

On considère le plus souvent que la fréquence fondamentale correspond au plus grand diviseur commun des fréquences des différents partiels. Il faut cependant être en mesure d'attibuer à chaque instrument ses partiels avant de s'autoriser l'utilisation de ce critère.

Plus subtil encore est le cas où plusieurs voix jouent une même note décalée d'une ou plusieurs octaves. Nous risquons de considérer que seule la note de fondamentale la plus grave est jouée, puisque tous les partiels de la note la plus haute seront confondus avec ceux de la note la plus grave, et pourront être considérées comme partiels de cette dernière.

Une fois encore, sans information à priori sur les instruments, il semble très difficile de détecter deux notes séparées d'une ou plusieurs octaves.

2.3.5 La réverbération

La réverbération présente dans les enregistrements pose un problème, en ceci qu'un long temps de réverbération agira de même qu'une source sonore suplémentaire, c'est-à-dire comme une nouvelle voix, qu plus est, bruitée.

Plus généralement, les conditions d'enristrements doivent être connues afin de maîtriser le plus grand nombre de paramêtres possible intervenant dans la transcription.

Chapitre 3

Eléments de la transcription automatique

3.1 Problématique de la reconnaissance de notes

Dans la musique occidentale, les instruments utilisés présentent tous des spectres caractéristiques, où les partiels sont tous multiples d'une fréquence dite fondamentale ; on les appelle alors harmoniques. Dans le cas où les harmoniques de plusieurs notes se trouvent mélangées, ou lorsqu'elles ne sont pas exactement multiples de la fondamentale, on les appelle plutôt **partiels**.

Les premiers travaux dans le domaine de la transcription automatique et portant principalement sur la reconnaissance de notes utilisaient des représentations temps-fréquence classiques qui ne tiennent pas compte de cette propriété. Elle ne servait qu'à déterminer la fondamentale, mais en aucun cas ne servait d'indication quant à la méthodologie la plus pratique.

La méthode de l'histogramme de Scrœder ([**?**]), examine les fréquences de toutes les harmoniques afin de trouver une fréquence fondamentale pouvant les engendrer toutes par comparaison avec des données tabulées préalablement.

Dix ans plus tard, Piszczalski et Galler proposent un système de poursuite de partiels (dans [**?**] et [**?**]) tandis que Terhardt met au point un algorithme de recherche de la fondamentale (cf. [**?**]) toujours utilisé aujourd'hui.

Les systèmes de poursuite de partiels utilisés travaillaient avec une analyse de Fourier classique. Pourtant, les représentations temps-fréquence classiques présentent des limites rédhibitoires.

Par contre, dans le domaine log-fréquentiel, les harmoniques ont des espacements indépendants de la fréquence fondamentale. Par exemple, l'écart entre la fondamentale (ou première harmonique) et la deuxième harmonique est de $log(2)$, l'écart entre la deuxième harmonique et la troisième vaut $log(3/2)$, etc. Une échelle logarithmique semble donc bien plus indiquée pour effectuer la poursuite des partiels, ce que ne permet pas une transformée de Fourier standard.

Cette propriété d'échelle est utilisée depuis quelques années dans les méthodes de reconnaissance de fondamentale.

Nous allons d'abord montrer où se posent le problème lié à l'utilisation de transformées de Fourier classiques, puis présenter les méthodes utilisant des transformées adaptées.

3.2 De la nécessité d'une représentation temps-fréquence adaptée à la problématique de la détection de partiels

Nous allons présenter ici quelques limites de l'analyse de Fourier classique, qui peuvent se lever en adoptant une représentation temps-fréquence adaptée, que l'on présentera ultérieurement.

3.2.1 Ségrégation des basses fréquences

Si l'on effectue une analyse de Fourier classique, la ségrégation de deux notes graves nécessite une grande précision en fréquence. Par exemple pour le piano, le *La* 0 à 27.5 Hz et le *Si b* 0 à 29.1 Hz ne diffèrente que de 1.6 Hz. Il faut alors au minimum $\frac{1}{1.6} = 0.6\ s$ d'échantillon sonore pour différencier ces deux notes distantes d'un demi-ton.

Cependant, dans une pièce musicale, une ligne mélodique (arpégée par exemple) se constitue aisément de 10 notes par secondes (et jusqu'à une vingtaine pour des solos), ce qui implique des durées de notes inférieures à 0.1 s, ce qui est très inférieur à la limite temporelle de différentiation de notes graves évoquée précédemment.

3.2.2 Espacement des hautes fréquences

Une fois la limite de ségrégation de bases fréquences fixées, on peut réaliser une analyse en fréquences classique. Cependant, le raffinement nécessaire en basses fréquences ne l'est plus en hautes fréquences.

Prenons à nouveau l'exemple du piano. Les deux dernières notes (séparées elles aussi d'un demi-ton) sont *Si* 7 (3951 Hz) et *Do* 8 (4186 Hz), c'est-à-dire distantes de 235 Hz, soit 146 bandes fréquentielles larges de 1.6 Hz. L'information fréquentielle présente dans ces 146 bandes est très peu utile.

3.2.3 Résolution fréquentielle

La résolution r en fréquence est constante et égale à la fréquence d'échantillonnage F_e divisé par la taille de la fenêtre d'analyse N (en nombre de points) : $r = F_e/N$. Ainsi, la précision sera trop faible en basses fréquences et exagérément élevée en hautes fréquences.

Prenons l'exemple d'une fenêtre d'analyse de $N = 1024$ points, et d'une fréquence d'échantillonnage de 44 100 échantillons par seconde. La résolution est alors de 43.1 Hz, ce qui ne permet pas de discerner les deux notes les plus basses du piano (séparées de 1.6 Hz). A titre d'exemple, voici les résolutions pour différentes notes séparées d'une octave :

Note	*La* 0	*La* 1	*La* 2	*La* 3	*La* 4	*La* 5	*La* 6	*La* 7
fréquence (Hz)	27.5	55	110	220	440	880	1760	3520
résolution relative	157%	78%	39%	20%	10%	5%	2.5%	1.2%

3.2.4 Solution envisagée

Les trois problèmes soulevés ici trouvent une solution dans l'utilisation de transformées particulières, dites à cœff icients de qualité constant (détaillées en **??**). Nous allons maintenant présenter les méthodes de reconnaissance de notes mises au point à l'aide ce ce type de transformation.

3.3 Sons monophoniques

La première étape consiste à savoir reconnaître la fréquence fondamentale d'une note seule. Bien que cela soit relativement simple de connaître les harmoniques d'un son dès que l'on utilise une transformée à Q constant, le problème de détermination de la fondamentale existe toujours. Il vient du fait qu'en fonction des différences de niveau entre les différentes harmoniques, on peut ne pas détecter la fondamentale de fréquence f_0, mais seulement le deuxième harmonique (de fréquence $2\,f_0$), ou parfois la sous-fondamentale (de fréquence $f_0/2$).

Les premières méthodes décrites (autocorrélation classique et étroite) font appel à une représentation temps-fréquence classique.Par la suite, plusieurs auteurs proposent d'appliquer une transformée à Q constant (que l'on trouvera décrite dans [**?**]) à des sons harmoniques monophoniques. Grâce à cette transformée, le domaine du logarithme des fréquences du son étudié est partagé en motifs de taille constante.

3.3.1 Corrélation classique

Prenons un signal périodique $f(t)$. Si l'on calcul la fonction d'autocorrélation de ce signal, on verra apparaître des pics pour des écarts temporels proportionnels à la période du signal. Si l'on applique le même procédé sur des sons quasi-harmoniques, ce qui est une bonne approximation des signaux musicaux, on obtient toujours des pics à condition de travailler sur de petites fenêtres temporelles (cela revient à segmenter le signal en durées inférieures ou égales à celle d'une note, de façon à observer un signal quasi-stationnaire).

A partir de ces pics, on peut déterminer quels sont les harmoniques présentes dans un son, et en extraire la fondamentale à l'aide d'un algorithme telle celui de Terhardt.

3.3.2 Corrélation étroite

En remplaçant la fonction d'autocorrélation du signal par une fonction plus générale, dite d'autocorrélation étroite (on se reportera à l'article pour une déf inition précise), Brown et Zhang comparent la méthode de corrélation classique avec celle de corrélation étroite dans [**?**] sur des sons synthétiques de piano, de violon et de flûte.

La méthode de corrélation étroite donne d'excellent résultats pour les phénomènes transitoires, là où la méthode classique effectue une moyenne des notes de part et d'autre de la transition. Cependant, cette méthode est plus coûteuse en temps de calcul. De plus, la longueur de la fenêtre d'analyse joue un rôle sur l'imprécision temporelle des événements.

3.3.3 Corrélation croisée

Un autre méthode consiste en un calcul de corrélation croisée (cross correlation) que l'on trouvera dans [?], et qui correspond aux méthodes de calcul de pitch en traitement de la parole. On compare le motif log-fréquentiel obtenu par la transformée aux motifs de différents instruments, et on recherche la position de la meilleure approximation en effectuant des translations du motif. Ainsi, la fréquence fondamentale du son étudié s'obtient lorque le maximum de corrélation est atteint.

Le défaut majeur de cette méthode est que des pics (de la fonction de corrélation) de même amplitude sont obtenus aussi bien pour la fondamentale que pour la deuxième harmonique. Par contre, aucun problème ne se pose pour la sous-fondamentale, contrairement à d'autres méthodes.

L'auteur propose une méthode efficace (mais heuristique) pour résoudre le problème du deuxième harmonique, qui consiste à assigner un signe à chaque pic, en alternant. Par des calculs de sommes de fonctions de corrélation, on évite directement le problème d'octave sur le deuxième harmonique.

Cette méthode a été mise en œuvre pour des sons monophoniques de piano, de flûte et de violon, et donne de très bons résultats, y compris sur des notes arpégées.

3.4 Sons polyphoniques

Plusieurs méthodes de transcriptions de sons polyphoniques existent, chacune appliquée dans une optique et des conditions distinctes.

La première consiste à capter les signaux midi des instruments, à l'aide de capteurs judicieusement placés. Par exemple, on utilise la sortie midi d'un synthétiseur, ou pour la guitareune série de six capteurs placés chacun sous une corde. Cela dit, cette méthode ne convient que pour des pièces dont on contrôle l'interprétation, dans le sens où cela ne peut plus s'effectuer à posteriori sur un enregistrement mixé.

D'autre part, quand on dispose d'un enregistrement en studio, on peut appliquer piste par piste, c'est-à-dire instrument par instrument, des méthodes adaptées aux sons monophoniques, voire dédiées à tel ou tel instrument. C'est de loin la méhode la plus efficace et pratique, mais aussi de moindre portée, puisqu'elle se limite aux enregistrements dont on dispose d'une version multi-pistes.

Enfin, quelques essais ont été effectués sur des sons d'instruments polyphoniques. Cela correspond beaucoup mieux à la problématique de la transcription automatique dans le cas le plus général, où l'on désire extraire une partition à partir d'un enregistrement stéréophonique contenant plusieurs voix.

3.4.1 Duo d'instruments

Maher s'est restreint à la transcription de duos dans lesquels les instruments ont des tessitures séparées tout au long du morceau ([?] et [?]). De cette façon, l'analyse se fait dans deux domaines distincts. Elle se base sur une procédure d'appariement dans deux directions simultanées. Pour chaque hypothèse de fréquence, , un désaccord est calculé entre les partiels déduits de l'hypothèse et ceux mesurés par les deux procédures d'appariement. Dans un premier temps, on calcule la différence fréquentielle entre chaque partiel de la série mesurée et son plus proche voisin dans la série supposée. On calcule ensuite la différence entre chaque partiel et son plus proche voisin de la série mesurée. Le désaccord correspond à la somme pondérée des deux erreurs.

Une recherche est d'abord menée en explorant toute l'étendue de la voix grave par demi-ton et en laissant la note aigüe fixe. une fois le minimum du désaccord trouvé, la fréquence fondamentale grave est précisée en recherchant plus finement autour de la valeur trouvée.

Ensuite, la note grave est laissée fixe et la recherche s'effectue sur la note aigüe. En pratique, cette recherche exhaustive n'est jamais menée pour toutes les fenêtres d'analyse. On al réalise à intervalles réguliers (aux instants de "recherche globale") espacés d'une durée inférieure à la durée de note minimum. Entre ces instants de recherche globale, la recherche est effectuée sur des intervalles de largeur d'un demi-ton de part et d'autre des fréquences précédemment trouvées (pour pouvoir suivre le vibrato, par exemple), à moins qu'entre deux instants de recherche global, les fréquences trouvées n'aient varié de plus d'un demi-ton. Dans ce cas, la recherche exhaustive reprend à partir du dernier instant global.

Le principal inconvénient de cette méthode réside en la connaissance à priori du nombre de voix (0, 1 ou 2) et de la zone de fréquence de séparation, indispensables au fonctionnement correct du modèle.

3.4.2 Modèle perceptif

Dans [**?**], S. Dixon propose une modèle d'identification de notes jouées par un seul instrument polyphonique. Pour ce faire, il utilise une transformée à coefficient de qualité Q constant, et des notions utilisées dans les modèles de fonctionnement de la perception auditive humaine.

Il utilise la théorie de la forme et les travaux de Bregman sur la séparation de flux sonores (bruit et son harmonique). Un modèle de source (à savoir de cordes frappées, applicable au piano, à la guitare, etc.) lui sert à prévoir le spectre sonore des différentes notes. Des méthodes de corrélation et d'autocorrélation servent à déterminer des pics. Par comparaisons des résultats obtenus dans les fenêtres successives, les différentes notes jouées simultanément sont lissées puis extraites.

La méthode donne de bons résultats pour des sons de guitare ; malheureusement, aucune indication sur le type de notes (de fondamentales proches ou non ; sons de synthèse ou réel) et le nombre de notes simultanées ayant servies aux tests n'est donnée.

3.4.3 Modèle avec contraintes

Afin de permettre une séparation précise des partiels découlant de notes distinctes, R. Mani et S. H. Nawab proposent dans [**?**] d'utiliser un système de contraintes pour initialiser et calibrer un ensemble de filtres à cœfficient de qualité constant. Ces contraintes portent sur le vibrato ou non de la note, la répartition des harmoniques, etc.Tout au long de l'analyse, les filtres sont adaptés en fonction de paramètres donnés par un modèle prédictif du signal.

Ensuite, une analyse spectrale et une recherche de pics associés à une poursuite sous plusieurs hypothèse portant sur les partiels (à travers un filtrage de Kalman) sont effectués. Après chaque analyse, on compare les prédictions et les résultats, puis des diagnostics sont établis quant à la présence de note, la présence de vibrato sur chaque note, etc.Une deuxième analyse peut être réalisée, et les filtres d'analyse recalibrés en fonction du diagnostic obtenu.

Les tests réalisés à l'aide de ce modèle portent sur deux notes de violon simultanées, séparées d'un demi-ton (seconde mineure) jusqu'à 9 demi-tons (sixte), et ceci pour des notes avec ou sans vibrato. Les résultats affichés sont satisfaisants ; cependant, les coûts de caluls sont non négligeables, dûs aux analyses répétées avec différents calibrages des filtres. Les auteurs utilisent un DSP dédié au calcul de ces filtres et réalisent ainsi un gain considérable en temps de calcul.

3.5 Proposition d'une méthodologie complète

De récents travaux proposent tout un cheminement pour traiter la musique polyphonique. Nous proposons d'en donner un rapide aperçu afin d'expliquer la direction suivie dans la suite de l'étude.

3.5.1 Séparation de sources

Nous avons vu que la principale difficulté dans la reconnaissance de sons multiphoniques réside en la ségrégation de partiels. Dans [?], S. Rossignol propose plusieurs étapes dans la transcription automatique. La première concerne la séparation du signal en sources. En effet, de nombreux modèles sonores existent : voix parlée ou chantée, instruments harmoniques entretenus (vents et cordes frottées) ou non (cordes frappés), etc. Si l'on commence par reconnaître les différentes sources, on pourra appliquer à chacun la méthode la plus adaptée.

3.5.2 Identification d'instruments

Dans le même ordre d'idées, des études sont réalisées au MediaLab du MIT afin d'identifier un instrument ([?]), que ce soit pour l'étude du timbre, le contrôle d'un instrument dans un orchestre ou la transcription automatique. Martin et Kim proposent une approche de reconnaissance de motif de l'enveloppe spectrale, autant sur les transitoires que sur la partie stationnaire d'un son.On imagine le gain d'une connaissance des instruments préalablement à la transcription : les tessitures, la répartition des harmoniques et le niveau de chaque instrument permettrait de mieux aborder la recherche des fondamentales présentes dans le signal.

3.5.3 Méthode proposée

Aux vues de tous ces éléments (qui ne constituent cependant qu'une bibliographie partielle sur le sujet), nous proposons le schéma de transcription suivant :

- séparation de sources ; traitement séparé de la voix parlée, de la voix chantée, des percussions et des sons harmoniques ;
- pour les sons harmoniques :
  - poursuite des partiels ;
  - recherche des fondamentales (synchronisation ou non des partiels, puis comparaison à des motifs harmoniques) ;
  - reconnaissance du rythme ;
  - écriture de la combinaison du rythme et de la mélodie trouvée ;
- un modèle d'apprentissage par l'exemple (à l'aide de réseaux de neurones) permettra d'affiner les capacités (et valeurs des paramètres) du système.

Le travail qui va être présenté par la suite se situe dans le cadre de la **poursuite des partiels**. Ceux-ci décroissent après l'attaque pour les sons non entretenus, plus ou moins vite selon l'instrument, selon la note, selon le jeu instrument, etc. Aussi, le rapport signal à bruit varie en décroissant au fur et à mesure que le temps s'écoule, et il faut être sûr de pouvoir détecter les partiels même dans un environnement bruité.

Après avoir explicité en détail le banc de filtres que l'on utilisera pour obtenir une représentation temps-fréquence à Q constant, nous présenterons une méthode originale de poursuite des partiels robuste au bruit ainsi qu'une représentation temps-fréquence faisant intervenir le rapport de vraisemblance de présence d'une harmonique.

Chapitre 4

Banc de filtres à facteur de qualité constant

Afin de pallier aux limites de l'analyse de Fourier classique, on utilise une transformée à cœfficient de qualité Q constant. Cette transformée calcule une transformée de Fourier dont la taille de la fenêtre d'analyse varie avec la fréquence analysée.

De cette façon, on peut analyser finement en basses fréquences, et plus grossièrement en hautes fréquences, tout en détectant aussi facilement un espacement "musical" (par exemple un ton, un quart de ton) en tout endroit de la gamme de fréquences audibles, et pourquoi pas en dehors.

Cette représentation est particulièrement bien adaptée à la musique occidentale, en ceci qu'elle respecte l'aspect géométrique de l'espacement entre les notes dans la gamme tempérée.

4.1 Description du banc de filtres

4.1.1 Définition du facteur de qualité

Le **facteur de qualité** Q correspond au rapport de la fréquence f que l'on cherche à analyser sur la précision (ou variation possible autour de cette fréquence) notée δf.

$$Q = \frac{f}{\delta f}$$

Si l'on réalise un banc de filtres vérifiant cette propriété, alors le facteur de qualité correspond au rapport de la fréquence centrale f_k^c du filtre sur la largeur L_k^c de ce filtre.

$$Q = \frac{f_k^c}{L_k^c} \tag{4.1}$$

4.1.2 Précision des filtres

On désire avoir un pouvoir séparateur d'un quart de ton, soit une résolution de 3%. En effet, une octave se composant de 12 demi-tons, ou encore de 24 quarts de tons, la résolution variable vaut $(2^{1/24} - 1) \approx 0.029$ fois la fréquence centrale du filtre.

Le facteur de qualité peut alors se calculer : $Q = f_k^c / L_k^c = f_k^c / 0.029 f_k^c = 34$. Les filtres ainsi réalisés se révèlent très sélectifs, ce qui répond à notre attente de distinguer aisément deux notes espacées d'un quart de ton, et ce quelque soit la gamme de fréquence dans laquelle nous travaillons.

Un filtre donné se définissant par les cœfficients de sa fonction de transfert, on remarquera que le nombre de cœfficients du filtre augmente avec son étroitesse. En effet, pour imposer de brusques variations à une courbe passant par un certain nombre de points, il faut prendre un très grand nombre de points.

Ainsi, le premier filtre du banc (celui de fréquence fondamentale la plus grave) aura plus de coefficients que le dernier (on passe pratiquement du simple au double).

4.1.3 Description des filtres utilisés

Idéalement, tout filtre $\mathcal{F}_k$ utilisé dans ce banc d'analyse est passe-bande parfait, c'est-à-dire rectangulaire (cf. fig. **??**). Il sera centré sur une fréquence f_k^c (sa fréquence centrale) et de largeur $L_k^c = 2\,\Delta f_k^c$ (où $\Delta f_k^c = 2^{1/48} f_k^c - f_k^c$ sera la demi-largeur de bande du filtre).

Fig. 4.1: filtre idéal

La réalisation d'un tel filtre selon un modèle à réponse impulsionnelle finie (RIF) est impossible. Aussi, on l'approche par un filtre passe-bande déterminé par un gabarit, qui consiste à se donner une certaine latitude sur l'amplitude du filtre sur la bande passante, une latitude sur l'amplitude du filtre sur la bande de coupure, ainsi qu'une largeur de décroissance, comme présenté fig. **??** .

Fig. 4.2: Gabarit d'un filtre avec ε_{bc} la latitude sur la bande de coupure, ε_{bp} la latitude sur la bande passante et l_{ba} la largeur de la bande atténuée

Fig. 4.3: filtre réel tel qu'implémenté dans notre modèle

Sachant que deux filtres successifs ne doivent en aucun cas laisser de "trou" fréquentiel et permettre la détection de n'importe quelle fréquence présente dans le signal, la fin de la bande passante d'un filtre $\mathcal{F}_k$ doit coïncider avec le début de la bande passante du filtre suivant $\mathcal{F}_{k+1}$.

De plus, on acceptera qu'une fréquence visible dans un filtre le soit dans une moindre mesure (c'est-à-dire avec un gain beaucoup plus faible) dans l'un seul de ses voisins. Ainsi, la latitude sur la largeur de décroissance correspondra à la demi-largeur de bande, à savoir Δf_k^c.

De cette manière, une fréquence présente dans le filtre $\mathcal{F}_k$ et comprise entre $f_k^c - \Delta f_k^c$ et f_k^c sera détectée, mais à un niveau très inférieur, dans le filtre $\mathcal{F}_{k-1}$. De même, une fréquence présente dans le filtre $\mathcal{F}_k$ et comprise entre f_k^c et $f_k^c + \Delta f_k^c$ sera détectée, toujours à un niveau très inférieur, dans le filtre $\mathcal{F}_k$ cette fois-ci.

Par ces choix de constructions des filtres, ont atteint aisément 700 cœfficients pour décrire la réponse impulsionnelle qui les caractérise (en ce qui concerne le premier filtre, sachant que c'est le plus étroit et donc celui caractérisé par le plus grand nombre de cœfficients).

4.2 Méthodologie d'analyse

L'analyse se fait octave par octave, avec le même banc de filtres (cf. fig. ?? gauche).

Fig. 4.4: Banc de filtres pour une analyse sur 1 octave (à gauche) et 2 octaves (à droite)

En effet, si notre banc de filtres couvre parfaitement une octave, une fois que l'analyse est faite pour une octave donnée, nous pouvons procéder à l'analyse de l'octave directement inférieure, et ainsi de suite. Aucune fréquence ne passera au travers de notre analyse lors du passage d'une octave à l'autre.

Pour ce faire, une fois l'analyse effectuée à l'octave n, on filtre passe-bas le signal puis on le décime d'une valeur sur deux. On lui applique alors le même filtrage à l'octave $n-1$.

Cela revient au même que d'appliquer un banc de filtre calculé spécialement pour l'octave $n-1$ (cf. fig. ?? droite), à ceci près qu'on évite de calculer des filtres de réponse impulsionnelle deux fois plus grande. Toutefois, n'oublions pas que la décimaton introduit une perte de précision selon l'axe temporel. Ainsi, si on représente les différents bancs de filtres utilisés en abcisse et le nombre d'échantillons temporels de signal analysés en ordonnée, on obtient le résultat proposé en fig. ?? gauche (remarquons l'homothétie).

Fig. 4.5: Modèle d'analyse de n octaves à l'aide du banc de filtres à Q constant (échelle linéaire à gauche et logarithmique à droite)

Autant en fréquences qu'en nombre d'échantillons temporels, la progression est logarithmique : une échelle logarithmique selon les deux axes est plus indiquée pour illustrer le fait que pour chaque octave, l'analyse se fait de la même manière (cf. fig. ?? droite).

4.3 Calibrage du banc de filtres

4.3.1 Première condition de non-repliement du spectre

L'analyse étant limitée aux fréquences inférieures à la moitié de la fréquence d'échantillonnage (ceci afin d'éviter le phénomène de repliement, ou d'aliasing), le dernier filtre de notre banc devra avoir pour fréquence de coupure "supérieure" une fréquence en deça de cette limite. Cela revient à dire que pour le filtre $\mathcal{F}_{24}$, sa fréquence centrale doit vérifier $f^c_{24} + 2\Delta f^c_{24} < 1/2F_e$. En effet, la fin de la zone de décroissance de la bande passante est située à une fréquence supérieure de 2 Δf^c_{24} à la fréquence maximale pour laquelle le gain est égal à 1 (soit $f^c_{24} + \Delta f^c_{24}$).

Afin d'alléger les notations par la suite, nous allons définir les fréquences limites de notre banc de filtres, à savoir f_- et f_+, comme suit :

$$f_- \quad = \quad f^c_1 - 2\,\Delta f^c_1$$

$$\begin{aligned}
&= \left(1 - 2(2^{1/48} - 1)\right)\ f_1^c \\
&= 0.9709\ f_1^c \\
f_+ &= f_{24}^c + 2\,\Delta f_{24}^c \\
&= 2^{23/24}\left(1 + 2(2^{1/48} - 1)\right)\ f_1^c \\
&= 1.9996\ f_1^c
\end{aligned}$$

avec f_1^c la fréquence centrale du premier filtre. La fréquence du vingt-quatrième et dernier filtre est définie par $f_1^c\, 2^{23/24}$ et la demi-largeur de bande par $\Delta f_{24}^c = (2^{1/48} - 1) f_{24}^c$.

La première condition de non repliement du spectre est alors la suivante :

$$f_+ < \frac{F_e}{2}$$

On peut l'écrire portant sur f_1^c, ce qui donne :

$$f_1^c < \frac{1}{2 \times 2^{23/24}\,(2^{49/48} - 1)} F_e$$

D'où l'on tire finalement $f_1^c \leq 0.25001\ F_e$. Ainsi, pour $f_1^c \leq \frac{1}{4} F_e$, la première condition de non repliement du spectre est toujours réalisée.

4.3.2 Seconde condition de non-repliement du spectre

Nous allons maintenant prendre en compte le filtrage passe-bas que l'on applique avant la décimation et le passage à l'octave inférieur.

Fig. 4.6: Banc de filtres à l'octave n

Une fois le signal analysé dans le banc de filtres à l'octave n entre f_- et f_+ (fig. ??), on effectue un certain filtrage passe-bas, puis on décime d'un facteur 2 le signal. On réitère l'analyse à l'octave $n-1$ entre $\frac{f_-}{2}$ et $\frac{f_+}{2}$ (fig. ??).

Fig. 4.7: Filtrage passe-bas avant le passage à l'octave inférieure

Le filtre passe-bas doit avoir pour fréquence de coupure $\frac{f_+}{2}$. Le facteur sur lequel nous pouvons exercer une influence est sa largeur de bande de décroissance Δf_{pb}. Sur cette bande, pour peu que nous l'ayons centrée sur $\frac{F_e}{2}$, un phénomène de repliement du spectre apparaît pour les fréquences comprises entre $\frac{f_+}{2}$ et $\frac{F_e}{4}$. Cependant, ces fréquences ayant déjà été analysées à l'octave n, on peut se permettre ce repliement sans que nos données soient affectées. Nous allons même en profiter pour obtenir la largeur Δf_{pb} la plus grande possible (dans les valeurs permises), ceci afin de baisser au maximum le nombre de cœfficients de ce filtre.

$$\begin{aligned}
\Delta f_{pb} &= 2\left(\frac{F_e}{4} - \frac{f_+}{2}\right) \\
&= \left(\frac{F_e}{2} - f_+\right)
\end{aligned}$$

Par ailleurs, si on implémente le filtre passe-bas à l'aide d'une fenêtre de Hamming, on sait que la largeur de la bande de décroissance du filtre est de l'ordre de 4 divisé par le nombre de cœfficients n_c.

$$\begin{aligned}\Delta f_{pb} &\approx \frac{4}{n_c} \\ &= \frac{F_e}{2} - 1.9996 f_1^c\end{aligned}$$

On obtient alors la seconde condition de non repliement sur les données à analyser :

$$n_c \geq \frac{4}{\frac{F_e}{2} - 1.9996 f_1^c} \tag{4.2}$$

Rappelons que la première condition nous donne $f_1^c \leq 0.2501$, où $f_1^c = f_{cal}$ est la fréquence centrale du premier filtre encore appelée fréquence de calibrage du banc de filtres. On désire conserver f_1^c la plus proche de cette valeur limite, mais comme le montre la figure **??**, n_c diverge lorsqu'on s'approche trop de 0.25, ce qui est normale : on tente de rendre pratiquement nulle la largeur de décroissance du filtre ; pour y arriver, il nous faut un très grand nombre de cœfficients.

Fig. 4.8: Nombre de cœfficients du filtre passe-bas en fonction de la fréquence de calibrage du banc

On ne désire pas plus de 100 cœfficients aussi on prendra finalement comme valeur maximale de fréquence de calibrage $f_{cal} = 0.23$.

4.3.3 Accordage du banc de filtres

La valeur précise de f_{cal} dépend de l'accordage du morceau à analyser. Il nous semble nécessaire de prendre en compte le fait que si la référence pour l'accordage du morceau n'a pas été le *La* 440 Hz, notre modèle doit pouvoir être adapté.

Ainsi, si nous remarquons dans une analyse que tous les partiels apparaissant sont doublés dans des filtres mitoyens, cela signifie que les fréquences analysées se placent à la limite entre deux filtres. Il suffira alors d'effectuer une seconde analyse après avoir effectué une translation du banc, en modifiant f_{cal} de $\pm\Delta f_1^c$.

4.4 Implémentation du banc de filtres et résultats en sortie

On génère un signal synthétique excitant les filtres de notre choix en leur fréquence centrale, puis on observe le signal en sortie du banc de filtres. Prenons pour exemple la configuration de la figure **??** gauche, où les filtres numéros 1, 5, 9, 13, 17, 21 sont excités.

Fig. 4.9: Sortie du banc de filtres (1, 5, 9, 13, 17, 21 à gauche, et 2, 6, 10, 14, 18, 22 à droite)

Fig. 4.10: Sortie du banc de filtres (3, 7, 11, 15, 19, 23 à gauche, et 4, 8, 12, 16, 20, 24 à droite)

Notons l'échelle de niveaux de gris pour ces graphiques, qui indiquent en clair les pics (positifs ou négatifs) des sinusoïdes et en foncé les faibles valeurs. On représente ainsi la valeur absolue des échantillons en sortie de chaque filtre.

Normalement, ces valeurs alternent tout au long de la sortie de chaque filtre : cela vient du fait qu'elles correspondent à la discrétisation d'une sinusoïde.

Cependant, nous remarquons tout de suite que des oscillations apparaissent (sur une plus grande échelle temporelle), dont l'amplitude dépend du numéro du filtre (en fait, de la largeur de ce filtre).

Cela correspond à une modulation d'amplitude, créée par chaque filtre $\mathcal{F}_k$ et dépendant de sa largeur L_k. Ainsi, par endroits, et ce dans chaque filtre, la sinusoïde s'aplatit jusqu'à disparaître, puis réapparaît plus loin. On est loin d'une alternance pure et simple de valeurs entre $+a_0$ et $-a_0$ (a_0 amplitude du signal).

Il va sans dire que ceci risque de nous poser des problèmes lorsque nous voudrons effectuer la poursuite des partiels. Ceci sera développé ultérieurement. Nous allons d'abord présenter une méthode de poursuite très efficace, et qui n'a pas encore été implémentée dans le cadre de la transcription automatique.

Chapitre 5

Méthode des "climbers" pour la recherche des crêtes

5.1 Motivation

Nous avons vu dans la description des étapes d'une transcription automatique de partitions qu'il est nécessaire de savoir repérer les endroits (autant en fréquence qu'en temps) où se répartie l'énergie dans le plan temps- fréquence.

A l'œil nu, dans le cas de signaux peu bruités, quiconque connaît la problématique posée sera capable de dire avec une aisance déconcertante quels sont ces secteurs. Malgré cela, une détection automatique, c'est-à-dire à l'aide d'algorithmes, demande des techniques éprouvées.

Pour ce faire, nous nous proposons d'utiliser la méthode dite des "climbers", ou "marcheurs fous". Elle a été développée par R. Carmona, W. Hwang et B. Torrésani (dans [**?**]) dans le cadre de la détection multi-crêtes de signaux bruités, ce qui correspond exactement à notre étude (plusieurs partiels présents dans la même zone temporelle, dans un contexte bruité).

5.2 Description de la méthode

5.2.1 Principe et cadre de l'utilisation de la méthode

La méthode se base sur une approche de type chaînes de Markov Monte-Carlo (MCMC, Markov Chain Monte Carlo). Cette approche utilise la distribution en énergie d'une représentation temps-fréquence du signal.

L'énergie du signal se concentre autour des courbes du plan temps-fréquence appelées **crêtes**. De ce fait, la chaîne de Markov est construite de sorte que les marcheurs aléatoires[1] sont attirés par ces structures à une dimension.

La procédure de détection est basée sur un algorithme original de chaîne de Markov Monte Carlo. Elle est conçues de telle manière que les densités d'occupation pondérés

[1] appelés ici "marcheurs fous", du fait que tout marcheur suivant ce type de déplacement dans un environnement accidenté sera à coup sûr considéré comme ne possédant plus tout ses moyens !

dessinent les crêtes du plan temps-fréquence. Plus important encore, sa robustesse au bruit est remarquable.

5.2.2 Représentation temps-fréquence

Dans l'article [?], la représentation temps-fréquence utilisée par Carmona est tantôt une transformée de Gabor, tantôt une transformée en ondelettes. Cela dit, elle peut être utilisée avec n'importe quelle représentation temps-fréquence. Aussi, c'est sans le moindre soucis que nous allons l'appliquer à notre sortie de banc de filtres.

Plaçons nous dans un plan temps-fréquence où l'on observe l'énergie. La sortie du banc de filtres étant constituée de valeurs temporelles, il nous font donc travailler avec une représentation qui en découle.

5.2.3 Notations

Nous noterons par la suite $M(\phi, b)$ cette représentation, avec ϕ la fréquence et b la fenêtre temporelle.

Le domaine D dans lequel on travaille reste borné, et les valeurs de M sont non négatives sur tout le domaine D.

L'intervalle temporel d'analyse du signal est décomposé en B éléments $\{b_0, b_1, \ldots b_{B-1}\}$ et la variable fréquentielle ϕ est discrétisée en un nombre fini de valeurs $\{\phi_0, \phi_1, \ldots \phi_{K-1}\}$ (correspondant aux 24 filtres de chaque banc d'analyse, pour n octaves analysés).

5.2.4 Principes de l'algorithme des marcheurs fous

Un grand nombre de particules (les marcheurs) sont initialement placés de façon aléatoire dans le domaine D au temps $t = 0$. Ensuite, chaque marcheur évolue selon une chaîne de Markov sur D avec un mécanisme de transition dépendant des valeurs locales de la fonction $M(b, \phi)$.

Cette chaîne est conçue pour relaxer une distribution qui est essentiellement concentrée sur les crêtes. La projection du déplacement selon l'axe b est une marche aléatoire symétrique classique. Verticalement, on encourage les marcheurs à gravir les obstacles pour rejoindre les crêtes par une méthode de pénalisation de Hastings-Metropolis et un réglage de température similaire à l'algorithme du recuit simulé. Cependant, contrairement au recuit simulé, l'algorithme du marcheur fou cherche tous les maxima locaux pour une valeur fixée de la variable temporelle, au lieu d'en chercher uniquement le maximum global.

5.2.5 Description de l'algorithme

Initialisation

A l'instant $t = 0$, nous initialisons les positions $X_\lambda(0)$ des N_m marcheurs sur la grille $\Gamma = \{0, \ldots, B-1\} \times \{0, \ldots, K-1\}$. Les marcheurs sont étiquetés par le paramètre $\lambda = 1, \ldots, N_m$. Les positions initiales sont choisies indépendamment les unes des autres et selon une distribution uniforme sur la grille Γ. Les marcheurs se déplacent indépendamment les uns des autres, selon la même loi.

Déplacements verticaux et horizontaux

Si un marcheur est positionné au point (j,k) à l'instant t, c'est-à-dire si $X_\lambda(t) = (j,k)$, alors sa position au temps $t+1$, notée $X_\lambda(t+1) = (j',k')$, est déterminée selon les lois édictées ci-après. On appellera variation d'énergie la fonction $\Delta M = M(j',k') - M(j,k)$, et on notera $T(t)$ la fonction de température.

- déplacement horizontal $(j \to j')$:
  - vers la droite : $j' = j+1$ avec la probabilité $p = \frac{1}{2}$;
  - vers la gauche : $j' = j-1$ avec la probabilité $p = \frac{1}{2}$;
  - avec des conditions de rebond élastique aux limites ;
- déplacement vertical $(k \to k')$:
  - vers le haut : $k' = k+1$ avec la probabilité $p = \frac{1}{2}$ (on note alors $k_0 = k+1$) ;
  - vers le bas : $k' = k-1$ avec la même probabilité $p = \frac{1}{2}$ (on note alors $k_0 = k-1$) ;
  - dans les deux cas,
    * si $\Delta M > 0$ alors $p(k' = k_0) = 1$;
    * si $\Delta M < 0$ alors $p(k' = k_0) = e^{\Delta M/T(t)}$ (déplacement malgré le fait que l'énergie décroisse dans cette direction, mais avec une probabilité décroissante et assez faible) et $p(k' = k) = 1 - e^{\Delta M/T(t)}$ (aucun déplacement dans ce sens puisque l'énergie décroît, avec une probabilité croissante et assez grande) ;
  - avec des conditions de rebond élastique aux limites.

On entend par conditions de rebond élastique le fait que si un déplacement doit se produire vers l'extérieur et que le marcheur se trouve précisément sur une frontière, il ne sort pas du domaine mais repart dans le sens opposé, avec la probabilité qui lui a été attribuée.

Au fur et à mesure que le temps passe, la température décroît, et les déplacements dans les directions opposées à celles qui permettent de rejoindre les crêtes sont de moins en moins favorisées. Ce phénomène s'appelle la **relaxation**. Dans un premier, les mouvements sont favorisés dans toutes les directions, ce qui permet aux marcheurs d'être présent dans toutes les zones où un maximum local existe. Dans un second temps, la température baissant, ils ne peuvent plus se déplacer que dans une zone restreinte, et se fixent finalement sur le maximum local.

Mesures d'occupation

Au fur et à mesure que l'algorithme se déroule, nous aimerions savoir où se trouvent les marcheurs, afin de savoir quand ils se seront immobilisés. De plus, une fois immobiles, nous voulons connaître leurs positions, qui correspondent dans la majeur partie des cas à celle des crêtes que nous désirons localiser (et parfois à des bruits prépondérants). Pour ce faire, il nous faut définir des mesures d'occupation du plan M.

A chaque instant t, on considère deux mesures d'occupation. La première $\mu_t^{(0)}$ est définie par :

$$\mu_t^{(0)} = \frac{1}{N} \sum_{\lambda=1}^{N_m} \delta_{X_\lambda(t)}$$

Elle est obtenue en associant un poids $\frac{1}{N}$ à tout endroit de la grille M où un marcheur est présent à l'instant t.

La seconde, μ_t, correspond à une mesure pondérée d'occupation, obtenue en pondérant par un poids égal à la valeur de la fonction M à la position où le marcheur est présent :

$$\mu_t = \sum_{\lambda=1}^{N_m} M(X_\lambda(t)) \delta_{X_\lambda(t)}$$

Nous donnons finalement les "mesures d'occupation intégrées" correspondantes, qui sont définies par les moyennes ergodiques :

$$\begin{aligned} \mu_I^{(0)} &= \frac{1}{T} \sum_{t=1}^{T} \mu_t^{(0)} \\ \mu_I &= \frac{1}{T} \sum_{t=1}^{T} \mu_t \end{aligned}$$

La première n'est donnée qu'à titre indicatif. En effet, μ_I permet d'obtenir de bien meilleurs résultats.

5.2.6 Réglage des paramètres de la relaxation

La fonction de température $T(t)$ est de la forme $T(t) = T_0 * \alpha^t$, avec $T_0 > 0,\ \alpha \in [0, 1]$ pour s'assurer de la convergence de l'algorithme. Aussi, selon la grandeur initiale du taux de décroissance T_0 et selon le facteur de décroissance α, les marcheurs vont se rendre plus ou moins vite vers les crêtes, en fonction des probabilités calculées à partir de ces paramètres.

De plus, s'ils sont mal réglés, par exemple si le marcheur rejette systématiquement de se rendre vers des zones de moindre énergie, il se peut qu'il passe à côté d'un minimum local difficile à atteindre (cas où $\alpha \approx 0$: la température baisse immédiatement, les marcheurs voient leurs positions gelées dès les premières itérations), mais correspondant cependant à un partiel digne d'intérêt pour la détection des notes.

De même, s'il accepte trop systématiquement de remettre en cause les pics d'énergie qu'il atteint, il pourra poursuivre très longtemps sa course sans s'arrêter (cas où $\alpha \approx 1$: la température ne baisse pas, les marcheurs remettent en cause leur position même lorsqu'elle correspond à un maximum, et ce $\forall t$).

5.2.7 Arrêt de l'algorithme

Pour éviter une marche d'une durée indéterminée, l'algorithme prévoit deux tests d'arrêt. Le premier prévoit que dans le cas où depuis certain nombre d'itérations N_0, $P_{arret}\%$ des marcheurs sont immobiles, on considère qu'on a atteint tous les minima locaux accessibles à nos N_c marcheurs.

Lorsque cette condition n'est jamais atteinte, on s'arrête une fois $N_{max} > N_0$ itérations réalisées.

5.3 Utilisation de la méthode des "climbers"

Appliquons la méthode des climbers sur un octave dans l'une des configurations proposées. Du fait des oscillations d'amplitude, les climbers viennent se loger de manière irrégulière sur chaque ligne de crête.

Nous proposons le résultat d'un test sur un signal de 1024 échantillons, pour lequel nous générons 50000 marcheurs, avec une température initiale $T_0 = 5$, un taux de décroissance $\alpha = 0.8$. L'algorithme s'arrête lorsque 95% des marcheurs sont immobiles depuis plus de 40 itérations, avec un maximum de 200 itérations dans tous les cas.

Cette fois-ci, on dessine la sortie temps-fréquence du banc de filtres avec des couleurs variant du foncé pour les valeurs négatives au clair pour les valeurs positives.

Les valeurs positives et négatives s'alternant, les climbers devraient être répartis tout au long de chaque crête.

Fig. 5.1: Résultat de l'algorithme des climbers (droite) sur la sortie du banc de filtres (gauche)

On observe (fig. **??** droite) la densité d'occupation pondérée du plan temps-fréquence. On arrive aisément à lire dans cette figure la même information que dans la sortie du banc de filtres : chaque filtre excité contient des marcheurs. Pour peu que cet algorithme soit bien paramétré avant utilisation, son efficacité reste vérifiée pour notre application. L'image du plan temps-fréquence qu'il nous donne est fidèle au plan temps-fréquence dans lequel les marcheurs évoluent.

Cependant, puisque la sortie du banc de filtres présente des défauts, l'algorithme des marcheurs présnte les mêmes erreurs.

Dans une écriture automatique de partitions, l'étape qui suit consiste, à partir de la détection des pics, à effectuer une poursuite des partiels, c'est-à-dire reconnaître, à partir de quelques "climbers" représentant un même partiel, où celui-ci commence et fini. Or, si nos partiels ne sont visibles que sous forme de pointillés, cette tâche sera alourdie de façon non négligeable.

Nous devons trouver une solution face à ce problème, par exemple un lissage du plan temps-fréquence avant l'utilisation des "climbers".

5.4 Solution proposée

Aussi, on se propose par la suite de développer un modèle de plan temps-fréquence qui prendra en compte la spécificité de chaque filtre et de ce fait nous donnera une représentation plus lissée, sans ces oscillations.

Ce modèle statistique va étudier la vraisemblance qu'un signal perçu dans un filtre corresponde à une sinusoïde. C'est l'objet du chapitre suivant.

Chapitre 6

Maximum de vraisemblance généralisé

6.1 Positionnement du problème

On se place à la sortie du banc de filtres à facteur de qualité constant. Pour chaque filtre, on désire savoir si l'énergie présente dans le signal après filtrage correspond à la présence d'un bruit ou à celle d'une sinusoïde, éventuellement mélangée à du bruit.

Nous allons dans un premier temps poser les hypothèses possibles sur un paramètre d'amplitude portant sur la sinusoïde, puis calculer les probabilités associées à chaque hypothèse. A partir de ces probabilités, nous calculerons le rapport de vraisemblance généralisé des hypothèses.

6.2 Notations

6.2.1 Signal d'entrée

Nous supposons que notre signal $x(n)$ suit le modèle "sinusoïde + bruit". La suite des échantillons avant passage dans le banc de filtres sera notée $x(n)$, avec :

$$x(n) = x_0(n) + b(n) \tag{6.1}$$

La partie sinusoïdale $x_0(n)$ peut s'écrire $a_0 \ \cos(2\pi f_0 n + \phi)$ ou encore :

$$x_0(n) \quad = \quad c_0 \ \cos(2\pi f_0 n) + s_0 \ \sin(2\pi f_0 n) \tag{6.2}$$

Cette écriture porte sur les échantillons temporels du signal, on parlera par la suite d'**écriture temporelle** du problème.

On note $b(n), \ n = 0, ..., N-1$ un ensemble de variables aléatoires modélisant le bruit en entrée du filtre, et h_k la réponse impulsionnelle du filtre $\mathcal{F}_k$. Il est supposé gaussien.

Fig. 6.1: Schéma du filtrage

6.2.2 Signal de sortie

En sortie du filtre $\mathcal{F}_k$, les échantillons $y_k(n)$ sont définis par :

$$y_k(n) = (x_0 * h_k)(n) + b_k(n) \tag{6.3}$$

le bruit $b_k(n) = (b * h_k)(n)$ est de spectre plat dans la bande $\mathcal{B}_k$ (en notant h_k la réponse impulsionnelle du filtre) : c'est ici qu'intervient l'hypothèse de non corrélation du bruit. En effet, lorsqu'on considère qu'il est toujours gaussien dans la bande, on remplace $b_k(n)$ par $b(n)$.

6.2.3 Notation vectorielle - Paramètre d'amplitude

Nous utilisons la notation vectorielle : $x_0 = \begin{pmatrix} x_0(0) \\ \vdots \\ x_0(N-1) \end{pmatrix}$, $y = \begin{pmatrix} y_k(0) \\ \vdots \\ y_k(N-1) \end{pmatrix}$, $\theta = \begin{pmatrix} c_0 \\ s_0 \end{pmatrix}$, et

$$D(f_0) = \begin{pmatrix} 1 & 0 \\ \cos(2\pi f_0) & \sin(2\pi f_0) \\ \vdots & \vdots \\ \cos\left(2\pi\left(N-1\right)f_0\right) & \sin\left(2\pi\left(N-1\right)f_0\right) \end{pmatrix} \tag{6.4}$$

Alors, on peut écrire $x_0 = \theta^H \, D(f_0)$, où θ est le **paramètre d'amplitude** de l'écriture temporelle.

6.3 Hypothèses

Nous nous plaçons en sortie du filtre $\mathcal{F}_k$. Nous allons tester deux hypothèses.

Hypothèse H_0

La première hypothèse, notée H_0, est que le signal en sortie du filtre est uniquement constitué de bruit : $y_k(n) = b_k(n)$. Le signal $x(n)$ ne contient aucune sinusoïde de fréquence $f_0 \in \mathcal{B}_k$. Autrement dit,

$$\theta = (0, 0)^T \tag{6.5}$$

Cette hypothèse est dite **simple**, du fait qu'il n'y a qu'une valeur possible de θ qui la satisfasse.

La probabilité de H_0 s'écrit :

$$\begin{aligned} p_0 &= P_{Y_0,\ldots Y_{N-1}/H_0}\left[y(0)\ldots, y(N-1); \theta\right] \\ &= P_{Y/H_0}(y; \theta) \end{aligned}$$

Hypothèse H_1

La seconde hypothèse, notée H_1, est que le signal se compose à la fois d'une sinusoïde à la fréquence $f_0 \in \mathcal{B}_k$ (f_0 restant à déterminer) et d'un bruit additif. Autrement dit :

$$\theta \neq (0,0)^T \tag{6.6}$$

C'est une hypothèse **composée** (plusieurs valeurs sont possibles).

La probabilité de H_1, est :

$$p_1 = P_{Y/H_1}(y;\theta)$$

6.4 Rapport de vraisemblance généralisé

6.4.1 Définition

On sait que dans un grand nombre d'approches statistiques, une bonne fonction de décision est le rapport de vraisemblance généralisé :

$$\Gamma = \frac{\max_{\theta \in H_1} \; p_{Y/H_1}(y;\theta)}{\max_{\theta \in H_0} \; p_{Y/H_0}(y;\theta)} \tag{6.7}$$

Afin de savoir laquelle de ces deux hypothèses est justifiée en présence d'un signal donné, il est nécessaire de le calculer dans chacun des filtres $\mathcal{F}_k$ $k = 1, \ldots, 24$.

6.4.2 Maximisation de p_0 et de p_1

Sous l'hypothèse H_0, $\theta = (0,0)^T$ donc l'expression de p_0 ne dépend pas de θ :

$$\max_{\theta \in H_0} \; p_{Y/H_0}(y;\theta) = p_{Y/H_0}(y) \tag{6.8}$$

D'autre part, maximiser p_1 revient à trouver le paramètre optimal

$$\overline{\theta} = \arg\max_{\theta \in H_1} p_1 \tag{6.9}$$

6.4.3 Valeur du rapport de vraisemblance généralisé

En utilisant les résultats de maximisation donnés précédemment, on obtient :

$$\Gamma = \frac{p_{Y/H_1}(y;\overline{\theta})}{p_{Y/H_0}(y)} \tag{6.10}$$

6.5 Ecriture fréquentielle

6.5.1 Motivation

Nous allons écrire à nouveau les deux hypothèses. Cette fois-ci, nous allons adopter une écriture fréquentielle. La loi de probabilité qui nous intéresse est celle du plus grand module de la valeur de Transformée de Fourier Discrète du signal de fréquence dans la bande $\mathcal{B}_k$ du filtre.

En fonction de la valeur de son module, et après l'estimation du rapport de vraisemblance généralisé, nous pourrons décider de ce qui est présent dans le signal : un signal utile de type sinusoïde bruité ou du bruit seul.

6.5.2 Notations

A un ensemble de N échantillons temporels $y(0), ..., y(N-1)$, on associe les L transformées de Fourier discrètes $\hat{y}(0), ..., \hat{y}(L-1)$ définies aux fréquences $f_i = \frac{i}{L}$ par :

$$\begin{aligned} \hat{y}(i) &= \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} y(n)\ \exp(-2j\pi n f_i) \\ \alpha_i &= \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} y(n)\ \cos(-2j\pi n f_i) \quad (6.11) \\ \beta_i &= \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} y(n)\ \sin(-2j\pi n f_i) \quad (6.12) \end{aligned}$$

La transformée inverse est alors définie avec le même facteur $\frac{1}{\sqrt{N}}$ de façon à normaliser les deux transformations.

On décompose $\hat{y}(i)$ en partie réelle et partie imaginaire :

$$\hat{y}(i) = \alpha_i + j\beta_i = \rho \exp(j\theta)$$

Dans la cas général, α_i (respectivement β_i) peut se décomposer en une partie fixe m_α (respectivement m_β) dûe à la présence d'une sinusoïde et une partie aléatoire ε_α (respectivement ε_β) dûe à la présence de bruit.

La partie réelle et la partie imaginaire de la TFD du bruit sont des variables gaussiennes centrées, de variances différentes :

$\varepsilon_\alpha \hookrightarrow \mathcal{N}(0, \sigma_\alpha^2) \quad \varepsilon_\beta \hookrightarrow \mathcal{N}(0, \sigma_\beta^2)$

La décomposition information-bruit s'énonce alors $\hat{y}(i) = (m_\alpha + \varepsilon_\alpha) + j(m_\beta + \varepsilon_\beta)$. Il existe un couple de réels (m, ϕ), $m \in \mathbb{R}^*$, $\phi \in [0, 2\pi[$ tels que l'on puisse écrire $m_\alpha + j\, m_\beta = m \exp(j\phi)$.

Ce paramètre m correspond au module de la sinusoïde qui nous intéresse. Nous utiliserons donc z_m comme **paramètre d'amplitude** θ dans l'écriture fréquentielle pour le calcul du rapport de vraisemblance généralisé.

$$z_m = \theta \quad (6.13)$$

De même, il existe un couple de réels (ε, ψ), $\varepsilon \in \mathbb{R}^* \ \psi \in [0, 2\pi[$ tels que l'on puisse écrire $\varepsilon_\alpha + j\,\varepsilon_\beta = \varepsilon \ \exp(j\psi)$.

Pour simplifier l'écriture des calculs, on introduit la notation vectorielle :

$$z = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}, \quad z_m = \begin{pmatrix} m_\alpha \\ m_\beta \end{pmatrix}, \quad z_\varepsilon = \begin{pmatrix} \varepsilon_\alpha \\ \varepsilon_\beta \end{pmatrix}$$

Ceci nous permet de réécrire z comme la somme : $z = z_m + z_\varepsilon$.

6.5.3 Hypothèse H_0 : bruit seul

Rappel de l'hypothèse

Sous l'hypothèse H_0, le signal en entrée du filtre $\mathcal{F}_k$ est un bruit blanc de variance σ^2. Ceci revient à dire que $\hat{y}(i) = \varepsilon e^{j\psi}$, ou $\alpha = \varepsilon_\alpha$ et $\beta = \varepsilon_\beta$, ou encore $m = 0$.

En utilisant la notation vectorielle, l'hypothèse H_0 revient à dire que

$$z_m = \theta = (0, 0)^T \tag{6.14}$$

La loi de probabilité conjointe de la partie réelle et de la partie imaginaire de $\hat{y}(i)$ s'exprime :

$$\begin{aligned} P_{A,B/H_0}(\alpha, \beta; \theta) &= P_{Z/H_0}(z; \theta) \\ &= P_{Z/H_0}(z) \end{aligned} \tag{6.15}$$

En respectant l'hypothèse H_0, les variables aléatoires α et β suivent toutes les deux une loi normale (de variance différente) :

$\alpha \hookrightarrow \mathcal{N}(0, \sigma_\alpha^2) \quad \beta \hookrightarrow \mathcal{N}(0, \sigma_\beta^2)$

Calcul de p_{R/H_0}

La loi de probabilité du module de $\hat{y}(i)$ est accessible par la connaissance de la loi conjointe de (α_i, β_i).

Soit J le jacobien de la transformation T qui à z associe $\begin{pmatrix} \rho \\ \omega \end{pmatrix}$.

$$T : \begin{pmatrix} \rho \ \cos\omega \\ \rho \ \sin\omega \end{pmatrix} \longrightarrow \begin{pmatrix} \rho \\ \omega \end{pmatrix}$$

Alors

$$\begin{aligned} p_{R,\Omega/H_0}(\rho, \omega) &= |J| \ p_{A,B/H_0}(\alpha_i, \beta_i) \\ &= |J| \ p_{R\cos\Omega, R\sin\Omega/H_0}(\rho \ \cos\omega, \rho \ \sin\omega) \\ &= |J| \ p_{Z/H_0}(z) \end{aligned} \tag{6.16}$$

$$J = \begin{pmatrix} \frac{\partial\alpha}{\partial\rho} & \frac{\partial\alpha}{\partial\omega} \\ \frac{\partial\beta}{\partial\rho} & \frac{\partial\beta}{\partial\omega} \end{pmatrix} = \begin{pmatrix} \cos\omega & -\rho\sin\omega \\ \sin\omega & \rho\cos\omega \end{pmatrix}$$

Le jacobien de T correspond au déterminant de la matrice J, soit :

$$|J| = \rho\ \cos^2\omega + \rho\ \sin^2\omega = \rho$$

Il vient :

$$p_{R,\Omega/H_0}(\rho,\omega) = \rho\ p_{Rcos\Omega,R\sin\Omega/H_0}(\rho\ cos\omega, \rho\ \sin\omega) \tag{6.17}$$

Pour obtenir la loi de probabilité p_0 du module de $\hat{y}(i)$, il suffit d'intégrer (**??**) par rapport à ω.

$$\begin{aligned} p_0 &= p_{R/H_0}(\rho) \\ &= \int_0^{2\pi} p_{R,\Omega/H_0}(\rho,\omega)d\omega \end{aligned} \tag{6.18}$$

6.5.4 hypothèse H_1 : bruit + sinus

Rappel de l'hypothèse

De même que pour sous l'hypothèse H_0, on décompose chaque valeur de TDF $\hat{y}(i)$ pour $i = 0, \ldots, L-1$ d'une suite de données $y(k)$ pour $k = 0, \ldots, N-1$ en partie réelle α et partie imaginaire β.

En respectant l'hypothèse H_1, les variables aléatoires α et β suivent toutes les deux une loi gaussienne (de variances différentes) :
$\alpha \hookrightarrow \mathcal{G}(m_\alpha, \sigma_\alpha^2)$
$\beta \hookrightarrow \mathcal{G}(m_\beta, \sigma_\beta^2)$

Loi de probabilité conjointe

Avec l'écriture vectorielle, la loi de probabilité conjointe de la partie réelle et de la partie imaginaire de $\hat{y}(i)$ est $P_{A,B/H_1}(\alpha,\beta;\theta,f_0)$.

Loi de probabilité du module

Exprimons maintenant la loi de probabilité en fonction du module ρ et de l'argument ω de $\hat{y}(i)$:

$$p_{R,\Omega/H_1}(\rho,\omega;\theta,f_0) = \rho P_{A,B/H_1}(\alpha,\beta;\theta,f_0)$$

La loi de probabilité du module p_1 s'écrit en intégrant selon ω, à savoir :

$$p_1 = \int_0^{2\pi} p_{R,\Omega/H_1}(\rho,\omega;\theta,f_0)\ d\omega$$

Rapport de vraisemblance généralisé

Notons $\overline{\theta} = arg\ \max_\theta p_1$. Dans ce cas, le rapport de vraisemblance généralisé s'écrit :

$$\Gamma = \frac{\max_{\theta \in H_1} \int_0^{2\pi} p_{R,\Omega/H_1}(\rho, \omega; \theta, f_0)\ d\omega}{\int_0^{2\pi} p_{R,\Omega/H_0}(\rho, \omega) d\omega} \tag{6.19}$$

6.6 Remarque sur le rapport signal sur bruit (RSB)

Dans toute l'étude présentée (y compris avec corrélation du bruit), nous utilisons l'hypothèse que le rapport signal sur bruit est connu. Même si ce n'est pas le cas, on remarquera que pour différents niveaux de bruit (raisonnablement choisis par rapport à des sons instrumentaux), le comportement des deux modèles reste cohérent.

Chapitre 7

Ecriture temporelle (bruit blanc)

7.1 Non corrélation du bruit

Pour simplifier les calculs et se faire une idée de la représentation du signal dans un plan Vraisemblance Temps-Fréquence, nous considérons dans un premier temps que les échantillons de bruit $b_k(n)$ ne sont pas corrélés. En réalité, le fait que les échantillons $b(n)$ soient passés au travers d'un filtre introduit une corrélation entre eux. Cependant, un modèle simplifié tel que celui proposé dans cette approche va nous permettre d'effectuer des calculs relativement simples, tout en aboutissant à un résultat satisfaisant sous certaines conditions.

La suite des échantillons avant passage dans le banc de filtres sera notée $x(n)$, avec :

$$x(n) = x_0(n) + b(n) \tag{7.1}$$

7.2 Hypothèses

Nous nous plaçons en sortie du filtre $\mathcal{F}_k$. Nous allons tester deux hypothèses.

Hypothèse H_0

Sous l'hypothèse H_0 (cf. **??**), La probabilité de H_0 s'écrit :

$$p_0 \quad = \quad \frac{1}{\sqrt{(2\ \pi\ \sigma^2)^n}}\ \exp\left(-\frac{1}{2\ \sigma^2} y^H y\right)$$

Ici, y^H correspond au vecteur y transposé conjugué (y étant réel, $y^H = y^T$). Cette probabilité ne dépend pas de θ.

Hypothèse H_1

Sous l'hypothèse H_1, la probabilité de H_1, est :

$$p_1 \quad = \quad \left(\frac{1}{\sqrt{2\ \pi\ \sigma^2}}\right)^n\ \exp\left(-\frac{1}{2\ \sigma^2}(y - x_0)^H\ (y - x_0)\right)$$

$$= \left(\frac{1}{\sqrt{2\ \pi\ \sigma^2}}\right)^n \exp\left(-\frac{1}{2\ \sigma^2}\left[y - D(f_0)\ \theta\right]^H\ \left[y - D(f_0)\ \theta\right]\right)$$

7.3 Rapport de vraisemblance généralisé

7.3.1 Maximisation de p_1

Maximiser p_1 revient à minimiser la quantité :

$$\begin{aligned} J(\theta) &= (y - x_0)^H\ (y - x_0) \\ &= (y - D(f_0)\ \theta)^H\ (y - D(f_0)\ \theta) \end{aligned}$$

Calculons la dérivée de $J(\theta)$:

$$\begin{aligned} \frac{\partial J(\theta)}{\partial \theta} &= \frac{\partial\ (y - D(f_0)\ \theta)^H\ (y - D(f_0)\ \theta)}{\partial \theta} \\ &= \sum_{n=0}^{N-1} \left[-2\ y(n) \cos(2\pi\, f_0\, n) + 2\, c_0\ \cos^2(2\pi\, f_0\, n)\right] \\ &+ \sum_{n=0}^{N-1} \left[2\, c_0\, s_0\ \cos(2\pi\, f_0\, n)\ \sin(2\pi\, f_0\, n)\right] \\ \frac{1}{N}\ \frac{\partial J(\theta)}{\theta} &= \frac{-2}{N} \sum_{n=0}^{N-1} y(n) \cos(2\pi\, f_0\, n) + \frac{2}{N} \sum_{n=0}^{N-1} c_0\ \cos^2(2\pi\, f_0\, n) \\ &+ \frac{2}{N} \sum_{n=0}^{N-1} c_0\, s_0\ \cos(2\pi\, f_0\, n)\ \sin(2\pi\, f_0\, n) \end{aligned}$$

Lorsque $N \longrightarrow +\infty$, les deux approximations suivantes sont vérifiées :

$$\begin{aligned} \frac{1}{N} \sum_{n=0}^{N-1} c_0\ \cos^2(2\pi\, f_0\, n) &= \frac{1}{2} \\ \frac{1}{N} \sum_{n=0}^{N-1} c_0\, s_0\ \cos(2\pi\, f_0\, n)\ \sin(2\pi\, f_0\, n) &= 0 \end{aligned}$$

A l'optimum, on note $\overline{\theta} = (\overline{c_0}, \overline{s_0})^T$ avec :

$$\begin{aligned} \overline{c_0} &= \frac{2}{N} \sum_{n=0}^{N-1} y(n)\ \cos(2\pi\, f_0\, n) \\ \overline{s_0} &= \frac{2}{N} \sum_{n=0}^{N-1} y(n)\ \sin(2\pi\, f_0\, n) \end{aligned}$$

On peut réécrire :

$$\overline{\theta} = \frac{2}{N} D^H(\overline{f_0})\ y \tag{7.2}$$

7.3.2 Calcul du rapport de vraisemblance généralisé

Une fois p_1 optimisé, sachant que p_0 ne dépend pas de θ, on peut calculer le rapport de vraisemblance généralisé :

$$\begin{aligned} \Gamma &= \exp\left(-\frac{1}{2\,\sigma^2}\ \left(D(\overline{f_0})\ \overline{\theta}\right)^H\ \left(2\ y - D(\overline{f_0})\ \overline{\theta}\right)\right) \\ &= \exp\left(\frac{1}{N^2}\, y^H\, D(\overline{f_0})\, D^H(\overline{f_0})\, y\right) \end{aligned}$$

et son logarithme népérien :

$$\ln(\Gamma) = \frac{1}{N^2}\, y^H\, D(\overline{f_0})\, D^H(\overline{f_0})\, y \qquad (7.3)$$

Nous implémenterons cette méthode et comparerons ses résultats à ceux de la méthode avec bruit corrélé dans le chapitre **??**.

7.3.3 Remarques à posteriori sur cette écriture

Dans le chapitre suivant, nous présenterons le calcul du rapport de vraisemblance généralisé dans un cas plus général, où l'on tient compte de la coloration du bruit. On remarquera que les résultats obtenus ici correspondent exactement à ceux du chapitre à venir en remplaçant la matrice de corrélation $R_b(f_k^c)$ par $\sigma^2 I$. La fréquence optimale $\overline{f_0}$ est celle qui optimise l'erreur entre le modèle et le signal, à savoir :

$$\overline{f_0} = \arg\max_{f_0} y^H\, D(f_0)\, D^H(f_0)\, y \qquad (7.4)$$

Chapitre 8

Ecriture temporelle (bruit coloré)

8.1 Positionnement du problème

Bien que les résultats obtenus sur des sons de synthèse simples soient suffisamment bons pour que l'on utilise le calcul du maximum de vraisemblance sans corrélation du bruit, nous aimerions tenir compte de l'effet de corrélation induit sur les parties bruitées des échantillons de signal lors du passage du filtre. Le modèle ainsi obtenu a de grandes chances d'être encore plus fiable en cas de fort bruit, et de donner des résultats encore plus fins (par exemple, de pouvoir détecter des partiels même lorsque le banc n'est pas accordé).

On se place à la sortie du banc de filtres à facteur de qualité constant. Pour chaque filtre, on désire savoir si l'énergie présente dans le signal après filtrage correspond à la présence d'un bruit corrélé (bruit blanc filtré), ou plutôt à la présence d'une sinusoïde, éventuellement mélangée à du bruit.

8.2 Notations

Nous conservons les notations posées en **??**. Le bruit $b(n)$ à l'entrée du banc de filtres est toujours supposé gaussien. En sortie du filtre $\mathcal{F}_k$, le bruit $b_k(n) = (b * h_k)(n)$ est corrélé (en notant h_k la réponse impulsionnelle du filtre).

Nous supposons que notre signal $x(n)$ suit le modèle "sinusoïde + bruit" (cf. definition en (**??**)).

La suite des échantillons avant passage dans le banc de filtres sera notée $x(n)$, avec $x(n) = x_0(n) + b(n)$ et $y_k(n)$ la sortie du k^e filtre ($k = 1, \ldots, 24$).

8.3 Bruit gaussien

Nous considérons $b(n)$ bruit gaussien, de moyenne statistique m_b, de matrice de corrélation R_b et de densité spectrale de puissance S_b. Après filtrage, le signal obtenu est lui aussi gaussien dans la bande du filtre.

Pour cette raison, nous considérons b_k bruit gaussien, de moyenne statistique m_{b_k}, de matrice de corrélation R_b et de densité spectrale de puissance $S_{b_k} = |H|^2.S_b$.

La matrice de corrélation R_b se constitue à partir des cœfficients de corrélation, définis comme suit :

$$r_b(l) = E[b_k(l+n)b_k(n)]$$

Dans le cas d'un signal réel, elle est symétrique. et s'exprime, pour un bruit constitué de N données $b(n)$:

$$R_b = \begin{pmatrix} r_b(0) & r_b(1) & \dots & r_b(N-1) \\ r_b(1) & r_b(0) & \dots & r_b(N-2) \\ \vdots & \vdots & \ddots & \vdots \\ r_b(N-1) & r_b(N-2) & \dots & r_b(0) \end{pmatrix}$$

La densité spectrale de puissance (d.s.p.) de b est la Transformée de Fourier Discrète de la fonction de corrélation r_b. Ainsi, on a :

$$\begin{aligned} S_{b_k}(e^{2j\pi f}) &= \frac{1}{N}\sum_{n=0}^{N-1} r_b(n)\ e^{-2j\pi n f} \\ r_b(l) &= \int_{-\frac{1}{2}}^{\frac{1}{2}} S_{b_k}(e^{2j\pi f})\ e^{-2j\pi l f} df \end{aligned}$$

On sait que la d.s.p. en sortie du filtre s'exprime en fonction de la d.s.p. en entrée, à savoir :

$$S_{b_k}(e^{2j\pi f}) = S_b(e^{2j\pi f})\ \left|H_k(e^{2j\pi f})\right|^2$$

où $\left|H_k(e^{2j\pi f})\right|^2$ est la réponse en fréquence du filtre $\mathcal{F}_k$ et $S_b(e^{2j\pi f}) = \sigma^2$. Tel qu'il a été construit, le filtre a pour réponse en fréquence :

$$\left|H_k(e^{2j\pi f})\right|^2 = \begin{cases} 1 \text{ si } f \in [-f_k^c - \Delta f_k^c; -f_k^c + \Delta f_k^c] \text{ ou } f \in [f_k^c - \Delta f_k^c; f_k^c + \Delta f_k^c] \\ 0 \text{ sinon} \end{cases}$$

Nous pouvons alors calculer $r_b(l)$:

$$\begin{aligned} r_b(l) &= \int_{-f_k^c-\Delta f_k^c}^{-f_k^c+\Delta f_k^c} \sigma^2\ e^{-2j\pi l f} df + \int_{f_k^c-\Delta f_k^c}^{f_k^c+\Delta f_k^c} \sigma^2\ e^{-2j\pi l f} df \\ &= \begin{cases} 4\,\sigma^2\ \Delta f_k^c & \text{si } k = 0 \\ \frac{2\,\sigma^2}{\pi\, l} \sin(2\,\pi\, l\,\Delta f_k^c)\ \cos(2\,\pi\, l\, f_k^c) & \text{sinon} \end{cases} \end{aligned}$$

Fig. 8.1: Valeurs de $r_b(k)$ pour une fréquence réduite $f_k^c = 0.25$, $\sigma^2 = 1$

On obtient des valeurs de $r_b(k)$ sinusoïde de fréquence $l\,\Delta f_k^c$ dont l'enveloppe est un cos à la fréquence $l f_k^c$ (cf. fig. **??**).

8.4 Rapport de vraisemblance généralisé

8.4.1 Hypothèse H_0

La probabilité de l'hypothèse H_0 est :

$$p_0 = \frac{1}{\sqrt{(2\pi)^n}\sqrt{\det(R_b(f_k^c))}} \exp\left(-\frac{1}{2} y^H \; {R_b}^{-1}(f_k^c) \; y\right) \tag{8.1}$$

8.4.2 Hypothèse H_1

La probabilité de l'hypothèse H_1 s'écrit :

$$p_1 = \frac{1}{\sqrt{(2\pi)^n}\sqrt{\det(R_b(f_k^c))}} \exp\left(-\frac{1}{2}[y - D(f_0)\theta]^H \; {R_b}^{-1}(f_k^c) \; [y - D(f_0)\theta]\right) \tag{8.2}$$

8.4.3 Rapport de vraisemblance généralisé

La connaissance du rapport de vraisemblance généralisé γ nécessite la maximisation de p_1 par rapport à θ.

8.4.4 Maximisation de p_1

Maximiser p_1 revient à minimiser la quantité

$$\epsilon^2(f_0) = [y - D(f_0)\;\theta]^H {R_b}^{-1}(f_k^c)\; [y - D(f_0)\;\theta] = \|y - D(f_0)\;\theta\|^2_{{R_b}^{-1}(f_k^c)}$$

On effectue les changements de variable suivants :

$$\begin{aligned} \tilde{y} &= {R_b}^{-1/2}(f_k^c)\; y \\ \tilde{D}(f_0) &= {R_b}^{-1/2}(f_k^c)\; D(f_0) \end{aligned}$$

Il est alors possible d'exprimer $\epsilon^2(f_0)$ sous la forme

$$\begin{aligned} \epsilon^2(f_0) &= \|\tilde{y} - \tilde{D}(f_0)\;\theta\|^2 \\ &= \left(\tilde{y} - \tilde{D}(f_0)\right)^H \; \left(\tilde{y} - \tilde{D}(f_0)\right) \end{aligned}$$

On sait que $\epsilon^2(f_0)$ est minimal lorsque sa dérivée par rapport à θ est nulle.

$$\begin{aligned} \frac{\partial}{\partial\theta}\left[\left(\tilde{y} - \tilde{D}(f_0)\;\theta\right)^H \left(\tilde{y} - \tilde{D}(f_0)\;\theta\right)\right] &= \frac{\partial}{\partial\theta^H}\left[\left(\tilde{y} - \tilde{D}(f_0)\;\theta\right)^H \left(\tilde{y} - \tilde{D}(f_0)\;\theta\right)\right]^H \\ &\quad + \frac{\partial}{\partial\theta^H}\left[\left(\tilde{y} - \tilde{D}(f_0)\;\theta\right)^H \left(\tilde{y} - \tilde{D}(f_0)\;\theta\right)\right] \\ &= 2\frac{\partial}{\partial\theta^H}\left[\left(\tilde{y} - \tilde{D}(f_0)\;\theta\right)^H \left(\tilde{y} - \tilde{D}(f_0)\;\theta\right)\right] \end{aligned}$$

$$
\begin{aligned}
&= 2\frac{\partial}{\partial\theta^H}\left[\left(\tilde{y}^H - \theta^H\ \tilde{D}(f_0)^H\right)\left(\tilde{y} - \tilde{D}(f_0)\ \theta\right)\right] \\
&= -2\frac{\partial}{\partial\theta^H}\left[\theta^H \tilde{D}(f_0)^H\left(\tilde{y} - \tilde{D}(f_0)\ \theta\right)\right] \\
&= -2\left[\tilde{D}(f_0)^H\left(\tilde{y} - \tilde{D}(f_0)\ \theta\right)\right] = 0
\end{aligned}
$$

On en tire finalement la valeur de θ optimal $\overline{\theta}$:

$$
\begin{aligned}
\overline{\theta} &= \left[\tilde{D}^H(f_0)\ \tilde{D}(f_0)\right]^{-1} \\
&= \tilde{D}^H(f_0)\ \tilde{y}
\end{aligned}
\tag{8.3}
$$

La connaissance de $\overline{\theta}$ nous donne l'expression de l'erreur quadratique :

$$
\begin{aligned}
\epsilon^2(f_0) &= \|\tilde{y} - \tilde{D}(f_0)\overline{\theta}\| \\
&= \left(\tilde{y} - \tilde{D}(f_0)\ \overline{\theta}\right)^H\left(\tilde{y} - \tilde{D}(f_0)\ \overline{\theta}\right) \\
&= \left(\tilde{y} - \tilde{D}(f_0)\ \overline{\theta}\right)^H \tilde{y} - \left(\tilde{y} - \tilde{D}(f_0)\ \overline{\theta}\right)^H \tilde{D}(f_0)\ \overline{\theta}
\end{aligned}
$$

or on sait que $(\tilde{y} - \tilde{D}(f_0)\ \overline{\theta})^H\ \tilde{D}(f_0)\ \overline{\theta} = 0$, puisqu'à l'optimum, ces deux vecteurs (dont on calcule ici le produit scalaire) sont orthogonaux. Nous pouvons donc poursuivre le calcul :

$$
\begin{aligned}
\epsilon^2(f_0) &= (\tilde{y} - \tilde{D}(f_0)\ \overline{\theta})^H\ \tilde{y} \\
&= \tilde{y}^H\ \tilde{y} - \overline{\theta}^H\ \tilde{D}^H(f_0)\ \tilde{y} \\
&= y^H\ {R_b}^{-1}(f_k^c)\ y - \overline{\theta}^H\ D^H(f_0)\ {R_b}^{-1}(f_k^c)\ y \\
&= y^H\ {R_b}^{-1}(f_k^c)\ y \\
&\quad -y^H\ {R_b}^{-1}(f_k^c)\ D(f_0)\ \left[D^H(f_0)\ {R_b}^{-1}(f_k^c)\ D(f_0)\ \right]^{-1} D^H(f_0)\ {R_b}^{-1}(f_k^c)\ y \\
&= y^H\ {R_b}^{-1}(f_k^c)\ y - y^H\ Q(f_0)\ y
\end{aligned}
$$

en posant $Q(f_0) = {R_b}^{-1}(f_k^c)\ D(f_0)\ \left[D^H(f_0)\ {R_b}^{-1}(f_k^c)\ D(f_0)\ \right]^{-1} D^H(f_0){R_b}^{-1}(f_k^c)$.

Minimisation de l'erreur quadratique

Il nous faut maintenant minimiser $\epsilon^2(f_0)$ par rapport à la fréquence f_0, ce qui revient à trouver la fréquence $\overline{f}_0 \in B_k$ (où B_k est la bande passante du filtre $\mathcal{F}_k$ définie par $[f_k^c - \Delta f_k^c; f_k^c + \Delta f_k^c]$) pour laquelle les amplitudes $\overline{c}_0$ et $\overline{s}_0$ données par $\overline{\theta}$ font coïncider au mieux notre modèle $x_0(n)$ avec le signal $x(n)$ que l'on analyse dans $\mathcal{F}_k$.

$$
\begin{aligned}
\overline{f}_0 &= \arg\ \min_{f_0}\ \epsilon^2(f_0) \\
&= \arg\ \max_{f_0}\ y^H\ Q(f_0)\ y
\end{aligned}
$$

Il s'agit ici d'une optimisation monodimensionnelle selon f_0. On calculera la quantité $y^H\,Q(f_0)\,y$ pour différentes valeurs de fréquences de la forme $\frac{i}{L}$ comprises dans B_k, on construira un polynôme passant par tous ces points (à l'aide de fonctions splines, par exemple) et on en extraira le maximum, duquel on déduira la fréquence $\overline{f}_0$ pour laquelle il est atteint.

Cela dit, une première implémentation dasn laquelle on approchait $\overline{f}_0$ par f_k^c donne les mêmes résultats lorsque la fréquence d'excitation et proche de f_k^c !

Rapport de vraisemblance généralisé

De la valeur optimale $\overline{f}_0$, nous obtenons la formule :

$$\overline{Q}(\overline{f}_0) = {R_b}^{-1}(f_k^c)D(\overline{f_0})\ \left[D^H(\overline{f_0})\ {R_b}^{-1}(f_k^c)\ D(\overline{f_0})\right]^{-1}\ D^H(\overline{f_0})\ {R_b}^{-1}(f_k^c)$$

Le rapport de vraisemblance généralisé prend finalement l'expression :

$$\begin{aligned}\Gamma &= \frac{\exp\left[-\left(\frac{1}{2}\ y^H\ R_b(f_k^c)\ y - y^H\ \overline{Q}(\overline{f_0})\ y\right)\right]}{\exp\left[-\frac{1}{2}\ y^H R_b(f_k^c) y\right]} \\ &= \exp\left[\frac{1}{2}\ y^H\ \overline{Q}(\overline{f_0})\ y\right]\end{aligned}$$

On préférera parler du logarithme népérien du rapport de vraisemblance généralisé, à savoir :

$$\ln(\Gamma) = \frac{1}{2}\, y^H\, \overline{Q}(\overline{f_0})\, y \tag{8.4}$$

8.4.5 Limites de l'écriture temporelle

Nous avons cherché à simplifier le calcul du rapport de vraisemblance. En effet, $R_b(f_k^c)$ est très mal conditionnée, et ne peut donc s'inverser. Par contre, si on arrive à montrer que $(D^H(\overline{f_0})\; {R_b}^{-1}(f_k^c)\; D(\overline{f_0}))^{-1}$ est proche de l'identité, le calcul de $\overline{Q}(\overline{f_0})$ s'en trouvera grandement simplifié.

La matrice $R_b(f_k^c)$ est Tœplitz, et pourrait donc sous certaines conditions s'approcher par une matrice circulante $C_B(f_k^c) = F^H\; \mathcal{S}\; F$ (cf.fig. **??** milieu), où $\mathcal{S}$ est une matrice diagonale (contenant le spectre du filtre de fréquence centrale f_k^c) et F la matrice de Fourier.

Dans ce cas, l'inverse de $C(f_k^c)$ est simple à calculer : $C(f_k^c) = F\; \mathcal{S}^{-1}\; F^H$. Il semble alors possible de trouver la simplification proposée pour l'écriture de $\overline{Q}(\overline{f_0})$ (nous passons ici les détails de calcul, n'étant pas utiles pour la suite du propos). Une autre approximation, $\frac{1}{2}\; (C_B(f_k^c) + C_B^T(f_k^c))$, rectifie le défaut de symétrie de$C_B(f_k^c)$ tout en étant elle aussi une matrice circulante (cf.fig. **??** droite).

Fig. 8.2: Matrice de corrélation $R_b(f_k^c)$ (à gauche), son approximation $C_B(f_k^c) = F^H\; \mathcal{S}\; F$ (au milieu) et $\frac{1}{2}\; (C_B(f_k^c) + C_B^T(f_k^c))$ (à droite) pour des fenêtres de 512 points et un bruit de variance $\sigma^2 = 1.$ avec $f_k^c = 0.233$

A ceci près que l'hypothèse de base n'est pas vérifiée ! La matrice $R_b(f_k^c)$ n'est pas suffisamment creuse (cf. fig. **??** pour des fenêtres temporelles de 512 points) pour être approchée par une matrice circulante. En effet, $R_b(f_k^c)$ ne se diagonalise pas sur la base de Fourier, ce qui implique que toute approximation de $R_b(f_k^c)$ par des matrices de type circulante donnera des résultats insuffisants (cette approximation se faisant pour des matrices relativement creuses et se décomposant sur la base de Fourier).

Nous devons donc, à contre-cœur, repousser cette hypothèse et attaquer notre problème sous un autre angle. Puisque nous étions en train de faire intervenir une écriture fréquentielle dans la formule (**??**), nous proposons de l'introduire dès les premières hypothèses, afin de la faire intervenir, non plus sur $R_b(f_k^c)$, mais sur des matrices plus petites, dont on espère qu'elle pourront s'inverser.

Chapitre 9

Ecriture fréquentielle

Dans ce chapitre, nous allons travailler avec le plus grand module des valeurs de la Transformée de Fourier Discrète comprises dans la bande du filtre. Une fois ce maximum déterminé, nous calculons le rapport de vraisemblance généralisé afin de savoir s'il s'agit du maximum du lobe principal de la TF d'une sinusoïde ou plutôt de bruit coloré.

9.1 Recherche du module maximum

9.1.1 Nombre de points fréquentiels L

Lorsqu'on calcule L transformées de Fourier discrètes sur toute la largeur du spectre de fréquence, seule un tout petit nombre de valeurs de $\hat{y}(i)$ sont dans la bande $\mathcal{B}_k$ du filtre $\mathcal{F}_k$.

Il est donc important de bien choisir cette valeur de L. Typiquement, pour 1024 points de transformées de Fourier, seuls 8 points sont compris dans la bande $\mathcal{B}_1$ du premier filtre, et presque le double dans le dernier filtre.

9.1.2 Nombre de points temporels N

On calcule chaque transformée de Fourier discrète à partir des N valeurs temporelles $y(i),\ i = 0, ..., N-1$. Si ce nombre est trop petit, cela revient à prendre peu d'arches de sinusoïde. On ne peut alors espérer avoir assez d'information sur cette sinusoïde pour obtenir une bonne résolution fréquentielle, et délimiter précisément les lobes de la sinusoïde.

Cependant, on désire connaître suffisamment bien la forme de ces lobes, de telle sorte que, lorsque le lobe principale est dans le filtre, le maximum qu'on détermine corresponde le mieux possible au maximum réel du lobe.

Cette condition s'exprime de la façon suivante : soit $t_{fenetre}$ le temps d'observation et soit ν_k^c la fréquence réduite de la sinusoïde observée. Observer plusieurs arches de la sinusoïde revient à dire que :

$$\nu_k^c.t_{fenetre} \gg 1 \tag{9.1}$$

D'autre part, on a :

$$\begin{aligned}
\nu_k^c &= \frac{f_k^c}{F_e} \\
t_{fenetre} &= \frac{N}{T_e} \\
&= N\,F_e
\end{aligned}$$

De ces définitions, nous tirons une autre expression de la condition (**??**) :

$$N\,f_k^c \gg 1$$

Rappelons que $f_k^c \in [f_-; f_+]$. Autrement dit, $f_k^c \geq f_- = 0.9709 f_{cal}$, où f_{cal} est la fréquence de calibrage du banc de filtre. Pour une analyse sur N_o, on aura aussi N points. Simplement, ces N points temporels correspondront, pour le l^e octave, à une durée temporelle $2^{N_o - l}$ supérieure à celle du plus haut octave. La condition (**??**), s'exprime alors :

$$0.9709\ N\ f_{cal} \gg 1$$

d'où l'on extrait la condition sur la taille minimale de la fenêtre en fonction de la fréquence de calibrage du filtre :

$$N \gg \frac{1}{0.9709 f_{cal}} \tag{9.2}$$

Fig. 9.1: Résolution du lobe principal d'une sinusoïde pour une fenêtre temporelle de 256 points (gauche) et 512 points (droite)

On donne dans les figures qui suivent des exemples de la qualité de restitution du lobe pour 256 points (fig. **??** gauche) et 512 points (fig. **??** droite).

Nous remarquons que pour de 256 points temporels, la restitution du lobe est mauvaise. Aussi, par la suite, nous prendrons des fenêtres de taille 512 (soit une dizaine d'arches de sinusoïde).

9.1.3 Limites de la détection qui en découle

La précision en fréquence qui nous est imposée est celle du modèle lui-même, à savoir le quart de ton. Par contre, la limite temporelle est fixée par les impératifs donnés en **??** : elle est divisée par deux à chaque octave inférieure que l'on analyse.

9.1.4 Méthode de recherche du maximum

En respectant les conditions données précédemment à la fois sur la taille de la TFD et sur la taille de la fenêtre temporelle, on obtient environ une dizaine de points dans chaque filtre. Lorsque le maximum de ces points est sur l'un des bords du filtre, on le prendra comme maximum. par contre, lorsqu'il sera au milieu du filtre, on recherchera le maximum sur une courbe **spline** contenant environ vingt fois plus de points, afin de connaître précisément la fréquence de la sinusoïde.

Fig. 9.2: Recherche du maximum, lorsqu'il est au milieu du filtre (gauche) ou sur un bord du filtre (droite)

Fig. 9.3: Recherche du maximum : TFD du signal, TFD du filtre et maximum

9.2 Hypothèse H_0 : bruit seul

9.2.1 Probabilité de l'hypothèse

La loi de probabilité conjointe de la partie réelle et de la partie imaginaire de $\hat{y}(i)$ s'exprime :

$$P_{A,B/H_0}(\alpha,\beta) = P_{Z/H_0}(z) = \frac{1}{2\pi\sqrt{\det(C_2(f_0))}} \ \exp\left(-\frac{1}{2} z_\varepsilon^T \ C_2^{-1}(f_0) \ z_\varepsilon\right) \tag{9.3}$$

où $C_2(f_0)$ est la matrice de covariance de α_ε et β_ε.

En respectant l'hypothèse H_0, les variables aléatoires α et β suivent toutes les deux une loi normale (de variance différente) :

$\alpha \hookrightarrow \mathcal{N}(0,\sigma_\alpha^2)$

$\beta \hookrightarrow \mathcal{N}(0,\sigma_\beta^2)$

Dans ce cas la matrice de covariance du processus aléatoire s'écrit quant à elle :

$$C_2(f) = E(z_\varepsilon \, z_\varepsilon^T) = E(z \, z^T) = \begin{pmatrix} \sigma_\alpha^2 & E(\alpha\,\beta) \\ E(\alpha\,\beta) & \sigma_\beta^2 \end{pmatrix}$$

9.2.2 Calcul exact de la matrice de covariance $C_2(f)$

En utilisant la matrice $D_N(f)$ définie en (**??**) (prise ici à la fréquence $f = -f_i = -i/L$), on peut exprimer le vecteur z_i comme suit :

$$z(i) = D_N^T(-i/L) \ y(i) = \ D_N^T(-f_i) \ y_i$$

La matrice de covariance se formule alors :

$$\begin{aligned} C_2(f_i) &= E\left(D_N^T(-f_i) \ y_i \ y_i^T \ D_N(-f_i)\right) \\ &= D_N^T(-f_i) \ E(y_i \ y_i^T) \ D_N(-f_i) \\ &= D_N^T(-f_i) \ R_b(f_k^c) \ D_N(-f_i) \end{aligned}$$

9.2.3 Approximation de la matrice de covariance $C_2(f)$ par $k.I_2$

La matrice de covariance peut encore s'écrire :

$$C_2(f) = \sigma_\alpha\sigma_\beta \begin{pmatrix} \frac{\sigma_\alpha}{\sigma_\beta} & \frac{E(\alpha\,\beta)}{\sigma_\alpha\sigma_\beta} \\ \frac{E(\alpha\,\beta)}{\sigma_\alpha\sigma_\beta} & \frac{\sigma_\beta}{\sigma_\alpha} \end{pmatrix}$$

Par la suite, on notera $\sigma_{\alpha\beta}^2 = \sigma_\alpha \sigma_\beta$, $\mu = \frac{E(\alpha\,\beta)}{\sigma_\alpha \sigma_\beta}$ et $r = \frac{\sigma_\alpha}{\sigma_\beta}$.

$$C_2(f) = \sigma_{\alpha\beta}^2 \begin{pmatrix} r & \mu \\ \mu & \frac{1}{r} \end{pmatrix}$$

Si l'hypothèse selon laquelle la matrice de covariance $C_2(f)$ est proportionnelle à $k.I_2$ se vérifie, alors $\mu \approx 0$ et $r \approx 1$.

Fig. 9.4: Etude de la matrice de covariance C_2 pour des fenêtres temporelles de 128 points

A l'aide des figures telles que fig. **??**, on vérifie ces propriétés. Cependant, il nous faut quantifier plus précisément la distance qui sépare $C_2(f)$ de $k.I_2$.

9.2.4 Définition d'un critère d'égalité de $C_2(f)$ et $k.I_2$

Dans [**?**] p.260, on trouve un critère d'estimation de l'hypothèse selon laquelle la matrice C_n, matrice définie positive, est proportionnelle à la matrice identité I_2.

La critère défini par :

$$d(C_n) = \frac{\sqrt[n]{\det(C)}}{\frac{1}{n}\ tr(C)}$$

est inférieure ou égale à 1, et sert de critère pour l'estimation de cette hypothèse. N'ayant pas l'utilité du résultat de ce test d'hypothèse, nous utiliserons cette "distance" (qui n'est cependant pas une distance au sens mathématique) à titre indicatif, sachant que l'expérimentation nous montre que pour des valeurs de $d(C_2)$ supérieures à 0.8, l'identification de C_2 à I_2 est satisfaisante.

D'après sa définition et en l'appliquant à notre cas, la distance de la matrice de covariance $C_2(f)$ à la matrice identité $k.I_2$ est :

$$d(C_2) = \frac{2\sqrt{\sigma_\alpha^2 \sigma_\beta^2}}{\sigma_\alpha^2 + \sigma_\beta^2} \tag{9.4}$$

Les fenêtres temporelles sur lesquelles nous seront amenés à travailler comptent 512 points au minimum. La distance $d(C_2)$ minimale est atteinte pour le dernier filtre du banc, mais ne descend cependant qu'à 0.92 pour 64 points (fig. **??** gauche), 0.97 pour 128 points, 0.993 pour 256 points et 0.996 pour 512 points (fig. **??** droite).

Fig. 9.5: Distance à la matrice identité pour 64 points temporels (gauche) et 512 points temporels (droite)

Ces valeurs minimales, qui caractérisent la plus grande distance entre $C_2(f)$ et $k.I_2$, sont donc très au-dessus de 0.8 : les résultats obtenus pour le calcul du maximum de vraisemblance qui utilise cette approximation seront donc très proche de la vraie valeur du maximum de vraisemblance. Nous validons ainsi l'approximation proposée, à savoir :

$$C_2 \approx \sigma_{\alpha\beta}^2\ I_2 \tag{9.5}$$

On remarquera au passage que dans le milieu de la bande, $\sigma^2_{\alpha\beta} \approx \sigma^2$. Sachant que lorsque le banc de filtres n'est pas accordé (cf. **??**), on peut décaler sa fréquence de calibrage f_{cal} pour l'accorder, on prendra cette approximation pour les calculs numériques.

9.2.5 Calcul de p_{R/H_0}

$$\begin{aligned}
p_{R,\Omega/H_0}(\rho,\omega) &= \rho\ p_{Rcos\Omega,R\sin\Omega/H_0}(\rho\ cos\omega,\rho\ \sin\omega) \\
&= \frac{\rho}{2\pi\sigma_{\alpha\beta}}\ \exp\left(-\frac{1}{2\ \sigma^2_{\alpha\beta}}\ z_\varepsilon^T\ I_2^{-1}\ z_\varepsilon\right)\ \mathbb{1}_{\rho\in[0,+\infty[}\ \mathbb{1}_{\omega\in[0,2\pi[} \\
&= \frac{\rho}{2\pi\sigma_{\alpha\beta}}\ \exp\left(-\frac{1}{2\ \sigma^2_{\alpha\beta}}\ z_\varepsilon^T\ z_\varepsilon\right)\ \mathbb{1}_{\rho\in[0,+\infty[}\ \mathbb{1}_{\omega\in[0,2\pi[} \\
&= \frac{\rho}{2\pi\sigma_{\alpha\beta}}\ \exp\left(-\frac{\alpha_\varepsilon^2+\beta_\varepsilon^2}{2\ \sigma^2_{\alpha\beta}}\right)\ \mathbb{1}_{\rho\in[0,+\infty[}\ \mathbb{1}_{\omega\in[0,2\pi[}
\end{aligned}$$

Sachant que selon l'hypothèse H_0, le signal entrant dans le filtre est un bruit seul, $\varepsilon^2_\alpha+\varepsilon^2_\beta = \alpha^2+\beta^2 = \rho^2$. De cette égalité, on obtient une expression simple de $p_{R,\Omega/H_0}(\rho,\omega)$.

$$p_{R,\Omega/H_0}(\rho,\omega) = \frac{\rho}{2\pi\sigma_{\alpha\beta}}\ \exp\left(-\frac{\rho^2}{2\ \sigma^2_{\alpha\beta}}\right) \tag{9.6}$$

Pour obtenir la loi de probabilité p_0 du module de $\hat{y}(i)$, il suffit d'intégrer (**??**) par rapport à ω.

$$\begin{aligned}
p_0 &= p_{R/H_0}(\rho) \\
&= \int_0^{2\pi} p_{R,\Omega/H_0}(\rho,\omega)d\omega \\
&= \int_0^{2\pi} \frac{\rho}{2\pi\sigma_{\alpha\beta}}\ \exp\left(-\frac{\rho^2}{2\ \sigma^2_{\alpha\beta}}\right)d\omega \\
&= \frac{\rho}{\sigma_{\alpha\beta}}\ \exp\left(-\frac{\rho^2}{2\ \sigma^2_{\alpha\beta}}\right)\int_0^{2\pi}\frac{1}{2\pi}d\omega
\end{aligned}$$

$$p_0 = \frac{\rho}{\sigma_{\alpha\beta}}\ \exp\left(-\frac{\rho^2}{2\ \sigma^2_{\alpha\beta}}\right) \tag{9.7}$$

9.3 hypothèse H_1 : bruit + sinus

9.3.1 Loi de probabilité conjointe

Avec l'écriture vectorielle, la loi de probabilité conjointe de la partie réelle et de la partie imaginaire de $\hat{y}(i)$ s'exprime :

$$P_{A,B/H_1}(\alpha,\beta;m,f_0) = \frac{1}{2\pi\sqrt{\det(C_2(f_0))}}\ \exp\left(-\frac{1}{2} z_\varepsilon^T\ C_2^{-1}(f_0)\ z_\varepsilon\right) \tag{9.8}$$

où C_2 est la matrice de covariance de α_ε et β_ε, f_0 et m étant à déterminer.

L'hypothèse en (**??**) selon laquelle $C_2(f_0) \approx \sigma_{\alpha\beta}^2\ I_2$ est encore une fois vérifiée, puisque les hypothèses faites sur la partie bruitée du signal sont les mêmes pour H_0 et H_1.

$$\begin{aligned}
P_{A,B/H_1}(\alpha,\beta;m,f_0) &= \frac{1}{2\pi\sqrt{\sigma_{\alpha\beta}^2}}\ \exp\left(-\frac{1}{2\sigma_{\alpha\beta}^2} z_\varepsilon^T\ I_2^{-1}\ z_\varepsilon\right)\\
&= \frac{1}{2\pi\sigma_{\alpha\beta}}\ \exp\left(-\frac{\varepsilon_\alpha^2+\varepsilon_\beta^2}{2\ \sigma_{\alpha\beta}^2}\right)\\
&= \frac{1}{2\pi\sigma_{\alpha\beta}}\ \exp\left(-\frac{(\alpha-m_\alpha)^2+(\beta-m_\beta)^2}{2\ \sigma_{\alpha\beta}^2}\right)\\
&= \frac{1}{2\pi\sigma_{\alpha\beta}}\ \exp\left(-\frac{(\alpha^2+\beta^2)+(m_\alpha^2+m_\beta^2)-2(\alpha\ m_\beta+\beta\ m_\alpha)}{2\ \sigma_{\alpha\beta}^2}\right)\\
&= \frac{1}{2\pi\sigma_{\alpha\beta}}\ \exp\left(-\frac{\rho^2+m^2-2\rho\ m\ \cos(\omega-\phi)}{2\ \sigma_{\alpha\beta}^2}\right)
\end{aligned}$$

9.3.2 Loi de probabilité du module

Exprimons maintenant la loi de probabilité en fonction du module ρ et de l'argument ω de $\hat{y}(i)$:

$$\begin{aligned}
p_{R,\Omega/H_1}(\rho,\omega;m,f_0) &= \rho P_{A,B/H_1}(\alpha,\beta;m,f_0)\\
&= \frac{\rho}{2\pi\sigma_{\alpha\beta}}\ \exp\left(-\frac{\rho^2+m^2-2\rho\ m\ \cos(\omega-\phi)}{2\ \sigma_{\alpha\beta}^2}\right)
\end{aligned}$$

La loi de probabilité du module p_1 s'écrit en intégrant selon ω, à savoir :

$$\begin{aligned}
p_1 &= \int_0^{2\pi} p_{R,\Omega/H_1}(\rho,\omega;m,f_0)\ d\omega\\
&= \frac{\rho}{\sigma_{\alpha\beta}}\ \exp\left(-\frac{\rho^2+m^2}{2\ \sigma_{\alpha\beta}^2}\right)\int_0^{2\pi}\frac{1}{2\pi}\ \exp\left(\frac{\rho\ m}{\sigma_{\alpha\beta}^2}\ \cos(\omega-\phi)\right)\ d\omega
\end{aligned}$$

L'intégrande $\frac{1}{2\pi}\ \exp\left(\frac{\rho\, m}{\sigma_{\alpha\beta}^2}\ \cos(\omega-\phi)\right)$ est une fonction 2π-périodique, aussi :

$$\int_0^{2\pi}\frac{1}{2\pi}\exp\left(\frac{\rho\ m}{\sigma_{\alpha\beta}^2}\ \cos(\omega-\phi)\right)\ d\omega = \int_0^{2\pi}\frac{1}{2\pi}\exp\left(\frac{\rho\ m}{\sigma_{\alpha\beta}^2}\ \cos(\omega-\phi)\right)\ d\omega$$

De plus, $\exp\left(\frac{\rho m}{\sigma^2_{\alpha\beta}} \cos(\omega-\phi)\right)$ est une fonction paire, donc :

$$\int_0^{2\pi} \frac{1}{2\pi} \exp\left(\frac{\rho\ m\ \cos\omega}{\sigma^2_{\alpha\beta}}\right) d\omega = \int_0^{\pi} \frac{1}{\pi} \exp\left(\frac{\rho\ m\ \cos\omega}{\sigma^2_{\alpha\beta}}\right) d\omega$$

On reconnaît une fonction de Bessel modifiée d'ordre 0, que l'on note $I_0\left(\frac{\rho m}{\sigma^2_{\alpha\beta}}\right)$. En effet, la fonction de Bessel modifiée d'ordre ν est définie par :

$$I_\nu(x) = \int_0^{\pi} \frac{1}{\pi} \cos(\nu\omega)\ \exp\left(x cos\omega\right) d\omega - \frac{\sin\ \pi\nu}{\nu} \int_0^{+\infty} \exp\left(-x\ \cosh t - \nu t\right) dt$$

Pour $\nu = 0$, on obtient $I_0(x) = \int_0^\pi \frac{1}{\pi} \exp\left(x\ \cos\omega\right) d\omega$. Nous obtenons l'expression (**??**) pour la loi de probabilité du module de $\hat{y}_i$.

$$p_1 = \frac{\rho}{\sigma_{\alpha\beta}}\ \exp\left(-\frac{\rho^2+m^2}{2\ \sigma^2_{\alpha\beta}}\right)\ I_0\left(\frac{\rho\ m}{\sigma^2_{\alpha\beta}}\right) \tag{9.9}$$

9.4 Rapport de vraisemblance généralisé

Notons $\overline{m} = arg\ \max_m p_1$. Dans ce cas, le rapport de vraisemblance généralisé Γ s'écrit :

$$\Gamma \quad = \quad \frac{\frac{\rho}{\sigma_{\alpha\beta}}\ \exp\left(-\frac{\rho^2+\overline{m}^2}{2\ \sigma^2_{\alpha\beta}}\right)\ I_0\left(\frac{\rho\ \overline{m}}{\sigma^2_{\alpha\beta}}\right)}{\frac{\rho}{\sigma_{\alpha\beta}}\ \exp\left(-\frac{\rho^2}{2\ \sigma^2_{\alpha\beta}}\right)}$$

En simplifiant les termes communs au numérateur et au dénominateur, on obtient :

$$\Gamma = \exp\left(-\frac{\overline{m}^2}{2\ \sigma^2_{\alpha\beta}}\right)\ I_0\left(\frac{\rho\ \overline{m}}{\sigma^2_{\alpha\beta}}\right) \tag{9.10}$$

9.4.1 Recherche de la valeur $\overline{m}$ qui maximise p_1

La forme générale de la courbe de la probabilité p_1 en fonction de m est donnée à titre indicatif (fig. **??** gauche). Cette probabilité possède donc un maximum en $\overline{m}$, cette valeur dépendant de ρ. Pour les faibles valeurs de ρ, la loi de probabilité p_1 possède son maximum en 0 (cf. fig. **??** droite).

Fig. 9.6: Loi de probabilité p_1 pour $\rho/\sigma_{\alpha\beta} = 0.8$ (gauche) et pour $\rho/\sigma_{\alpha\beta} = 2.2$ (droite)

Le maximum de p_1 est atteint lorsque sa dérivée s'annule. Pour dériver (**??**), il nous faut connaître la dérivée de la fonction de Bessel modifiée d'ordre 0.

$$
\begin{aligned}
\frac{d\,I_0(a\,x)}{d\,x} &= \frac{d}{d\,x}\left(\frac{1}{\pi}\int_0^\pi \exp(ax\ \cos\theta)\ d\theta\right)\\
&= \frac{1}{\pi}\int_0^\pi a\,\cos\theta\ \exp(a\,x\ \cos\theta)\ d\theta\\
&= a\left[\frac{1}{\pi}\int_0^\pi \cos\theta\ \exp(a\,x\ \cos\theta)\ d\theta\right]\\
&= a\ I_1(a\,x)
\end{aligned}
$$

Nous pouvons donc calculer $\frac{d\,p_1}{d\,m}$. Auparavant, nous renormalisons $I_0\left(\frac{\rho\ m}{\sigma^2_{\alpha\beta}}\right)$. Pour ce faire, notons $r_a = \frac{\rho}{\sigma_{\alpha\beta}}$ le rapport signal sur bruit connu à posteriori et $r_b = \frac{m}{\sigma_{\alpha\beta}}$ le rapport signal (sinusoïde) sur bruit à priori. A l'optimum, m est noté $\overline{m}$; r_b est quant à lui noté $\overline{r_b}$.

Avec les notations proposées, $\frac{d\,p_1}{d\,m} = 0 \Longleftrightarrow \frac{d\,p_1}{d\,r_b} = 0$

$$
\begin{aligned}
p_1 &= r_a\ \exp\left(-\frac{r_a^2+r_b^2}{2}\right)\ I_0(r_a\,r_b)\\
\frac{d\,p_1}{d\,r_b} &= \frac{1}{\sigma_{\alpha\beta}}\,r_a\left[r_a\,I_1(r_a\,r_b) - r_b\,I_0(r_a\,r_b)\right]\ \exp\left(-\frac{r_a^2+r_b^2}{2}\right)
\end{aligned}
$$

Nous cherchons à résoudre l'équation implicite

$$
r_a\,I_1(r_a\,\overline{r_b}) - \overline{r_b}\,I_0(r_a\,\overline{r_b}) = 0 \tag{9.11}
$$

en fonction de $\overline{r_b}$.

Par le nouveau changement de variable $u = r_a\,\overline{r_b}$ et en multipliant **??** à gauche et à droite par $\overline{r_b}$, on peut exprimer $\overline{r_b}$ et $r_a = \frac{u}{\overline{r_b}}$ en fonction de u, à savoir :

$$
\begin{aligned}
\overline{r_b} &= \sqrt{\frac{u\,I_1(u)}{I_0(u)}}\\
&= s(u)\\
r_a &= \sqrt{\frac{u\,I_0(u)}{I_1(u)}}\\
&= t(u)
\end{aligned}
$$

La méthode de calcul de $\overline{r_b}$ est simple :

1. tabuler la fonction $t(u)$;
2. déterminer la valeur u telle que $t(u) = r_a$.
3. calculer $s(u) = \overline{r_b}$.

Le résultat obtenu est donné en fig. **??**. On remarque que la courbe se découpe en trois zones :

- si $r_a \in [0; 1.41]$ alors $\overline{r_b} = 0$. Cela signifie que si le module de la TFD maximale mesurée dans le filtre n'est pas suffisamment élevé par rapport à la puissance du bruit, la probabilité que l'on ait mesuré la TFD d'une sinusoïde est faible ;
- si $r_a \in [1.41; 3]$, la valeur mesurée est dans une plage d'indécision ; le signal mesuré n'est pas franchement une sinusoïde ;
- si $r_a \geq 3$ alors asymptotiquement $\overline{r_b} = r_a$. Pour un module mesuré suffisamment grand, on considérera qu'il s'agit bien d'une sinusoïde.

Fig. 9.7: $\overline{r_b} = f(r_a)$

9.4.2 Calcul effectif du rapport de vraisemblance généralisé

Nous avons ainsi obtenu la valeur optimale $\overline{m}$ en fonction du module ρ et de la racine de la variance $\sigma_{\alpha\beta}$, et pouvons évaluer le rapport de vraisemblance dans le cas où l'on considère le bruit corrélé à l'aide de (**??**).

Cependant, pour de grandes valeurs du paramètre $\frac{\rho\,\overline{m}}{\sigma^2_{\alpha\beta}}$, la fonction $I_0(\frac{\rho\,\overline{m}}{\sigma^2_{\alpha\beta}})$ diverge très vite et n'est plus calculable sous Matlab (pour des valeurs de $\frac{\rho}{\sigma_{\alpha\beta}} > 25$, ce qui est largement dépassé lorsque l'on détecte la présence d'un signal dans un filtre).

Par contre, dans [**?**] et dans [**?**], on trouve un développement en série asymptotique de $I_\nu(x)$ lorsque $x \longrightarrow +\infty$:

$$\begin{aligned} I_\nu(x) &= \frac{e^x}{\sqrt{2\pi\, x}} \left[1 + \sum_{m=1}^{M} (-1)^m \frac{\prod_{k=1}^{m} \left[4\nu^2 - (2k-1)^2\right]}{m!\,(8x)^m} \right] \qquad (9.12) \\ I_\nu(x) &\approx \frac{e^x}{\sqrt{2\pi\, x}} \end{aligned}$$

Fig. 9.8: Approximation de $\ln(\Gamma)$ par son développement asymptotique

Puisque la quantité qui nous intéresse est le logarithme népérien du rapport de vraisemblance généralisé, on a :

$$\ln(\Gamma) \approx \frac{1}{2}\,\frac{\overline{m}^2}{\sigma^2_{\alpha\beta}} - \frac{1}{2}\,\ln\left(2\pi\frac{\overline{m}^2}{\sigma^2_{\alpha\beta}}\right) \qquad (9.13)$$

Nous avons testé cette approximation en la comparant au calcul réalisé par la fonction correspondant de Matlab (besseli(0,x)), et observons (fig. **??**) qu'elle est tout à fait correcte, même pour des valeurs de x proches de 1.

Pour de faibles valeurs de ρ, on préférera utiliser :

$$\ln(\Gamma) = -\frac{1}{2}\,\frac{\overline{m}^2}{\sigma_{\alpha\beta}^2} + \ln\left[I_0\left(\frac{\rho\,\overline{m}}{\sigma_{\alpha\beta}^2}\right)\right] \tag{9.14}$$

Chapitre 10

Résultats obtenus par les deux méthodes

10.1 Avertissement

Lorsque nous calculons un rapport de vraisemblance généralisé, on utilise (la plupart du temps) ensuite un seuil. Lorsque le rapport de vraisemblance Γ est inférieur au seuil, on refute l'hypothèse selon laquelle le signal est une sinusoïde bruitée. Par contre, dès que Γ est supérieur au seuil, on accepte l'hypothèse. La détermination du seuil n'est pas une mince affaire, car il faut d'abord calculer la probabilité de fausse alarme (on détecte une sinusoïde qui n'existe pas) et la probabilité de non détection (on ne détecte pas une sinusoïde qui exsite, cette fois).

La procédure de calcul de ces probabilités est très longue, puisqu'il faut, pour chaque filtre, exécuter toute une série de simulations, pour lesquels une partie des paramètres se règlent à l'aveugle. Nous avons donc décidé, dans le temps qui nous était imparti, de proposer quelques résultats “bruts”, et nous nous laissons le plaisir de remettre à plus tard ces simulations. L'analyse proposée ici est donc qualitative.

10.2 Sons de synthèse utilisés

Nous avons généré des signaux contenant les fréquences qui nous intéressent pour cibler des phénomènes particuliers sur nos filtres. Nous avons la possibilité d'ajouter du bruit (et quantifiant le rapport signal sur bruit), de décaler les fréquences du signal par rapport aux fréquences centrales des filtres.

Les sons utilisés comporte en général 2048 échantillons (avec une fréquence d'échantillonnage $F_e = 44\,000\ Hz$), et l'étude ne porte que sur le premier octave dans la plupart des cas, pour diminuer le temps de calcul.

10.3 Limites du programme utilisé

Dans la version du programme utilisée pour les résultats qui vont suivre, le retard de chaque filtre n'est pas corrigé ; aussi, nous verrons apparaître un décalage des transitoires en fonction des fréquences.

10.4 Nombre de cœfficients des filtres du banc

Nous avons expliqué en **??** que nous pouvions modifier plusieurs paramètres du gabarit des filtres, afin d'abaisser le nombre de cœfficients tout en sachant que nous perdrions en précision. Nous allons tester dans un cas simple l'effet de ces paramètres.

Fig. 10.1: Largeur de décroissance minimale (Δf_k^c) : sortie du banc de filtres (gauche), vraisemblance sans corrélation du bruit (milieu) et avec corrélation du bruit (droite), $RSB = 20\ dB$; 4 fenêtres non recouvrantes

Dans un premier cas, nous prendrons les filtres les plus sélectifs (de largeur L_k, de bande de décroissance à gauche et à droite $\Delta f_k^c = \frac{1}{2} L_k$, et de précision 0.001 sur la bande passante), ce qui donne jusqu'à 771 cœfficients pour le premier filtre du banc. Nous allons comparer les résultats donnés par ce banc de filtres (cf. fig. **??**) avec ceux d'un banc dont on aura élargi la bande de décroissance (pour chaque filtre) à 1.15 Δf_k^c (soit de 15%) avec une précision en bande passante de 0.01, et dont le premier filtre sera défini par 499 cœfficients seulement.

Fig. 10.2: Modification de la largeur de décroissance (1.15 Δf_k^c) : sortie du banc de filtres (gauche), vraisemblance sans corrélation du bruit (milieu) et avec corrélation du bruit (droite), $RSB = 20\ dB$; 4 fenêtres non recouvrantes

Les figures **??** milieu et droite, obtenues par calcul du maximum de vraisemblance en sortie d'un banc de filtres moins sélectifs que prévu, sont identiques à celles obtenues avec le banc de filtres (cf.fig. **??**) défini au chapitre **??**. On pourra donc par la suite se permettre de relâcher les contraintes sur les filtres. Cependant, afin de bien découpler les phénomènes, nous avons effectué les simulations suivantes avec le banc de "référence".

10.5 Différences de seuillage

Nous noterons que les résultats affichés par les deux méthodes présentent de grandes différences quant à la valeur du maximum de vraisemblance. Pour la méthode sans corrélation du bruit, cette valeur est impédante du filtre où on la mesure (cf. fig. **??** gauche), puisqu'aucune hypothèse sur le filtre n'est faite lors des calculs.

Fig. 10.3: Différence de seuillage : sortie du banc de filtres (gauche), vraisemblance sans corrélation du bruit (milieu) et avec corrélation du bruit (droite), $RSB = 20\ dB$; 4 fenêtres non recouvrantes

Par contre, pour le modèle avec corrélation du bruit, la valeur du maximum dépend de la largeur du filtre (cf. fig. **??** droite). Ceci revient à dire que lorsqu'un calcul de seuils sera mis en place, les figures affichées après seuillage auront la même allure pour les deux méthodes, ce qui n'est pas encore le cas puisque le calcul de seuil n'est pas encore réalisé. Les filtres du banc n'ont pas été renormalisés en gain, c'est de là que vient ce phénomène particulier. Dans une prochaine étape, on prendra soin d'apporter une correction de gain adaptée.

Nous prendrons en compte cette remarque dans les analyses des phénomènes de bruitage, de transition et de désaccordage.

Dans l'exemple donné fig. **??**, nous excitons les filtres 1, 5, 9, 13, 17 et 21. Le rapport signal sur bruit est de 20 dB. Les fenêtres sur lesquelles nous calculons le rapport de vraisemblance sont recouvrantes.

10.6 Bruitage des signaux

Nous désirons savoir si nos modèles sont robustes en cas de signaux bruités. Pour étudier ce phénomène seul, nous allons générer un signal ne contenant qu'une sinusoïde à la fréquence centrale du 12^e filtre (on dira qu'on excite le 12^e filtre). Nous ne regardons pas pour l'instant le phénomène de transition, aussi un seul filtre est excité (régime supposé stationnaire).

Fig. 10.4: Phénomène de bruitage ($RSB = 5\ dB$) : sortie du banc de filtres (à gauche), vraisemblance sans corrélation du bruit (milieu) et avec corrélation du bruit (droite) ; 4 fenêtres non recouvrantes

Rappelons la définition du rapport signal sur bruit (RSB) :

$$RSB = 10\ \log_{10}\left(\frac{P_{signal}}{P_{bruit}}\right) \tag{10.1}$$

Ici, nous prendrons pour puissance du signal P_{signal} la puissance des 6 sinusoïdes présentes dans le signal. C'est une définition "pessimiste" ; en effet, nous aurions pu prendre pour puissance du signal la puissance d'une seule sinusoïde (en considérant qu'elles aient toute la même puissance).

Jusqu'à un rapport signal sur bruit de 5 dB, on observe que les deux modèles ne détecte que la sinusoïde. Comme on peut le voir fig. **??** pour un RSB de 2 dB, les deux méthodes ne donnent pas les mêmes résultats. On s'attendrait à voir de meilleurs résultats pour la méthode avec corrélation du bruit, et l'œil nous dit le contraire. Cependant, méfions nous de nos yeux : les résultats présentés ne font pas intervenir de seuil de vraisemblance !

Fig. 10.5: Phénomène de bruitage ($RSB = 2\ dB$) : sortie du banc de filtres (à gauche), vraisemblance sans corrélation du bruit (milieu) et avec corrélation du bruit (droite)

Fig. 10.6: Phénomène de bruitage ($RSB = 2\ dB$) : sortie du banc de filtres (à gauche), vraisemblance sans corrélation du bruit (milieu) et avec corrélation du bruit (droite)

Si l'on n'avait qu'un partiel à détecter (cf. fig. **??**), les résultats seraient bien différents. En effet, jusqu'à un RSB de 2 dB, on détecte très bien la sinusoïde par les deux méthodes.

De manière générale, on note que les deux méthodes détectent bien les sinusoïde lorsque le rapport signal sur bruit est supérieur ou égal à 5 dB. Dans les signaux musicaux composés uniquement de sons harmoniques, l'imperfection des sinusoïde (modélisées ici par le bruit) peut être considérée comme donnant un rapport signal sur bruit de cet ordre.

10.7 Effet de transition

Nous proposons d'abord de voir quelles indications nous fournissent nos deux modèles pour des signaux plus ou moins bruités dont la fréquence augmente puis diminue d'un quart de ton, sur des durées inférieures à la taille de la fenêtre de calcul du rapport de vraisemblance.

Le son de synthèse comporte 2048 échantillons. Pendant les 500 premiers, les filtres 3 et 10 sont excités. Pendant les 400 suivants, ce sont les filtres 4 et 11, puis à nouveau les filtres 3 et 10 pendant 300 échantillons seulement, et enfin les filtres 2 et 9 sur les 848 échantillons restant.

10.7.1 Signaux non bruités

Fig. 10.7: Phénomène de transition ($RSB = 20\ dB$) : sortie du banc de filtres (gauche), vraisemblance sans corrélation du bruit (milieu) et avec corrélation du bruit (droite) ; 4 fenêtres non recouvrantes

A partir du moment où le rapport signal sur bruit est supérieur au rapport de l'amplitude du lobe principale sur celle du lobe secondaire de la transformée de Fourier de chaque sinusoïde, on considère que le bruit est nul. C'est le cas pour un rapport signal sur bruit de 20 dB (cf. fig. **??** gauche). Dans ce cas, les résultats semblent tout à fait correct.

Fig. 10.8: Phénomène de transition ($RSB = 20\ dB$) : vraisemblance sans corrélation du bruit (gauche) et avec corrélation du bruit (droite) ; 49 fenêtres recouvrantes

D'autre part, le transitoire est assez bien assimilé à une sinusoïde dans le premier modèle, alors que le second rejette à raison cette hypothèse. Une première différence notable entre les deux méthodes est d'ejà mise à jour.

Bien que nous n'ayons pas déterminé les seuils de vraisemblance, on remarque que les zones sombres sont raisonnables pour des fenêtres non recouvrantes (cf. fig. **??**), et très précises pour des fenêtres recouvrantes (cf. fig. **??**).

On ne peut cependant dire laquelle des deux méthodes donne les meilleurs résultats : en effet, même si l'on est tenté d'effectuer un jugement "à l'œil", tant que l'étude statistique des modèles n'aura pas été faite, nous n'aurons aucun élément sérieux de discrimination.

10.7.2 Signaux bruités

On effectue ensuite la même étude sur de signaux assez bruités (le rapport signal sur bruit passe à 5 dB). On n'utilise cette fois-ci uniquement des fenêtres recouvrantes pour gagner en précision de détection temporelle.

Fig. 10.9: Phénomène de transition ($RSB = 5\ dB$) : sortie du banc de filtres (gauche), vraisemblance sans corrélation du bruit (milieu) et avec corrélation du bruit (droite) ; 13 fenêtres recouvrantes

Fig. 10.10: Phénomène de transition ($RSB = 5\ dB$) : vraisemblance sans corrélation du bruit (gauche) et avec corrélation du bruit (droite) ; 49 fenêtres recouvrantes

Nous avons testé les modèles pour des rapports signal sur bruit de 5 dB (cf. fig. **??**) et 1 dB (cf. fig. **??**), soit la même puissance de signal utile et de bruit. Malgré cela, les deux modèles fonctionnent : celui sans corrélation de bruit n'indique que les partiels, mais à priori un peu moins précisément que le modèle avec corrélation du bruit (qui lui, par contre, détecte une sinusoïde de très courte durée dans les hautes fréquences alors qu'il n'y en a pas : ce genre d'erreurs se corrige aisément lors de la prise de décision de présence ou non de sinusoïde en fonction du rapport de vraisemblance et de la durée des événements mesurés).

10.8 Effet de décentrage ou désaccord

10.8.1 Signaux non bruités

On décentre petit à petit la fréquence de la sinusoïde par rapport à la fréquence centrale du filtre. Pour comparaison, on excite deux filtres : l'un sans décalage (le numéro 2), l'autre avec le décalage variable (le numéro 13). Dans tous les cas, puisqu'on ne regarde plus l'effet de transition (quoiqu'il soit présent en début de signal), seules 4 fenêtres d'analyse non recouvrantes sont utilisées.

Le décalage indiqué est exprimé en fonction de Δf_k^c. Ainsi, un décalage de 20 % correspond à une fréquence d'excitation $f_0 = f_k^c + 0.2\Delta f_k^c$, et ainsi de suite.

Dans le modèle sans corrélation du bruit, nous avons considéré que la fréquence centrale du filtre faisait une bonne approximation de la fréquence qui optimise p_1. Cette hypothèse n'est évidemment plus vérifiée lorsqu'on désaccorde le banc de filtres et les résultats s'en ressentent. Nous ne sommes donc pas surpris de voir que pour un décentrage de l'ordre de 20 % de Δf_k^c, plus rien ne soit détecté dans le modèle sans corrélation du bruit.

Par contre, le modèle avec corrélation du bruit passe progressivement d'un filtre à l'autre ! On peut donc aisément détecter si le banc de filtres est accordé ou non.

10.9 Remarque sur la précision temporelle

Nous proposons un petit calcul "récréatif", donnant, pour chaque octave, la fréquence de calibrage du banc F_c, la gamme de fréquence concernée $[F_-; F_+]$, la précision temporelle t

Fig. 10.11: Phénomène de transition ($RSB = 1\ dB$) : vraisemblance sans corrélation du bruit (gauche) et avec corrélation du bruit (droite) ; 49 fenêtres recouvrantes

Fig. 10.12: Phénomène de désaccord (20%) : sortie du banc de filtres (à gauche), vraisemblance sans corrélation du bruit (milieu) et avec corrélation du bruit

Fig. 10.13: Phénomène de désaccord (40%) : sortie du banc de filtres (à gauche), vraisemblance sans corrélation du bruit (milieu) et avec corrélation du bruit

et le nombre de notes par seconde N_{notes} auquel correspond une taille de fenêtre d'analyse de F_{fen} points pour une fréquence d'échantillonnage de 44 100 Hz.

octave	F_c	F_-	F_+	t	N_{notes}	F_{fen}
10	10279.7	10130.7	20555.4	0.01	86	512
9	5139.8	5065.3	10277.7	0.02	43	1024
8	2569.9	2532.7	5138.8	0.04	21	2048
7	1285.0	1266.3	2569.4	0.09	10	4096
6	642.5	633.2	1284.7	0.18	5	8192
5	321.2	316.6	642.3	0.37	2.7	16384
4	160.6	158.3	321.2	0.7	1.3	32768
3	80.3	79.1	160.6	1.5	0.7	65536
2	40.1	39.6	80.3	3.0	0.34	131072
1	20.1	19.8	40.1	6.0	0.16	262144

Nous sommes relativement exigeants, puisqu'en général, les méthodes recherchant les partiels s'arrêtent à 7 ou 8 octaves. Cela dit, dans la gamme de fréquence supérieures à 1 285 Hz, on obtient une précision d'au moins 10 notes par secondes. Cette vitesse d'exécution ne concerne que les solistes ou des notés arpégées, et les notes jouées à cette vitesse sont plutôt aigües ; en tout cas dans cette gamme de fréquence. Nos modèles permettent donc d'atteindre une précision de détection de notes très intéressante.

10.10 Utilisation de la méthode des "climbers"

Nous avons enfin implémenté l'algorithme des "climbers" sur le plan vraisemblance temps-fréquence, à la fois sans corrélation du bruit (cf. fig. **??** milieu et droite) et avec corrélation (cf. fig. **??**), pour un signal assez difficile à lire, puisqu'il contient sur une seule octave 6 partiels, avec un rapport signal sur bruit de 5 dB et des notes durant entre 300 et 500 échantillons (soit entre 6.8 et 11.7 ms).

Fig. 10.14: Phénomène de désaccord (60%) : sortie du banc de filtres (à gauche), vraisemblance sans corrélation du bruit (milieu) et avec corrélation du bruit

Fig. 10.15: Méthode des climbers : sortie du banc (gauche), plan vraisemblance temps-fréquence sans corrélation du bruit (milieu), crêtes trouvées à l'aide de 5000 marcheurs

Fig. 10.16: Méthode des climbers : plan vraisemblance temps-fréquence avec corrélation du bruit (gauche), crêtes trouvées à l'aide de 5000 marcheurs

10.11 Commentaires

Ces résultats ne semblent pas surprenants. Le modèle sans corrélation est volontairement simple. Il ne fonctionne plus lorsque le banc est désaccordé, mais donne de bones indications dans le cas contraire, pour un temps de calcul très faible devant celui de la méthode avec corrélation. Cette dernière est plus efficace à la fois pour détecter un désaccord et pour déterminer où ont lieu les transitions lorsque le signal est peu bruité.

Par contre, pour des signaux bruités, il est difficile de donner des résultats même qualitatifs. Il serait plus judicieux d'attendre d'avoir mis en œuvre le calcul des seuils avant de s'exprimer.

La méthode des "marcheurs fous" trouve ici sa place, pour détecter les crêtes du plan vraisemblance temps-fréquence. Il ne nous restera que quelques réglages à effectuer sur le modèle (normaliser le gain de chaque filtre, entre autres) avant de pouvoir l'utiliser sur plusieurs octaves et sur des sons réels.

Chapitre 11

Conclusions - Perspectives

Cette étude propose plusieurs éléments de réponse à la problématique de la poursuite des partiels qui se pose lorsque l'on veut réaliser automatiquement la transcription de pièces de musiques.

Après la mise en place d'un banc de filtres à facteur de qualité constant, nous avons appliqué une méthode de recherche des crêtes dans le plan temps-fréquence (méthode des "climbers", ou "marcheurs fous") qui donne de très bons résultats pour les signaux contenant plusieurs partiels, ce qui est le cas en musique.

De plus, deux modèles statistiques de représentation temps-fréquence de l'information contenue dans un signal harmonique bruité sont proposés. Ils utilisent un calcul de rapport de vraisemblance généralisé sur l'hypothèse de présence d'un signal harmonique, et ceci dans chaque filtre. Le premier modèle ne prend pas en compte la corrélation du bruit induite par le filtre, tandis que la deuxième, plus lourde à mettre en œuvre, corrige cet effet de corrélation.

Enfin, à défaut d'une étude statistique complète des deux modèles, un ensemble de tests sont réalisés, à la fois sur le banc de filtre et sur les deux méthodes.

Ainsi, on sait que les contraintes imposées sur les filtres sont trop rigides et peuvent être relâchées. De plus, les deux modèles fonctionnent correctement en présence de fort bruit. Les phénomènes de transitions (changement de notes) sont pris en compte à l'aide de fenêtres d'analyse glissantes, et permettent une bonne localisation de ces transitions sans trop alourdir les calculs. Notons aussi que le modèle avec corrélation du bruit est très fiable pour détecter un désaccordage du banc de filtres, et que ceci se corrige à l'aide d'un simple paramètre.

L'étude se poursuivra par des tests sur des sons réels, puis par l'étude statistique des modèles et l'élaboration de méthodes de regroupement des partiels pour trouver les différentes fréquence fondamentales présentes dans le signal.

Annexe A

Présentation des programmes réalisés sous Matlab

A.1 Organigramme

On donne en figure fig. **??** l'organigramme des programmes, représentés par leurs noms sans l'extension ".m".

Fig. A.1: Organigramme des programmes

Le programme principal **tests.m** génère au besoin le banc de filtres, le son de synthèse à analyser, puis effectue l'analyse du plan Temps-Fréquence selon la méthode temporelle avec bruit blanc et la méthode fréquentielle avec bruit coloré. Ensuite, dans chacun des plans Vraisemblance-Temps-Fréquence obtenus, l'utilisateur peut appliquer l'algorithme des climbers.

A.2 Description des programmes

Succintement, nous présentons ici les principes des routines écrites durant ce stage, afin de faciliter la compréhension de celui ou ceux qui s'aventurerai(en)t à poursuivre le projet. Les routines concernant la mise en place du banc de filtres et des climbers ont été réalisées par Carine Duval, et en partie remaniées par moi. Toutes les autres ont été entièrement écrites par mes soins.

A.2.1 Calcul des cœfficients du banc de filtres

Les cœfficients de chaque filtre du banc de filtres sont calculés de façon à ce que le filtre respecte le gabarit donné par l'utilisateur. On vérifie d'abord que la fréquence de calibrage (fréquence centrale du premier filtre) respecte les conditions de construction du banc. Les gabarits sont ensuite corrigés de façon à normaliser la puissance de chaque filtre. Enfin, les cœfficients de chaque filtre sont calculés, et renvoyés à l'utilisateur.

A.2.2 Synthèse du son d'analyse

Son synthétique

La routine **GenerateTune.m** génére un son synthétique, somme de sinusoïde de même amplitude, dont les fréquences sont données par l'utilisateur. Pour un rapport signal sur bruit (RSB) donné, la puissance du bruit additionnel est calculée, les échantillons de bruit sont ensuite normalisés avant d'être ajoutés aux échantillons du signal "somme de sinusoïde".

Fréquences d'excitation

Les fréquences correspondent en général à des partiels harmoniques (ou non) connus au préalable. Toutefois, dans le but de tester le banc de filtres et les routines de calcul de plan vraisemblance-temps-Fréquence, une routine permettant d'exciter les filtres qui nous intéressent, en leur fréquence centrale ou non. Cette routine s'appelle **GenerateFrequencies.m**, et utilise les variables suivantes :

- **NFilter** : numéros des filtres concernés ;
- **DecalageF** : décalage de chaque fréquence dans la bande du filtre, par rapport à sa fréquence centrale, en pourcentage ;
- **NOctaves** : nombres d'octaves concernés ;
- **Indic** : indicateur de présence ou non du partiel, octave par octave.

Les indicateurs de présence permettent de moduler, en fonction de chaque octave, les filtres excités, ce qui donne plus de souplesse à l'utilisateur désireux de tester le banc de filtres.

A.2.3 Plan Vraisemblance-Temps-Fréquence : éléments communs aux deux implémentations

Pour les deux méthodes, le plan Vraisemblance-Temps-Fréquence est calculé de la manière suivante :

```
pour chaque octave
  pour chaque filtre
    pour chaque fenetre glissante
      calculer le rapport de vraisemblance generalisé
    fin
  fin
fin
```

A.2.4 Plan Vraisemblance-Temps-Fréquence par la méthode temporelle

On applique ici les résultats des calculs présentés au chapitre **??** : dans une première étape, on calcule les cœfficients d'amplitude du modèle à la fréquence centrale du filtre, puis on calcul le rapport de vraisemblance.

A.2.5 Plan Vraisemblance-Temps-Fréquence par la méthode fréquentielle

Tout d'abord, on recheche le bin de fréquence de la FFT dont le module est maximum dans la bande du filtre concerné. Ceci se fait à l'aide de la routine **ModuleMaxBande.m**.

Ensuite, selon les valeurs du rapport entre ce module maximum $\overline{m}$ et le niveau de bruit σ^2, on donne la valeur du module du signal utile p (cf.**??**).

- pour $\overline{m} < 1.41$, ;
- pour $1.41 \leq \overline{m} < 3$, on calcule p par interpolation dans les valeurs d'une table calculée au préalable (correspondant à la figure **??**), et ceci à l'aide de **ValeurApprocheeNopt.m**;
- pour $\overline{m} \geq 3$, on prend $p = \overline{m}$.

Enfin, on calcule le rapport de vraisemblance, soit par la formule directe pour p petit, soit á l'aide de son développement asymptotique pour $p \geq 3$.

A.2.6 Climbers

La routine **MoveClimbers.m** met à jour les densités de climbers selon les deux mesures, une fois le déplacement des climbers réalisé par **CalcClimbersDensites.m**. L'algorithme proposé en [**?**] est scrupuleusement respecté.

A.2.7 Affichages

Pour afficher les résultats des plans Vraisemblance-Temps-Fréquence et du plan des climbers, nous avons choisi de stocker chaque plan temps fréquence dans un tableau de 24 lignes (une par filtre), et suffisamment de colonnes pour contenir toutes les données des n octaves représentés. Les variables utilisées sont :

- **Tableau** contient les données à afficher ;
- **Taille** contient le nombre de colonne des données de chaque octave ;
- **Temps** correspond aux instants temporels, placés en abcisse ;
- **Temps** correspond aux fréquences, placées en ordonnée ;
- **NumFigure** indique le numéro de la figure générée.

A.3 Fonctions d'affichage de résultats

Des fonctions d'affichage des résultats ont été écrite, en grande partie pour créer les figures du rapport. Il s'agit de :

- **Affiche_2e_condition.m** : nombre de cœfficients du filtre passe-bas de Hamming en fonction de la fréquence de calibrage du banc ;

- **Affiche_Approx_Gamma.m** : approximation du rapport de vraisemblance (par la méthode fréquentielle avec prise en compte de la coloration du bruit) par son développement asymptotique ;

- **Affiche_Banc_Filtres.m** : pour afficher la sortie d'un banc de filtres, il faut initialiser plsuieurs paramètres : longueur du signal temporel **TuneLength**, données **eInitial**, nombre d'octave d'analyse **NOctaves**, fréquence d'échantillonnage **Fe** et puissance du bruit (σ^2) **Sigma** ;

- **Affiche_CalcCoefficientsBancFiltres.m** : il faut absolument donner la bonne valeur pour F_{cal}, la fréquence de calibrage du banc, sinon le gabarit ne sera pas bien positionné par rapport au filtre ;

- **Affiche_Climbers.m** : pour afficher le résultat de l'algorithme des climbers sur la sortie d'un banc de filtres ; il faut initialiser plsuieurs paramètres : longueur du signal temporel **TuneLength**, données **eInitial**, nombre d'octave d'analyse **NOctaves**, fréquence d'échantillonnage **Fe**, puissance du bruit (σ^2) **Sigma**, nombre de climbers **Nc**, température **T**, taux de décroissance **alpha**, nombre d'itération sans test **Niter**, nombre d'itérations maximal **Narret**, pourcentage de climbers immobiles **PourcArret** pour arrêter l'algorithme ;

- **Affiche_Matrice_Covariance_C.m** : pour observer la distance de la matrice C_2 à l'identité I_2, ainsi que chacun de ses cœfficients ;

- **Affiche_MaximisationP1.m** : courbe de R_b en fonction de R_a lors de la résolution de l'équation implicite **??** (méthode fréquentielle) ;

- **Affiche_module_max_bande.m** : recherche du module maximum du spectre dans la bande du filtre (méthode fréquentielle) ;

- **Affiche_Plages_Frequences.m** : présente les plages de fréquence du banc de filtres pour n itérations de l'analyse en échelle linéaire et en logarithmique ;

- **Affiche_Rb.m** : matrice de covariance pour la méthode temporelle avec bruit coloré, et comparaison avec son approximation ; il faut préciser la fréquence centrale du filtre (f_k^c) **f0**, la fréquence de la sinusoïde **f**, la puissance du bruit (σ^2) **Sigma** et la taille de la fenêtre d'analyse **N** (fixée à 512 dans nos calculs) ;

A.4 Fonctions de test des routines développées

Plusieurs fonctions de test ont été écrites au fur et à mesure dans le but de valider les routines. Notons les plus utiles :

- **TestApplyBankFilter.m**
- **TestCalcClimbersDensites.m**
- **TestCalcCoefficientsBancFiltres.m**
- **TestEstimation.m**

- **TestGraphiqueTempsFrequence2.m**
- **TestValeur_Approchee_Nopt.m**

Il peut être utile de les essayer afin de se donner une idée de ce que fait chaque routine présentée.

Annexe B

Programmes

B.1 Routines de calcul

B.1.1 CalcClimbersDensites.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%  Calcul de la densite et de la densite ponderee
% (par la matrice M) de climbers, apres leur mvt
%
%   Paramemetres d'entree :
%
%   M      Description sous forme matricielle du plan tps-freq
%   Nc      Nbre de climbers
%   T Facteur de decroissance
%   alpha   Taux de decroissance
%   Niter    Nbre d'iterations, 1er critere d'arret
%   NArret Nbre d'iterations prises en compte lors du 2eme critere d'arret
%   PourcArret Pourcentage (compris entre 0 et 1) de climbers qui doivent rester
% immobiles pour satisfaire le 2eme critere d'arret
%
%
%   Densite      de climbers apres leur mvt ds le plan tps-freq
%   DensitePonderee  par M de climbers apres leur mvt ds le plan tps-freq

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


function [Densite,DensitePonderee] = CalcClimbersDensites( M, Nc, T, ...
   alpha,  Niter, NArret, PourcArret)

%[F,T] = size(M)
[ct, cf] = MoveClimbers( M, Nc, T, alpha, Niter, NArret, PourcArret);
[F,T] = size(M);

Densite = zeros(F,T);
DensitePonderee = zeros(F,T);

 for k=1:Nc
    Densite(cf(k),ct(k)) = Densite(cf(k),ct(k)) + 1 ;
    DensitePonderee(cf(k),ct(k)) = DensitePonderee(cf(k),ct(k)) + M(cf(k),ct(k));
 end
```

B.1.2 CalculCoefficientsBancFiltres.m

```
function CalculCoefficientsBancFiltres(fc, Fe, precBP, precBC, FactDeltaF)
```

```
%::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
% function CalculCoefficientsBancFiltres(fc, Fe, precBP, precBC, FactDeltaF)
%::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
%
%  Calcul des coeffs des 24 filtres d'un banc analysant un octave
% de la gamme tempérée.
%
%::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
%  Gabarit d'un filtre:
%     - fc          frequence centrale
%     - largeur     1/4 de ton
%     - precBP      precision sur la bande passante
%     - precBC      precision surla bande de coupure
%    - FactDeltaF  facteur correctif sur DeltaF (+ qq %)
%::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
% CoefficientsBancFiltres  Matrice de 23 lignes. Chacune d'elle represente
%        un filtre du banc par son ordre (en 1ere colonne) et ses coefficients.
% A savoir que le nbre de coeffs d'un filtre est egale a
% son ordre+1.
%::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
% V. Verfaille (juillet 2000)
%::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::


fprintf('\n\n ==> Calcul des coefficients du banc de filtre\n\n');
fprintf('     frequence centrale (reduite)         : %5.3f\n',fc/Fe);
fprintf('     precision en bande passante          : %5.3f\n', precBP);
fprintf('     precision en bande de coupure        : %5.3f\n', precBC);
fprintf('     facteur correctif sur DeltaF optimal : %5.3f\n\n', FactDeltaF);

Max = 1. / (2. * 2^(23/24) * (2^(49/48)-1));
N = 1024;

% Passage en frequences reduites
fc = fc / Fe;

if (fc < Max)

  fc = 2. * fc; % Respect de la condition Matlab (pb de symetrie)
  QuartTon = 2^(1/24);
  DemiQuartTon = 2^(1/48);
  FInf = fc / DemiQuartTon;
  FSup = fc * DemiQuartTon;
  DeltaF = FactDeltaF * ( fc * DemiQuartTon - fc);

  % Calcul des coefficients du 1er filtre

  [ordre,GabFreq,ao,w] = remezord([FInf-DeltaF FInf FSup FSup+DeltaF],...
     [0 1 0],[precBC precBP precBC]);
  fprintf('\n     filtre n  1 : %4d coefficients\n', ordre);
  CoefficientsBancFiltres = zeros(24,ordre+1);

  b = remez(ordre, GabFreq, ao, w);
  [h,f] = freqz(b,1,N,1);
  FacteurNormalisation = sqrt (abs(h')*abs(h));
  filtre = b; %  1. / sqrt(abs(h')*abs(h)) * b;
  CoefficientsBancFiltres(1,1:ordre+2) = [ordre, filtre];

  % Calcul des coefficients des 23 autres filtres

  for k=1:23

     fc = fc*QuartTon;

     FInf = fc/DemiQuartTon;
```

```
     FSup = fc*DemiQuartTon;
     DeltaF = FactDeltaF * (fc*DemiQuartTon-fc);

     [ordre,GabFreq,ao,w] = remezord([FInf-DeltaF FInf FSup FSup+DeltaF],...
        [0 1 0],[precBC precBP precBC]);
     fprintf('     filtre n %2d : %4d coefficients\n',k+1, ordre);
     b = remez(ordre,GabFreq,ao,w);  % length(b)=ordre+1
     % renormalisation du filtre
     [h,f] = freqz(b,1,N,1);
     filtre = FacteurNormalisation / sqrt (abs(h')*abs(h)) * b;
     CoefficientsBancFiltres(k+1,1:ordre+2) = [ordre, filtre];

  end

  save CoefficientsBancFiltres CoefficientsBancFiltres;

else
   fprintf('\n --> Erreur : la frequence centrale du bank de filtre "fc" doit\n',...
      '--> etre inferieure a %f\n\n', Max);
end
```

B.1.3 Estimation_bruit_non_correle.m

```
function [LogRDVG] = Estimation_bruit_non_correle (y, Fc, Fe, Sigma)

%
% function [LogRDVG] = Estimation_bruit_non_correle (y, Fc, Fe, Sigma)
%
% Estimation du Ln du Rapport de Vraisemblance Generalise
% sous l'hypothese y(n) = c0 * cos(2 pi Fc n) + s0 * sin(2 pi Fc n) + b(n)
%
%:::::::::::::::::::::::::::::::::::::::::::::::::::::
%  y         : signal temporel
%  Fc        : frequence centrale du Filtre
%  Fe        : frequence d'echantillonnage
%  Sigma     : puissance du bruit
%:::::::::::::::::::::::::::::::::::::::::::::::::::::


WindowLength = length(y);
expo=zeros(size(y));
expo(1:WindowLength) = exp(2*i*pi*Fc*(0:WindowLength-1));
c0 = 2. / WindowLength * y * real(expo)';
s0 = 2. / WindowLength * y * imag(expo)';
modele = c0 * real(expo) + s0 * imag(expo);
LogRDVG = 1./(2.*Sigma) * modele * (2.*y - modele)';
```

B.1.4 Estimation_bruit_colore.m

```
function [logRDVG] = Estimation_bruit_colore ...
   (y, Fc, Fe, Fcalibrage, Sigma, NumFiltre, h, L, tab_P, tab_Nopt, ...
   Nlength, RbComplet)
%::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
%function [logRDVG] = Estimation_bruit_colore ...
%   (y, Fc, Fe, Fcalibrage, Sigma, NumFiltre, h, L, tab_P, tab_Nopt, ...
%   Nlength, RbComplet)
%::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
%
% Estimation du Rapport de Vraisemblance Generalise
% sous l'hypothese y(n) = c0 * cos(2 pi Fc n) + s0 * sin(2 pi Fc n) + b(n)
% en tenant compte de la correlation du bruit
%
%::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
```

```
% y         : signal temporel
% Fc        : frequence centrale du Filtre
% Sigma     : puissance du bruit
% NumFiltre : Numero du filtre
% h         : fonction de transfert du filtre
% L         : nombre de points frequentiels
% tab_P     : P = Ra * SigmaAlphaBeta
% tab_Nopt  : N = Rb * SigmaAlphaBeta
% Nlength   : longueur des tableau tab_P, tab_Nopt
%::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::


WindowLength = length(y);

% recherche du module maximum dans la bande et de sa frequence associee
[Yk, Fk] = ModuleMaxBande(y, NumFiltre, h, WindowLength, L, Fcalibrage/Fe);

%% calcul de la variance SigmaAlphaBeta^2 = sqrt(SigmaAlpha.SigmaBeta)
%D  = matrice_D (Fk, WindowLength);
%C  = D' * RbComplet * D;
%Sigma2carre = sqrt(C(1,1)*C(2,2));
%Sigma2 = sqrt(Sigma2carre);
Sigma2 = Sigma;

% calcul de RDVG
p = Yk/Sigma2;
if (p < 1.41)
   logRDVG = 0;
else if (p<3.)
      n = ValeurApprocheeNopt(p, tab_P, tab_Nopt, Nlength);
      logRDVG = log(besseli(0,p*n)) - n^2/2.;
   else
      n = p;
      logRDVG = n^2/2 - 0.5 * log(2*pi*n^2);
   end
end
```

B.1.5 EstimationsCompareesEtClimbers.m

```
function [OutFilterMultiOctave, LogEstimeNonCorreleMultiOctave, ...
   LogEstimeCorreleMultiOctave, Freq] = EstimationsCompareesEtClimbers...
   (TuneLength, e, NOctaves, Fe, Sigma, fcBanc, NCoefPasseBas, ...
   TailleFenetre, Nc, T, alpha, Niter, NArret, PourcArret)
%
%function [OutFilterMultiOctave, LogEstimeNonCorreleMultiOctave, ...
%   LogEstimeCorreleMultiOctave, Freq] = EstimationsCompareesEtClimbers...
%   (TuneLength, e, NOctaves, Fe, Sigma, fcBanc, NCoefPasseBas, ...
%   TailleFenetre, Nc, T, alpha, Niter, NArret, PourcArret)
%::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
%
% Calcul du rapport de vraisemblance par les deux methodes
% et recherche des cretes par l'algorithme des climbers
%
%::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
% parametres en sortie :
%   OutFilterMultiOctave           : sortie du banc de filtres
%   LogEstimeNonCorreleMultiOctave : log du RDVG par la methode temporelle
%                                    (bruit blanc)
%   LogEstimeCorreleMultiOctave    : log du RVDG par la methode frequentielle
%   Freq                           : liste des frequences centrales des filtres
% parametres d'entree :
%   TuneLength     : longueur du signale e
%   e              : le signal a l'entree du banc
%   NOctaves       : nombre d'octaves d'analyse
%   Fe             : frequence d'echantillonnage
```

```
%   Sigma          : puissance du bruit
%   fcBanc         : frequence de calibrage du banc
%   NCoefPasseBas  : nombre de coefficients du filtre passe-bas
%   TailleFenetre  : taille de la fenetre d'analyse
%   Nc             : nombre de Climbers
%   T              : temperature
%   alpha          : taux de decroissance
%   Niter          : nombre d'iteration avant test 1
%   NArret         : nombre d'iteration maximal
%   PourcArret     : pourcentage de Climbers immobiles
%::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::

L = 2048; % nmbre points frequentiels

F1 = fcBanc/Fe/2^(NOctaves-1);  % frequence centrale du 1er filtre

%:::::: courbe de Nopt en fonction de p ::::::

tab_P = [];
tab_Nopt = [];

% 1ere partie de la courbe, de p=1.4 a p=1.5
load CourbeNoptimal_zone2a P Nopt
tab_P = P;
tab_Nopt = Nopt;

% 2eme partie de la courbe, de p=1.5 a p=3.
load  CourbeNoptimal_zone2b P Nopt
tab_P = [tab_P, P(2:length(P))];
tab_Nopt = [tab_Nopt, Nopt(2:length(Nopt))];

Nlength = length(tab_Nopt);


%:::::: lecture des coefficients des filtres du banc ::::::
[CoefficientsBancFiltres, Retard] = LectureBancFiltres;

%:::: affichage des filtres du banc et de la TF du signal :::
figure(10);
plot((1:L)/L, 20*log10(abs(fft(e,L))));
hold on

H = [];
for k=1:24
   End = CoefficientsBancFiltres(k,1)+1;
   [h,f] = freqz(CoefficientsBancFiltres(k,2:End),1,L,1);
        % h : reponse frequentielle du filtre  CoeffsFilterBank(f,:)
        % f : frequences
   H = [H, abs(h)];
end
for l=1:NOctaves
  plot(f/2^(l-1),H);
end
hold off

%:::::: fenetres avec recouvrement ::::::

Pas           = floor(TailleFenetre);
NbreFenetres  = floor((TuneLength - TailleFenetre) / Pas) + 1; %40;
% nbre fenetres = cte pour tout octave
IndicesFenetresCor = 0.5:NbreFenetres+0.5;
IndicesFenetres = 1:NbreFenetres;

OutFilterMultiOctave = [];
LogEstimeNonCorreleMultiOctave  = [];
```

```
LogEstimeCorreleMultiOctave     = [];
Freq = [];
Taille=[];
TailleOutFilter=[];
DensiteNonCorreleMultiOctave = [];
DensiteCorreleMultiOctave = [];


%::::::: comparaison des estimations :::::::

for l=1:NOctaves

   % passage du signal dans le banc de filtres
   [OutFilterBank, TL2] = PassageBancFiltres...
      (e, TuneLength, CoefficientsBancFiltres, Retard);
   figure(10)
   hold on

   Pas = floor((TL2-TailleFenetre)/(NbreFenetres-1))
   TailleOutFilter = [TailleOutFilter, TL2];
   fprintf('\nTraitement de l''octave %d  (%d points, %d fenetres)\n   ',...
      l, TuneLength, NbreFenetres);

   % tableau contenant l'estimation par le max de vraisemblance
   LogEstimeNc = zeros(24,NbreFenetres);
   LogEstime   = zeros(24,NbreFenetres);
   Freq_octave_n = [];
   DensitePondereeOneOctaveNonCorrele = [];
   DensitePondereeOneOctaveCorrele = [];

   % pour chacun des 24 filtres du banc...
   for k=1:24 %24

      fprintf(' %2d',k);
      Fc = fcBanc / Fe * 2^((k-1)/24);
      F0 = Fc/2^(l-1);
      Freq_octave_n = [Freq_octave_n  F0];
      End = CoefficientsBancFiltres(k,1)+1;
      h = CoefficientsBancFiltres(k,2:End);
      RbComplet = [];
      % sur chacun des fenetres, on estime le max de vraisemblance
      for j=1:NbreFenetres
         LogEstimeNc(k,j) = EstimationBruitBlanc...
            (OutFilterBank(k,Pas*(j-1)+1:Pas*(j-1)+TailleFenetre),...
            Fc, Fe, Sigma);
         LogEstime(k,j)   = EstimationBruitColore...
            (OutFilterBank(k,Pas*(j-1)+1:Pas*(j-1)+TailleFenetre),...
            Fc, Fe, fcBanc,...
            Sigma, k, h, L, tab_P, tab_Nopt, Nlength, RbComplet);
      end

   end

  % sauvegardes
   fprintf('\n   Sauvegarde sorties de banc de filtre et RDVG\n');
   OutFilterMultiOctave = [OutFilterMultiOctave, OutFilterBank];
   LogEstimeNonCorreleMultiOctave = ...
      [LogEstimeNonCorreleMultiOctave, LogEstime];
   LogEstimeCorreleMultiOctave = ...
      [LogEstimeCorreleMultiOctave, LogEstime];
   Taille = [Taille NbreFenetres];
   Freq = [Freq_octave_n  Freq];

   % lancement des Climbers
   fprintf('   Climbers (non correle)\n');
```

```
   [DensiteOneOctaveNonCorrele, DensitePondereeOneOctaveNonCorrele] = ...
      CalcClimbersDensites( LogEstimeNcOctave_n, Nc, T, alpha,...
      Niter, NArret, PourcArret);
   fprintf('   Climbers (correle)\n');
   [DensiteOneOctaveCorrele, DensitePondereeOneOctaveCorrele] = ...
      CalcClimbersDensites( LogEstimeOctave_n, Nc, T, alpha, ...
      Niter, NArret, PourcArret);

   % sauvegarde Multi Octave
   DensiteNonCorreleMultiOctave = ...
      [DensitePondereeOneOctaveNonCorrele ;
      DensiteNonCorreleMultiOctave];
   DensiteCorreleMultiOctave = ...
      [DensitePondereeOneOctaveCorrele ;
      DensiteCorreleMultiOctave];

   % filtrage passebas du signal d'entree
   %decimation du signal d'entree
   fprintf('   Sauvegardes, decimation...\n');
   e = decimate (e,2, NCoefPasseBas, 'FIR');

   % calcul des nouveau parametres relatifs aux fenetres
   TuneLength = TuneLength/2;
%   NbreFenetres  = floor ((TuneLength - TailleFenetre) / Pas + 1);
end


figure(1)
imagesc(OutFilterMultiOctave);

GraphiqueTempsFrequence2(LogEstimeNonCorreleMultiOctave, Taille, NOctaves, ...
   IndicesFenetres, Freq, 2);

GraphiqueTempsFrequence2(LogEstimeCorreleMultiOctave, Taille, NOctaves, ...
   IndicesFenetres, Freq, 3);
```

B.1.6 GenerateFrequencies.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%
%  Generation des frequences d'excitation d'un signal d'analyse
%
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


function [freq] = GenerateFrequencies...
   (NFilter, DecalageF, NOctaves, Indic, FcBanc, Fe)

%::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
% NFilter   : numeros des filtres excites [n1 n2 n3 n4 ...]
% DecalageF :  decalage de la frequence par rapport a la frequence
%               centrale de chaque filtre   [dec1 dec 2 dec3 dec4 ...]
% NOctaves  : nombre d'octaves sur lesquels on excite les filtres
% Indic     : indicateur d'excitation ou non de chaque filtre a chaque
%                octave  [[1  0  0  1...] [...] [...] [...] ...]
%                          ^  ^  ^  ^
%                filtre   n1 n2 n3 n4
%                          +-----------+   +-+   +-+   +-+
%               num octave     1e           2e    3e    4e
%  Fe :  frequence d'echantillonnage
%::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
```

```
NbFiltresExcites = length(NFilter);
f=zeros(NbFiltresExcites);
freq = [];

% frequences centrales des filtres excites
DemiQuartTon = 2^(1/48);

for i=1:NbFiltresExcites
  f(i) = FcBanc / Fe * 2.^((NFilter(i)-1)/24);
  DeltaF = f(i) * DemiQuartTon - f(i);
  f(i) = f(i) + DecalageF(i)*DeltaF;
end


% Generation des partiels sur N octaves

for no=1:NOctaves
  for nf=1:NbFiltresExcites
     if (Indic(no,nf)==1)
        freq = [freq f(nf)];
     end
  end
  f(:) = f(:)/2.;
end
```

B.1.7 GenerateTune.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%
%  Generation du signal d'analyse
%
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


function [e, Sigma] = GenerateTune (TuneLength, Freq, RSB)

%::::::::::::::::::::::::::::::::::::::::::::::::::::
% TuneLength : longueur du signal d'analyse a creer
% Freq : liste des frequences presentes
% RSB : rapport signal/bruit
%::::::::::::::::::::::::::::::::::::::::::::::::::::


Amplitude = 1.;

% Generation des partiels sur N octaves
DeuxPi = 2.*pi;
e(1:TuneLength) = zeros(size(TuneLength));
NFreq = length(Freq);

for nf=1:NFreq
   e(1:TuneLength) = e(1:TuneLength) + Amplitude * ...
      cos(DeuxPi*(0:TuneLength-1)*Freq(nf)) ;
end


%:::::::::: AJOUT DE BRUIT ::::::::::::::::::::::
% perturbation aleatoire, loi Normale Mu=0., Sigma=1.

for i=1:TuneLength
  bruit(i) = Amplitude * randn(1.);
end
```

```
% normalisation de 'bruit' telle que sa puissance Sigma^2 = 1
PuissanceBruit = 1./TuneLength * bruit*bruit';
bruit = 1./PuissanceBruit * bruit;

% Calcul du gain Sigma de facon a ce que
% 10 log10 (Psignal / Sigma) = RSB fixe par l'utilisateur
PuissanceSignal = 1./TuneLength * e*e';
Sigma = PuissanceSignal * 10^(-RSB/10.);

% correction du bruit par le facteur 'Sigma' et ajout du bruit au signal
e = e + Sigma*bruit;
```

B.1.8 GraphiqueTempsFrequence2.m

```
function GraphiqueTempsFrequence2(Tableau, Taille, NOctaves, Temps, ...
Frequences, NumFigure)
%
% function GraphiqueTempsFrequence2(Tableau, Taille, NOctaves, Temps, ...
%   Frequences, NumFigure)
%:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
%
% Affichage d'un domaine Temps-Frequence, stocke dans un tableau
% octave par octave de la maniere suivante :
%
%  +------------------+-------------------+
%  | 1ere octave      | 2eme octave       | 24 filtres
%  | N1 elts          | N2<N1 elts        |
%  +------------------+-------------------+
%
%::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
% Tableau(nl,nc) : valeurs a afficher, avec
%    nl = 24*NOctaves lignes (ie bandes frequentielles
%    nc colonnes (intervalles temporels)
% Taille(NOctaves) : nb intervalles temporels a chaque octave
% NOctaves : nombre d'octaves
% Temps : abcisses
% Frequences : ordonnees
% NumFigure : numero de la figure pour affichage
%::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::

OutTableau = [];
TailleMax = Taille(1:1);
OutTableau = [Tableau(1:24,1:TailleMax); OutTableau];
NCoefResample = 20; % par defaut, 10
indiceDepart = TailleMax+1;

for l=2:NOctaves
   TailleCourant = Taille(l:l);
   OutTableauOctave_n = zeros(24,TailleMax);
   for k=1:24
      OutTableauOctave_n(k:k,1:TailleMax) = ...
        resample( Tableau(k:k,indiceDepart:indiceDepart+TailleCourant-1), ...
        TailleMax, TailleCourant, NCoefResample);
   end
   indiceDepart = indiceDepart + TailleCourant;
   OutTableau = [OutTableauOctave_n ; OutTableau];
end

figure(NumFigure);
imagesc(Temps, Frequences, OutTableau);
colormap(flipud(gray));
colorbar;
set(gca, 'YDir', 'normal');
```

B.1.9 LectureBancFiltres.m

```
function [CoefficientsBancFiltres, Retard] = LectureBancFiltres
%:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
% function [CoefficientsBancFiltres, Retard] = LectureBancFiltres
%:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
% Lecture des coefficients des RIF du banc de filtres
% et calcul des retards a introduire afin de synchroniser
% les sorties
%
%:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
% CoeffsBancFiltres : coefficients du banc de filtres
% Retard :
%:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::



load CoefficientsBancFiltres CoefficientsBancFiltres;
Retard = zeros(1,24);
Longueur = zeros(1,24);

% pour chacun des 24 filtres du banc...
for k=1:24
   Longueur(k) = CoefficientsBancFiltres(k,1)+1;
end

Maxi = Longueur(1);
% retard de chaque filtre = 1/2 * nbre de coefficients du filtre
for k=1:24
   Retard(k) = floor((Maxi - Longueur(k))/2); %- (Maxi-Mini)/2);
end
```

B.1.10 ModuleMaxBande.m

```
function [Yk, Fk] = module_max_bande(x, NumFiltre, h, N, L, Fcalibrage)

% filtrage
y = filter(h,1,x);

% TFD de la matrice contenant les realisations de y
Y = abs(fft(y',L));
A = (1:L) / L;  % abcisse pour les graphiques, en frequence reduite

HuitiemeTon = 2^(1/48);

% frequences min et max du  filtre
fc = Fcalibrage * 2^((NumFiltre-1)/24); % frequence centrale de ce filtre
Fmin   = fc / HuitiemeTon;
Fmax   = fc * HuitiemeTon;

% indices des "bin" de frequences comprises dans le filtre
Imin = floor(Fmin*L)+1;
Imax = ceil(Fmax*L)+1;

% recherche du bin de plus grand module de chaque ligne
[m,Ind] = max( Y(Imin:Imax) );
abc = Imin + 1 * [0:(Imax-Imin)];

% v------
%figure(3)
%plot(abc/L, 20*log10(Y(Imin:Imax)),'r.');
%hold on
%plot(A, 20*log(abs(fft(h,L))));
%%plot(A, 20*log(abs(fft(x,L))), 'y');
% ^------
```

```
if(Ind == 1)
   Yresu = Y(Imin+Ind-1,:);
   Fresu = Imin;
else
   if (Ind == Imax-Imin+1)
      Yresu = Y(Imin+Ind-1,:);
      Fresu = Imax;
   else
      ord = Y(Imin:Imax);
      facteur = 30;
      abc2 = Imin + (Imax-Imin)/facteur * [0:facteur];
      ord2 = spline(abc, ord, abc2);
% v------
%%      plot(abc2 / L, 20*log10(ord2),'r');
% ^------
      [Yresu,Fresu] = max(ord2);
      Yresu = abs(Yresu);
      Fresu = Imin + (Imax-Imin)/facteur*(Fresu-1);
   end
end

% v------
%plot(A, 20*log10(Y),'r');
%%plot(Fresu*ones(21)/L, log10(Yresu)*(0:20), 'g');

%%legend('fonction de transfert du filtre', 'signal emis', ...
%%'approximation par les splines', 'bin de frequences dans Fk', ...
%%'maximum detecte');
%legend('bin de frequences dans Fk', 'fonction de transfert du filtre');
hold off
% ^------

Yk = Yresu/sqrt(N);
Fk = (Fresu-1)/L;
```

B.1.11 MoveClimbers.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%  Deplacement des climbers dans le plan temps-frequence.
%
%:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
% Paramemetres d'entree :
%   M      Description sous forme matricielle du plan tps-freq
%   Nc      Nbre de climbers
%   T Facteur de decroissance
%   alpha   Taux de decroissance
%   Niter     Nbre d'iterations, 1er critere d'arret
%   NArret Nbre d'iterations prises en compte lors du 2eme critere d'arret
%   PourcArret Pourcentage (compris entre 0 et 1) de climbers qui doivent rester
% immobiles pour satisfaire le 2eme critere d'arret
%
% Parametres de sortie :
%   ct vecteur des abcisses (ie temps) des Nc climbers
%   cf vecteur des ordonnees (ie frequence) des Nc climbers
%   sfm  somme des facteurs de mouvement vertical
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


function [ct, cf] = MoveClimbers( M, Nc, T, alpha, Niter, NArret, PourcArret)

[ Nf, Nt] = size(M);
```

```
% initialisation des coordonnees des Nc climbers

ct = floor(rand(1,Nc)*Nt)+1;
cf = floor(rand(1,Nc)*Nf)+1;
TpsFreq = zeros(2,Nc);

dM = zeros(1,Nc); % vecteur de variation du module des OutFilter

sfm = zeros(1,Niter); % somme des facteurs de mouvement vertical,
% 2eme critere d'arret (avec Niter)

% boucle iterative

for n=1:Niter

   T = alpha*T;
   dct = sign(rand(1,Nc)-0.5); % deplacement elementaire selon le temps = +1 ou -1
   ct = ct + dct;
   ct = ct+0.5*(sign(Nt-ct+0.5)-sign(ct-0.5));

   dcf = sign(rand(1,Nc)-0.5);
   dcf = dcf + 0.5*(sign(Nf-(cf+dcf)+0.5)-sign((cf+dcf)-0.5));
       % condition aux limites du plan
   newcf = cf + dcf;

   %dM = M(newcf(1:Nc),ct(1:Nc)) - M(cf(1:Nc),ct(1:Nc));
% BAD car = matrice Nc*Nc on ne voudrait que sa diagonale
   for k=1:Nc
      dM(k) = M(newcf(k),ct(k)) - M(cf(k),ct(k));% eviter la boucle:calcul matriciel
   end

   Pm = exp((dM.*(1-sign(dM)))/(2*T)); % proba de mvt selon l'axe des frequences
   fm = (sign(Pm-rand(1,Nc))+1)/2; % facteur de mouvement = 1 si mvt
%         = 0 sinon
   sfm(n) = sum(fm);
   dcf = dcf.*fm;
   cf = cf + dcf;
   SfmSeuil = zeros(1,NArret);

   if n>NArret

      SfmSeuil = sign(Nc*PourcArret-sfm(n-NArret:n))-1;
% critere d'arret: si PourcArret des climbers
       % sont immobiles pdt NArret iterations
      if all(SfmSeuil)
         fprintf('\n\n --> Arret :\n -->        %f %% des %d climbers',...
            ' sont immobiles depuis %d iterations\n\n', 100*PourcArret, Nc, n);
         break
      end
   end

end
```

B.1.12 PassageBancFiltres.m

```
function [OutFilterBank, TuneLength2] = PassageBancFiltres...
   (e, TuneLength, CoeffsFilterBank, Retard)
%::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
% function [OutFilterBank, TuneLength2] = PassageBancFiltres...
%    (e, TuneLength, CoeffsFilterBank, Retard)
%::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
%
%  Applique le banc de filtre dont les
%  coeffs ont ete calcules dans CalculCoefficientsBancFiltres.
%
```

```
%::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
%  e             : le signal a l'entree du banc
% OutFilterBank : les sorties temporelles de chaque filtre
%           du banc rangees dans une matrice 23*(Length de e)
%::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
% C. Duval (fevrier 2000)
% V. Verfaille (juillet 2000) => prise en compte du retard !
%::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::

Maxi = Retard(24);
TuneLength2 = TuneLength - Maxi;
OutFilterBank = zeros( 24, TuneLength2);

for k=1:24

   End = CoeffsFilterBank(k,1)+1;
   b = CoeffsFilterBank(k,2:End);
   s = filter(b,1,e);
   OutFilterBank(k,:) = s(Maxi-Retard(k)+1:TuneLength-Retard(k));

 end
```

B.1.13 ValeurApprocheeNopt.m

```
function [n] = Valeur_Approchee_Nopt(p, tab_P, tab_Nopt, Nlength)
%
% calcul de la valeur Nopt en fonction de p dans la zone 2
% (forte variation en debut de zone)
% par interpolation dans une table



ind_p = 1;
i_min = 1;

% recherche de l'indice du dernier element de tab_P inferieur a p
while tab_P(ind_p) < p
   i_min = ind_p;
   ind_p = ind_p + 1;
end
i_max = ind_p +1;

if(i_max > Nlength)
   n = tab_Nopt(Nlength:Nlength);
else
   n = tab_Nopt(i_min) + (p-tab_P(i_min)) * (tab_Nopt(i_max) - tab_Nopt(i_min))...
          / (tab_P(i_max) - tab_P(i_min));
end

%figure(1);
%plot([tab_P(i_min), p, tab_P(i_max)], [tab_Nopt(i_min), n, tab_Nopt(i_max)]);
```

B.1.14 diagonale_S.m

```
function [S] = diagonale_S (Ry,N)

S = fft(Ry,N);
```

B.1.15 matrice_D.m

```
function [D] = matrice_D (f0,N)

% matrice D d'analyse a la frequence f0,
```

```
% contenant 2 colonnes : [  :    : ]
%                       [ cos  sin]
%                       [  :    : ]


arg = - 2. * pi * f0 * (0:N-1);
D = 1./sqrt(N) * [cos(arg)' sin(arg)'];
```

B.1.16 matrice_Fourier.m

```
function [F] = matrice_Fourier (N)

% calcul de la matrice de Fourier definie par
%    1/sqrt(N) * exp (2 j pi k l)
%    k,l = 0, ..., N-1


F = zeros(N,N);
UnSurN = 1./ sqrt(N);
DeuxJPiSurN = 2.*pi*j/N;
for k=1:N
   for l=1:N
      F(k,l) = UnSurN * exp (DeuxJPiSurN*(k-1)*(l-1));
   end
end
```

B.1.17 mycolmap.m

```
function M = mycolmap()
% table des couleurs (color map) personnalisee : noir -> blanc -> noir

col = (1:127)/127;
A = [col, 1, fliplr(col)];
M = [A', A', A'];
```

B.1.18 tests.m

```
%::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
%
%                 PROGRAMME PRINCIPAL
%
%::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
%
% APPEL DES ROUTINES POUR :
%  - LE CALCUL DES COEFFICIENTS DU BANC DE FILTRES
%  - LA GENERATION DU SIGNAL SYNTHETIQUE
%  - LE CALCUL DU RDVG PAR LES DEUX METHODES
%  - L'UTILISATION DE L'ALGORITHME DES CLIMBERS
%::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
% V. Verfaille (juillet 2000)
%::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::


%TuneLength = 8192;   % cas du son de piano

%---------- Coefficients des filtres du banc -----------------
%TestCalcCoeffsFilterBank
Fe = 44100.;
FcBanc = 0.2331*Fe;         % frequence centrale         (<=0.2331)
precBP = 0.0125;    % precision bande passante   (0.001 -> 0.0125)
precBC = 0.01;    % precision bande de coupure (0.01)
FactDeltaF = 1.15;  % + 10%                    (1.  -> 1.15)
NCoefPasseBas = 100;
```

```
% calcul des coefficients des filtres du banc de filtres
%CalculCoefficientsBancFiltres (FcBanc, Fe, precBP, precBC, FactDeltaF);


%--------- excitations des filtres ----------------------------
NOctaves = 4;

% Numéros des filtres excitéses
NFilter = [2 6 10 14 18 22];

% decalage par rapport a la fréquence centrale, en %
DecalageF = [0.  0.  0.  0.  0.  0.];
%DecalageF = [0.  0.1  0.2  0.3  0.4  0.5];

% indicateurs d'excitation aux differents octaves
Indic =[[1 1 1 1 1 1]; [1 1 1 1 1 1]; [1 1 1 1 1 1]; [1 1 1 1 1 1]; [1 1 1 1 1 1];...
    [1 1 1 1 1 1]; [1 1 1 1 1 1]];

% rapport signal / bruit (en dB)
RSB = 5.;


%---------- CLIMBERS ----------------------------------------------
Nclimbers = 1;  % Nombre total de climbers
T         = 5.;    % facteur de decroissance
alpha     = 0.9;   % taux de decroissance
Niter     = 250;   % nbre d'iteration total
NiterStop = 40;    % nbre d'iteration successives avec immobilisation (2e critere)
PourcCl   = 0.95;  % pourcentage de climbers immobiles

%=== Generation d'un signal de synthese ===
%Freq = GenerateFrequencies (NFilter, DecalageF, NOctaves, Indic, FcBanc, Fe);
%[e, Sigma] = GenerateTune (TuneLength, Freq, RSB);


%----------        ----------------------------------------------

%TuneLength = 2048;
%Freq = GenerateFrequencies (NFilter, DecalageF, NOctaves, Indic, FcBanc, Fe);
%[e1, Sigma] = GenerateTune (1000, Freq, RSB);
%Freq = GenerateFrequencies (NFilter+1, DecalageF, NOctaves, Indic, FcBanc, Fe);
%[e2, Sigma] = GenerateTune (500, Freq, RSB);
%Freq = GenerateFrequencies (NFilter, DecalageF, NOctaves, Indic, FcBanc, Fe);
%[e3, Sigma] = GenerateTune (500, Freq, RSB);
%Freq = GenerateFrequencies (NFilter-1, DecalageF, NOctaves, Indic, FcBanc, Fe);
%[e4, Sigma] = GenerateTune (2096, Freq, RSB);
%e = [e1 e2 e3 e4];

%=== son de piano ===tests
load piano2_17.dat -ascii;
e = piano2_17(1:8192)';
size(e)
%e = decimate(e, 4, NCoefPasseBas, 'FIR');
%e = [e e];
TuneLength = length(e);
%
PuissanceSignal = 1./TuneLength * e*e';
Sigma = PuissanceSignal * 10^(-RSB/10.)


WindowLength = 512;

Mini =  WindowLength * 2^(NOctaves-1);
if (TuneLength < Mini)
```

```
   fprintf('\nERREUR : le signal comporte %d points, \n   alors qu''un minimum,...
      ' de %d points\n   est requis pour une analyse sur %d octaves !\n\n',...
      TuneLength, Mini, NOctaves);
else
   % ---------- application simple du banc de filtre ----------
%     TestApplyBankFilter2(e, NOctaves, NCoefPasseBas, FcBanc, Fe);
%     TestApplyBankFilterRms(e, NOctaves, WindowLength, NCoefPasseBas, FcBanc, Fe);
%     Affiche_Banc_Filtres(TuneLength, e, NOctaves, Fe, Sigma)

   % ---------- climbers seuls ----------
%     Affiche_Climbers(TuneLength, e, NOctaves, Fe, Sigma, Nclimbers, T,...
%        alpha, Niter, NiterStop, PourcCl)


   % ---------- estimations + climbers ----------
   [OutFilterMultiOctave, LogEstimeNonCorreleMultiOctave, ...
      LogEstimeCorreleMultiOctave, Freq] = EstimationsCompareesEtClimbers...
      (TuneLength, e, NOctaves, Fe, Sigma, FcBanc, NCoefPasseBas, ...
      WindowLength, Nclimbers, T, alpha, Niter, NiterStop, PourcCl);
end
```

B.1.19 vecteur_toeplitz_Rb.m

```
function [Rb] = vecteur_toeplitz_Rb (f0, sigma2, N)

%:::: calcul des Rb(k) par la formule algebrique :::::::
% Rb(k) = 2 * sigma * sigma / pi / k * sin (2 pi Df k) * cos (2 pi f0 k)
%   f0 : frequence centrale du filtre
% sigma2 : variance du bruit
%   N  : nombre de points de la fenetre d'analyse


% initialisation des constantes

DemiQuartTon = 2^(1/48);
DeltaF = f0 * (DemiQuartTon - 1);
Rb = zeros(1,N);
DeuxPiDeltaF = 2.*pi*DeltaF;
DeuxPiF0 = 2.*pi*f0;

% calcul des Ry(k)

coef  = 2. * sigma2 / pi;
Rb(1) = 4. * sigma2 * DeltaF;

for k=1:N-1
   Rb(k+1) = coef / k * sin(DeuxPiDeltaF*k) * cos(DeuxPiF0*k);
end
```

B.2 Routines d'affichage

B.2.1 Affiche_2e_condition.m

```
% respect de la 2e condition de non repliement du spectre
%
%  Nc = 4 / (Fe/2 - (2^(49/48) - 1) 2^(23/24)) f_1^c
%
%  avec Nc le nombre de coefficients du filtre passe-bas
%      et f_1^c la frequence centrale du 1er filtre
%      (= frequence de calibrage du banc de filtres)

Freq = [];
Nc = [];
```

```
coef = (2^(49/48) - 1) * 2^(23/24);

for f = 0.2:0.001:0.248
   Freq = [Freq, f];
   Nc = [Nc, 4./(0.5-coef*f)];
end

plot(Freq, Nc);
xlabel('f_1^c : frequence de calibrage');
ylabel(' n_c : nombre de coefficients du filtre');
grid
```

B.2.2 Affiche_Approx_Gamma.m

```
N = [];
G1 = [];
G2 = [];
pas = 0.5;

for n=pas:pas:35
   N = [N, n];
   G1 = [G1, log(besseli(0,n*n)) - n^2/2.];
   G2 = [G2, n^2/2 - 0.5 * log(2*pi*n^2)];
end

close all
figure(1)
x = 1:length(G1);
plot(pas*x,G1);
hold on
plot(N, G2, 'r.');
hold off
legend('calcul direct', 'calcul asymptotique');
xlabel('R_a = \rho / \sigma_{\alpha \beta}');
ylabel('ln (\Gamma)');
```

B.2.3 Affiche_Banc_Filtres.m

```
function [] = Affiche_Banc_Filtres(TuneLength, eInitial, NOctaves, Fe, Sigma)

e = eInitial;
Length = TuneLength;

L = 1024; % nmbre points frequentiels

% correction eventuelle de la taille de fenetre
% pour respecter la condition  Fc * Taille >> 1

indic_print='non';
Fc = 1./4. /2^(NOctaves-1);  % frequence centrale du 1er filtre


%:::::: lecture des coefficients des filtres du banc ::::::

[CoeffsFilterBank, Retard] = BancFiltres;


%:::::: fenetres avec recouvrement ::::::

OutFilterMultiOctave = [];
Freq = [];
```

```
%::::::: comparaison des estimations ::::::::

for l=1:NOctaves

   % passage du signal dans le banc de filtres
   OutFilterBank = ApplyBankFilter(e, TuneLength, CoeffsFilterBank, Retard);

   Freq_octave_n = [];

   % pour chacun des 24 filtres du banc...
   for k=1:24
      Fc = 1./4. * 2^((k-1)/24);
      F0 = Fc/2^(l-1);
      Freq_octave_n = [Freq_octave_n  F0];
      End = CoeffsFilterBank(k,1)+1;
      h = CoeffsFilterBank(k,2:End);

      % interpolation des valeurs estimees afin de creer l'image
      OutFilterOctave_n(k,:)  = interp( OutFilterBank(k,:), 2^(l-1));%
   end

   OutFilterMultiOctave           = [ OutFilterOctave_n ; OutFilterMultiOctave];
   Freq                           = [Freq_octave_n  Freq];

   % decimation du signal d'entree et calcul des nouveau parametres relatifs aux fenetres
   e = decimate (e,2);
   TuneLength = TuneLength/2;
end



figure(1);
imagesc(1/Fe*(1:Length), Freq, OutFilterMultiOctave);
%title('Sortie du banc de filtres');
xlabel('temps (s)');
ylabel('frequence reduite');
colormap(mycolmap);
colorbar;
set(gca, 'YDir', 'normal'); %, 'YScale', 'log', 'YTick', [0. 0.125 0.25 0.5 1]);
```

B.2.4 Affiche_CalcCoefficientsBancFiltres.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%  Test du calcul des coeffs des filtres du banc par
% le trace des fonctions de transfert H
%  ==> pour affichage
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Fcal = 0.2331

load CalcCoeffsFilterBank CoeffsFilterBank

H = [];

N=1024; % nbre de points de la reponse en frequence

Num = 1;

for k=Num:Num

   End = CoeffsFilterBank(k,1)+1;
   [h,f] = freqz(CoeffsFilterBank(k,2:End),1,N,1);
   % h : reponse frequentielle du filtre  CoeffsFilterBank(f,:)
   % f : frequences
```

```
   H = [H, 20*log10(abs(h))];
end

DemiQuartTon = 2^(1/48);
fc = Fcal*2^(Num-1);
FInf = fc/DemiQuartTon;
FSup = fc*DemiQuartTon;
FactDeltaF = 1.;
DeltaF = FactDeltaF * (fc*DemiQuartTon-fc);

hold off
plot(f, H);
hold on
plot([0., FInf-DeltaF], [-40., -40], 'r-.',[FSup+DeltaF, 0.5], [-40., -40.], 'r-.');
plot([FInf-DeltaF, FInf-DeltaF, FSup+DeltaF, FSup+DeltaF], [-40., 0., 0., -40.], 'r-.');
plot([FInf, FInf], [-70., 0], 'r-.',[FSup, FSup], [-70., 0], 'r-.');
plot([fc, fc], [-60, 0], 'r');
xlabel('frequences reduites');
ylabel('Gain du filtre (dB)');
```

B.2.5 Affiche_Climbers.m

```
function [] = Affiche_Climbers(TuneLength, eInitial, NOctaves, ...
   Fe, Sigma, Nc, T, alpha, Niter, NArret, PourcArret)
%function [] = Affiche_Climbers(TuneLength, eInitial, NOctaves, ...
%   Fe, Sigma, Nc, T, alpha, Niter, NArret, PourcArret)
%::::::::::::::::::::::::::::::::::::::::::::::::::
% Affichage du resultat de l'algorithme des Climbers
%::::::::::::::::::::::::::::::::::::::::::::::::::
% TuneLength : longueur de eInitial
% eInitial   : donnees temporelles
% NOctaves   : nombre d'octaves d'analyse
% Fe         : frequence d'echantillonnage
% Sigma      : puissance du bruit
% Nc         : nombre de Climbers
% T          : temperature
% alpha      : taux de decroissance
% Niter      : nombre d'iteration avant test 1
% NArret     : nombre d'iteration maximal
% PourcArret : pourcentage de Climbers immobiles
%::::::::::::::::::::::::::::::::::::::::::::::::::

e = eInitial;
Length = TuneLength;

L = 1024; % nmbre points frequentiels

% correction eventuelle de la taille de fenetre
% pour respecter la condition  Fc * Taille >> 1

indic_print='non';
Fc = 1./4. /2^(NOctaves-1);  % frequence centrale du 1er filtre


%:::::: lecture des coefficients des filtres du banc ::::::

[CoeffsFilterBank, Retard] = BancFiltres;


%:::::: fenetres avec recouvrement ::::::

OutFilterMultiOctave = [];
Freq = [];

%:::::: comparaison des estimations :::::::
```

```
for l=1:NOctaves

   % passage du signal dans le banc de filtres
   OutFilterBank = ApplyBankFilter(e, TuneLength, CoeffsFilterBank, Retard);

   Freq_octave_n = [];

   % pour chacun des 24 filtres du banc...
   for k=1:24
      Fc = 1./4. * 2^((k-1)/24);
      F0 = Fc/2^(l-1);
      Freq_octave_n = [Freq_octave_n  F0];
      End = CoeffsFilterBank(k,1)+1;
      h = CoeffsFilterBank(k,2:End);

      % interpolation des valeurs estimees afin de creer l'image
      OutFilterOctave_n(k,:)  = interp( OutFilterBank(k,:), 2^(l-1));%
   end

   [Densite, DensitePonderee] = ...
      CalcClimbersDensites( abs(OutFilterOctave_n), Nc, T, alpha, Niter, NArret, PourcArret);

   OutFilterMultiOctave            = [ OutFilterOctave_n ; OutFilterMultiOctave];
   Freq                            = [Freq_octave_n  Freq];

   % decimation du signal d'entree et calcul des nouveau parametres relatifs aux fenetres
   e = decimate (e,2);
   TuneLength = TuneLength/2;
end


figure(1);
imagesc(1/Fe*(1:Length), Freq, OutFilterMultiOctave);
%title('Sortie du banc de filtres');
xlabel('temps (s)');
ylabel('frequence reduite');
colormap(mycolmap);
colorbar;
set(gca, 'YDir', 'normal');

%figure(2);
%imagesc(1/Fe*(1:Length), Freq, Densite);
%%title('Sortie du banc de filtres');
%xlabel('temps (s)');
%ylabel('frequence reduite');
%colormap(mycolmap);
%colorbar;
%set(gca, 'YDir', 'normal');

figure(3);
imagesc(1/Fe*(1:Length), Freq, DensitePonderee);
%title('Sortie du banc de filtres');
xlabel('temps (s)');
ylabel('frequence reduite');
colormap(flipud(gray));
colorbar;
set(gca, 'YDir', 'normal');
```

B.2.6 Affiche_Matrice_Covariance_C.m

```
clear all

% calcul de la matrice de correlation C (2x2)
```

```
sigma2 = 1.;
N = 256; % nbre points temporels
NbPtsFrequentiels = 20;    % nombre de points resolution frequentielle
DeuxPi = 2. * pi;

DeltaF = 2^(1/48);

% frequences min et max du banc de filtre
Fmin   = 1./4. / DeltaF;
Fmax   = 1./4. * DeltaF^47;

% choix d'un indice l tq l/N appartienne a [Fmin ; Fmax]
l=floor(Fmax*NbPtsFrequentiels)-1;
ecart =  DeltaF;

% initialisation graphique
close all;
figure(1);
subplot(4,1,1);
hold on;
subplot(4,1,2);
hold on;
subplot(4,1,3);
hold on;
subplot(4,1,4);
hold on;

SigmaAB_tot = [];
Rho_tot     = [];
Freq_tot    = [];
Dist_tot    = [];
Sigma2_tot  = [];



% Ie filtre du banc de filtres

for i=0:23
   %calcul des bornes du ie filtre
   f0 = 0.25*2^(i/24);
   F1 = f0/DeltaF;
   F2 = f0*DeltaF;

   % calcul de Rb(f0), pour la frequence centrale du filtre
   Rb = vecteur_toeplitz_Rb (f0, sigma2, N);
   RbComplet = toeplitz(Rb);
   SigmaAB = [];
   Rho = [];
   Freq = [];
   Dist = [];
   Sigma2 = [];
   pas = (F2-F1)/NbPtsFrequentiels;

   % calcul de C(f) = D'(f) * Rb(f0) * C(f)
   for f=F1:pas:F2
      D  = matrice_D (f, N);
      C  = D' * RbComplet * D;
      SigmaAB = [SigmaAB,  sqrt(C(1,1)/C(2,2))];
      Rho     = [Rho,    C(1,2)/sqrt(C(1,1)*C(2,2))];
      Sigma2  = [Sigma2,  C(1,1)*C(2,2)];
      d       = 2.*sqrt(C(1,1)*C(2,2) - C(1,2)^2) / (C(1,1) + C(2,2));
      Dist    = [Dist,  d];
      Freq    = [Freq,  f];
   end
```

```
   % sauvegarde incrementale des donnees
   Freq_tot   = [Freq_tot ,  Freq];
   Rho_tot    = [Rho_tot,   Rho];
   Sigma2_tot = [Sigma2_tot ,  Sigma2];
   SigmaAB_tot = [SigmaAB_tot,   SigmaAB];
   Dist_tot   = [Dist_tot ,  Dist];
end


% affichage des courbes de    SigmaAlpha / SigmaBeta,
% et de    rho / SigmaAlpha / SigmaBeta

subplot(4,1,1);
plot(Freq_tot, SigmaAB_tot);
subplot(4,1,2);
plot(Freq_tot, Rho_tot);
subplot(4,1,3);
plot(Freq_tot, Sigma2_tot);
subplot(4,1,4);
plot(Freq_tot, Dist_tot);


figure(1);
% titres des graphiques
subplot(4,1,1);
ylabel('\sigma_{\alpha} / \sigma_{\beta}');
grid;
hold off

subplot(4,1,2);
ylabel('\rho / (\sigma_{\alpha} \sigma_{\beta})');
grid;
hold off

subplot(4,1,3);
ylabel('{\sigma_{\alpha\beta}}^2 = \sigma_{\alpha} . \sigma_{\beta}');
grid;
hold off

subplot(4,1,4);
ylabel('d(C_2,I_2)');
xlabel('frequence reduite');
grid;
hold off



figure(2);
plot(Freq_tot, Sigma2_tot);
ylabel('{\sigma_{\alpha\beta}}^2 = \sigma_{\alpha} . \sigma_{\beta}');
xlabel('frequence reduite');
grid;

figure(3);
plot(Freq_tot, Dist_tot);
ylabel('d(C_2,I_2)');
xlabel('frequence reduite');
grid;
```

B.2.7 Affiche_MaximisationP1.m

```
%:::::: 1ere methode :::::::::::::::::::::
% lecture des donnees dans la table (interpolation)

tab_P = [];
```

```
tab_Nopt = [];

% 1ere partie de la courbe, de p=1.4 a p=1.5
load CourbeNoptimal_zone2a P Nopt
tab_P = P;
tab_Nopt = Nopt;

% 2eme partie de la courbe, de p=1.5 a p=3.
load  CourbeNoptimal_zone2b P Nopt
tab_P = [tab_P, P(2:length(P))];
tab_Nopt = [tab_Nopt, Nopt(2:length(Nopt))];

figure(1);
plot(tab_P, tab_Nopt);
%hold on


%:::: 2eme methode de calcul ::::::::::::::::

u = 0.001:0.1:50.;
Ra = sqrt (u .* besseli(0,u) ./ besseli(1,u));
Rb =  u ./ Ra;

%plot(Ra, Rb);

xlabel('R_a = \epsilon / \sigma_{\alpha \beta}');
ylabel('R_b = m_{opt} / \sigma_{\alpha \beta} ');
title('R_b = f(R_a)');

hold off
```

B.2.8 Affiche_Plages_Frequences.m

```
NOctaves = 10;
Fe = 1;
Fplus = 2^(23/24) * (1 + 2 * (2^(1/48) - 1) );
Fmoins = (1 - 2 * (2^(1/48) - 1) );
F0 = 0.2331;

figure(2)
hold off
figure(3)
hold off

for n=1:NOctaves
   x = [F0*Fe*Fmoins, F0*Fe*Fmoins, F0*Fe*Fplus, F0*Fe*Fplus];
   y = [0.1, 2^(NOctaves-n+1), 2^(NOctaves-n+1), 0.1];
   figure(2);
   plot(x,y);
   hold on
   figure(3)
   loglog(x,y);
   hold on
   F0 = F0/2;
end

figure(2);
hold off
xlabel('frequences')
ylabel('nombre de points temporels')
figure(3);
hold off
xlabel('frequences')
ylabel('nombre de points temporels')
```

B.2.9 Affiche_Rb.m

```
function [ApproxRb] = Approx_Rb (f0, f, sigma2, N);

% approximation de la matrice d'autocorrelation Rb
% par F' (S + JSJ -C0)


% calcul de la diagonale de Ry -> coefficients de correlation
diag_Rb = vecteur_toeplitz_Rb (f0, sigma2, N);

% calcul de la diagonale de S, ie des valeurs du spectre de Ry
diag_S = diagonale_S (diag_Rb, N);

F = matrice_Fourier(N);
C0 = diag_Rb(1,1) * eye(N);
%C0(1,1)
S = diag(diag_S);


%ApproxRy = F' * ( S + J*S*J - C0 ) * F';
% construction de F*F = F'*F' = J

K = fliplr(eye(N));
J = [K(:,N) K(:,1:N-1)];
C1 = S; %  +  J*S*J -  C0;
ApproxRb1 = F' * C1 * F;
C2 = (S +  J*S*J) -  C0;
ApproxRb2 = F' * C2 * F;



%::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
%                          TESTS
%::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::

% comparaison de Rb et de son approximation par Cb + Cb' -C0

colormap('default');
couleur = colormap;
gris = gray;
figure(1);
Rb = toeplitz(diag_Rb);
imagesc(Rb);
drawnow
colormap(mycolmap);
colorbar('vert');
%title('matrice Rb');
figure(2);
imagesc(real(ApproxRb1));
drawnow
colormap(mycolmap);
colorbar('vert');
figure(3);
imagesc(real(ApproxRb2));
drawnow
colormap(mycolmap);
colorbar('vert');
```

B.3 Routines de test

B.3.1 TestApplyBankFilter.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%
%  Test de la fonction PassageBancFiltres
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%   parametres d'entree:
%
% TuneLength  Taille du signal d'entree test (TuneLength>500 pour depasser le
%     transitoire)
% NFilter     Numeros des filtres devant etre excites,
%     avec NFilter0<NFilter1<NFilter2)
%     (ie:harmonique presentes (sur un seul octave) dans le signal
%     d'entree)
%
%
%   Plot des sorties des (NFilter2+2) 1ers filtres du banc
%
%
% warning: Stabilite des filtres
%    w0 = 2*pi*fc : sauver fc depuis CalcCoeffsFilterBank
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function OutFilterMultiOctave = TestApplyBankFilter ( e, NOctaves, ...
   NCoefPasseBas, FcBanc, Fe)


ei = e;
[CoeffsFilterBank, Retard] = BancFiltres;

OutFilterMultiOctave = [];
Freq = [];
TuneLengthMax = length(e);
TuneLength = TuneLengthMax;
Taille = TuneLength;

for n=1:NOctaves

   OutFilter = ApplyBankFilter (e, TuneLength, CoeffsFilterBank, Retard);
   Freq_octave_n = [];

   for k=1:24

      OutFilterOctave_n(k,:) = interp( OutFilter(k,:), 2^(n-1));
      Fc = FcBanc /Fe * 2^((k-1)/24);
      F0 = Fc/2^(n-1);
      Freq_octave_n = [Freq_octave_n  F0];

   end

   OutFilterMultiOctave = [ OutFilterOctave_n ; OutFilterMultiOctave];
   Freq = [Freq_octave_n  Freq];
   e = decimate (e,2, NCoefPasseBas, 'FIR');
   TuneLength = TuneLength / 2;
   Taille = [Taille, TuneLength];

end

figure(1)
Temps = 1/Fe*(1:TuneLength)
imagesc(Temps, Freq, OutFilterMultiOctave);
xlabel('Temps (secondes)');
ylabel('Frequence reduite');
```

B.3.2 TestCalcClimbersDensites.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%  Test de CalcClimbersDensites
%
%
%   Paramemetres d'entree :
%
%   Length duree temporelle du signal d'entree (ie: nb de colonnes de la mat M)
%   Nc      Nbre de climbers
%   T Facteur de decroissance
%   alpha   Taux de decroissance
%   Niter     Nbre d'iterations, 1er critere d'arret
%   NArret Nbre d'iterations prises en compte lors du 2eme critere d'arret
%   PourcArret Pourcentage (compris entre 0 et 1) de climbers qui doivent rester
% immobiles pour satisfaire le 2eme critere d'arret
%   TestNumber Choix de l'une des 4 matrices M test (M = plan tps-freq)
%
%
%
%   figure(1) imagesc(M)
%   figure(2) plot(cf,'o'), position frequentielle des Nc climbers
%   figure(3) imagesc(Densite), densites de climbers dans le plan tps-freq
%   figure(4) imagesc(DensitePonderee), densites ponderee par M de climbers dans
%  le plan tps-freq
%
%
%
% warning: L'axe des ordonnees est inverse
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


function TestClimbersDensites( Length, Nc, T, alpha, Niter, NArret, PourcArret, TestNumber)

M = zeros(20,Length);
if TestNumber==1
   M(10,1:Length/2) = 1:Length/2;
M(10,Length/2+1:Length) = Length-(Length/2+1:Length);
end

if TestNumber==2
   M(10,1:Length/2) = 1:Length/2;
M(15,Length/2+1:Length) = Length/2+1:Length;
end

if TestNumber==3
   M(10,1:Length/2) = 1:Length/2;
   M(15,Length/4:Length) = Length/4:Length;
end

if TestNumber==4
   M(10,1:Length/4) = 1:Length/4;
   M(15,Length/2:Length) = Length/2:Length;
end

figure(1)
imagesc(M)
[Densite,DensitePonderee] = CalcClimbersDensites( M, Nc, T, alpha, Niter, NArret, PourcArret);

figure(2)
imagesc(Densite)

figure(3)
```

```
imagesc(DensitePonderee)
```

B.3.3 TestCalcCoeffsFilterBank.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%  Test du calcul des coeffs des filtres du banc par
% le trace des fonctions de transfert H
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


function TestCalcCoeffsFilterBank()


load CalcCoeffsFilterBank CoeffsFilterBank

H = [];

N=1024; % nbre de points de la reponse en frequence

for k=1:24

   End = CoeffsFilterBank(k,1)+1;
   [h,f] = freqz(CoeffsFilterBank(k,2:End),1,N,1);
        % h : reponse frequentielle du filtre  CoeffsFilterBank(f,:)
        % f : frequences
   H = [H, (abs(h))];
%   plot(f, H);
%   pause
end

plot(f, H);
%plot(f/2, H, f, H);
```

B.3.4 TestEstimation

```
%function  [e] = TestEstimation(NumTest)

NumTest = 1;


TuneLength = 4096;
Fe = 1.;


% Test 1 : un seul filtre d'un seul banc excite
if (NumTest == 1)
NOctaves  = 1;
NFilter   = [1];
DecalageF = [0.];
Indic     = [[1]];
   RSB       = 200.;
   Freq = GenerateFrequencies (NFilter, DecalageF, NOctaves, Indic, Fe);
e(1:TuneLength) = GenerateTune (TuneLength, Freq, RSB);

figure(1);
title('Excitation d''une seule frequence d''un seul octave');
subplot(2,1,1);
   imagesc (TestApplyBankFilter(e, NOctaves) );
   title('Sortie du banc de filtres');
   colorbar

Seuil = 0.;
```

```
   [LogRDVG, X, Y, OutEstime] = EstimationMultiOctave (e, NOctaves, Seuil);
   figure(1);
   subplot(2,1,2);
   imagesc (10000.*LogRDVG);
   title('Estimation du Log du rapport de Vraisemblance');
   colorbar


   % Test 2 : un seul filtre excite sur plusieurs octaves

else if (NumTest == 2)

   NOctaves  = 3;
NFilter   = [10];
DecalageF = [0.];
Indic     = [[1] [1] [1]];
RSB       = 200.;
   Freq = GenerateFrequencies (NFilter, DecalageF, NOctaves, Indic, Fe);
e(1:TuneLength) = GenerateTune (TuneLength, Freq, RSB);

figure(2);
title('Excitation d''une seule frequence sur plusieurs octaves');
subplot(2,1,1);
imagesc (TestApplyBankFilter(e, NOctaves) );

Seuil = 0.;
subplot(2,1,2);
imagesc (EstimationMultiOctave (e, NOctaves, Seuil) );


% Test 3 : excitation a F differente de Fc

else if (NumTest == 3)

NOctaves  = 3;
NFilter   = [1];
DecalageF = [0.6];
Indic     = [[0] [1] [0]];
RSB       = 200.;
   Freq = GenerateFrequencies (NFilter, DecalageF, NOctaves, Indic, Fe);
e(1:TuneLength) = GenerateTune (TuneLength, Freq, RSB);

figure(3);
title('Excitation d''une seule frequence d''un seul octave');
subplot(2,1,1);
imagesc (TestApplyBankFilter(e, NOctaves) );

Seuil = 0.;
subplot(2,1,2);
imagesc (EstimationMultiOctave (e, NOctaves, Seuil) );


% Test 4 : transitions

else if(NumTest == 4)

NOctaves  = 3;
NFilter   = [7];
DecalageF = [0.];
Indic     = [[0] [1] [0]];
RSB       = 200.;

decentrage = floor (TuneLength/9);
e(1:TuneLength/2-decentrage) = GenerateTune...
           (TuneLength/2-decentrage, NFilter, DecalageF, NOctaves, Indic, RSB);
```

```
e(TuneLength/2-decentrage+1:TuneLength/2+decentrage) = ...
           GenerateTune (2*decentrage, NFilter+2, DecalageF, NOctaves, Indic, RSB);
e(TuneLength/2+decentrage+1:TuneLength) = GenerateTune ...
           (TuneLength/2-decentrage, NFilter+4, DecalageF, NOctaves, Indic, RSB);

figure(4);
title('Transitions');
subplot(2,1,1);
imagesc (TestApplyBankFilter(e, NOctaves) );

Seuil = 0.;
subplot(2,1,2);
imagesc (EstimationMultiOctave (e, NOctaves, Seuil) );

   end
  end
 end
end
```

B.3.5 TestGraphiqueTempsFrequence2.m

```
function TestGraphiqueTempsFrequence2

TuneLength = 100;
NOctaves = 3;
Tableau = zeros(24,TuneLength*2);
Tableau(3:3,1:TuneLength)=ones(1,TuneLength);

Taille = zeros(24,1);
Taille(1:1) = TuneLength;
Taille(2:2) = floor(TuneLength/3);
Taille(3:3) = floor(TuneLength/5);

Tableau(13:13,Taille(1:1)+1:Taille(1:1)+Taille(2:2))=ones(1,Taille(2:2));
Tableau(7:7,Taille(1:1)+Taille(2:2)+1:Taille(1:1)+Taille(2:2)+Taille(3:3))=ones(1,Taille(3:3));

Temps = (1:TuneLength)/10/TuneLength;
Frequences = 0.25 +(1:24)/24/4;
NumFigure=1;


figure(2)
imagesc(Tableau)
colorbar;
colormap(flipud(gray));
set(gca, 'YDir', 'normal');


GraphiqueTempsFrequence2(Tableau, Taille, NOctaves, Temps, Frequences, NumFigure)
```

B.3.6 TestValeur_Approchee_Nopt.m

```
tab_P = [];
tab_Nopt = [];

% 1ere partie de la courbe, de p=1.4 a p=1.5
load CourbeNoptimal_zone2a P Nopt
tab_P = P;
tab_Nopt = Nopt;

% 2eme partie de la courbe, de p=1.5 a p=3.
load  CourbeNoptimal_zone2b P Nopt
tab_P = [tab_P, P(2:length(P))];
```

```
tab_Nopt = [tab_Nopt, Nopt(2:length(Nopt))];

figure(1);
hold off;
plot(tab_P, tab_Nopt);

Nlength = length(tab_Nopt);

Ninter = [];
x = 1.4:0.02:3.;
for i=x
   n = Valeur_Approchee_Nopt(i, tab_P, tab_Nopt, Nlength);
   Ninter = [Ninter, n];
end

figure(1);
hold on
plot(x, Ninter, 'r');
```

Bibliographie

[Anderson 1957] T. W. Anderson, "An Introduction to Multivariate Statistical Analysis", *Ed. Wiley*, Stanford 1957.

[Brown 1991] J. C. Brown, "Calculation of a Constant Q Spectral Transform", *Journal of the Acoustic Society of America, vol 89, no. 1, 425-434*, 1991.

[Brown, Zhang 1991] J. C. Brown, B. Zhang, "Musical frequency tracking using methods of conventional and "narrowed" autocorrelation", *Journal of the Acoustic Society of America, vol. 92, no. 3, 1394-1402*, 1991.

[Brown 1992] J. C. Brown, "Musical fundamental frequency tracking using a pattern recognition method", *Journal of the Acoustic Society of America, vol. 92*, 1992.

[Carmona 1999] R. Carmona, W. Hwang, B. Torrésani, "Multi-ridge detection and Time-Frequency reconstruction", *IEEE Transactions on signal processing, vol. 47, no. 2*, 1999.

[Charbit 1996] M. Charbit, "Eléments de théorie du signal : les signaux aléatoires", *Ellipse*, 1996.

[Doval 1994] B. Doval, "Estimation de la fréquence fondamentale des signaux sonores", *Thèse de doctorat, Laforia 94/02*, 1994.

[Dixon 1996] S. Dixon, "Multiphonic Note Identification", *Proceedings of the 19th Australasian Computer Science*, Melbourne, Australia, January 31 - February 2 1996.

[Gradshteyn 1980] I. S. Gradshteyn, I. M. Pyzhik, "Table of Integrals, Series and Products", *Academic Press*, 1980.

[Maher 1990] R. C. Maher, "Evaluation of a method for separating digitized duet signals", *Journal de l'Acoustical Society of America, vol 38, no. 12, 956-979*, 1990.

[Maher 1994] R. C. Maher, "Fundamental frequency estimation of musical signals using a two-way mismatch procedure", *Journal de l'Acoustical Society of America, vol 95, no. 4, 2254-2263*, 1994.

[Mani, Nawab 1995] R. Mani, S. H. Nawab, "Integration of DSP algorithms and musical constraints for the separation of partials in polyphonic music", *Report for the ECE Department, Boston University*, 1995.

[Martin, Kim 1999] K. D. Martin, Y. E. Kim, "Musical instrument identification: a pattern-recognition approach", *136e rencontres de l'Acoustical Society of America*, 1999.

[Piszczalski, Galler 1977] M. Piszczalski, B. F. Galler, "Automatic music transcription", *Computer Music Journal 1*, 1977.

[Piszczalski, Galler 1979] M. Piszczalski, B. F. Galler, "Predicting musical pitch fom component frequency ratios", *Journal of the Acoustic Society of America*, 1979.

[Polyanin 1998] A. Polyanin, A. Manzhiroz, "Handbook of integral equations", *CRC Press*, 1998.

[Rossignol 2000] S. Rossignol, "Segmentation, indexation et manipulation des signaux sonores", *Rapport de Thèse*, 2000.

[Scheirer, Slaney 1997] E. D. Scheirer, M. Slaney, "Construction and evaluation of a robust multifeature speech/music discriminator", *ICASSP Proceedings*, 1997.

[Schrœder 1968] M. R. Schrœder, "Period histogram and product spectrum: new methods for fundamental-frequency measurements", *Journal of the Acoustic Society of America*, 1968.

[Terhardt 1979] E. Terhardt, "Calculating virtual pitch", *Hearing Research 1, 155-182*, 1979.

[Wold, Blum, Keislar 1996] E. Wold, T. Blum, D. Keislar, "Content-based classification, search and retrieval of audio", *IEEE Multimedia vol.3, no. 3, 27-36*, 1996.