| | | | |
|---|---|---|---|
| Grenoble INP | ENSERG<br>Ecole Nationale Supérieure d'Electronique et de Radioélectricité de Grenoble | TELECOM<br>ENSIMAG | MINATEC POLE D'INNOVATION |

PROJET numéro 88 ANNEE UNIVERSITAIRE **2007/2008**

# PROJET DE TROISIEME ANNEE

## RAPPORT FINAL

**JUIN 2008**

TITRE DU SUJET :

**Transformation des sons musicaux avec préservation des articulations musicales**

*Proposé par l'Entreprise :*

*IRCAM*

*Adresse :*

*1,place Igor Stravinski , 75004 Paris*

*Nom et Prénom du Responsable : Roebel Axel*

*Nom et Prénom du Tuteur INPG : Girin Laurent*

*Noms, prénoms et option des Etudiants : Maller Simon*

*Option TST - Signal/image*

| | |
|---|---|
| ***Ecole Nationale Supérieure d'Electronique et de Radioélectricité de Grenoble***<br>Grenoble INP – Minatec, 3 Parvis Louis Néel - B.P. 257<br>*38016 GRENOBLE Cedex 1 – France*<br>*Tél : 33 (0)4 56 52 91 00 - Fax : 33 (0)4 56 52 91 03* | ***Télécom - ENSIMAG***<br>*681 rue de la Passerelle – Domaine Universitaire – BP 72*<br>*38402 ST MARTIN D'HERES Cedex – France*<br>*Tél : 33 (0)4.76.82.72.22 – Fax : 33 (0)4.76.82.72.50* |

# Sommaire


Glossaire ..... 3
Abstract ..... 4
1.Introduction ..... 5
Présentation du laboratoire ..... 7
Rappel du cahier des charges ..... 11
2. Partie théorique ..... 13
2.1 Système d'analyse/synthèse ..... 13
2.2 Formalisation mathématique de la transformée de Fourier à court terme ..... 15
2.2.1 Quelques rappels sur l'analyse de Fourier discrète ..... 15
2.2.2 La transformée de Fourier à court terme ..... 19
2.2.3 Fonctionnement du vocodeur de phase ..... 24
2.3 Modifications du son avec le vocodeur de phase ..... 28
2.3.1 Le « time stretching » ..... 28
2.3.2 La transposition ..... 33
2.3.3 Le saut temporel ..... 34
2.3.4 Le reverse/repeat ..... 37
2.4 Détection du fondamental ..... 38
2.5 Estimation de l'enveloppe spectrale ..... 41
3. Travail réalisé ..... 44
Etat de l'art et cadre de travail ..... 44
Démarche générale ..... 45
Plan de travail effectué et comparaison avec le prévisionnel ..... 45
3.1 Transposition des modulations ..... 46
Introduction ..... 46
3.1.1 Fonctions développées ..... 48
3.1.2 Le cas de la sinusoïde pure ..... 52
3.1.3 Le cas des sons réels ..... 54
3.2 Modifications temporelles des modulations ..... 62
3.2.1 Méthode ..... 62
3.2.1 Exemples ..... 65
3.2.3 Perspectives ..... 67
Conclusion ..... 68
Remerciements ..... 69
Bibliographie ..... 70
Lexique ..... 71
Diagramme de Gantt ..... 72
Annexes Techniques ..... 74
A1. AudioSculpt ..... 74
A2. Les Analyses utilisées ..... 75
A3. Exemple de fonction : fonction trans.m ..... 77

# Glossaire

*Audiosculpt* : Interface pour le traitement sonore développé par l'Ircam.

*« F0 »* : algorithme de détection de la *fréquence fondamentale.*

*Format SDIF (Standard Data Interframe File)* : format de fichier dédié aux descripteurs audio, dans lequel les données sont organisées en trames temporelles successives.

*IRCAM* : Institut de Recherche et Coordination Acoustique/Musique.

*SuperVP* : logiciel d'analyse synthèse sonore développé par l'Ircam , basé sur la technique du vocodeur de phase.

# Abstract

I have done my final study internship at IRCAM (Institute for music/acoustic research and coordination, Paris) in the sound Analysis and synthesis team of the research and development department.
Ircam was created by the composer Pierre Boulez in 1969 and is associated to the national center of art and culture George Pompidou. Its fundamental principle is to encorage productive interaction between scientific research, technological development and contemporary music production.

Its activities are structured by themes, entrusted to specialized team, which assume research, software developement, collaborative projects and diffusion.

The sound analysis and synthesis team I am working with carries out activity in sound analysis, transformation, and the synthesis of sound signal. Technics of transformation and synthesis are initially created to respond to the needs of musicians in the production of new sounds and new music.

Methods used for sound transformations are more and more efficient and provide natural sounding results.
However, there is still a problem for sounds with musical ornements, like vibratos or tremolos, which often lose their significance when the sound is being transformed.

The main objectives of the internship will be to develop methods of transformation wich allow preservation of these modulations.
Thus, I will have to extract parameters describing the modulation, using mainly Matlab and SuperVP.
Technics implemented will have to work with AudioSculpt and SuperVP environment, so that in the future, users could easily make treatments by themselves through the interface.

# 1.Introduction

## Contexte du sujet

Les techniques actuelles d'analyse et de synthèse du signal sonore permettent de transformer les sons musicaux de façon très naturelle. Pouvoir modifier presque sans limites un son préenregistré a représenté une véritable révolution dans le processus de création musicale, et continue de susciter l'intérêt des utilisateurs.

Le logiciel SuperVp développé par l'Ircam est capable de telles opérations.

Un des problèmes restant est celui des gestes musicaux et des modulations des sons (notamment le vibrato et le tremolo) qui, normalement, sont modifiés et perdent alors leur signification dans le contexte du son.
Le vibrato est une technique qui consiste à appliquer une variation rapide de la hauteur du son autour de sa tonalité (la mélodie).
Le tremolo consiste à faire varier l'intensité de la note autour d'une valeur moyenne en conservant la hauteur de départ.
Ces modulations apportent une expressivité particulière selon la façon dont elles sont effectuées : vite ou lentement (fréquence de modulation), de façon fluide ou saccadée (forme de la modulation), de façon plus au moins prononcée (extension de la modulation).
Dans le souci d'amélioration et d'extension de ses outils, l'équipe analyse/synthèse de l'Ircam souhaite parvenir à conserver cette expressivité au cours des transformations du son.

## Objectifs

L'objectif de ce stage est de développer des méthodes permettant la conservation du sens musical des modulations au cours des transformations sonores.
Ces méthodes devront être adaptées à l'environnement de travail de l'équipe, et utiliseront l'outil SuperVP, qui permet de transformer un son de façon dynamique, et d'estimer plusieurs paramètres intéressants (fréquence fondamentale, énergie locale).
D'une façon générale, il faut pouvoir extraire l'information de la modulation par analyse, pour ensuite modifier cette modulation indépendamment des autres caractéristiques du son.
Il peut par exemple s'agir d'une réduction/amplification de la modulation, d'une dilatation/compression temporelle, ou d'autres modifications…

Après intégration par les développeurs de l'Ircam, mon travail devra aboutir à une fonctionnalité supplémentaire dans Audiosculpt dédiée aux transformations des sons avec modulation.

## Plan du rapport

Après une présentation du laboratoire et un rappel du cahier des charges, le présent rapport sera organisé en deux grandes parties.
La première a pour objectif d'expliquer les outils utilisés, notamment l'analyse/modification/synthèse du son avec le vocodeur de phase. On y trouvera une justification théorique et une formulation propre des méthodes décrites dans la seconde partie, qui décrit le travail réalisé durant ce stage.

# Présentation du laboratoire

Ce paragraphe a pour but de présenter l'Institut et ses différentes activités, ainsi que l'équipe Analyse/synthèse, dans laquelle je réalise ce stage.

## Missions

L'Ircam a pour mission fondamentale de susciter une interaction féconde entre recherche scientifique, développement technologique et création musicale contemporaine. Cette articulation constitue, depuis sa création en 1969 par le compositeur Pierre Boulez, le principal axe structurant l'ensemble de ses activités.

Cette médiation entre recherches et créations musicales comporte en particuliers le développement d'outils logiciels pour les musiciens (compositeurs, interprètes, musicologues), à partir de modèles et prototypes élaborés par des équipes de recherche travaillant dans les différents domaines en rapport avec la musique : informatique (langages, IHM, temps réel, base de données), traitement du signal et automatique, acoustique, perception et psychologie cognitive de l'audition, musicologie…

Les travaux reposent ainsi sur l'articulation de deux types d'activités complémentaires, la recherche et le développement.

***La recherche*** vise l'élaboration de connaissances en rapport avec les problématiques musicales. Elle inscrit son activité sous la forme de nombreuses collaborations avec des laboratoires français et étrangers, avec des organismes d'enseignement supérieur et avec des partenariats institutionnels (notamment le CNRS) et privés. L'accueil d'élèves-chercheurs et ingénieurs dans le cadre de thèses de doctorat, de stages de Master et d'école d'ingénieurs contribue à la formation par la recherche.

Les compétences développées trouvent de nombreuses applications au-delà des problématiques musicales, et font l'objet de projets réalisés en collaboration avec des industriels ou dans le cadre de programmes nationaux, européens, et internationaux.

***Le développement*** effectue l'adaptation des connaissances, modèles et prototypes issus de la recherche sous la forme d'environnements logiciels. Les principales applications visent la réalisation d'outils pour la création musicale, à travers la mise en œuvre d'environnements ouverts et programmables, afin de pouvoir répondre à des approches esthétiques très diverses,

et d'intégrer les modèles issus des travaux de recherche au fur et à mesure de leur avancement. La mise en place du forum Ircam en 1993, club d'utilisateurs des logiciels Ircam, a favorisé la diffusion de ceux-ci auprès d'une communauté internationale de professionnels de la musique et du son, évalués à plus de 5000 utilisateurs depuis sa création. Des cessions de licences sont également accordées à des partenaires extérieurs, pour leur utilisation propre ou pour la commercialisation des logiciels.

La valorisation de la recherche et la diffusion fait également partie des objectifs de l'institut, qui organise en permanence des missions de création, de formation et de transmission. Ceci sous-entend le développement de nombreux projets, services, publications et évènements.

## Liens institutionnels

L'ircam, association à but non lucratif reconnue d'utilité publique, est associé au centre national d'art et de culture Georges Pompidou et placé sous la tutelle du ministère de la Culture et de la Communication.

Au delà des nombreuses collaborations avec des laboratoires de diverses disciplines, les principaux axes de structurations concernent :

- L'unité mixte de recherche STMS (Science et Technologie de la Science et su Son - UMR 9912), regroupant chercheurs du CNRS et de l'Ircam.
- Le parcours ATIAM (Acoustique, Traitement du signal et informatique appliqués à la musique), accueilli et coordonné par l'Ircam, organisé dans le cadre du Master Sciences et Technologies de l'université Paris-6.
- La participation du département R&D de l'Ircam comme laboratoire d'accueil des écoles doctorales de l'université Paris-6 dans des domaines de compétences, en particuliers EDITE (Ecole doctorale d'informatique, télécommunications et électronique de Paris, et SMAE (Sciences mécaniques, acoustique et électronique).

## Organisation

L'ircam regroupe plusieurs départements : création et diffusion, médiathèque, pédagogie et action culturelle, médiations recherches-création, et enfin recherche et développement, auquel nous nous intéresserons. C'est actuellement Frank Maldener qui préside l'institut dans son ensemble.

Le mode d'organisation mis en œuvre au sein du département R&D repose sur une répartition thématique des activités entre équipes spécialisées, chacune d'elles intégrant l'ensemble des la chaîne de travaux correspondant à son domaine : recherche, développement logiciel, contrats et projets extérieurs, diffusion. L'ensemble des collaborateurs (90 chercheurs, ingénieurs, doctorants, techniciens et administratifs), regroupés au sein du département Recherche et Développement de l'Ircam dirigé par Hugues Vinet, se répartissent selon les équipes ci-dessous :

- *Anayse et synthèse des sons (resp. Xavier Rodet)*
- *Acoustique instrumentale (resp. René Caussé)*
- *Acoustique des salles (resp. Olivier Warusfel)*
- *Interactions musicales temps réel (resp. Norbert Schnell)*
- *Représentations musicales (resp. Gerard Assayag)*
- *Perception et design sonore (resp. Patrick Susini)*
- *Analyse des pratiques musicales (resp. Nicolas Donin)*
- *Services en lignes* (resp. Jérôme Barthélemy)

## L'équipe Analyse et synthèse des sons

Ce paragraphe présente plus en détail les activités de l'équipe ainsi que les principales thématiques qu'elle aborde.

L'équipe est composée d'ingénieurs-chercheurs, de doctorants, et de développeurs. Son repsondable est Xavier Rodet. Mon tuteur de stage, Axel Roebel, est ingénieur-chercheur.

### *Activités*

L'équipe analyse et synthèse des sons effectue des recherches et des développement en analyse, transformation et synthèse des signaux sonores.

L'analyse des sons comprend les méthodes permettant l'extraction ou la structuration automatiques de divers types d'informations provenant du signal, comme la fréquence fondamentale ou les évolutions spectrales déterminant la hauteur et le timbre du son perçu.

Les méthodes utilisées reposent sur le traitement du signal, l'analyse statistique, la théorie de l'information, les techniques d'apprentissage et de reconnaissance des formes, mais aussi sur la connaissance de la perception auditive et de la production sonore par les systèmes acoustiques.

Analyse et synthèse reposent sur la conception d'une part de modèles de signaux (modélisation des effets produits en termes de signaux), d'autre part de modèles physiques (modélisation acoustique des causes& de production en tant que source sonore).
Ces modèles sont implantés sous la forme de logiciels pour PC ou macintosh, dotés d'interfaces graphiques spécifiquement conçues à l'intention d'utilisateurs professionnels ou non, musiciens, mais aussi ingénieurs du son, acousticiens et amateurs.

***Principales thématiques***

Les principales thématiques de l'équipe sont regroupées ci-dessus. Certains termes techniques seront expliqués dans l'annexe du rapport.
*Modèles de signaux* : analyse/synthèse additive, traitement par vocodeur de phase (cf A1).
*Caractérisations des sons* : indexation automatique
*Modèles physiques* : inversion de modèles physiques, projet Windset (modèles physique d'instruments à vent sous forme de plugin commerciaux, en collaboration avec Arturia).
*Analyse, reconnaissance, transformation et synthèse de la voix et de la parole*
Logiciels développés par l'équipe : Audiosculpt (cf A1), SuperVP(cf A2), Diphone Studio.

# Rappel du cahier des charges

Pour modifier les vibrato/tremolo, je me sers de fonctions déjà développées dans le logiciel SuperVP par l'équipe Analyse/synthèse de l'Ircam.(cf partie **1**.)

On cherche à agir sur les modulations de type vibrato/tremolo indépendamment de la partie lente du signal. Deux grandes familles de transformations ont été retenues :

1) Les *transformations temporelles* reliées à la fréquence et la durée de la modulation. Exemple : modifier la fréquence de vibrato d'une note sans en changer la durée, ou allonger une note jouée avec vibrato sans en changer la fréquence.

2) La *transposition* reliée à l'extension de la modulation par rapport à sa valeur moyenne (f0 pour le vibrato, énergie pour le tremolo). Exemple : annuler une modulation en annulant son extension à la valeur moyenne calculée, ou accentuer une modulation en augmentant cette extension.

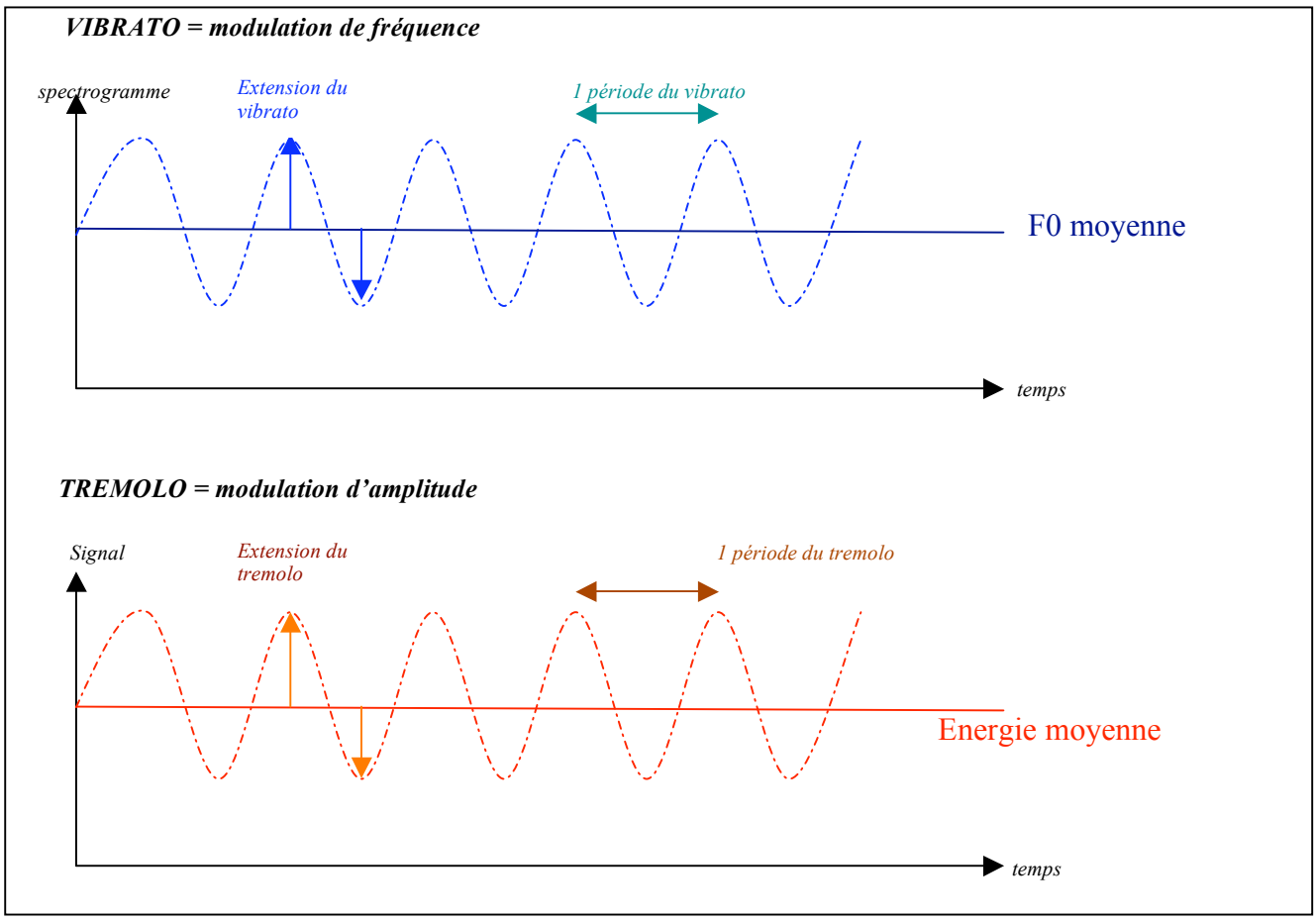



*fig 1. Les paramètres des modulations*

Les méthodes de transformation se servent de SuprVP pour l'analyse et la synthèse du signal modifié (cf. partie **1**.) Mon travail consiste à trouver les paramètres de la modification en question (facteur de dilatation du time stretch, position des segments à recopier, facteur de transposition, etc.), et ceci de façon automatique.

Dans ce contexte, je devrais avoir réalisé quatre fonctions d'ici la fin de mon stage:

- -1 fonction d'amplification/réduction de la modulation , déclinée en deux sous fonctions : une pour le vibrato (analyse f0), une pour le tremolo (analyse d'énergie locale) (cf partie **2.**)
- -1 fonction de modification temporelle de la modulation, déclinée en deux sous fonctions également : conservation de la fréquence de modulation sans changer la durée de la note, ou modification de la durée de la modulation sans en changer la fréquence.

# 2. Partie théorique

Cette partie a pour objectif d'expliquer les fondements théoriques sur lesquels se basent les transformations du son dans le vocodeur de phase SuperVP.
Après avoir détaillé les principes de base de l'Analyse/synthèse, je présenterai la technique du vocodeur de phase, et les différentes modifications du son qu'elle permet d'effectuer.
Enfin, j'expliquerai les méthodes de détection du fondamental (algorithme f0) et d'estimation d'enveloppe spectrale (algorithme « True Enveloppe »).

## 2.1 Système d'analyse/synthèse

**Généralités**

L'analyse/synthèse est un procédé numérique utilisé pour modifier un signal. Une hypothèse faite la structure du signal permet d'adapter les méthodes de modification et de les rendre performantes.
Tous les systèmes d'analyse/synthèse reposent sur le même schéma :

- phase d'analyse servant à acquérir l'information sur le signal.
- modification du signal en agissant sur les données de l'analyse
- synthèse du signal avec ses nouvelles caractéristiques

**L'analyse synthèse par transformée de Fourier à court terme :**

La transformée de Fourier  a court terme (STFT pour Short time Fourier Transform) consiste à effectuer des transformées de Fourier Discrète sur des portions de signal régulièrement espacées. Chaque portion (appelée « trame ») est obtenue en appliquant au signal une fenêtre de pondération de longueur L, puis en décalant la fenêtre d'un certain pourcentage de cette fenêtre, si bien qu'il y a un recouvrement (overlapping) de l'information analysée [1].
Dans chaque trame, on applique une transformée de Fourier discrète, qui fournit une description fréquentielle du signal entre deux instants d'analyse. L'overlapping permet un lissage des données analysées.

La synthèse du signal est obtenue en faisant une transformée  de Fourier inverse de chaque trame, en pondérant chacune de celles-ci par une fenêtre d'analyse, et en ajoutant les portions qui se superposent (« overlap-add »).

L'intérêt de la STFT est d'obtenir une description fréquentielle du signal à un instant donné. Cette représentation permet de réaliser des transformations complexes, basées sur la modification des trames dans le domaine fréquentiel.
C'est cette technique d'analyse/modification/synthèse qui est utilisée dans le vocodeur de phase.

## 2.2 Formalisation mathématique de la transformée de Fourier à court terme

Après un bref rappel de l'analyse de Fourier discrète, ce paragraphe donne le formalisme de la STFT, qui servira notamment à comprendre le fonctionnement du vocodeur de phase.

### 2.2.1 Quelques rappels sur l'analyse de Fourier discrète

Nous rappelons ici les principes de l'analyse de Fourier pour les signaux discrets, et introduisons la Transformation de Fourier Discrète (DFT pour « Discrete Fourier transform »).

***A) La transformée de Fourier pour les signaux discrets***

La transformée de Fourier $X(\omega)$ pour un signal discret $x(n)$ est donnée par :

$$X(\omega) = \sum_{n=-\infty}^{\infty} x(n) e^{-j\omega n} \quad (1)$$

où $X$ indique l'amplitude et la phase complexes du signal $x$ à la fréquence $\omega$.

La transformée de Fourier inverse s'ecrit alors : $x(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} X(\omega) e^{j\omega n} d\omega$

Comme $e^{-j\omega}$ est 2π-périodique en $\omega$, le spectre sera aussi $2\pi$-périodique.

De plus, pour des signaux réels, on a :

Ce qui signifie que la partie réelle (resp. imaginaire) du spectre est symétrique (resp. antisymétrique) par rapport à $\omega=0$. Par conséquent toute l'information spectrale est contenue entre la fréquence zéro et $\pi$. (Pour des signaux échantillonnés à Fe, on a $\omega = 2\pi f = 2\pi \frac{fa}{Fe}$, où $fa$ est la fréquence analogique).

***B) La transformée de Fourier Discrète (DFT)***

Pour que la somme dans (1) converge, il faut que $x(n) = \frac{1}{N}\sum_{k=0}^{N-1} X(k\Omega_N) e^{j\Omega_N kn} \leq \infty$, ce qui exclut le cas des signaux périodiques, non absolument sommables.

La DFT utilise la propriété suivante : un signal périodique de période *N* peut être représenté avec *N* fonctions de base $e^{j\omega n}$ de même périodicité que le signal, et dont les fréquences valent $\omega = k\frac{2\pi}{N} = k\Omega_N$.

On va considérer que la portion de signal analysé est le résultat d'une sélection par une **fenêtre rectangulaire de taille *N*,** d'un signal dit « périodisé », qui serait une répétition de ces portions de taille *N*.

La DFT d'un signal de période *N* est ainsi donnée par :

$$X(k\Omega_N) = \sum_{n=0}^{N-1} x(n) e^{-j\Omega_N kn} \quad (2)$$

La transformée de Fourier discrète inverse (IDFT) s'écrit :

$$x(n) = \frac{1}{N}\sum_{k=0}^{N-1} X(k\Omega_N) e^{j\Omega_N kn} \quad (3)$$

Lorsque $N = 2^k$, il existe une implémentation très efficace de la DFT (et utilisée dans superVP) : la FFT (Fast Fourier Transform).

Pour la FT la portion de signal analysé est $\Re$, elle n'est plus que de *N* points pour la DFT.

Pour comprendre le lien entre la FT et la DFT, on peut voir que la DFT échantillonne le spectre obtenu par la FT aux positions $\omega = k\Omega_N$.

Ainsi, si on augmente *N*, on augmente le nombre d'échantillons, et donc la précision de la DFT. Ceci est la base de la technique du « *zero padding* » qui rajoute un certain nombre d'échantillons de valeur nulle pour avoir une taille de FFT plus grande (si la fenêtre d'analyse recouvre *K* échantillons, on met *N*-*K* échantillons à 0 pour avoir une taille de FFT *N*, cf. fig2).

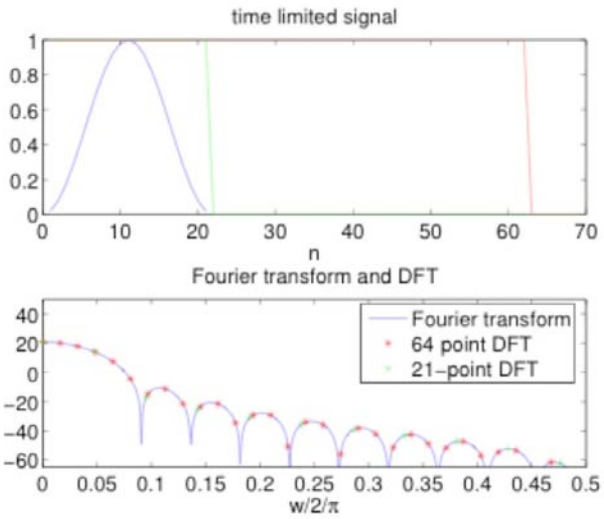



**Figure 2 : FT et DFT. En rouge, le zero padding. (source : A.Roebel, AMT part1)**

## *C) Fenêtrage*

Pour les signaux de durée infinie, on va couper le signal en segments considérés comme stationnaires. Pour ce faire, on multiplie le signal par une courte fenêtre d'analyse.

***Choix de la taille de la fenêtre***

La largeur de cette fenêtre détermine la résolution fréquentielle de l'analyse. Plus la fenêtre sera large, plus la résolution fréquentielle sera élevée. Mais moins bonne aussi sera la résolution temporelle du signal. Ce compromis de la résolution temps/fréquence est récurrent dans l'analyse STFT. (cf 2.2.2).

***Choix du type de fenêtre***

La multiplication dans le domaine temporel étant équivalente à une convolution dans le domaine spectral, le choix de la fenêtre utilisée (type et taille) aura une influence importante sur l'analyse.

Les fenêtres couramment utilisées se comportent comme des filtres passe-bas dans le domaine fréquentiel. Elles sont choisies selon le compromis entre la largeur de ce filtre (largeur de bande du lobe principal), et l'atténuation des lobes secondaires : (cf. fig 3)

- plus le lobe principal est étroit, meilleure sera la précision fréquentielle, de sorte que deux sinusoïdes voisines pourront être vues comme deux pics spectraux distincts.
- plus l'atténuation des lobes secondaires est grande, moins il y aura d'interférences entre des sinusoïdes voisines.

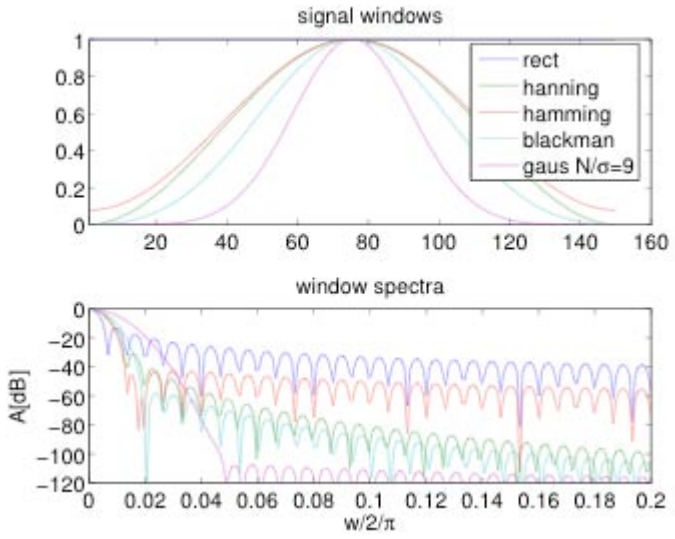



**Figure 3: différentes fenêtres utilisées et leur spectre**

### *D) Analyse d'une sinusoïde complexe fenêtrée*

Le vocodeur de phase faisant l'hypothèse du modèle sinusoïdal (le signal est vu comme une superposition de sinusoïdes, cf. 2.2.2), il est intéressant de savoir ce que va être le spectre d'une sinusoïde après fenêtrage.

Soit le signal, et la fenêtre d'analyse *w(n)* de longueur *M* et de spectre $W(k\Omega_N)$.

La TF du signal *x(n)* fenêtrée à la position *m* (début de fenêtre à l'instant m) est alors : [5]

$X(m,\omega) = e^{j((m+\frac{M-1}{2})\Omega+\Phi)} e^{-j(\frac{M-1}{2})\omega} |W(\omega-\Omega)|$ (4) , où Ω est la fréquence de la sinusoïde analysée.

. $e^{j((m+\frac{M-1}{2})\Omega+\Phi)}$ correspond à la phase de la sinusoïde à la position centrale de la fenêtre.

. $e^{-j(\frac{M-1}{2})\omega}$ est un déphasage introduit par le fait que les fonctions de base de la DFT n'ont pas leur origine au centre de la fenêtre mais au début. Ce facteur ne dit rien sur le signal et provient du calcul seul. Il devra être compensé.

. La réponse en amplitude est celle de la fenêtre centrée à la fréquence de la sinusoïde.

### 2.2.2 La transformée de Fourier à court terme

#### *A) L'analyse : principe et formulation*

On vient de voir que l'on pouvait obtenir de l'information localement sur le signal en lui appliquant une fenêtre d'analyse à un instant donné. L'idée pour l'analyse temps-fréquence est d'utiliser une séquence de ces trames pour obtenir une analyse spectrale du signal variant au cours du temps.
La formulation mathématique pour l'analyse est la suivante :

$$X(n,k) = \sum_{m=n}^{N+n-1} w(m-n)x(m)e^{-j\Omega_N k(m-n)} \qquad (5)$$

où $n$ est la position de la trame dans le signal, et $k$ la fréquence observée. On parle du 'bin' $k$.

Pour les fenêtres symétriques : [5]

$$\begin{aligned} X(n,k) &= \sum_{m=n}^{N+n-1} w(m-n)x(m)e^{-j\Omega_N k(m-n)} \\ &= \sum_{-\infty}^{\infty} x(m)(w(n-m)e^{j\Omega_N k(n-m)}) = x(n) * (w(n)e^{j\Omega_N kn}) = x(n) * h_w(n,k) \end{aligned} \qquad (6)$$

On peut ainsi interpréter la STFT comme la convolution du signal avec $N$ filtres passe-bande de fréquence centrale $k\Omega_N$ et de fonction de transfert le spectre d'amplitude de la fenêtre, centré à cette fréquence. C'est la vision « banc de filtres », qui servira notamment à comprendre comment fonctionne le vocodeur de phase (cf. 2.2.3)
$X(n,k)$ représente le signal à la position $n$ et à la fréquence $k\Omega_N$.

### *B) Les paramètres de la STFT*

Le but est d'obtenir une information précise sur l'évolution des sinusoïdes (aussi appelée « trajectoire des sinusoïdes ») qui constituent le signal (la STFT se base sur le modèle sinusoïdal), pour obtenir l' évolution fréquentiel du signal au cours du temps. Nous voyons dans ce paragraphe les différents paramètres qui sont à choisir dans l'analyse STFT.

- La taille et le type de la fenêtre d'analyse :

La taille de la fenêtre détermine la résolution temps-fréquence de l'analyse, et doit être choisie judicieusement pour que les trajectoires des sinusoïdes soient extraites correctement.

Une fenêtre trop courte aura une mauvaise résolution fréquentielle, car le lobe principal de son spectre sera large. Au sein du même bin, il pourra se trouver plusieurs pics fréquentiels du signal observé :

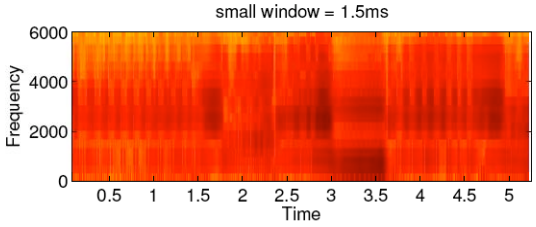



***Figure 4 : taille de fenêtre trop petite***

À l'inverse, une fenêtre trop grande n'aura pas une résolution temporelle suffisante pour que l'évolution des sinusoïdes au cours du temps soit visible :

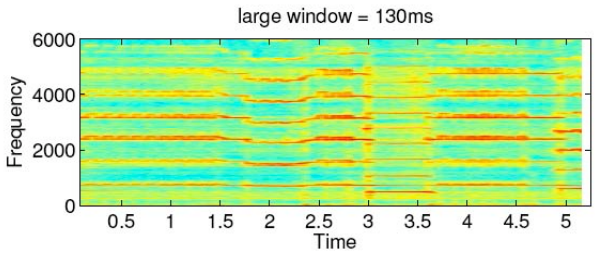



***Figure 5 : taille de fenêtre trop grande***

Pour les sons harmoniques, on choisit une taille de fenêtre qui comporte au moins 4 périodes du signal. Mais *F0* variant avec le temps, si on veut une taille de fenêtre fixe au cours de l'analyse, on choisit celle dont on est sûr qu'elle comporte 4 périodes, soit :

$$L = \frac{4}{\min(F0(t))}$$

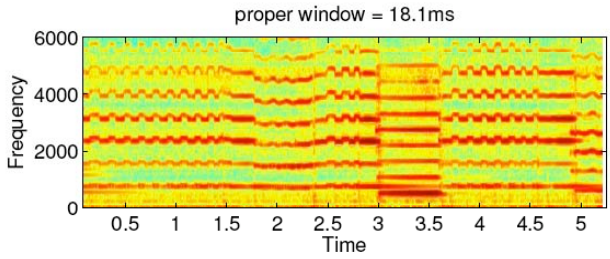



***Figure 6 : taille de fenêtre adaptée***

- Taille de la FFT

La taille de la FFT donne la résolution de la représentation de l 'analyse (densité de la grille des bins ). Elle est choisie généralement plus grande que la taille de la fenêtre (technique du zéro padding).

Le fait de fenêtrer le signal temporel revient à une convolution dans le domaine fréquentiel entre le spectre du signal et celui de la fenêtre. Le résultat de cette convolution, suivant la taille et le type de fenêtre choisie, donne une résolution fréquentielle plus ou moins bonne. La taille de la FFT n'influe pas sur cette résolution, mais bien sur la densité de la grille de visualisation du résultat de la convolution (il y a autant de « cases » dans cette grille que de points dans la FFT).

- Pas d'avancement (Hop size) :

Le pas d'avancement *I* est la durée en échantillons qui sépare deux fenêtres d'analyse successives.

Si *I* est trop grand, on n'a pas une bonne représentation du spectrogramme, du fait que l'obtention du spectrogramme se fait en sous - échantillonnant (i.e en prenant une valeur tous les *I* instants) un spectrogramme idéal de pas d'avancement « 1 échantillon ».

(une fenêtre d'analyse tous les échantillons)

Faisons alors le parallèle avec l'échantillonnage d'un signal, qui doit respecter le théorème de Shannon : $Fe \geq 2F\max$ .

Sauf qu'ici, le signal à échantillonner est le spectrogramme idéal : $Fe$ correspond alors à $\frac{1}{I}$et $Fmax$ à la fréquence maximale du spectrogramme idéal. Pour trouver cette fréquence *Fmax,* il faut voir que le spectrogramme idéal est l'observation de l'évolution des sinusoïdes du signal. Donc au sein de chaque bin *k,* le spectrogramme ne peut varier que dans la bande spectrale donnée par la taille de la fénêtre : $\left[k\Omega_N + \frac{B\omega}{2}, k\Omega_N - \frac{B\omega}{2}\right]$, où $k\Omega_N$ est le bin observé, et $B\omega$ la largeur spectrale du 1er lobe de la fenêtre. [5]

Ainsi, « *Fmax* » correspond à $\frac{B\omega}{2}$, et pour respecter $Fe \geq 2F\max$, on prendra:

$$I = \frac{1}{Bw}$$

Si cette condition n'est pas respectée, et que les fenêtres sont trop éloignées (ce qui correspondrait à un taux d'échantillonnage trop faible), il va y avoir un recouvrement fréquentiel (phénomène d'aliasing) de l'évolution des X(n,k) pour chaque k.

Pour la fenêtre de Hanning, on trouve I=L/4 , avec L=4/F0.

- Formulation mathématique d'une STFT avec un pas d'avancement *I*, une fenêtre de taille *L*, et une FFT de taille *N*:

$$\forall k \in \{0,...,N-1\}, X(lI,k) = \sum_{m=lI}^{lI+L-1} w(m-lI)x(m)e^{-jk\Omega_N(m-lI)}, l \in \{0,\pm1,\pm2,...\} \qquad (7)$$

***C) Synthèse en l'absence de modifications***

On considère ici que les trames n'ont pas été modifiées.

La transformée de Fourier discrète inverse pour une trame est donnée par (d'après (3)) :

$$w(n')x(n'+lI) = \frac{1}{N} r_N(n') \sum_{K=0}^{N-1} X(lI,k) e^{j\Omega_N k n'}$$

où $r_N$ est une fenêtre rectangulaire de longueur $N$ qui sélectionne une seule période du signal périodique de la transformée inverse. $\{n'\}$ est le système de coordonnée au début de la fenêtre.

En prenant $n = n' + lI$, on obtient :

$$w(n - lI)x(n) = \frac{1}{N} r_N(n - lI) \sum_{K=0}^{N-1} X(lI,k) e^{j\Omega_N k(n-lI)} \qquad (8)$$

Notons $x(n,lI) = w(n - lI)x(n)$.

On additionne maintenant toutes les trames :

$$\frac{1}{N} \sum_{l=-\infty}^{\infty} (r_N(n - lI) \sum_{K=0}^{N-1} X(lI,k) e^{j\Omega_N k(n-lI)}) = \sum_{l=-\infty}^{\infty} x(n,lI)$$

$$= \sum_{l=-\infty}^{\infty} w(n - lI)x(n)$$

$$= x(n) \sum_{l=-\infty}^{\infty} w(n - lI)$$

Pour tous les n tels que $C(n) = \sum_{l=-\infty}^{\infty} w(n - lI) \neq 0$ , on a :

$$x(n) = \frac{\frac{1}{N} \sum_{l=-\infty}^{\infty} (r_N(n - lI) \sum_{K=0}^{N-1} X(lI,k) e^{j\Omega_N k(n-lI)})}{C(n)} \qquad (9)$$

La division par ce coefficient de normalisation *C(n)* sert à compenser la fenêtrage lors de l'analyse, qui crée une modulation d'amplitude régulière sur le signal reconstruit sans cette étape.

Pour *N>L>I , C(n)* est non nul, et on peut reconstruire le signal à partir de la STFT.

***D) Synthèse après modification temps-fréquence***

Lors des transformations dans le vocodeur de phase, on modifie les trames de la STFT.

À cause du recouvrement des fenêtres d'analyse, les informations contenues dans la TFCT ne sont pas indépendantes entre elles, et il n'est donc pas forcément possible de trouver un signal temporel correspondant à une TFCT donnée arbitrairement. [1]

A partir d'une TFCT dont on ne sait pas si elle est valide (si elle correspond effectivement

un signal temporel), on va chercher à minimiser l'erreur quadratique entre *Y(lI,k)* , la STFT modifiée, et *X(lI,k),* la STFT du signal *x(n)* qu'on cherche à synthétiser. On cherche donc un signal *x(n)* tel que :

$$\sum_{l=-\infty}^{\infty}\sum_{k=0}^{N-1}\left|X(lI,k)-Y(lI,k)\right|^2 = MIN.$$

La solution optimale de ce problème a été donné par Griffin et Lim en 1984 [2] :

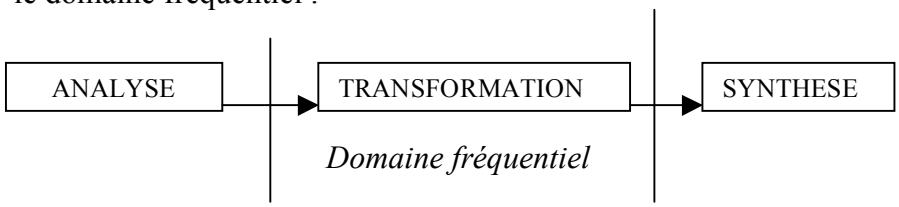
$$x(n) = \frac{\sum_{l=-\infty}^{\infty} w(n-lI)\,y(n-lI)}{\sum_{l=-\infty}^{\infty} w(n-lI)^2} \qquad (10)$$

Notons que cette formule règle également la normalisation dans le cas d'une STFT non modifiée. C'est la formule de synthèse utilisée dans SuperVP.

### 2.2.3 Fonctionnement du vocodeur de phase

***A)Principe***

Le vocodeur de phase permet de transformer les sons en faisant des modifications dans le domaine fréquentiel :




On fait l'hypothèse que le signal d'entrée est une somme de sinusoïdes (modèle additif), et que chacune de ces sinusoïdes va pouvoir être observée au sein de chaque bin, à un offset près (cf. B). L'analyse permet d'extraire l'évolution de la phase et de l'amplitude de ces sinusoïdes en fonction du temps. Les valeurs des phases ainsi obtenues permettent le calcul de la **fréquence instantanée** : ainsi, si on déplace une trame lors d'un time stretch (cf 2.3.1) par exemple, on pourra déduire sa nouvelle phase à partir de l'estimation de cette fréquence et de la valeur de la phase de la trame précédente (cf 2.3). C'est de cette façon que la cohérence des phases est assurée.

***B) Point de vue banc de filtres***

On a vu d'après 2.2.2 - A (6) que l'on pouvait interpréter la STFT comme *N* filtres passe-bande centrés sur la fréquence du bin observé [3].Pour comprendre comment on peut obtenir l'évolution de la phase et de l'amplitude d'une sinusoïde au cours du temps, nous allons étudier le fonctionnement d'un de ces filtres pas à pas.

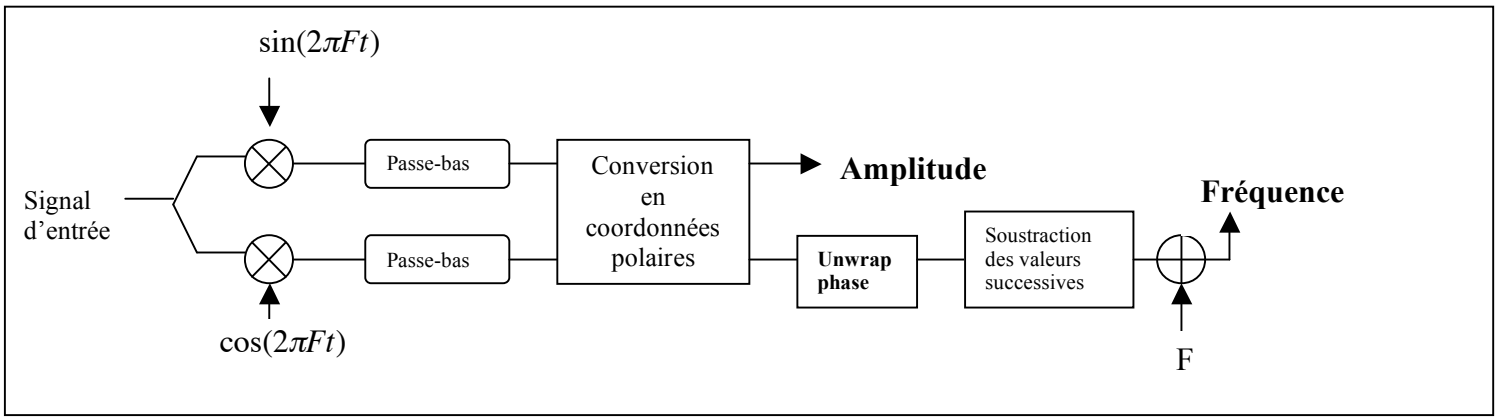
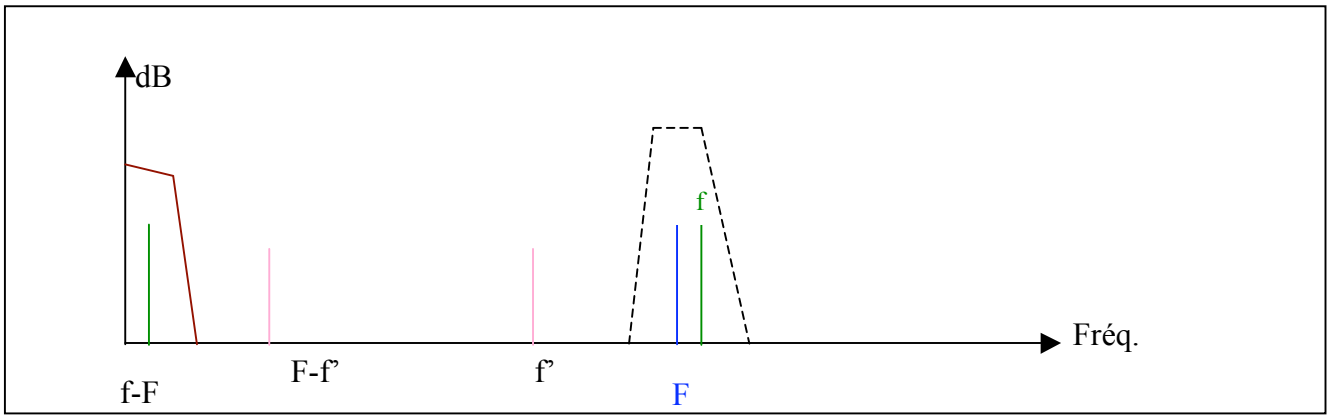



**Fig.7 : Strcuture du filtre passe bande**

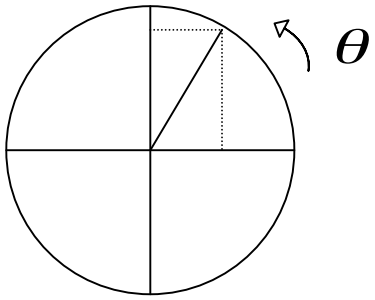
1. La première étape consiste à multiplier le signal par $\sin(2\pi Ft)$ sur une voie, et $\cos(2\pi Ft)$ sur une autre voie, *F* étant la fréquence centrale du filtre passe-bande étudié.
Cette modulation déplace le spectre du signal d'entrée de $+F$ et $-F$.
Ensuite, on effectue un filtrage passe-bas sur chacune des deux voies, pour ne sélectionner que les fréquences qui étaient proches de F (structure hétérodyne) :




Remarque : On réalise ainsi le filtrage passe-bande qui correspond à celui de la vision banc de filtre. Les caractèristiques du filtrage (raideur et fréquence de coupure) correspondent alors aux caractéristiques de la fenêtre d'analyse STFT.

2. Les deux signaux à la fréquence *F-f* obtenus sont déphasés de $\pi/2$ et peuvent être vues comme les coordonnées horizontales (cos) et verticales (sin) sur le cercle unité. On passe en coordonnées polaires pour avoir facilement l' évolution de la phase :

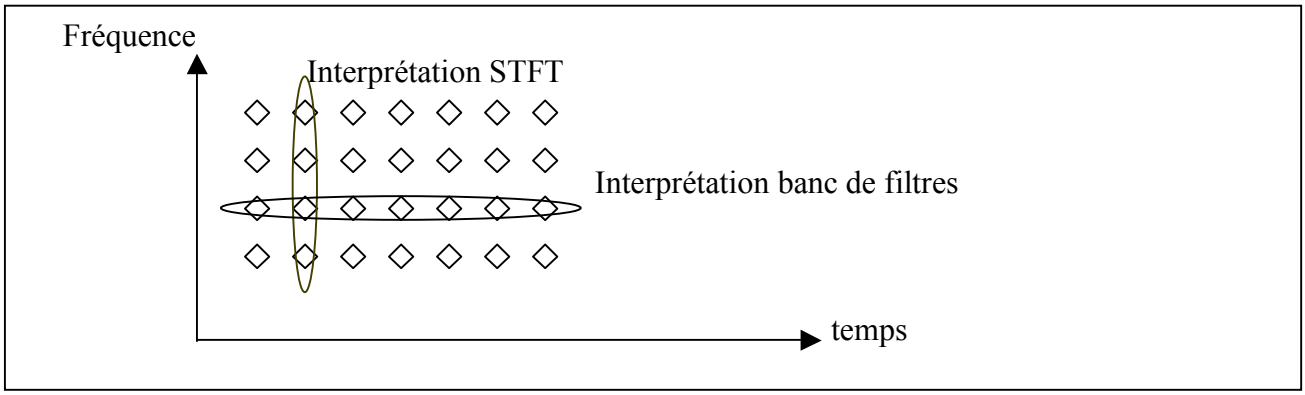
$$\theta = \arctan\left(\frac{x0}{y0}\right),$$
$$r = \sqrt{x0^2 + y0^2}$$




2. Pour calculer la fréquence instantanée (cf. 2.3), il suffit de prendre 2 valeurs successives de la phase, puis de diviser par le temps qui sépare ces deux valeurs. Ex : si on a relevé une valeur de 45° à t et de 90 ° à *t+T*, la fréquence vaut *90-45/T=45/T*. Si on a relevé à deux instants successifs *315 et 0*, le problème est qu'on calcule *0-315/T*, ce qui ne correspond pas à une fréquence. Pour que l'estimation se fasse correctement, il faut changer *0* en *360.* En fait, il faudra ajouter *360* à chaque fois qu'un tour de cercle aura été fait : c'est le « phase unwrapping ».

3. On ajoute la fréquence *F* à la valeur estimée, afin d'obtenir la fréquence recherchée.

Le vocodeur de phase n'est pas implémenté selon cette interprétation, mais utilise la STFT et l'algorithme efficace FFT pour les calculs de DFT. Ces deux points de vue sont finalement identiques pour l'interprétation du vocodeur de phase, et on peut les résumer avec le schéma suivant [3]



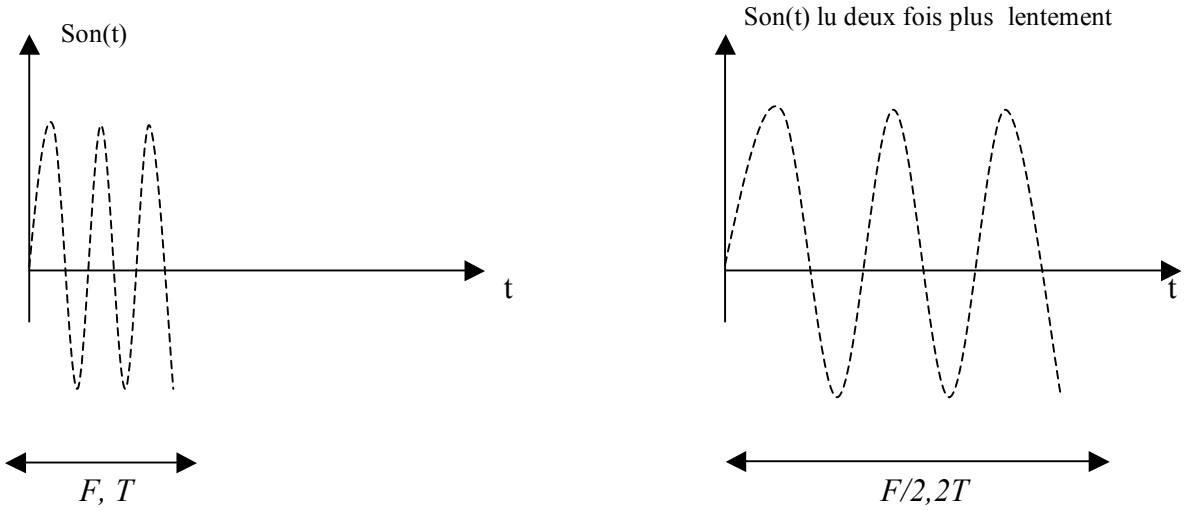
On retrouve que le nombre de filtres passe-bande est en fait la taille de la FFT.
Les caractéristiques de coupure de ces filtres sont déterminées par le type de la fenêtre utilisé par la STFT;  la raideur de coupure augmentant avec la durée de la fenêtre.

## 2.3 Modifications du son avec le vocodeur de phase

On a vu que la connaissance de la phase instantanée permettait d'estimer la fréquence des sinusoïdes, considérées comme stationnaires au sein d'une trame. À partir de cette estimation, on peut calculer la phase d'une trame déplacée lors de la transformation (ex : time stretch, cf. 2.3.1), et ainsi assurer la cohérence des phases lors de la reconstruction du signal.

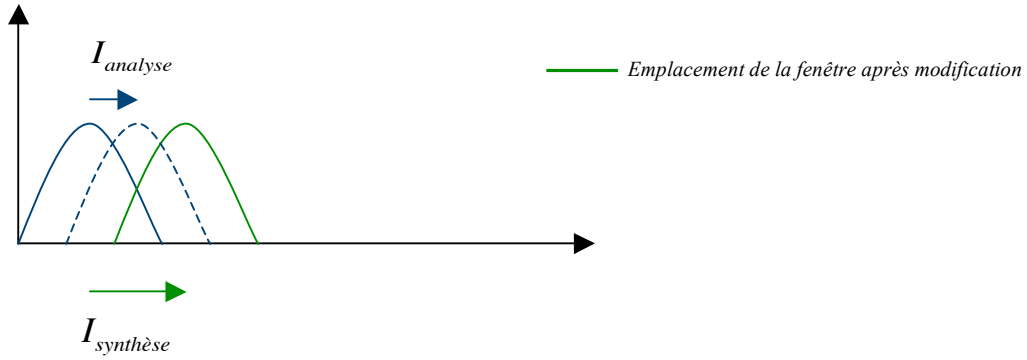
### 2.3.1 Le « time stretching »

Ce paragraphe présente la technique du time stretching, qui consiste à étirer un signal sans en changer la tonalité. Lorsqu'on lit un disque à une vitesse plus lente que l'enregistrement (disons deux fois plus lente), le son est plus grave. En effet, une note à la fréquence 100 Hz par exemple, qui a été jouée pendant une durée *T* à l'enregistrement, est jouée pendant une durée *2T* à la lecture. L'évolution du signal étant deux fois plue lente, la fréquence est divisée par 2 :




Le vocodeur de phase permet de séparer l'information fréquentielle de l'information temporelle ; il va donc être possible de réaliser un time stretch sans changer la tonalité. L'idée consiste à ajuster la fréquence de synthèse des trames STFT en fonction du facteur de dilatation, et de corriger la valeur des trames déplacées pour assurer la cohérence des trames successives.

Pour une dilatation facteur 2, les trames *Y(lI,k)* et *Y((l+1)I,k)* de la STFT modifiée seront espacées de *2I* et non plus de *I* :




Nous allons voir qu'il faudra faire attention à deux choses différentes pour la reconstruction d'un signal sans artefacts : la cohérence temporelle des phases (synchronisation horizontale), et la synchronisation des phases de chaque bin (synchronisation verticale), introduite par Laroche et Dolson en 1999 [4].

***A) Synchronisation horizontale***

Regardons comment évolue la phase des bins de la STFT.
Pour cela, considérons une sinusoïde *x(n)* de fréquence $\Omega_N$

$x(n) = e^{j(\Omega_N + \Phi)}$, *w(n)* la fenêtre d'analyse de DFT W(k).

D'après 2.2.D (4) qui donne la DFT d'une sinusoïde fenêtrée, on a, à la position *lI* et pour le bin *k* :

$$X(lI,k) = e^{j((lI + \frac{N-1}{2})\Omega + \Phi)} e^{-j(\frac{N-1}{2})k} \left|W(k-\Omega)\right| = Ke^{jlI\Omega} \quad (11)$$

après compensation du terme $e^{-j(\frac{N-1}{2})k}$ introduit par filtrage. $K = e^{j((\frac{N-1}{2})\Omega + \Phi)} \left|W(k-\Omega)\right|$
Ainsi, pour une même sinusoïde, tous les bin ont la même phase, et celle-ci évolue linéairement avec *lI*.

On peut déduire $\Omega$ en dérivant deux valeurs de phases successives.

Mais pour des valeurs de $\Omega$ trop grandes, la phase va recouvrir $2\pi$ et on retrouve le problème du recouvrement de phase. L'adaptation du pas d'avancement I fait que l'erreur est limitée aux régions où l'amplitude des sinusoïdes sera faible :

$$\Omega_{\lim} = \pm\frac{\pi}{I} \quad [5]$$

On va donc considérer que l'amplitude de la STFT n'est pertinente que dans la bande située autour de la fréquence $\Omega$, et on va ainsi estimer la fréquence uniquement au voisinage de la fréquence du bin [5].

Notons $\Theta_k$, l'offset entre le pic de la sinusoïde et la fréquence du bin $k\Omega_N$ :

$$X(lI,k) = Ke^{jlI(\Theta_k + k\Omega_N)}$$

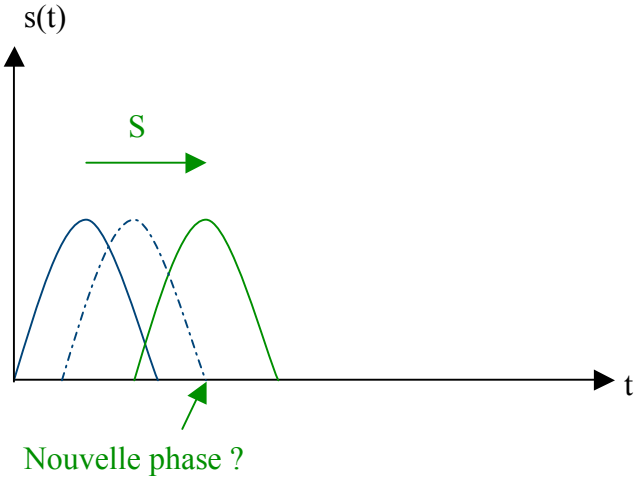
Le calcul de $\Theta_k$ est donné par : [5]

$$\Theta_k = \frac{\left[\arg(X((l+1)I,k)) - \arg(X(lI,k) - I\frac{2\pi k}{N}\right]_{2\pi}}{I} \quad (12)$$

où $[\Phi]_{2\pi} = \left(\frac{\Phi}{2\pi} - round(\frac{\Phi}{2\pi})\right)(2\pi)$ sélectionne la valeur de l'argument entre - $\pi$ et $\pi$.

(Cette étape ne se fait que pour les sinusoïdes qui sont loin de la fréquence du bin, et qui ont une amplitude faible).

Une fois la fréquence $\Theta_k$ ainsi calculée, on va pouvoir trouver quelle sera la phase d'une trame déplacée pour le time stretch :




Soit S le déplacement entre la trame (l-1)I et lI après le time stretch.

La nouvelle phase pour la trame (l,k), déplacée de S, s'obtient en sommant la phase de la trame (l-1,k) et la différence de phase dû au déplacement S :

$$\Phi_S(l,k) = S \times (\Theta_k + k\Omega_N) + \Phi(l-1,k) \qquad (13)$$

En corrigeant ainsi la phase des trames déplacées, on assure la synchronisation horizontale du signal synthétisé (fig. 7) :

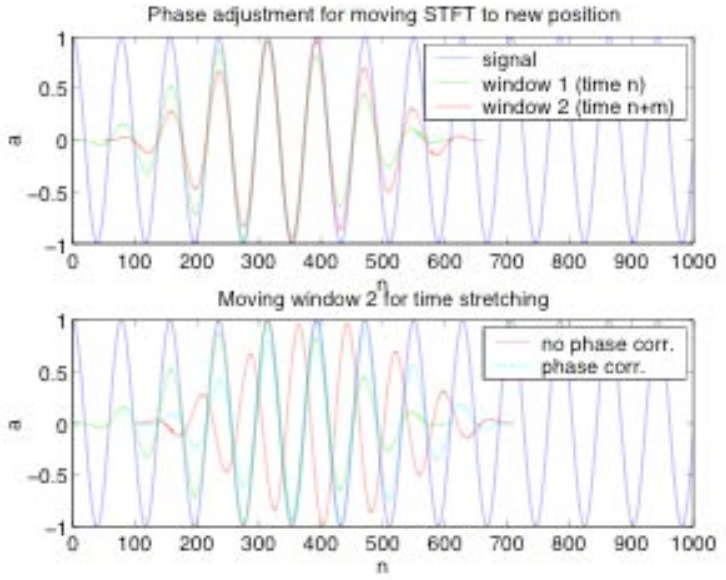



***fig8 : cohérence horizontale assurée***

### *B) Synchronisation verticale*

On voit d'après la formule (11) que l'évolution de la phase est constante pour tous les bins qui représentent la même sinusoïde.

Du fait de l'instabilité de l'intégration de la phase, une erreur dans l'estimation de la fréquence (due à l'interférence entre sinusoïdes voisines), va produire des fréquences incohérentes dans les bins qui sont liés à une même sinusoïde (fig. 8)

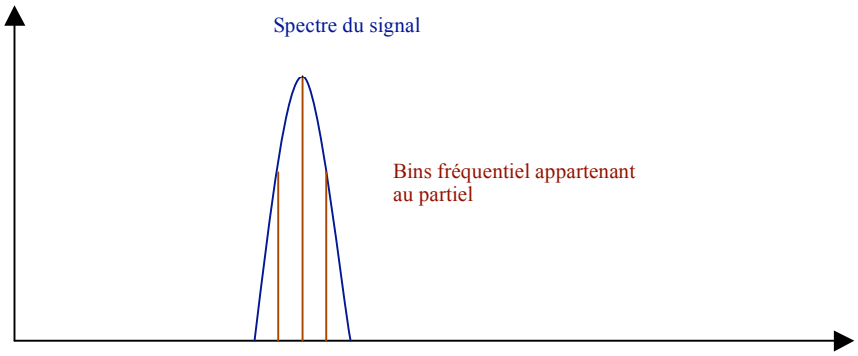



***fig 9. partiel présent dans plusieurs bins.***

La synchronisation verticale n'est plus respectée, et une modulation d'amplitude va apparaître dans le signal synthétisé.

La solution à ce problème a été proposée par Laroche et Dolson [3] ; et consiste à :

- calculer la nouvelle valeur de la phase seulement pour le centre du pic spectral.
- forcer la synchronisation verticale entre le centre du pic et les bin voisins, en copiant simplement les différences de phases donnée dans la trame analysée.

### 2.3.2 La transposition

La transposition est l'opération qui consiste à déplacer dans le domaine spectral tous les pics en conservant le rapport harmonique.
Elle est effectuée par rééchantillonnage temporel dans SuperVP, et sera utilisée pour transposer un vibrato (cf 3.1).

Supposons un signal *x(n)* et sa FT $X(\omega) = \sum_{n=-\infty}^{\infty} x(n)e^{-j\omega n}$.

On va effectuer la procédure suivante :

$y(n) = x(\frac{n}{L})$ pour $n = 0, \pm L, \pm 2L, \ldots$ , et $y(n) = 0$ sinon, de sorte qu'on obtient le signal

$y(n) = \sum_{-\infty}^{\infty} x(k)\delta(n - kL)$.

On change le taux d'échantillonnage *SR* en *L*SR*.
La FT de *y(n)* s'écrit alors :

$$Y(\omega) = \sum_{n=-\infty}^{\infty} y(n)e^{-j\omega n} = \sum_{n=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} x(k)\delta(n - kL)e^{-j\omega n}$$

En prenant *n'L=n,* il vient [5] :

$$Y(\omega) = \sum_{n'=-\infty}^{\infty} x(n')e^{-j\omega n'L} = \sum_{n'=-\infty}^{\infty} x(n)e^{-j\omega n'L} = X(\omega L)$$

Ainsi, on voit que tous les pics du signal *x(n)* sont multipliés par le facteur de ré échantillonnage *L*.
Si on lit ce nouveau signal avec le taux d'échantillonnage original, la hauteur du son est transposée du rapport des 2 taux d'échantillonnage. Mais la durée du son est également changée de ce même rapport.

Si bien qu'on doit faire précéder une transposition par un time stretch qui compensera ce changement de durée. Au final, on aura un son transposé de la valeur souhaitée, mais qui aura conservé sa durée originale.

### 2.3.3 Le saut temporel

On peut effectuer des sauts dans le signal en utilisant un fichier de <posfile> qui indique au vocodeur de phases la position des trames à analyser. Le saut est utilisé pour changer la fréquence d'un vibrato sans changer sa durée (ex : après un time stretch facteur 2, on coupe une période sur 2). Dans ce contexte, on va placer l'arrivée du saut une période plus loin dans le spectrogramme (fig 10), c'est-à-dire un instant dont les pics spectraux du signal sont presque à la même position que dans l'instant de saut. Prenons un exemple :

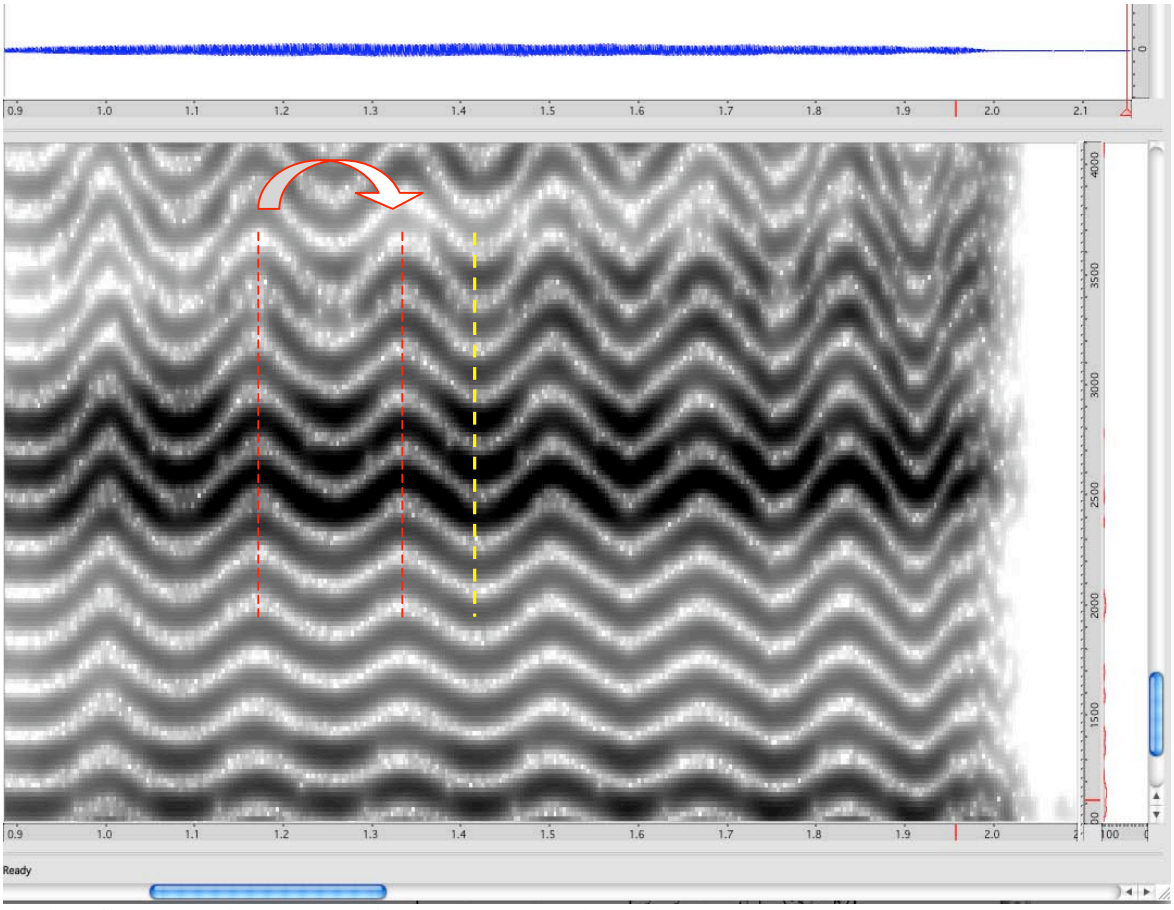
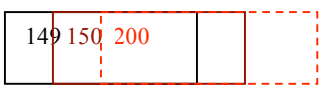
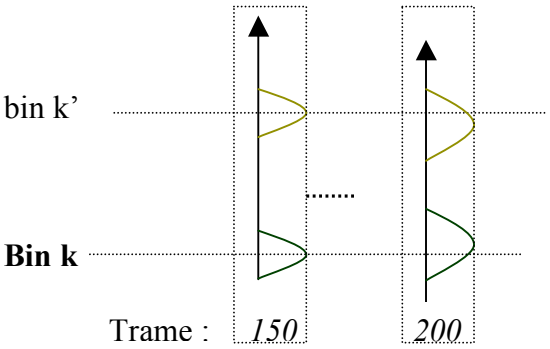



*fig.10 : illusatrtion d'un saut sur le spectrogramme d'un vibrato.*

Disons que l'instant de saut est arbitrairement la trame 150, et que l'instant d'arrivée est la trame 200. L'analyse du signal fournit une séquence de trames continue, et l'on connaît notamment toute l'information entre les trames d'analyse 150 et 200. À la synthèse, on placera la trame 200 après la trame 150 :



Comme nous sommes dans le cas où les fréquences des sinusoïdes sont presque identiques, si on se place dans le bin  k de la trame 150, la fréquence estimée sera très proche de celle du bin k de la trame 200 :



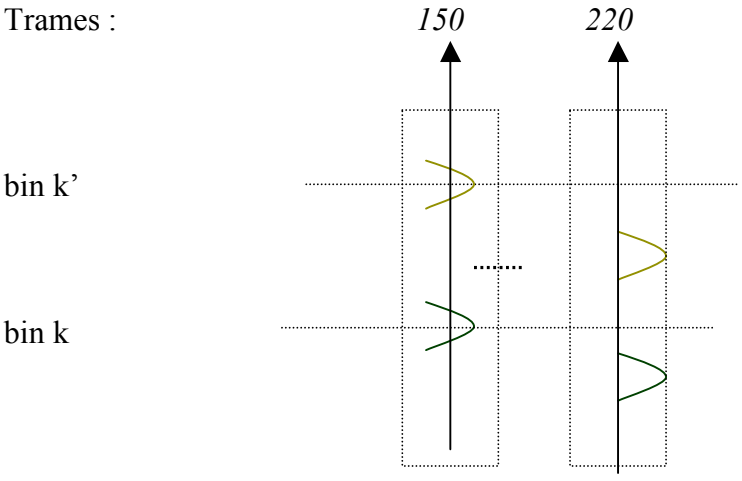

Pour assurer la cohérence des phases à la synthèse entre les trames 150 et 200, on va calculer la phase de la trame 151 avec la fréquence estimée entre les trames 199 et 200, selon (13) :

$\Phi_S(151,k) = \Phi_S(150,k) + I \times (F_i(200))$ où l'indice « S » désigne les trames de synthèse.

*Fi(200)* est la fréquence instantanée de la sinusoïde estimée dans le bin k de la trame 200 obtenue lors de l'analyse.

En faisant ainsi pour chaque bin, on peut synthétiser un signal sans « craquements », et le saut n'est pas audible. (un craquement est perçu lorsqu'il y aune discontinuité brutale dans le signal).

Dans le cas où on aurait mal situé le saut (en jaune sur la fig.9), ce qui correspond au cas où les pics spectraux des deux trames mises en jeu sont éloignées, on n'a plus continuité du vibrato :



En faisant ceci, on juxtapose deux bin dont les fréquences sont différentes. On n'aura cependant pas de cassure brusque sur le signal temporel, car les deux sinusoïdes de fréquences différentes vont s'overlapper. Et du fait du fenêtrage, l'une va disparaître progressivement, et l'autre va apparaître progressivement.
Ceci vient du fait que l'on additionne que des signaux lisses dans le vocodeur de phase, et le résultat de toute opération ne peut donc donner qu'un signal lisse également.

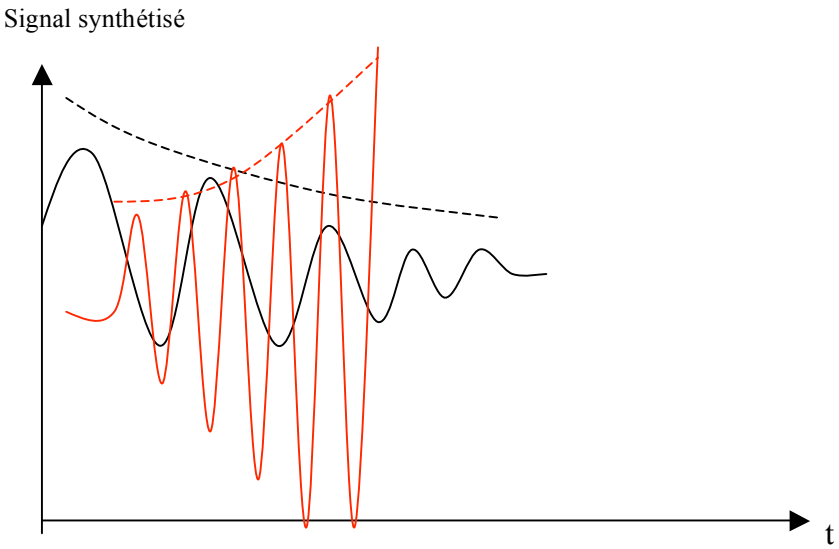
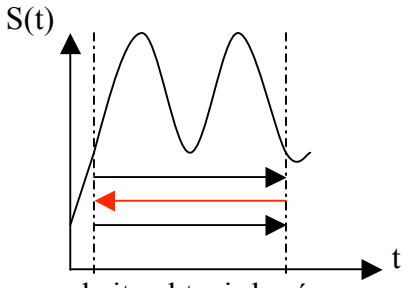



***Fig.10 : deux sinusoïdes de fréquences différentes se chevauchant***

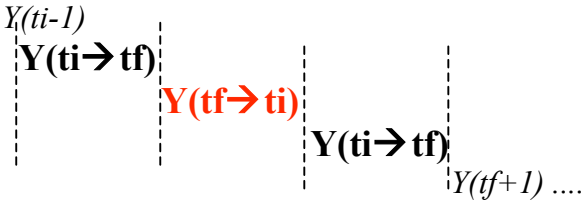
Par contre, le vibrato ne sera pas conservé, et il faudra donc bien choisir les positions de sauts.

### 2.3.4 Le reverse/repeat

Le reverse/repeat permet de répéter un segment du signal, avec lecture avant (‘forth’)et arrière(‘back’) de celui-ci. Par exemple, avec l’option ‘forth and back = 3’, on réalise une analyse du signal comme suit :




Ainsi, on souhaite obtenir la séquence de trames suivantes pour la synthèse : (ti et tf désignent les instants de début et de fin du segment où s’applique le reverse repeat)




Pour réaliser ceci, il faut adapter les phases des trames déplacées, de la même façon que pour le time stretching (13). L’analyse en arrière (back) se fait avec un overlap négatif, et il faudra faire attention à ré inverser ce signe dans les phases à la synthèse.

## 2.4 Détection du fondamental

Ce paragraphe décrit la méthode de détection de la fréquence fondamentale « f0 Feature Scoring» de superVP. Elle est utilisée pour fournir le signal *f0(t)* qui sert à la détection des vibratos. Les algorithmes utilisés reposent sur des méthodes fréquentielles (par oppostion aux méthodes temporelles qui se fondent sur le signal original pour la détection de la fondamentale [1]), et succèdent à une analyse STFT préalable.

Deux critères pondérables pour évaluer la fréquence fondamentale :

- le critère **"Spectral Match" :** les harmoniques de la fréquence fondamentale doivent s'accorder avec les pics spectraux observés, de façon à bien « expliquer » la répartition d'énergie du spectre.

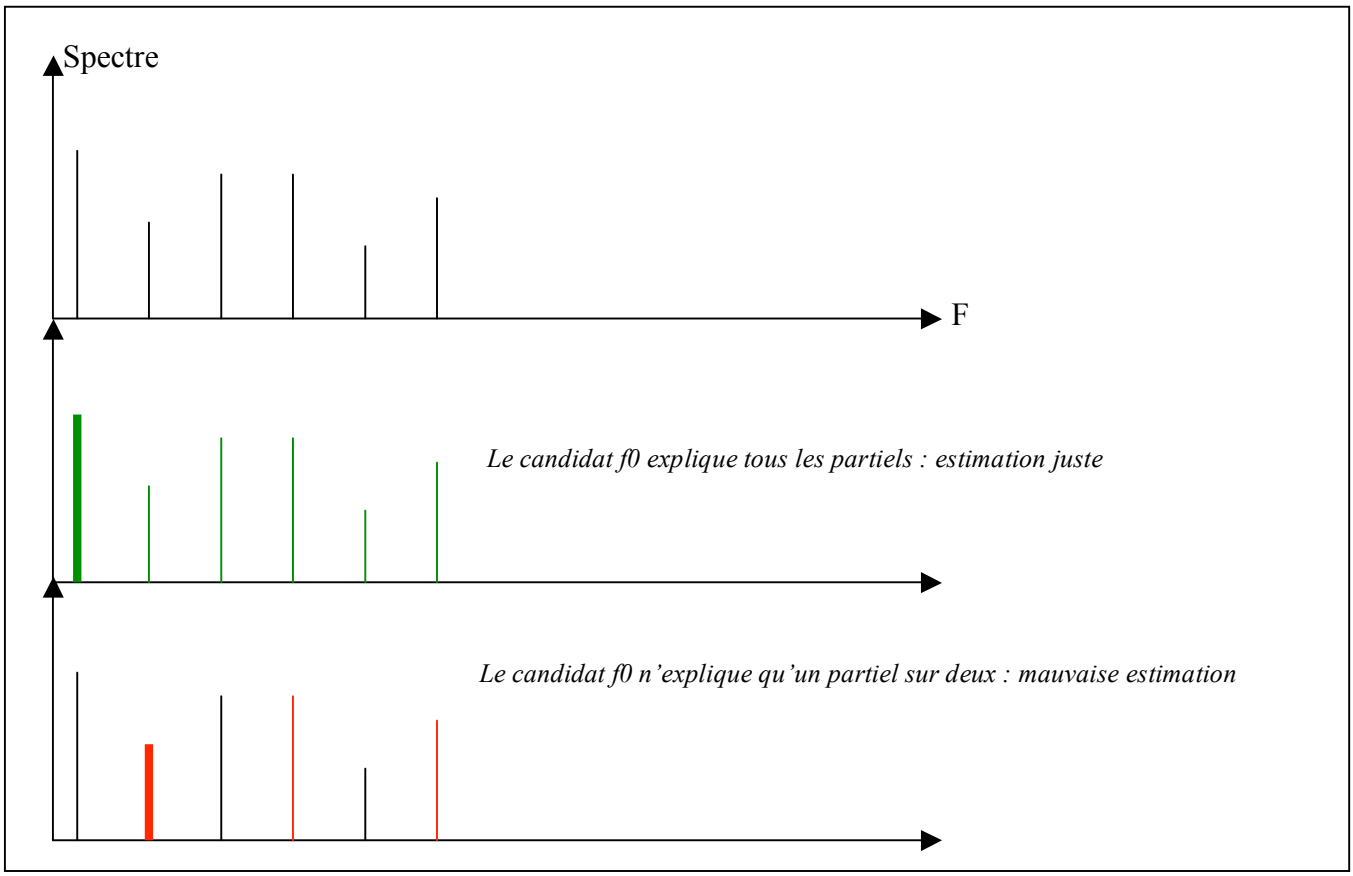



***Figure 11 : le critère « Spectral match » . En gras, le « candidat » pour f0.***

- le critère **"Enveloppe Smoothness" :** la séquence des amplitudes des harmoniques qui seront attribuées à la fréquence fondamentale doit créer une enveloppe spectrale raisonnable pour une source sonore naturelle. Nous rappelons que l'enveloppe spectrale est la courbe qui relie les pics du spectre d'amplitude.

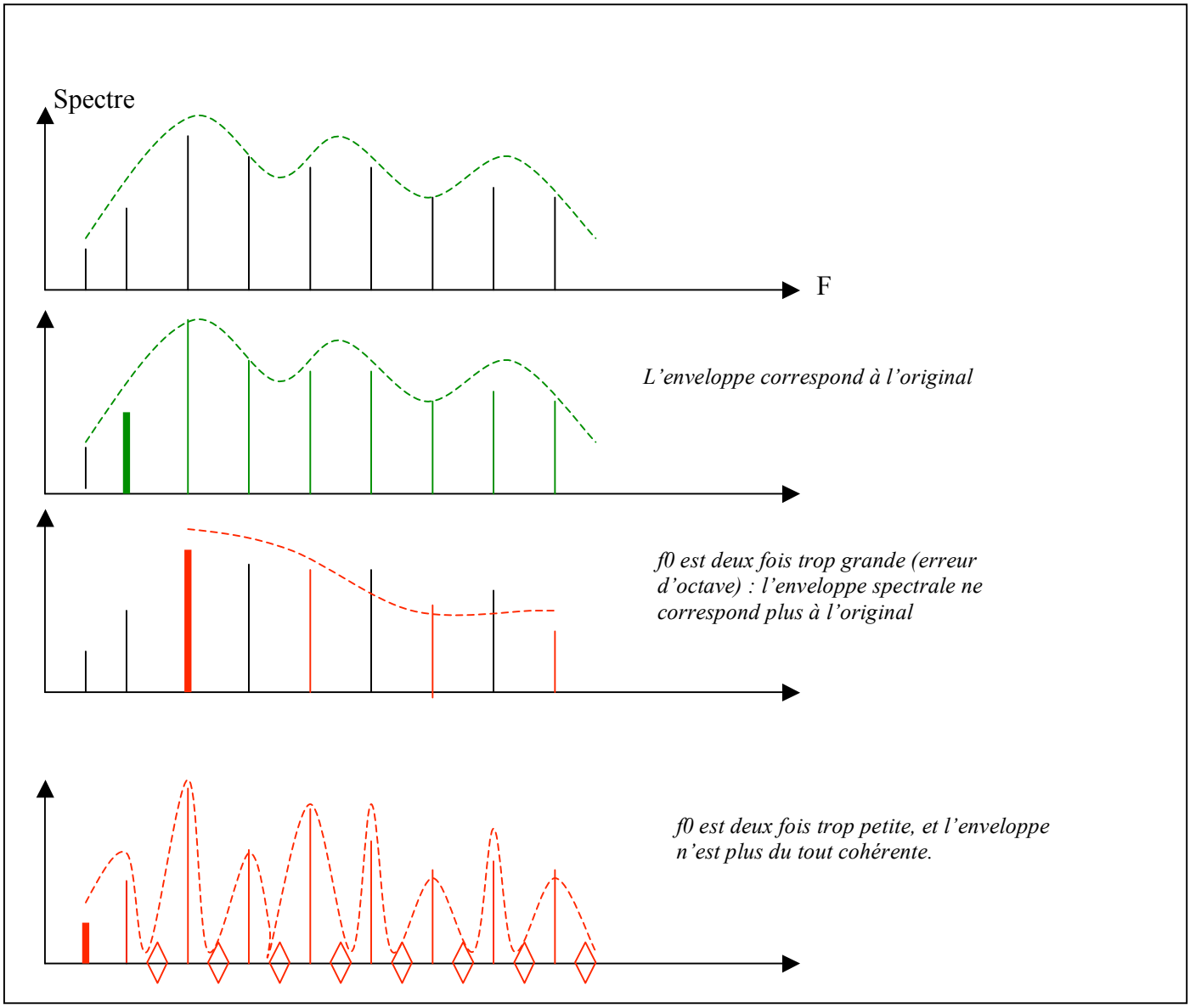



***Figure 12 : le critère « Envelope Weight »***

L'analyse "Feature Scoring" permet une adaptation par l'utilisateur du poids accordé à chacun des critères (via les paramètres : "Envelope Weight", "Spectral Match Weight") et permet ainsi d'adapter l'algorithme à des instruments spécifiques, dont on connaît l'enveloppe spectrale.

Les différents paramètres de l'analyse sont (cf annexe A2 : exemple d'analyse f0) [6]

• **“Fundamental Frequency Range” en Hz :** définition de la bande dans laquelle chercher la fréquence fondamentale : seuil minimal de fréquence fondamentale en dessous duquel la fréquence ne sera pas recherchée (par défaut : 50 Hz) et seuil maximal de fréquence fondamentale au-dessus duquel la fréquence ne sera pas recherchée (par défaut : 1000 Hz).

• **“Maximum Frequency in Spectrum” en Hz :** l’analyse ne prendra pas en compte les pics spectraux situés au-dessus de ce seuil (par défaut 4000 Hz).

• **“Smooth Order” :** ordre du filtre de post-traitement pour le lissage des fréquences (par défaut 3). C’est le lissage temporel des valeurs trouvées de F0 par l'application d'un filtre médian sur une fenêtre glissante de n valeurs consécutives de F0 observées. Cet ordre doit être de valeur impaire. Plus il est grand, plus la courbe F0 est lissée, mais moins il y a de détails temporels. En contrepartie, cela permet de limiter les erreurs locales de F0.

• **“Relative Noise Threshold” en dB :** spécifie un niveau de bruit : si la différence en amplitude entre un pic et le pic le plus haut est plus grande que cette valeur, ce pic est négligé (par défaut 50 dB).

• **“Expert Settings – Feature Scoring Weights"**

Pondération des deux critères “Spectral Match” et l'“Enveloppe Smoothness”

- “Envelope Weight” (par défaut 0.14),
- “Spectral Match Weight” (par défaut 0.26).

La somme de ces deux valeurs doit être inférieure à 1.

## 2.5 Estimation de l'enveloppe spectrale

L'analyse « True Enveloppe » de SuperVp permet de visualiser l'évolution de l'enveloppe spectrale au cours du temps.
Elle se base sur le filtrage cepstral. Avant d'expliquer cette méthode, il est nécessaite d'introduire la notion de cepstre.

*Définition et propriétés*

Le cepstre est un signal temporel issu du spectre du signal de départ. Il est défini par la FFT inverse du logarithme du module de la FFT du signal :

$$C(n) = FFT^{-1}(\log(FFT(x_n)) = \frac{1}{2\pi}\int_{-\pi}^{\pi} \log(\left|X(e^{j\omega})\right|)e^{j\omega n} d\omega$$

Il a les propriétés suivantes [1] :

- (1) Le cepstre de la réponse impulsionnelle d'un filtre causal peu résonnant est localisé autour de zéro.
- (2) Le cepstre de la réponse impulsionnelle d'un filtre en peigne $H(z) = \frac{1}{1 - c^K z^K}$ *(c coefficient réel)* est un peigne d'impulsions d'amplitudes décroissantes localisées aux valeurs temporelles $lK$ avec $l>0$.

*Utilisation*

Une multiplication dans le domaine spectral correspond ainsi à une addition dans le domaine fréquentiel. Or, on peut décomposer le spectre d'un signal en deux composantes multiplicatives [1] :
-un spectre de raies plat *P(f)* (un peigne fréquentiel où les raies sont espacées de 1/T0)
-une enveloppe spectrale *H(f)* multiplicative qui est définie à chaque harmonique par l'amplitude de l'harmonique dans le signal original.

La première composante correspond temporellement à un peigne de Dirac *p(t)* espacé de T0. Cette composante seule retient l'information sur la fréquence du signal.
La seconde composante peut être considérée comme la fonction de transfert en amplitude d'un filtre *h(t*) peu résonnant. Elle contient l'information sur le timbre du signal (ses maxima d'amplitude sont appelés les formants.)
La multiplication dans le domaine spectral correspond à la convolution dans le domaine temporel entre *p(t)* et *h(t)*.
L'intérêt de cepstre est de pouvoir retrouver les informations sur la fréquence et sur le timbre de façon additives. Ainsi, si les deux spectres *P(f)* et *H(f)* ont des caractéristiques différentes, il devient possible de les séparer.

D'après les propriétés 1 et 2 précédentes, on peut appliquer la technique du cepstre pour détecter facilement l'information d'enveloppe. Le signal est modélisé par un train d'impulsions filtré par un filtre de réponse impulsionnelle suffisamment amortie. On peut considérer le train d'impulsion d'espacement K comme la réponse impulsionnelle d'un filtre en peigne $H(z) = \frac{1}{1 - cz^K}$ avec *c* proche de 1.

D'après ce qui précède, le cepstre du signal est la somme du train d'impulsions espacé de K (propriété 2) et du cepstre de la réponse impulsionnelle du filtre, de support limité poche de zéro (propriété 1). Il devient donc aisé de différencier la contribution du filtre (aspect formantique) et la période du signal, représentée par des pics espacés autour de *T0=K*. (Remarque : cette méthode est souvent utilisée aussi pour la détection du fondamental).

*Estimation de l'enveloppe par le filtrage cepstral*

On ne veut retenir que l'information de l'enveloppe, localisée autour de zéro. On ne va donc garder que les coefficients du cepstre proches de zéro. La méthode est la suivante :

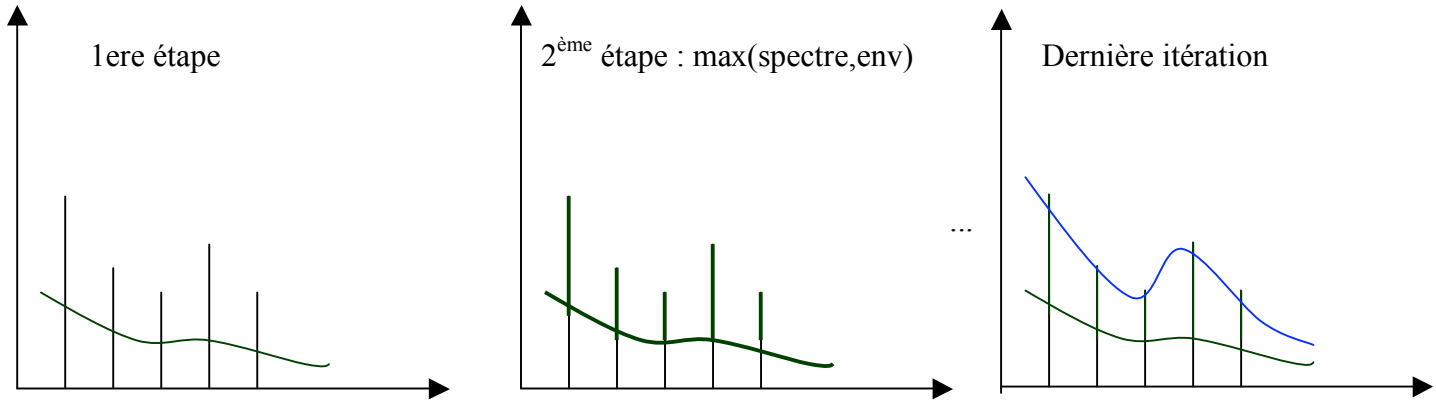
1. calcul de cepstre $C(n)$ du signal.
2. Fenétrage du cepstre : $C(n) = 0$ pour $n > n0$, nombre de coefficients cepstraux retenus.
3. Reconstruction de l'enveloppe spectrale par FFT de $C(n)$.

Le choix de *n0* est déterminant pour une bonne estimation. S'il est choisi trop grand, on va prendre, l'enveloppe aura une périodicité trop grande, et fera apparaître des creux entre les

harmoniques (or l'enveloppe doit « englober » les pics). S'il est trop petit, l'enveloppe devient très vite erroné.

L'enveloppe ainsi obtenue ne passe pas par les sommets des pics. Dans superVP, une étape supplémentaire est rajoutée , qui consiste à prendre le maximum entre le spectre et l'enveloppe, puis de réitérer l'opération jusqu'à avoir un nombre satisfaisant de pics englobés. (paramètre utilisateur).

*Spectre*



# 3. Travail réalisé

## Etat de l'art et cadre de travail

D'un point de vue traitement du signal, on peut identifier le vibrato à une modulation de fréquence, et le tremolo à une modulation d'amplitude [7].
Plusieurs modèles ont été proposés pour représenter les paramètres de ces modulations.
Le plus utilisé est le modèle additif, qui consiste à considérer les trajectoires des paramètres comme une somme de sinusoïdes, appelées « partiels », variant au cours du temps. La connaissance de ces partiels, avec leur amplitude, et de leur évolution permet un contrôle très fin de la modulation [8].
On peut également citer le modèle polynomial–sinusoïdal qui considère en plus la partie polynomiale du signal [9].
Les analyses basées sur ces différents modèles fournissent une information très précise sur les signaux, avec un très grand nombre de paramètres, mais introduisent une complexité de calcul importante. Il n'est pas forcément nécessaire de connaître avec une telle précision tous les paramètres de la modulation pour conserver son sens musical lors d'une transformation.
La simple connaissance de la *fréquence* de modulation obtenue avec l'analyse des variations de F0 (pour le vibrato) et d'énergie (pour le tremolo) pourrait suffire pour réaliser des transformations correctement.

Les méthodes doivent être réalisables par superVP, car elles sont destinées à y être intégrées.
Le cadre de travail se limite aux sons monophoniques (une seule source sonore), et harmoniques.

## Démarche générale

Les méthodes de transformation se servent de SuprVP pour l'analyse et la synthèse du signal modifié. Mon travail consiste à trouver les paramètres de la modification en question (facteur de dilatation du time stretch, position des segments à recopier, facteur de transposition, etc.), et ceci de façon automatique. Voici comment je procède:

- Je fais l'**analyse** avec SuperVp, depuis Matlab. Il peut s'agir d'une analyse F0, d'énergie locale, … et je récupère les paramètres de l'analyse dans Matlab.
- J'écris une **fonction Matlab**, avec en entrée les paramètres issus de l'analyse (présents dans un fichier SDIF), et en sortie les paramètres de transformation calculées par ma fonction. (Ex : facteur de transposition à appliquer au signal pour annuler un vibrato)
- Ces paramètres sont retranscrits dans des **fichiers** qui sont appelés lors de la commande SuperVP.
- Je **synthétise** le signal modifié avec SuperVP.

## Plan de travail effectué et comparaison avec le prévisionnel

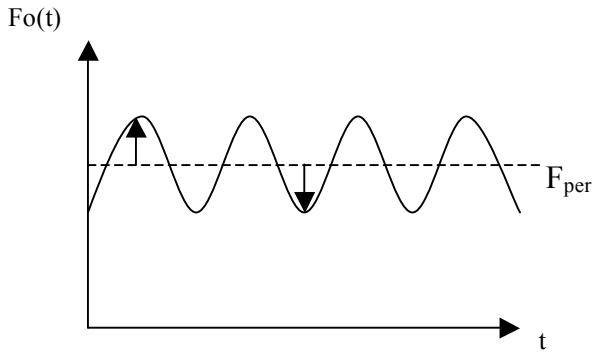
Le plan de travail est divisé selon les deux types de transformations envisagés : modifications par transposition, et modifications temporelles. (cf Diagramme de Gantt).
Dans le planning prévisionnel, nous avions prévu de traiter toutes les transformations liées au vibrato, et ensuite toutes celles liées au tremolo. Mais il s'avère que le tremolo est beaucoup moins fréquent que le vibrato, et faute d'exemples, et de temps, je l'ai surtout étudié dans le cadre du vibrato (tremolo induit).

## 3.1 Transposition des modulations

### Introduction

La transposition permet d'agir sur l'extension des modulations. Après estimation du F0 moyen pour le vibrato, on pourra soit annuler, soit amplifier la modulation. La transposition s'applique pour le vibrato, et la modification du gain pour le tremolo.

Dans le cas du vibrato, notons $F_0$ la fréquence locale, et $F_{per}$ la fréquence perçue (ie la fréquence centrale du vibrato) :




*Le facteur de transposition*

L'oreille identifie des intervalles, alors que la physique identifie des rapports de fréquences.

Par exemple, une octave entre deux notes correspond à un rapport de fréquence de deux.

Si on prend trois sons *A, B ,C* tels que les intervalles relatifs *B-A* et *C-B* sont perçus comme identiques, on *a* $h_b$- $h_{a=}$ $h_c$-$h_b$ (h désigne la hauteur), et $\frac{Fa}{Fb} = \frac{Fb}{Fc}$.

Afin de pouvoir quantifier cette notion de hauteur, on utilise la fonction logarithme qui permet de passer de $\frac{Fb}{Fa} = \frac{Fc}{Fa}$ à $\log(Fb) - \log(Fa) = \log(Fc) - \log(Fb)$. Cette dernière équation est équivalente à $h_b$- $h_{a=}$ $h_c$-$h_{b,}$, et satisfait donc à la notion de hauteur.

La conséquence pour la transposition est que pour accentuer un vibrato d'un facteur 2, on souhaite doubler au sens de la perception l'écart *Fo(t)-Fperc*, et mathématiquement cela

correspond à multiplier *Fo(t)* par le rapport de fréquence $\frac{F_0(t)}{F_{perc}}$. Pour annuler un vibrato, on fait l'opération inverse, à savoir multiplier Fo(t) par $\frac{F_{per}}{F_0(t)}$.

Le facteur de transposition est donc défini par

$$\alpha = \left(\frac{F_{perc}(t)}{F_0(t)}\right)^K$$

où *K* dépend de l'opération choisie : il est négatif pour amplifier un vibrato, il est positif pour abaisser un vibrato, il vaut 0 lorsqu'on ne fait rien.

L'unité utilisée en musique pour quantifier une transposition est le cent. Sa définition provient de la décomposition en 12 demi-tons de la gamme tempérée occidentale.

***facteur en cent = 1200*log(rapport en Hz)***

(Dans une octave, il y a 12 demi-tons. Un rapport d'une octave vaut 2 et correspond à 1200 cents, un rapport d'un demi-ton vaut $2^{\frac{1}{12}}$ et correspond à 100 cents )

*Utilisation avec SuperVP*

La transposition s'effectue avec la commande –trans, et avec en argument un fichier, <trans> qui contient les informations pour la synthèse du signal transposé (les facteurs de transposition à appliquer pour chaque instant) .C'est ce fichier qui devra être modifié en sortie de la fonction matlab.

L'option –transke permet d'effectuer une transposition avec préservation de l'enveloppe spectrale : c'est celle-ci qui sera utilisée, notamment pour tenir compte du tremolo induit par le résonateur lorsque la source produit un vibrato. (cf 3.1.2)

Dans le cas du tremolo, on utilise la modification locale du gain, disponible via la fonction « breakpoint for gain » : -ggain <bpfgain> . L'estimation d'amplitude se fait par convolution entre le signal et une fenêtre de Hamming, dont la durée est plus grande que l'inverse de F0 local, et plus petite que la période d'un tremolo.

### 3.1.1 Fonctions développées

***Extraction du vibrato : « vibextract.m »***

Principe

Pour trouver le facteur de transposition, défini par $\left(\frac{F_0(t)}{F_{perc}}\right)^K$, il faut connaître $F_{perc}(t)$.
Cette fréquence perçue peut être vue comme la composante continue du signal *F0(t)*.

La première étape est d'importer le résulat de l'analyse « f0 » dans Matlab ,à partir du fichier « son.f0.sdif ».
Une fois que l'on dispose de ce signal, on va chercher sa composante continue, en faisant l'hypothèse que la fréquence fondamentale du vibrato est de 6Hz (un vibrato est toujours compris dans la bande [5-7HZ]). L'extraction de la composante se fait avec un filtrage RIF défini comme ceci :

- on multiplie le signal *f0(n)* par une fenêtre de hanning normalisée de taille N : *h= hanning(N)/sum(hanning(N))* *(cf fig. 13)*

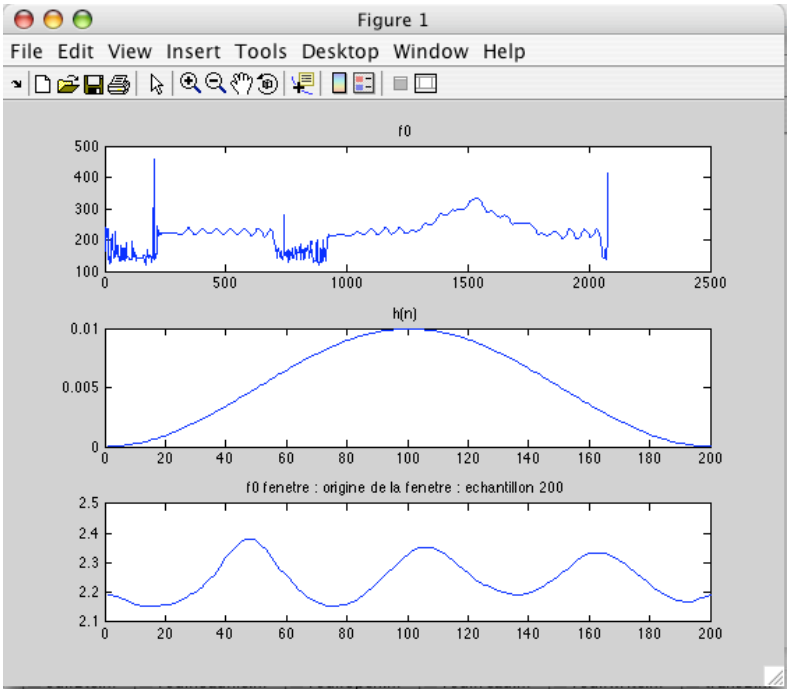



***figure 13 : f0(n) et une f0(n) fenétré avec h(n).***

- Si on regarde la FFT du signal qui résulte de cette opération, on voit les différents pics du vibrato : (fig 14)

-

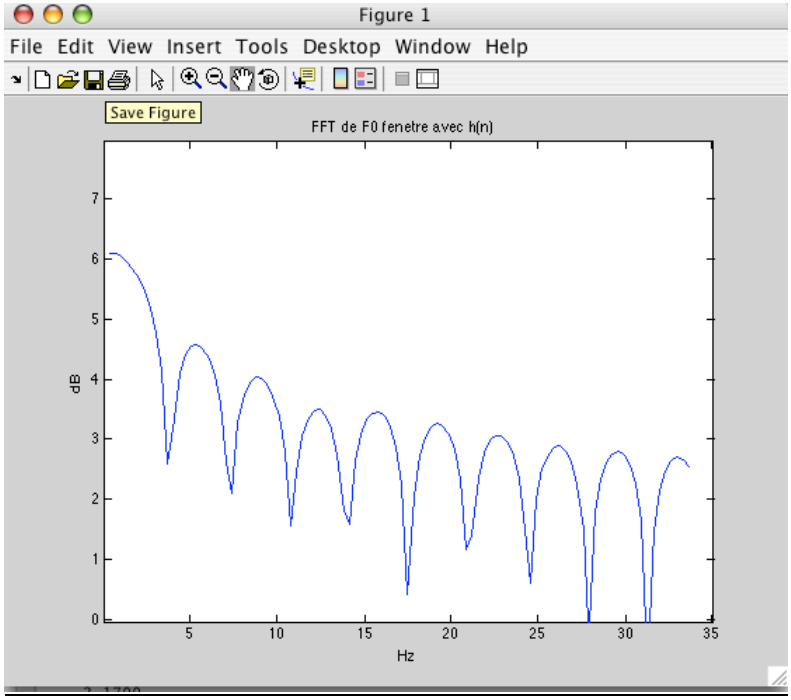



***fig 14 : fft de f0 fenetré(on voit bien un pic autour de 6Hz)***

- La fréquence fondamentale est l'amplitude du pic à la valeur k=0, qui est donnée par : $F0(k=0) = \sum_{n=0}^{N-1} f0(n)h(n)e^{-j2\pi n*0} = \sum_{n=0}^{N-1} f0(n)h(n)$. où *F0* représente la DFT de *f0(n)*.
- On avance cette fenêtre d'un certain pas d'avancement I, avec recouvrement pour lisser le résultat, et on calcule la nouvelle valeur de DC(t).

En interpolant les valeurs de DC obtenues (on a en effet une valeur tous les I instants, ce qui représente *longueur(f0)/I* valeurs), on obtient l'évolution de la composante continue du signal *DC_out(t)* (cf fig.16).

La longueur du filtre, *N*, détermine sa résolution fréquentielle : plus *N* est grand, mieux la composante continue calculée avec la formule sera estimée (car alors le pic en 0 ne va pas aller interférer avec le pic de la fondamentale du vibrato). Mais une fenêtre trop longue n'aura pas une bonne résolution temporelle, on retrouve le compromis de l'analyse temps-fréquence.

L'intérêt de ce filtrage est que l'on connaît très bien ses carcatéristiques, et qu'on va pouvoir l'appliquer sur des signaux *f0* venant de sons différents, en adaptant simplement sa taille *N*.

Description de la fonction :

La fonction permet de choisir les différents paramètres du filtrage :

- « fvib » : la valeur du fondamental du vibrato : notre hypothèse est de prendre fvib = 6Hz.
- « WS » pour « window size » : paramètre qui permet de choisir la longueur de la fenêtre, en nombre de périodes du vibrato (1/fvib) . On choisit WS=3.5.
- Wstep : pas d'avancement de la fenêtre.

On récupère en sortie le signal DC :

```
[DC_out]= vibextract2(f0,fvib,OL,WStep,WS)
```

***Fonction de création du fichier de transposition : « trans.m »***

Principe :

Une fois que l'on a fait l'extraction de *DC*, on peut facilement calculer le coefficient de transposition à appliquer à *f0(t)*, en prenant à tous les instants la valeur *DC(n)/f0(n)*.
Il faut encore adapter ce coefficient de transposition à l'opération que l'on souhaite réaliser. La convention adoptée est la suivante : le paramètre qu'on rentre est l' « état » du vibrato après transposition. 0 doit annuler le vibrato, 1 doit le laisser intact, 2 doit l'accentuer ; -1 doit donner le vibrato opposé en phase,etc.
Par rapport au coefficient K de l'introduction de la partie 3.1, c'est le contraire. Le nouveau coefficient en exposant est maintenant K'=1-K

$$\alpha = \left( \frac{F_{perc}(t)}{F_0(t)} \right)^{1-K}$$

Il reste à convertir ce rapport en cents pour que superVP puisse l'utiliser, et on sauvegarde dans un fichier (en ASCII), la matrice faite des deux colonnes trames temporelles et valeur du coefficient en cent à appliquer pour chaque trame.

Description de la fonction :

```
[mat] = trans(f0_sdif_filename , DC , alpha)
```

La fonction prend en paramètre le fichier SDIF de l'analyse f0, la valeur de DC calculée avec la fonction vibextract, et alpha, le coefficient de transposition. La matrice résultat est disponible en sortie.

***La fonction trans_svp : appel de superVP depuis matlab***

Principe :

Elle regroupe les deux fonctions précédentes, avec en plus la possibilité de ne traiter que certains segments du signal, choisis préalablement par l'utilisateur dans AudioSculpt, et qui sont stockés dans un fichier de marqueurs SDIF.
L'avantage est qu'en un seul appel, elle combine : extraction de DC, création du fichier de transposition , et appel à SuperVP depuis Matlab, qui réalise alors la transposition (on choisit l'option –ke).
On obtient en fin d'exécution le fichier son résultat, annoté avec les paramètres de la transposition et du filtrage utilisé (facteur de transposition, taille du filtre).

Utilisation :

```
trans_svp(f0_sdif_filename,sndname,outname,Fvib,Wstep,WS,alpha
,typename,mrk_filename)
```

sndname est le fichier son à transposer, outname est le nom du fichier de sortie (auquel seront accolés les valeurs de différents paramètres) ; typename est pour l'option 'trans' ou 'transke' ; et enfin mrk_filename est le fichier de marqueurs qui contient les segments à traiter.

### 3.1.2 Le cas de la sinusoïde pure

Ces différentes fonctions ont d'abord été testées pour le cas de sinusoïdes pures modulées en fréquence :

$$x\_fm(t) = \cos(2\pi F0t + \beta \frac{F0}{fm} \cos(2\pi fmt)),$$

où $\beta$ est l'extension de la FM, F0 la fréquence de la sinusoïde, et fm la fréquence de modulation.

$\beta$ est donné en cent dans une fonction 'vib.m' qui génère cette sinusoïde :

$$\beta = 2^{\frac{extent\_en\_cents}{1200}-1}$$

On teste la transposition « alpha = 0 » avec une sinusoïde à 400 Hz ( un La4 = 440 Hz), une Fm de 6 Hz, et on choisit un extent de 200 cents pour le cas le plus critique du vibrato.( 200 cents représente un ton.). Ci – dessous, cette sinusoïde et son analyse f0 dans Audiosculpt.

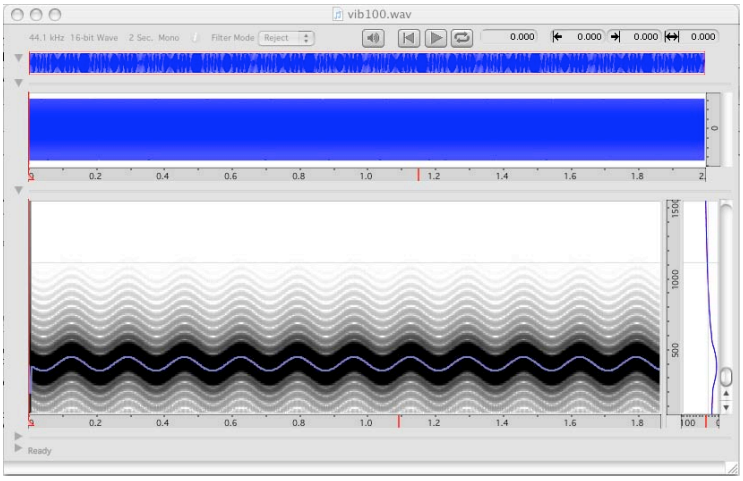



***fig.15 : une sinusoide pure modulée en fréquence. F0 = 100 Hz, Fm =6Hz, $\beta$=200cents***

On applique les deux fonctions *vibextract* et *trans* au signal f0 qui provient du fichier 'wav100.f0.sdif'. On peut voir les différents signaux résultats sur la figure 13 :
Les modulations sur le signal DC s'expliquent par le fenêtrage. Le son étant parfaitement sinusoïdal, il y a une modulation régulière dans l'estimation de DC, la valeur de la somme faisant des variations selon la phase de la sinusoïde. Elle reste très légère (1% d'erreur).

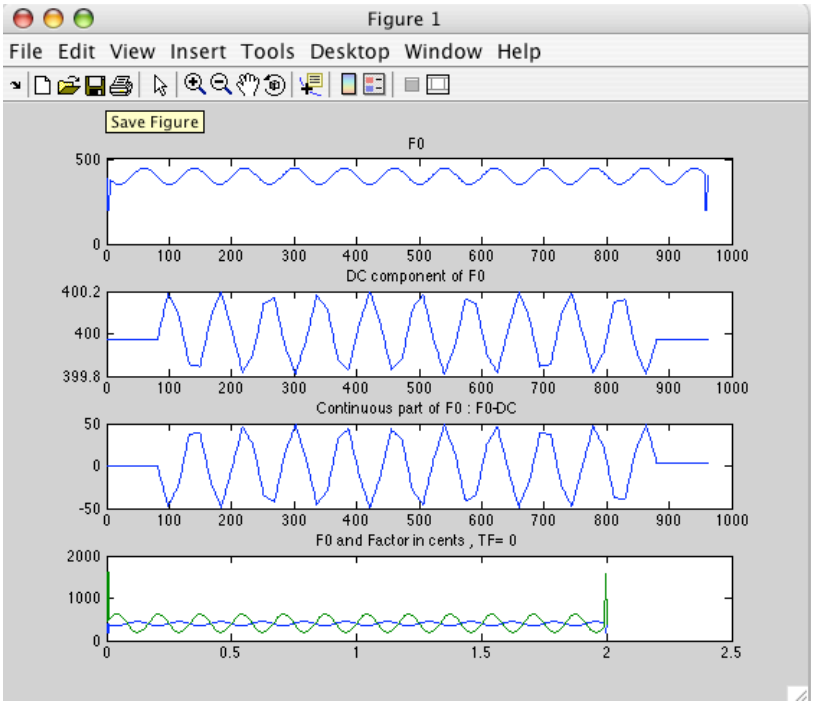



*fig 16 : résultats de 'vibextract' et 'trans'*

On peut voir sur la figure 15 qu'après transposition (fonction *trans_svp*), le signal est sans FM . On obtient ce qu'on souhaite.

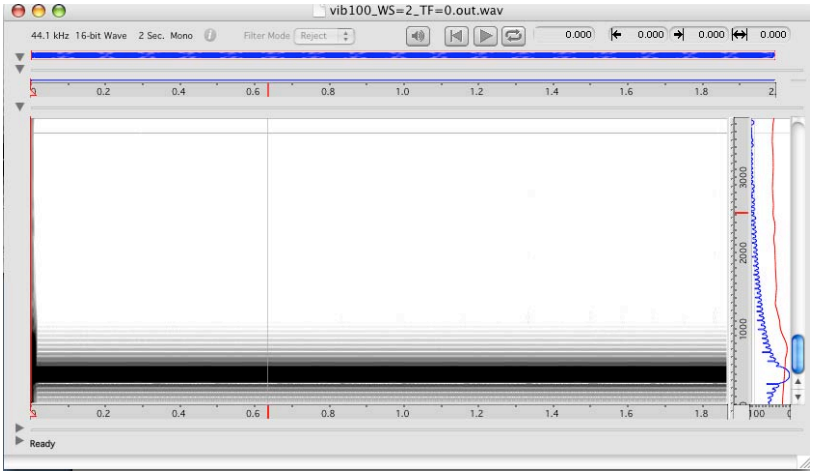



*fig 17. Disparition de la FM.*

### 3.1.3 Le cas des sons réels

On se place maintenant dans le cas des sons rééls. Nous avons utilisé des sons monophoniques et harmoniques, de type instrument ou voix.

#### *A) Le problème du trémolo induit*

***Rappel du principe***

Une prémière étude, qui avait été présentée dans le rapport de mars, mettait en évidence le trémolo induit par un vibrato. Il s'agissait de modéliser le comportement d'un signal modulé en fréquence lorsqu'il traverse le conduit vocal.

Le conduit vocal est modélisé par filtre est un filtre IIR de Butterworth passe-bande d'ordre 2, de fréquences de coupures f1=1500Hz et f2=1800Hz.(figure 18)

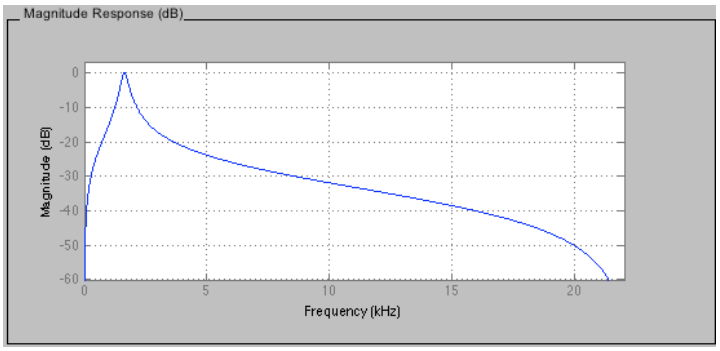



***figure 18 : réponse en fréquence du filtre IIR.***

Le signal de départ x(t) est une sinosoîde de fréquence la fréquence de résonance du filtre (F0 = 1636 Hz) et d'amplitude 1.

$$x\_fm(t) = \cos(2\pi F0t + \beta \frac{F0}{fm} \cos(2\pi fmt)) \qquad \text{(spectrogramme fig.19)}$$

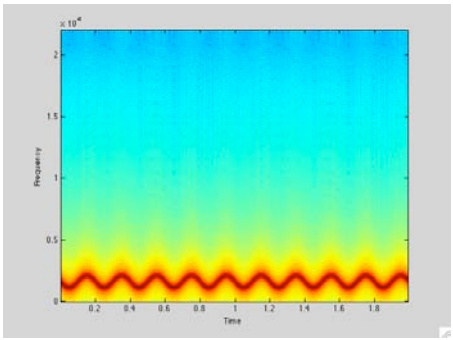


*figure 19 : spectrogramme*

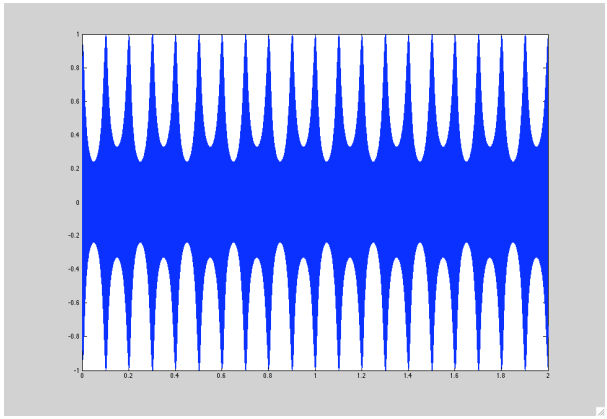


*figure 20 : sortie du filtre*

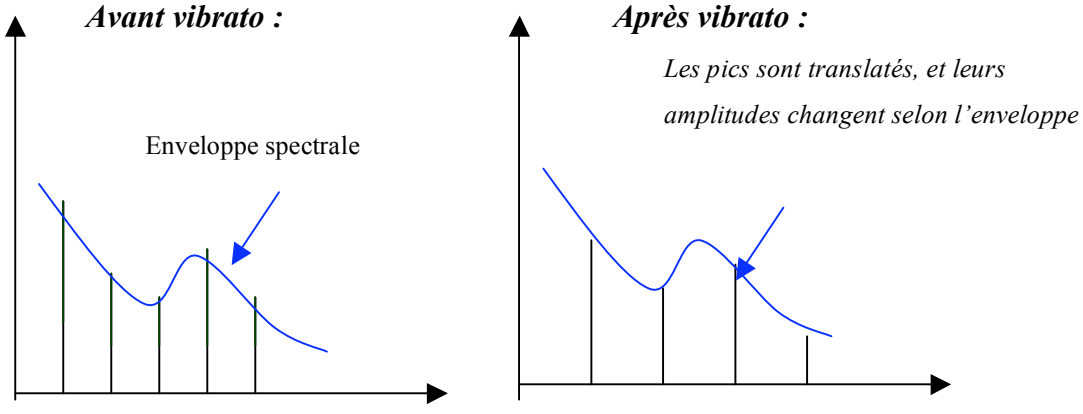
Signal d'origine → **modulation _FM.m** → signal modulé en fréquence(fig.1) → **filtre IIR ordre 2** (fig.2) → signal modulé en fréquence et en amplitude (fig.3)

On observe une AM sur le signal de sortie. En effet, les aller-retour à droite et à gauche de la fréquence de résonance induisent un filtrage qui change en conséquence au cours du temps, et qui est à la base de l'AM induite (ie à la base d'un tremolo dans le contexte du conduit vocal).

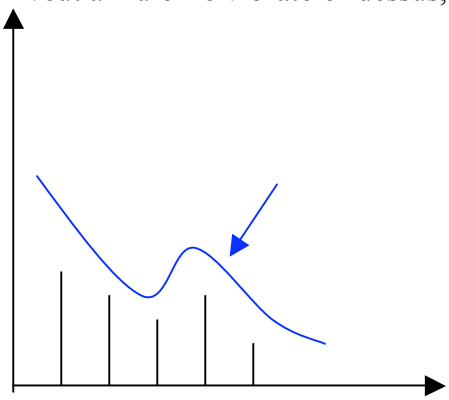
### *L'option « -transke ».*

Lorsqu'on transpose le signal pour annuler le vibrato, on aimerait annuler aussi le tremolo induit.

On a vu qu'un vibrato induisait un tremolo, visible par un changement des amplitudes des pics lorsque ceux-ci bougent sous l'enveloppe spectrale :



Si on veut annuler le vibrato ci-dessus, on va transposer tous les pics vers leur valeur initiale :



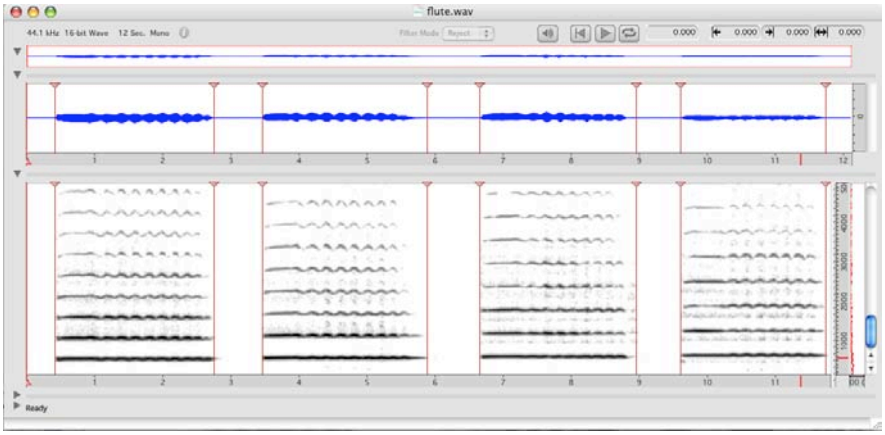
On voit qu'il y a un problème avec les amplitudes des pics, qui ne sont plus celles d'origine : Ce sont celles du signal avec vibrato, car on a juste translaté les pics, et pas modifié leur amplitude.

Pour éviter ce problème, il faudrait réadapter l'amplitude des pics à l'enveloppe spectrale, pour obtenir le signal de départ : on aurait ainsi enlevé le vibrato, et le trémolo induit.

Ceci est possible dans le vocodeur de phase grâce à l'option « transke », qui transpose le signal avec préservation de l'enveloppe spectrale.

***Limites du modèle source-résonnateur***

Le problème avec ce modèle est qu'on considère que l'enveloppe spectrale est constante au cours du temps. Or, il existe une corrélation entre l'enveloppe d'un instrument, et le geste du musicien (coup d'archet pour un instrument à cordes,façon de souffler dans une flûte, ouverture de la bouche pour le chant,etc.). L'étude de cette corrélation n'est pas abordée dans le cadre du stage. Mais le fait que l'enveloppe du son change au cours du temps est un point essentiel à prendre en compte si on veut éviter les modulations d'amplitude sur chaque partiel du son. Regardons l'enveloppe spectrale d'un son de flûte sur lequel il y a un vibrato (fig 21) :




En rouge, les segments à traiter, stockés dans un fichier SDIF.

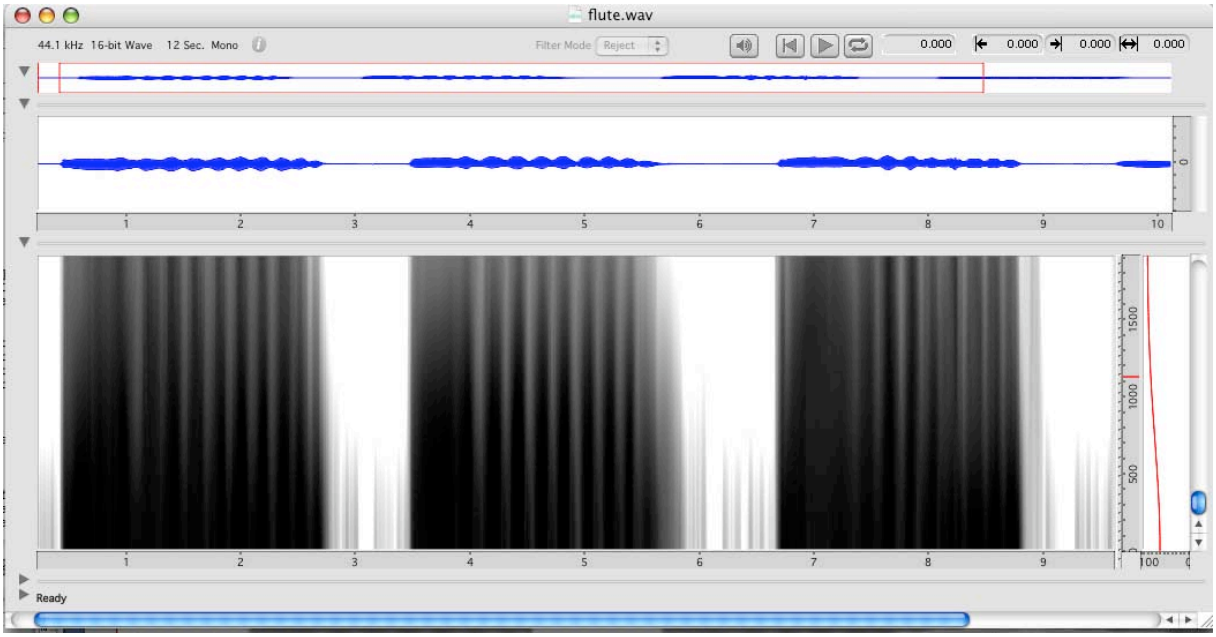



*fig 21 : enveloppe spectrale (analyse « true enveloppe » ) : on voit qu'elle change au cours du temps.*

La variation régulière de l'enveloppe spectrale s'explique par la variation régulière du geste instrumental qui réalise un vibrato. La modulation d'amplitude que cette variation induit est ajoutée au tremolo induit : ce sont deux choses distinctes. C'est en ceci qu'elle représente un nouveau problème. Une solution peut être d'estimer l'énergie et d'appliquer un gain dynamique comme si on traitait un tremolo pur ; seulement le problème va être que les différents partiels ont un trémolo décalé en phase. Mais choisir le deuxième partiel et y appliquer cette opération devrait réduire la modulation d'amplitude par exemple. (Ces méthodes n'ont pas encore été testées durant le stage).

Observons le résultat de la transposition avec préservation d'enveloppe sur ce son de flûte, avec N=2*période_du_vibrato, et $\alpha$=0. (fonction trans_svp) :

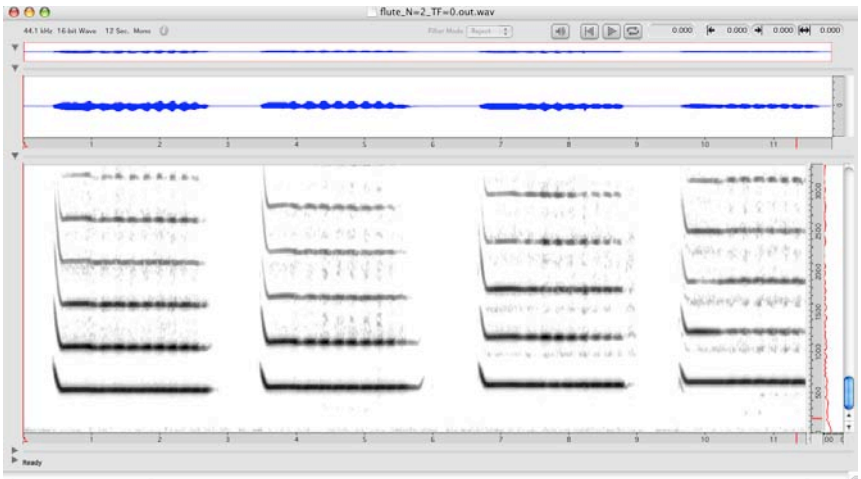



*fig 22 : son de flûte transposé*

Le vibrato est bien supprimé, mais on aperçoit les modulations d'amplitude sur le deuxième par exemple, liée au changement de l'enveloppe :

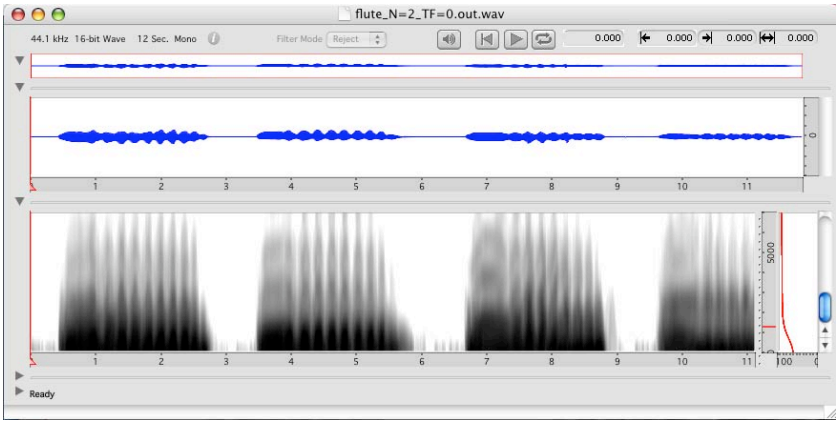
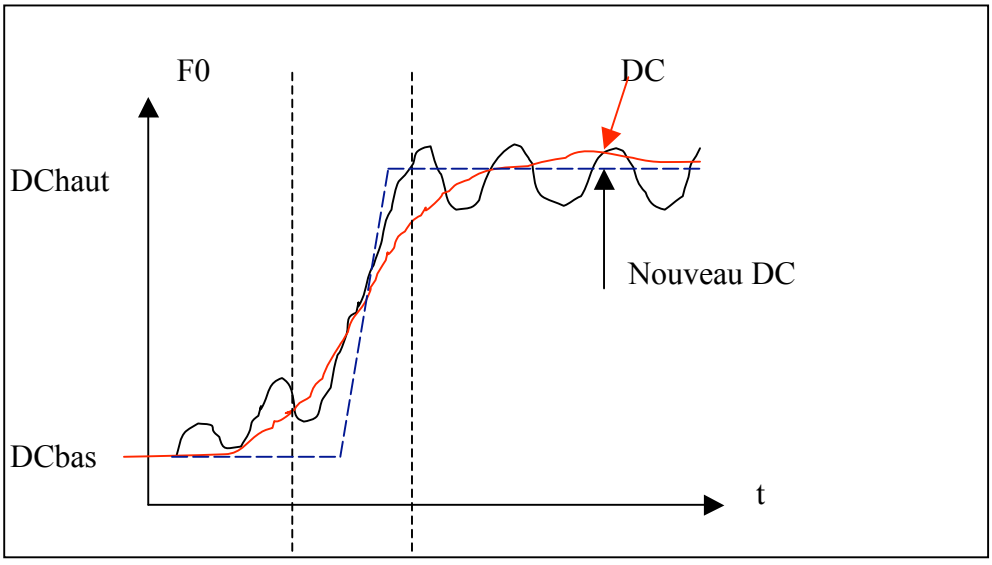



***fig 23 : analyse « true enveloppe » sur le son de flûte transposé.***

### *B) Le traitement des bords des vibratos*

Un autre problème restant est celui des bords des modulations, qui représentent une transition souvent brusque dans le signal. On voit ceci sur la figure 23 : cela provient du filtrage fait pour obtenir DC, qui lisse le signal sur les transisitions. Plus *N* va être élevé, plus cette zone sera grande. Cela va se répercuter sur le facteur de transposition, calculé en fonction de DC. C'est problématique pour un vibrato qui ne contient que 2 périodes par exemple.

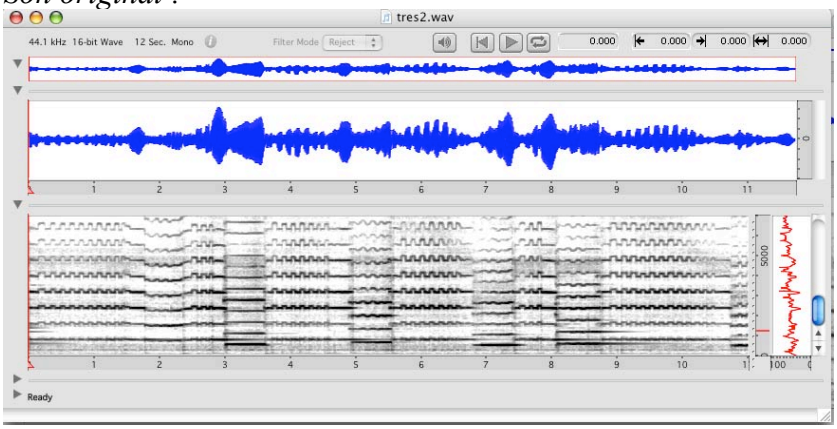
Pour pallier à ce problème, on il faut pouvoir détecter les zones de transitions, et changer le signal DC pour qu'il soit plus raide dans ces zones. Cette étude est encore en cours, mais l'idée est de dériver *f0* pour détecter les transitions, de définir une zone autour de la transition détectée (choisie avec un seuil), et de connecter les valeurs de DCbas et DChaut dans cette zone :



### *B) Différents résultats :*

*B1.Son de violon :*

*Son original :*




*Son transposé avec $\alpha$=0, N=2.*

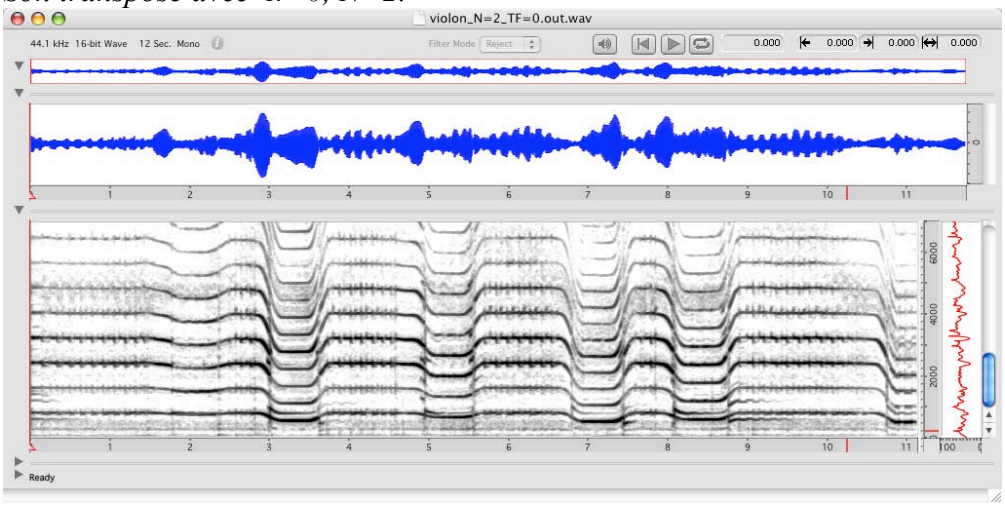
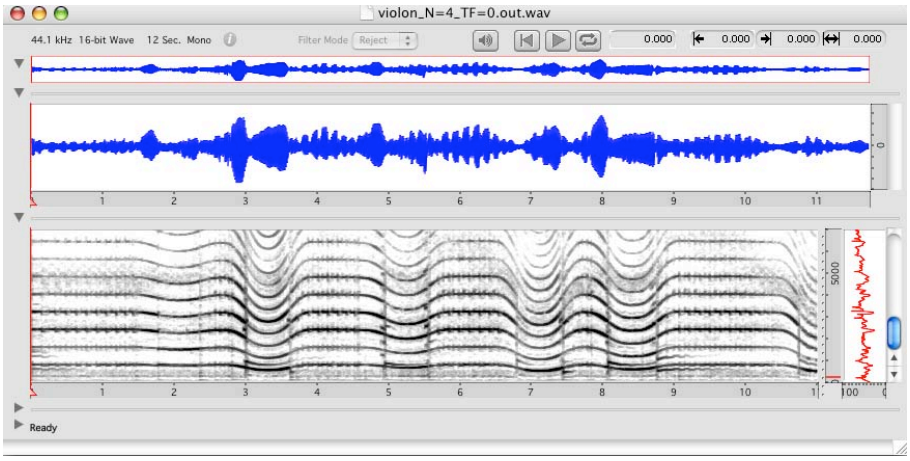


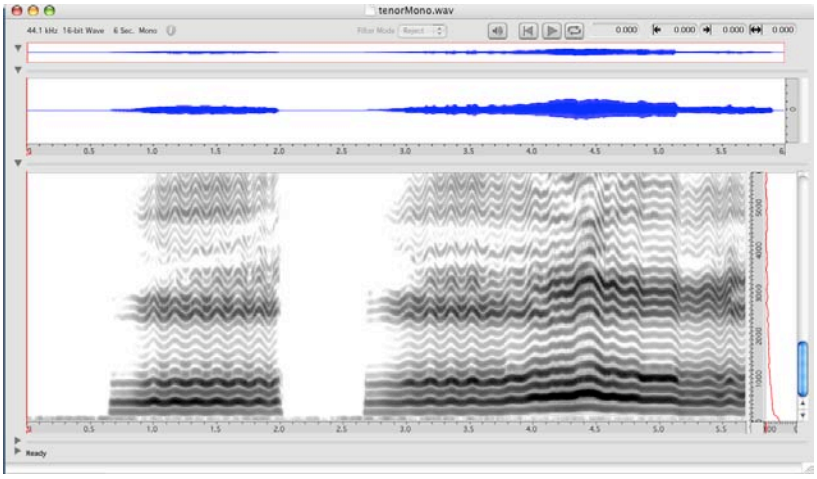

***fig 24 : un son de violon sur lequel on a annulé le vibrato.***

On voit bien sur cet exemple le problème des transitions, car le son présente beaucoup de sauts de notes. Avec N=4, c'est encore plus marqué :




*B2.Son de voix :*

Son original :




Son transposé avec $\alpha=0$ *et* $N=2$ *:*

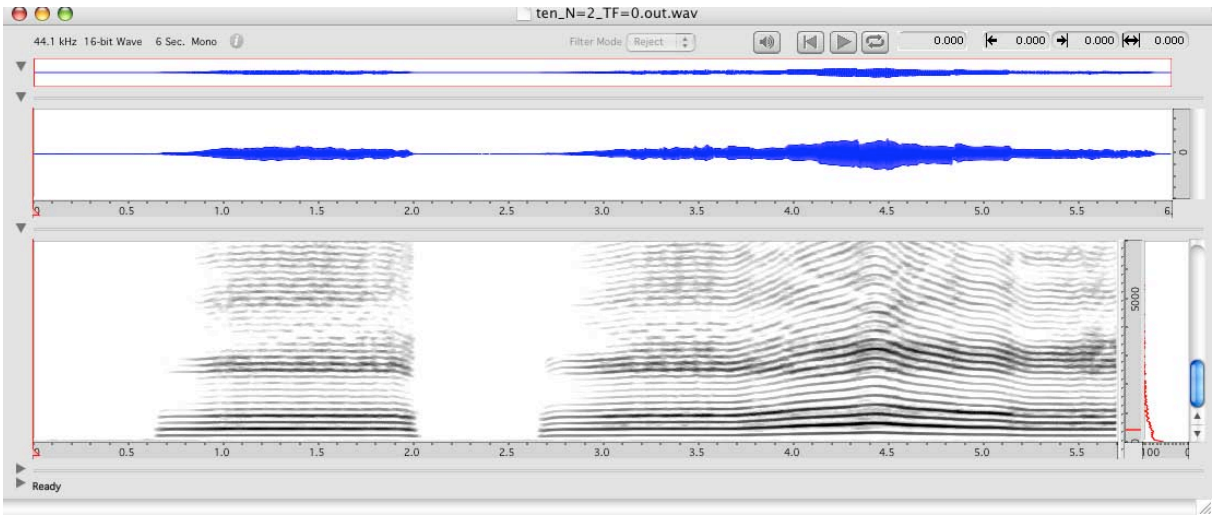
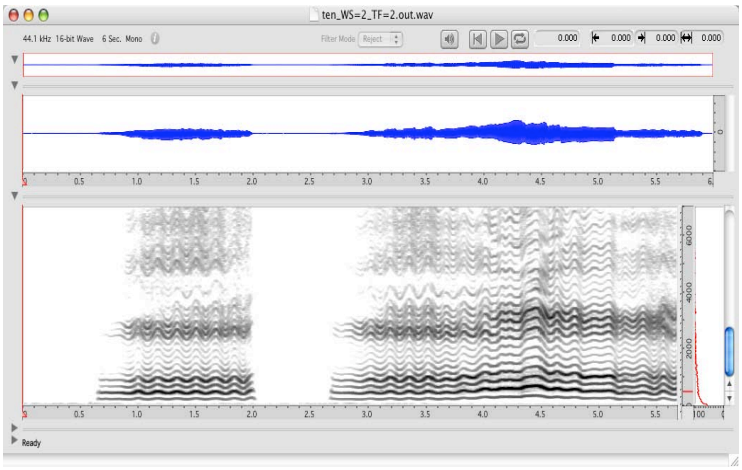



***fig 24 : vibrato sur la voix annulé***

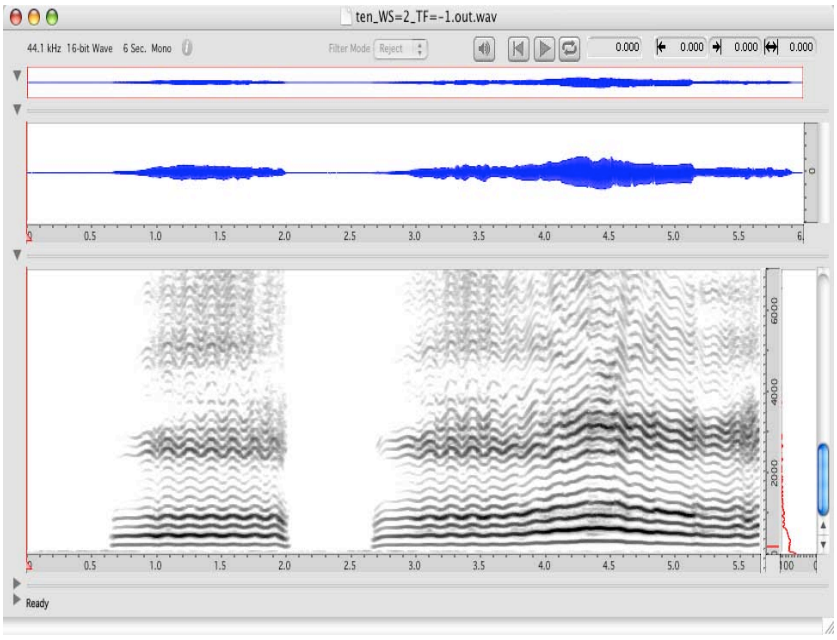
Ici le son ne présente pas de transition rapide, et le problème des bords n'apparaît presque pas. Par contre, on voit bien sur les partiels supérieurs la modulation d'amplitude résiduelle due au changement de l'enveloppe spectrale au cours du temps.

On peut voir que d'autres valeurs de $\alpha$ donnent des résultats satisfaisants :

Son avec amplification du vibrato : $\alpha = 2$ *et* $N=2$




Son avec vibrato en oppostion de phase : $\alpha = -1$ *et* $N=2$



## 3.2 Modifications temporelles des modulations

### 3.2.1 Méthode

Les outils de SuperVP utilisés sont le « *TimeStretch* », les *sauts temporels*, et le « *Reverse/Repeat* ».

***A) Le TimeStretch***

Il fonctionne selon le principe détaillé dans l'étude du vocodeur de phase : les trames analysées sont déplacées selon le facteur de dilatation/compression souhaité.

Il s'utilise via la commande :

**-D[coefficient] ou -D<filename>** dans le cas d'un time stretch dynamique (qui varie en fonction du temps).

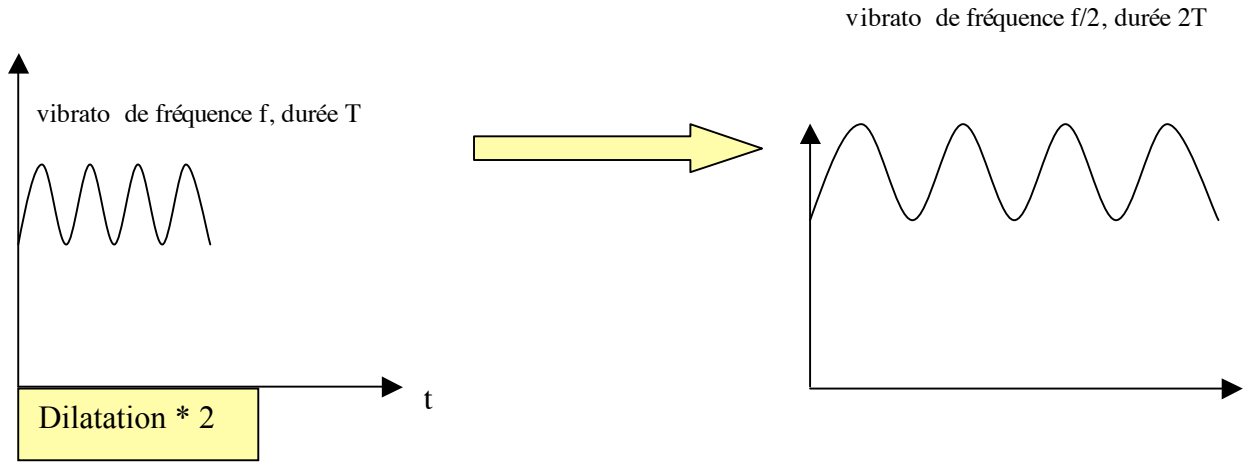
Ex : `supervp -Z -S"/Users/snd/tenorMono.aif" -Afft -Np0 -M0.03406s -oversamp 8 -Wblackman -D"$USERHOME/Temp/`**`timefile`**`" "/Users/maller/AS/Sounds/tenorMono.out 2.aif"`

*Explication des différents champs :*

-**Z** : indique qu'on réalise une synthèse ( IFFT et overlapp/add)
-**S** : fichier source
-**A** fft : analyse de type FFT
-**Np** 0 : taille de la fft = prochaine puissance de 2 de ( $2^0$ * taille de la fenêtre).
-**M** : taille de la fenêtre d'analyse
-**oversamp** : taux de recouvrement des fenêtres.
-**W** : type de fenêtre utilisée.
-**D** : time strectch utilisant le fichier <timefile>
-"/Users/maller/AS/Sounds/tenorMono.out.aif" : fichier audio en sortie

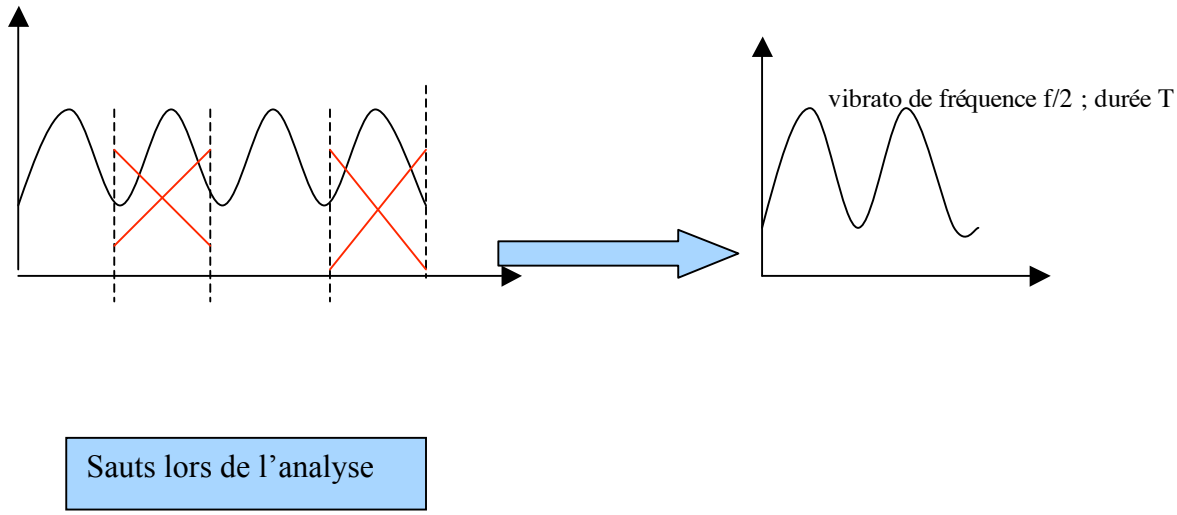
Dans notre cadre, le time stretch est utilisé pour augmenter ou réduire la fréquence d'un vibrato (ou éventuellement d'un tremolo).
Si on étire une période de vibrato d'un facteur 2 par exemple, on ne change pas la valeur du pitch, mais le vibrato est deux fois plus lent :

Spectrogramme




***B) Les sauts temporels***

Si maintenant on veut que la durée du vibrato soit conservée, il faut « enlever » des périodes du nouveau vibrato. On peut par exemple en enlever une sur deux lorsque le facteur de dilatation vaut deux. (on pourrait aussi enlever les N/2 premières ou dernières périodes des N périodes totales du vibrato)




La cohérence des phases dans le cas de tels sauts est expliquée en 2.3 ; elle est basée sur une estimation de la fréquence locale à l'endroit du saut.

Les positions des sauts sont spécifiées dans un fichier <posfile> ,appelé avec la commande – I par superVP,
Ce fichier <posfile> est constitué de labels « u » et « c » suivis d'une position.

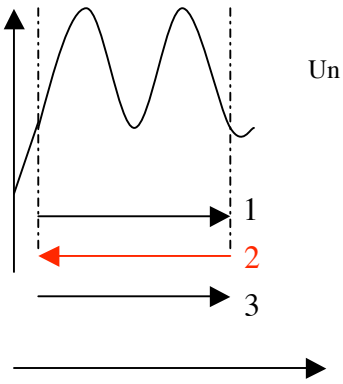
« c *pos*» indique un repositionnement de la trame d'analyse. Le pointeur de lecture est déplacé sans créer de trame d'analyse. C'est ce qui permet de sauter dans le fichier sans ajouter d'échantillons dans le fichier de sortie.

« u *pos*» indique une position cible : une séquence de trames dont la dernière est centrée à l'échantillon *pos* est crée.

On peut ainsi faire l'analyse du fichier sonore comme on le souhaite.

***C) Le reverse-repeat***

C'est une fonctionnalité de SuperVp qui permet de faire un (ou plusieurs) aller-retour sur un segment du signal :




Un reverse-repeat *Forth and back = 3*

Elle s'utilise de la même façon que les « sauts temporels », à savoir avec le fichier <posfile>.
Mais ici, l'analyse est possible vers arrière, et le fichier est ensuite lu à l'envers.

On se sert du reverse-repeat pour recopier des périodes des modulations, et ainsi allonger un vibrato ou un tremolo sans en changer la fréquence. Suivant le facteur de dilatation, il faudra choisir le nombre de périodes à traiter et le nombre de reverse-repeat à faire sur chaque période.

### 3.2.1 Exemples

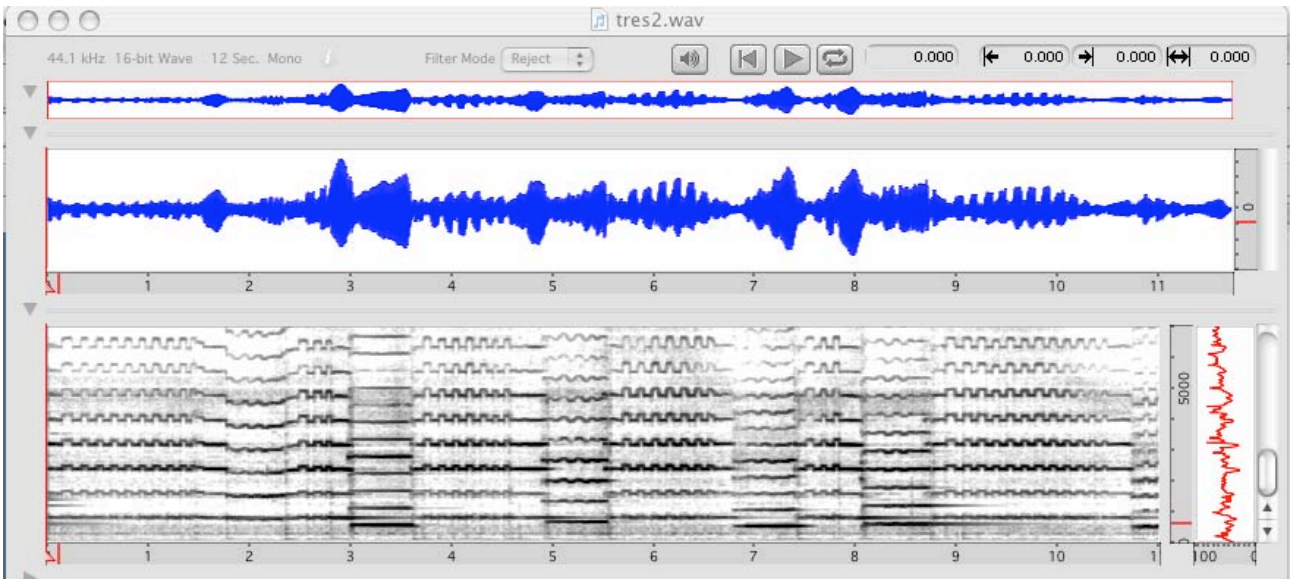
*Exemple de fichier posfile pour un reverse-repeat :*

```
c 0
u 162812
u 169426  // → 1
u 162812  // ← 2
u 169426  // → 3
u 175530
u 169426
u 175530
u 181562
u 175530
u 181562
u 518836
```

Le fichier à modifier automatiquement avec les fonctions matlab est <posfile>.

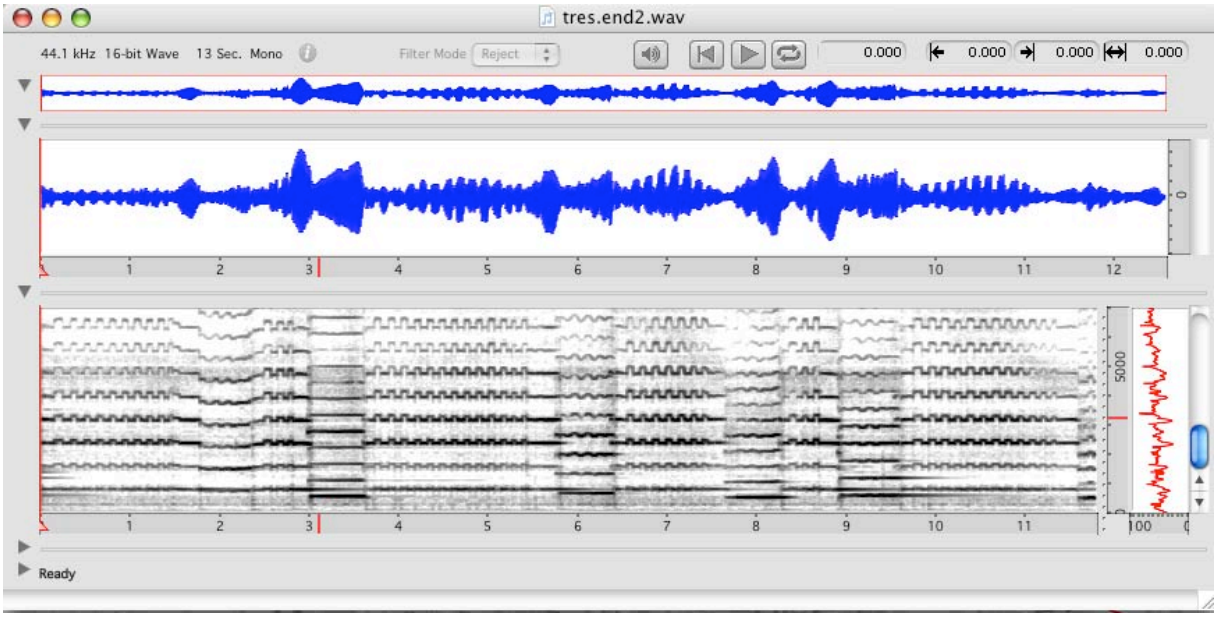
*Exemples réalisés sur le son de violon*

*son original :*



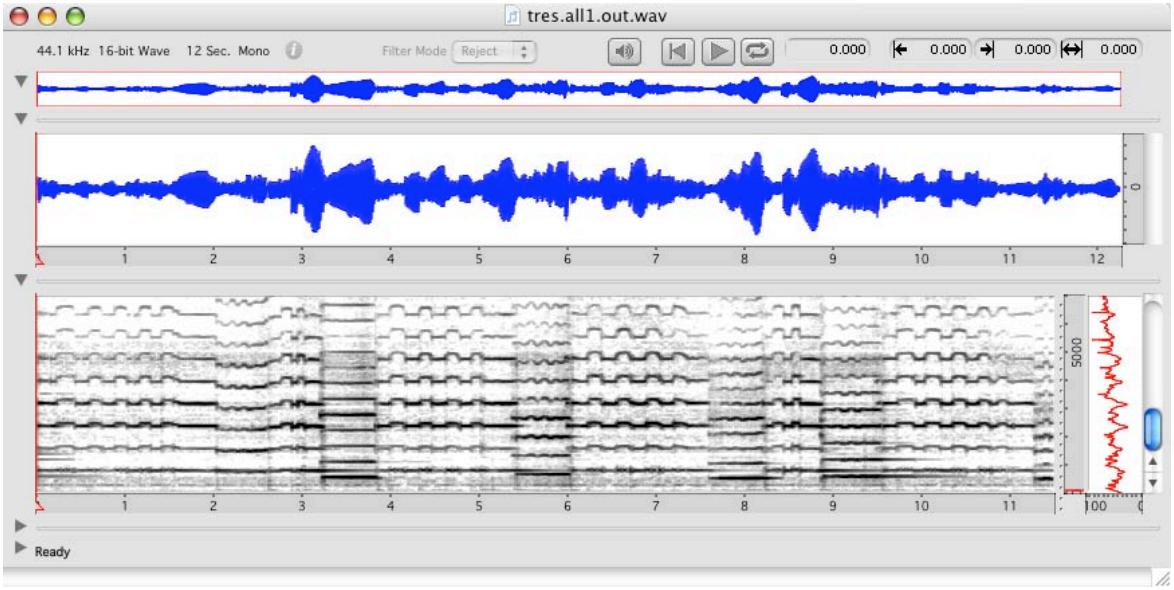

On applique sur ce son deux transformations, en éditant les fichiers <posfile> manuellement.

La première est l'allongement du troisième vibrato , sans en changer sa fréquence :




On voit sur le spectrogramme que le deuxième vibrato est effectivement plus long que sur le son original.

La deuxième transformation est la diminution de la fréquence du vibrato facteur 2, sans en changer la durée :



### 3.2.3 Perspectives

Il s'agira de développer des fonctions permettant la génération automatiques de fichiers de positions, suivant ce que l'on veut faire comme transformation. Les questions pour les transformations temporelles portent entre autres sur

- La bonne estimation de la fréquence du vibrato/tremolo pour sélectionner les bonnes périodes.
- Les positions des marqueurs : doit-on couper le signal lors d'un passage par zéro, d'un maximum, d'un minimum…
- Le choix des segments à recopier/supprimer : adjacents ou alternés.

On n'aura un mauvais résultat quand la naturalité du son ne sera plus respectée. Cette notion étant subjective dans la plupart des cas (on n'a pas de mesure pour quantifier la qualité des transformations sonores), on fera appel à des compositeurs pour l'évaluer.

Nous avons déjà pensé à une méthode pour faciliter la sélection des segments sur la courbe *f0*. Elle consiste à récupérer la fondamentale de *f0* afin d'éliminer les hautes fréquences qui peuvent gêner pour placer les segments.

# Conclusion

On a vu que les transformations des modulations avec le vocodeur de phase peuvent donner des résultats satisfaisants. Si certains défauts doivent encore être supprimés, le travail réalisé est plutôt encourageant.

Il me reste encore plus d'un mois et demi pour résoudre les problèmes du trémolo induit, et concevoir les fonctions de modification temporelles. Je suis plutôt enthousiaste quant à cette fin de stage : j'ai à ce stade une bonne maîtrise des outils utilisés, et une bonne connaissance du sujet.

J'ai passé beaucoup de temps sur l'étude du vocodeur de phase, ce qui m'a ralenti par rapport au prévisionnel, mais c'était selon moi important de bien maîtriser l'outil de base de ce stage.

De plus, d'un point de vue personnel, cette étude m'a apporté des connaissances théoriques enrichissantes, et j'ai le sentiment d'avoir appris beaucoup de nouvelles choses en traitement du signal audio durant ce stage.

# Remerciements

Je tiens à remercier tout particulièrement Axel Roebel qui m'a encadré durant ce stage.

Son expérience et son sens de la pédagogie ont été d'une grande aide. Je dois souligner la patience dont il a souvent fait preuve, restant parfois jusque tard le soir pour m'expliquer des concepts nouveaux pour moi. J'ai appris beaucoup de choses grâce à lui.

Je veux remercier également Laurent Girin, mon tuteur Telecom, pour être venu à l'Ircam en mai, et pour m'avoir corrigé la partie théorique de ce rapport.

Merci à Xavier Rodet, responsable de l'équipe Ananlyse/synthèse, pour avoir donné suite à ma candidature de stage.

Je remercie bien sûr toute l'équipe analyse/synthèse, en particuliers Juan Jose Burred et Carmine Emanuele Cellaqui m'ont accueilli dans leur bureau, et qui en plus de m'avoir aidé volontiers à plusieurs reprises, se sont intéressés à mon travail.

Merci enfin à l'équipe administrative de l'Ircam pour avoir organisé l'accueil de mon stage.

# Bibliographie


[1] **Laroche J.**
*Traitement des signaux Audio-Fréquences* – Department Signal, groupe Acoustique Telecom Paris, Février 1995.

[2] **Griffin D.W., Lim J.S.**
*Signal estimation from modified Short Time Fourier Transform* – IEEE Transactions on Acoustics, Speech, and signal processing, Vol. ASSP 32, n°2, April 1984.

3] **Dolson M.**
*The Phase Vocoder : a tutorial* – Computer audio Research Laboratory, Univerity of California, San Diego. URL : http://www.panix.com/~jens/pvoc-dolson.par, November 17, 2000.

[4] **Dolson M., Laroche E.**
*Improved phase vocoder – Times scale modification of Audio.* – IEEE Transactions on Speech and audio processing, vol. 7, n°3, May 1999.

[5] **Roebel A.**
*Fundamental of discrete Fourier Analysis, Analysis-resynthesis using the short time fourier transform, Signal modification using the STFT* – summer 2006 lecture on analysis, modeling and transformation of audio signals.

[6] **Ircam , Centre Pompidou, Paris.**
*Audiosculpt : Manuel de l'utilisateur (Lithaud A.)*
Audioculpt 2.9

[7] **Verfaille V., Guastavino C., Depalle P.**
*Perceptual Evaluation of Vibrato models* – Proc. Of the Conference on Interdisciplanry Musicology (CIM05) Montréal (Québec) Canada, 10-12/03/2005.

[8] **Gilbert J., Simon L., Terroir J.,**
*Vibrato of saxophones* – Acoustical Society of America, Vol.118, n°4, p. 2469-2655, October 2005.

[9] **Raspaud M., Marchand S., Girin L.**
*A generalized polynomial and sinusoidal model for partial tracking and time stretching* – Proc. Of the 8th Int. Conference on digital Audio Effects (DAFx'05), Madrid, Spain, September 2005.

# Lexique

*Fréquence fondamentale* : fréquence du premier *harmonique* du son considéré, qui correspond à la hauteur perçue d'une note.

*Harmoniques* : ensemble des fréquences qui composent le son, toutes multiples de la fondamentale.

*Sons harmoniques* : désigne un son composé uniquement de fréquences multiples de la fondamentale, par oppositions aux sons inharmoniques, qui peuvent contenir d'autres fréquences.

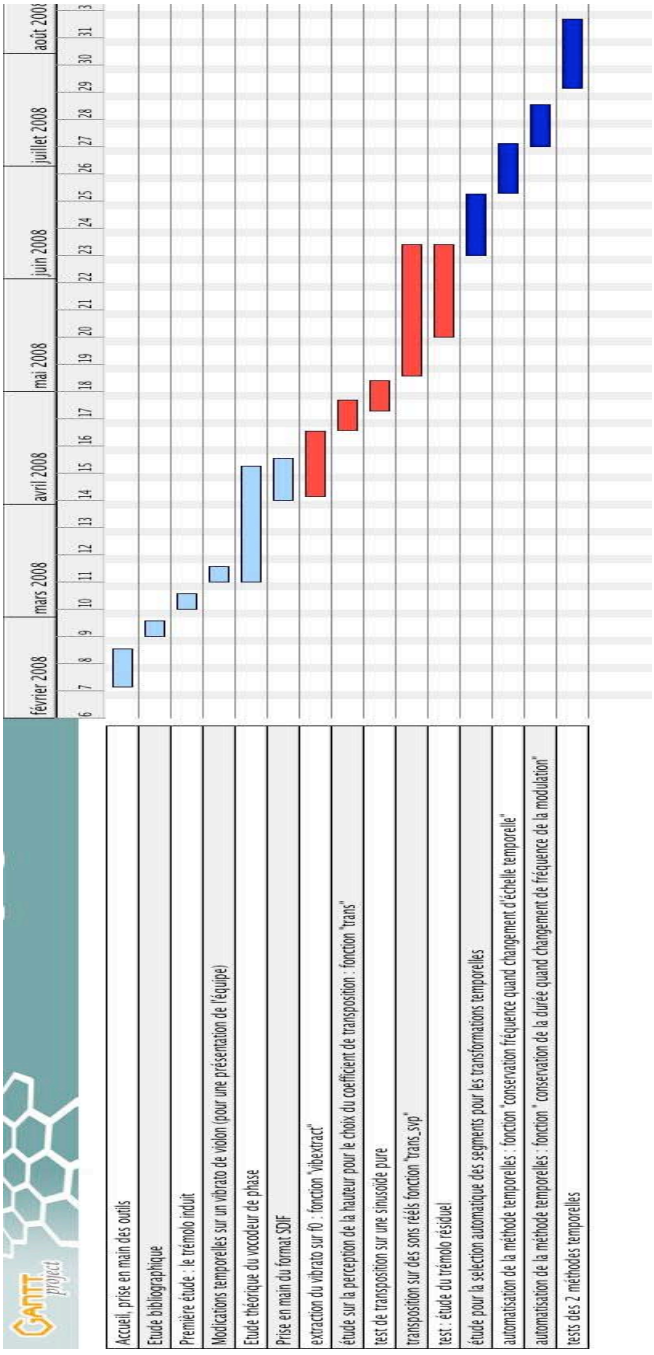
*Spectrogramme* : représentation de l'évolution fréquentielle d'un signal en fonction du temps.

*Timbre* : caractéristique du son d'un instrument, d'une voix, etc. définie comme la répartition des amplitudes de chaque harmonique dans le spectre. Si deux instruments différents jouent une note à la même hauteur, le timbre est ce qui fait qu'on entendra deux sons bien distincts.

# Diagramme de Gantt

Le  plan de travail est divisé selon les deux types de transformations envisagés : modifications par transposition (en rouge sur le Diagramme de Gantt), et modifications temporelles (en bleu sur le diagramme de Gantt).

***Diagramme de Gantt :***



***Évaluation du coût du projet :***

Durée : 24 semaines – 120 jours. Matériel investi : Mac G4, estimé à 4€/jour.

Ingénieur-chercheur impliqué : Axel Roebel. Coût du projet en équivalent ingénieur: 2500 €,

*Coût total : 2980 €.*

# Annexes Techniques

## A1. AudioSculpt

Audiosculpt permet d'afficher et d'analyser un son via une interface graphique. Le son s'affiche alors sous la forme d'un sonagramme , et l'utilisateur peut dessiner les modifications qu'il veut lui appliquer.Les fonctions principales sont l'affichage/édition, l'analyse, les annotations, les traitements sonores (séquenceur de traitement permettant de regrouper des pistes de différents traitements et d'écouter leurs effets avant en temps réel avant de générer le résultat), le filtrage, la compression/expansion, la transposition, etc.

*Caractéristiques techniques :*
Audiosculpt fonctionne avec l'outil d'analyse/synthèse SuperVP pour la plupart des analyses et modifications de son. On peut inspecter les commandes issues de SuperVP, les modifier et contrôler SuperVP par ligne de commande.
Audiosculpt accepte les sons multipistes et de haute qualité (jusqu'à 32-bits integer ou flottants/192 kHz) et utilise les formats SDIF pour les analyses, qui sont échangeables avec les autres logiciels.

*Applications :* composition, design sonore, postproduction, cinéma, multimédia, acoustique, enseignement,etc.

# A2. Les Analyses utilisées

### *Analyse de sonogramme*

Elle permet d'afficher un spectrogramme.

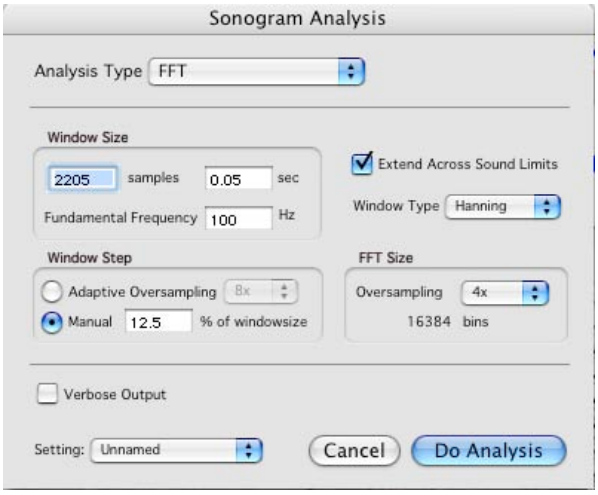



***figure 24. les paramètres de l'analyse de sonogramme***

On peut choisir les paramètres de la fenêtre d'analyse (champ *Window size*), le pas d'avancement (*Window Step),* le type de fenêtre, et la taille de la FFT. On retrouve ici les paramtres de l'analyse STFT.

### ***Analyse f0***

Les paramètres ont été détaillées en 2.4 ; on les retrouve dans la fenêtre d'interface d'Audiosculpt :

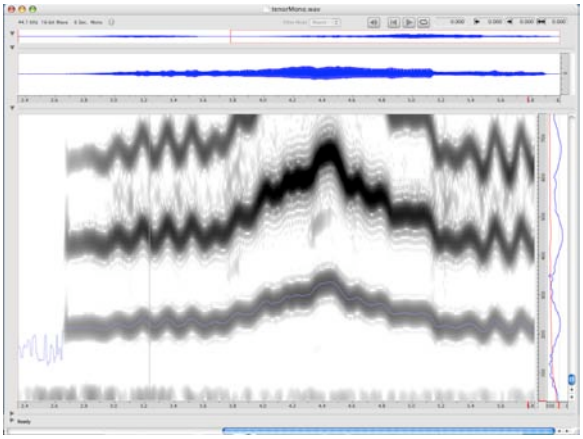


*fig 25 :Analyse f0 superposée au spectrogramme*

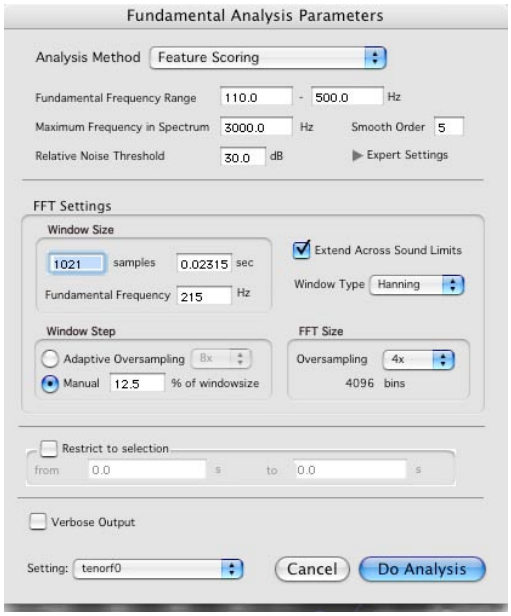



*fig. 26 : paramètres de l'analyse*

### *Analyse « True enveloppe »*

L'analyse « True Enveloppe » est réalisée avec la fonction *sonogram analysis* également, en choisissant l'option « true enveloppe ». On retrouve les paramètres de l'analyse FFT, avec en plus un chois pour la valeur de férquence fondamentale maximale (cf 2.5).

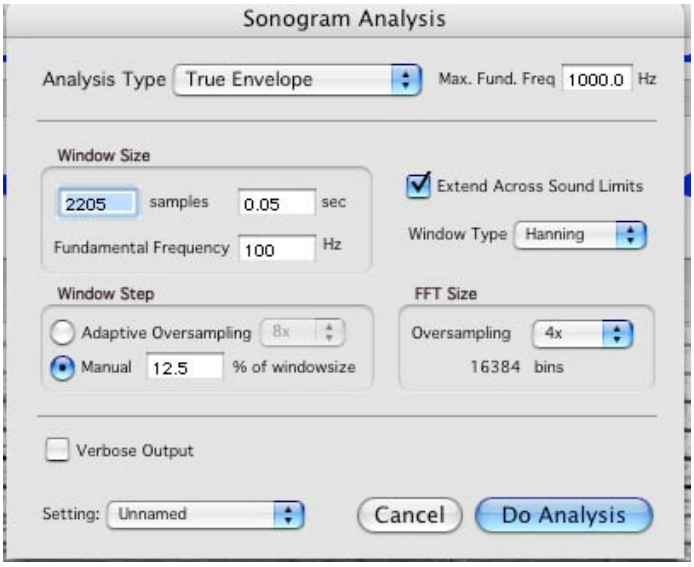



*fig 27 : paramètres de l'analyse true enveloppe*

# A3. Exemple de fonction : fonction trans.m

```
%NAME
%   trans.m - generates a transposition file for supervP "transpose"
%   operation.
%%USAGE
%[mat,fact] = trans (sdif_filename,DC,alpha)
%alpha is the transposition parameter (can be a float):
%alpha = 0 : cancel the vibrato , alpha =1 : none effect, alpha >1 :
%accuentuates the vibratos, alpha<0 : inverse the vibrato

function [mat,fact] = trans (f0_sdif_filename , DC , alpha , mrk_sdif_filename)

   %appel à la bibliothèque de gestion des fichiers sdif pour récupérer f0
   [data,H] = Fsdifloadfile(f0_sdif_filename);

   %H est le vecteur colonne des temps associe a chaque trame de f0
   H=H(:,1);

   %f0 est le vecteur des valeurs de f0 pour chaque trame, OL est le sous echantillonnage de
f0 par rapport au signal
   [f0,OL]=sdif2f0(data);

   % on prépare les vecteurs qui contiendront le facteur de transposition
   res = zeros (size(f0));
   res_cent = zeros(size(f0));

   %beat est le coefficient de transposition
   beta      =  1-alpha;

   % boucle de traitement
   for ind = 1:length(DC),

        %on évite le cas f0=0 qui correspond à un point critique dans l'analyse f0
         if (f0(ind)==0) f0(ind) = min(f0(f0>0)); end

        %facteur de transposition
         res (ind)  =  DC(ind)/f0(ind);


        %conversion en cents
         fact(ind) =  beta * (1200*log(res(ind)))/log(2);

   end

   %En dehors des segments, on met le facteur de transposition à 1
   if mrk_sdif_filename ~= ' '
   %la fonction « segment » convertit les positions des marqueurs en sec. et les place dans un
vecteur Sg
   Sg=segment(mrk_sdif_filename,OL);

   for ind = 2:2:(length(Sg)-1)

         Tdeb=Sg(ind);
         Tfin=Sg(ind+1);
         Hdeb = round((44100/OL)*Tdeb);
         Hfin = round((44100/OL)*Tfin);

         for k = Hdeb : Hfin

             fact(k) = 1;
         end
   end
   Tdeb = Sg(end);
   Hdeb = round((44100/OL)*Tdeb);

  for k = Hdeb:H(end)

  fact(k)=1;

  end

  mat = [H fact'];
  end

  mat = [H fact'];
 end
```