



**HAL**  
open science

# Synthèse de sons musicaux : extraction de paramètres à partir d'une gamme en vue de la resynthèse par distorsion non linéaire et filtrage

Patrick Sanchez

## ► To cite this version:

Patrick Sanchez. Synthèse de sons musicaux : extraction de paramètres à partir d'une gamme en vue de la resynthèse par distorsion non linéaire et filtrage. Acoustique [physics.class-ph]. Université Aix-Marseille II - Institut de Mécanique de Marseille, 1992. Français. NNT: . tel-01863489

**HAL Id: tel-01863489**

**<https://hal.science/tel-01863489>**

Submitted on 28 Aug 2018

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

**Université d'Aix-Marseille II  
Institut de Mécanique de Marseille**

# **THESE**

Présentée par

**Patrick SANCHEZ**

Pour obtenir le grade de

**Docteur de l'Université d'Aix-Marseille II  
Spécialité Acoustique et Dynamique des Vibrations**

## **SYNTHESE DE SONS MUSICAUX: EXTRACTION DE PARAMETRES A PARTIR D'UNE GAMME EN VUE DE LA RESYNTHESE PAR DISTORSION NON LINEAIRE ET FILTRAGE**

Rapporteurs

E. BIANCO  
J. CHOWNING  
G. DE POLI

Membres du jury

E. BIANCO  
G. DE POLI  
D. ARFIB

Directeur de thèse

J.C. RISSET



## TABLE DES MATIERES

		page
I.	INTRODUCTION GENERALE	1
I.1.	CONTENU DU MEMOIRE	3
I.1.a.	Synthèse: aspect logiciel	3
I.1.b.	Synthèse: aspect matériel	4
I.1.c.	Analyse-synthèse	4
I.2.	UN BREF TOUR D'HORIZON DE L'EVOLUTION DES TECHNIQUES	5
I.2.a.	Informatique Musicale : Logiciels	5
I.2.b.	Informatique Musicale : Matériels (hardware)	6
II.	ETUDE D'UNE METHODE DE COMPILATION DU LANGAGE MUSICV	9
II.1.	LE SYSTEME	9
II.2.	LA MEMOIRE	11
II.3.	AUTOJECTIVITE	11
II.4.	BOUCLE A OUVERTURE	13
II.5.	COMPILATION	14
II.6.	PARTITION MUSICV	14
II.7.	DEROULEMENT	16
II.8.	CODES OPERATOIRES	18
II.9.	DEFINITION DES MODULES	19
II.10.	OPERATEURS	19
II.11.	COMPILATION	21
II.11.a.	Grammaire de MUSICV	21
II.11.A.1.	LE TEXTE.	21
II.11.A.2.	LES DECLARATIONS.	21
II.11.A.3.	LES PARAMETRES.	21
II.11.A.4.	LES INSTRUMENTS.	21
II.11.b.	Structure du compilateur	22
II.11.c.	Variables d'états	24
II.11.d.	Exemple de compilation	24
II.12.	AUTOJECTION ET SYSTEME	27
III.	BIBLIOGRAPHIE	29
IV.	CARTE DE CONTROLE DU SFG-05	31
IV.1.	ORIGINE DU STANDARD MIDI	31
IV.2.	PRINCIPE DE FONCTIONNEMENT DU SFG-05	35
IV.3.	FONCTIONS DE BASE	37
IV.4.	CONFIGURATION POUR LA PROGRAMMATION	37
IV.5.	PLAN DETAILLE DE LA CARTE (SFG-05)	38
IV.6.	POSSIBILITE DU MONTAGE	40
IV.6.a.	mapping de la mémoire	40
IV.6.A.1.	DECODAGE ADRESSE	41
IV.6.A.2.	DECODAGE DU SIGNAL D'ECRITURE	42
IV.7.	EXEMPLES DE PROGRAMME	42
V.	BIBLIOGRAPHIE	52

	page	
VI.	EXTRACTION DE PARAMETRES SONORES EN VUE DE LA RESYNTHESE ET SEPARATION SOURCE-RESONANCE	53
VI.1.	POSITION DU PROBLEME	53
VI.2.	LES OUTILS: METHODES D'ANALYSE ET DE SYNTHESE DE SIGNAUX	56
VI.2.a.	La synthèse	56
VI.2.a.1.	SYNTHESE SIMPLE	56
VI.2.a.2.	SYNTHESE ADDITIVE	56
VI.2.a.3.	SYNTHESE SOUSTRACTIVE	57
VI.2.b.	Synthèse globale ou non-linéaire	58
VI.2.b.1.	MODULATION D'AMPLITUDE	58
VI.2.b.2.	MODULATION DE FREQUENCE	59
VI.2.b.3.	DISTORSION NON-LINEAIRE	59
VI.2.c.	L'analyse	66
VI.2.c.1.	ENERGIE DU SIGNAL	66
VI.2.c.2.	ANALYSE SPECTRALE SYNCHRONE	68
VI.2.c.2.1.	extraction des périodes	68
VI.2.c.2.2.	sur-échantillonnage	68
VI.2.c.3.	ESTIMATION DE LA FREQUENCE FONDAMENTALE	71
VI.2.c.3.1.	cas du signal périodique	71
VI.2.c.3.2.	cas où la fréquence du fondamental est une fonction du temps	72
VI.3.	SEPARATION SOURCE-RESONANCE	75
VI.3.a.	estimation des formants	75
VI.3.a.1.	METHODES CLASSIQUES	76
VI.3.a.1.1.	maxima locaux du spectre	76
VI.3.a.1.2.	représentation homomorphique: cepstrale	76
VI.3.a.1.3.	Modèle auto-regressif: prédiction linéaire	78
VI.4.	APPROCHE ORIGINALE DU PROBLEME	81
VI.4.a.	cas du spectre d'excitation constant	86
VI.4.a.1.	CALCUL D'UNE GRILLE DE FREQUENCE COMMUNE	90
VI.4.a.2.	ESTIMATION DE L'ENVELOPPE SPECTRALE	94
VI.4.a.3.	MODELISATION DE L'ENVELOPPE SPECTRALE	103
VI.4.a.3.1.	modélisation de la phase: phase minimale	103
VI.4.a.3.2.	estimation des coefficients du filtre	105
VI.4.a.3.3.	Réduction à un problème linéaire	108
VI.4.a.3.4.	méthode de Marquardt	112
VI.4.a.3.5.	méthode de Laguerre	114
VI.4.a.3.6.	Stabilité	115
VI.4.a.3.7.	Synthèse du filtre	115
VI.4.b.	Exemple de résultat issu de signaux réels: voix chantée.	116
VI.4.c.	Inversion du filtre résonant	121
VI.4.d.	modélisation du signal d'excitation par distorsion non-linéaire	121
VI.4.d.1.	MOMENTS SPECTRAUX	121
VI.4.d.2.	BARYCENTRE DU SPECTRE	122
VI.4.e.	estimation de la fonction de distorsion et de l'index de distorsion	127
VI.5.	APPLICATIONS	133
VI.5.a.	modélisation des familles d'instruments acoustiques: lutherie informatique	133
VI.5.b.	Expérimentations	135
VI.5.c.	construction d'un exemple académique et recherche des paramètres	135
VI.5.d.	signaux réels	147
VI.5.d.1.	Contrôle de l'excitation	148
VI.5.d.1.2.	Type d'instrument	148
VI.5.e.	applications sonores	153
VII.	CONCLUSION	155
VIII.	BIBLIOGRAPHIE	159

		page
VI.	EXTRACTION DE PARAMETRES SONORES EN VUE DE LA RESYNTHESE ET SEPARATION SOURCE-RESONANCE	53
VI.1.	POSITION DU PROBLEME	53
VI.2.	LES OUTILS: METHODES D'ANALYSE ET DE SYNTHESE DE SIGNAUX	56
VI.2.a.	La synthèse	56
VI.2.a.1.	SYNTHESE SIMPLE	56
VI.2.a.2.	SYNTHESE ADDITIVE	56
VI.2.a.3.	SYNTHESE SOUSTRUCTIVE	57
VI.2.b.	Synthèse globale ou non-linéaire	58
VI.2.b.1.	MODULATION D'AMPLITUDE	58
VI.2.b.2.	MODULATION DE FREQUENCE	59
VI.2.b.3.	DISTORSION NON-LINEAIRE	59
VI.2.c.	L'analyse	66
VI.2.c.1.	ENERGIE DU SIGNAL	66
VI.2.c.2.	ANALYSE SPECTRALE SYNCHRONE	68
VI.2.c.2.1.	extraction des périodes	68
VI.2.c.2.2.	sur-échantillonnage	68
VI.2.c.3.	ESTIMATION DE LA FREQUENCE FONDAMENTALE	71
VI.2.c.3.1.	cas du signal périodique	71
VI.2.c.3.2.	cas où la fréquence du fondamental est une fonction du temps	72
VI.3.	SEPARATION SOURCE-RESONANCE	75
VI.3.a.	estimation des formants	75
VI.3.a.1.	METHODES CLASSIQUES	76
VI.3.a.1.1.	maxima locaux du spectre	76
VI.3.a.1.2.	représentation homomorphique: cepstrale	76
VI.3.a.1.3.	Modèle auto-regressif: prédiction linéaire	78
VI.4.	APPROCHE ORIGINALE DU PROBLEME	81
VI.4.a.	cas du spectre d'excitation constant	86
VI.4.a.1.	CALCUL D'UNE GRILLE DE FREQUENCE COMMUNE	90
VI.4.a.2.	ESTIMATION DE L'ENVELOPPE SPECTRALE	94
VI.4.a.3.	MODELISATION DE L'ENVELOPPE SPECTRALE	103
VI.4.a.3.1.	modélisation de la phase: phase minimale	103
VI.4.a.3.2.	estimation des coefficients du filtre	105
VI.4.a.3.3.	Réduction à un problème linéaire	108
VI.4.a.3.4.	méthode de Marquardt	112
VI.4.a.3.5.	méthode de Laguerre	114
VI.4.a.3.6.	Stabilité	115
VI.4.a.3.7.	Synthèse du filtre	115
VI.4.b.	Exemple de résultat issu de signaux réels: voix chantée.	116
VI.4.c.	Inversion du filtre résonant	121
VI.4.d.	modélisation du signal d'excitation par distorsion non-linéaire	121
VI.4.d.1.	MOMENTS SPECTRAUX	121
VI.4.d.2.	BARYCENTRE DU SPECTRE	122
VI.4.e.	estimation de la fonction de distorsion et de l'index de distorsion	127
VI.5.	APPLICATIONS	133
VI.5.a.	modélisation des familles d'instruments acoustiques: lutherie informatique	133
VI.5.b.	Expérimentations	136
VI.5.c.	construction d'un exemple académique et recherche des paramètres	136
VI.5.d.	signaux réels	148
VI.5.d.1.	Contrôle de l'excitation	149
VI.5.d.1.2.	Type d'instrument	149
VI.5.e.	applications sonores	154
VII.	CONCLUSION	155
VIII.	BIBLIOGRAPHIE	159



Je tiens à remercier en priorité M. Jean-Claude RISSET qui m'a accepté dans son équipe et qui a fait preuve d'énormément de patience à mon égard surtout vers la fin de mes travaux.

Je remercie également Daniel ARFIB qui est à l'origine des idées développées dans ce document, tel un gourou il a su éveiller ma passion pour l'Informatique Musicale.

Je remercie plus particulièrement Richard KRONLAND, qui au moment le plus difficile a su m'épauler pendant la dernière ligne droite.

Je ne veux pas oublier dans mes remerciements tous les membres de l'équipe d'Informatique Musicale qui de près ou de loin ont contribué sur des chemins parallèles à améliorer les connaissances de l'équipe et avec qui j'ai passé de très bons moments (surtout en dehors des heures de travail): Frédéric BOYER, Nathalie DELPRAT, Philippe GUILLEMAIN, Simon BOLSINGER.

Sans oublier les personnes extérieures à l'équipe sans lesquelles le LMA ne serait pas le LMA et qui par leur présence et souvent leurs conseils ont contribué au soutien des thésards : l'éternel Frédéric MAZEROLLE, la dynamique Danielle CATTHIAUX, Jacques JELSCH, Martine CULIOLI, les indispensables M. et Mme. AVENI. Tous ont contribué à rendre mon séjour très agréable.

Une mention spéciale pour Sylvie qui a supporté mes angoisses, une autre pour ma Maman et Pablo.

Dédicace: Pascal et Brigitte GERMAIN, Anne-Marie ALBRAND, Daniel ARFIB, M et MME AUDOYE, Edmond et Marie-Jo BIANCO, Paul BOJIKIAN, Dominique BOZON, Jacques CHATRON, Anne CHAUVET, Marc CHOSTAKOFF, Christian CLOT, Les membres du CRIM, Agnès DAHL, Sabine DAVID, Jean-Pierre DE VITA., M et MME DE VITA., Jacques DIENNET, Société DIGILOG, Didier LEVIN, Jérôme AUDIBERT, Lionel SANDRE, Famille DONNET, MR et MME DREVET, Philippe DREVET, Laurence et Eric DUBOT, Jacques DUDON, Famille EMIREN, Jean-Louis FORTUNATO, Nicole FRANCKENBERGER, Marcel et Anette GALL, Flora GAUTHIER, Patrick GAUTHIER, Joel GELYS, Marie\_France et Martin GONZALVEZ, Jean-Pierre HENNEQUIN, famille HENNEQUIN, Hélène HERVIER, Gérard HILAIRE.et Rose, Claude HILEZIAN, François LACOMME, Andrée LALOGÉ., M. et MME LEDUC, Béatrice LENORMAND, Blanche et Jean-denis LUER, Augustin MARCOTORCHINO, Clodine MARTEL, Thierry MARTEL, Thierry MAUCCI, Serge MENSAH, Didier MESTRE, famille MEYER-HEINE, Roger MONTEAU, Simone MONTFORTE, Suzy NAAS, Jacques NOLANE, Mireille OLIVER, Jean-Claude ROUVE et Jojo, Jean-Claude SABETTA, Pauline SANDOZ, François SANGALETTI, Jean SCHMUTZ, Gérard SEGHETTO, Jules SLAMA, Robert SOULAKIAN, Charles VIAN, famille WAGNER, Krikor SOUKIASSIAN.



# I. INTRODUCTION GENERALE

L'informatique musicale sonore représente un domaine d'activité très vaste dont la vocation est de mettre à la disposition des musiciens un ensemble de techniques logicielles et matérielles destinées à la production de matériaux musicaux. Les principales études se rapportant à ce domaine peuvent être décomposées en trois sous-groupes intimement liés, à savoir: la synthèse algorithmique de sons, l'analyse et l'extraction de paramètres physiques et psychoacoustiques à partir de signaux naturels, la transformation de sons réels (ce qui nécessite des procédures d'analyse et de resynthèse). Chacun de ces champs d'application fait intervenir à la fois les aspects logiciels et matériels de l'informatique.

Certains domaines ont été plus étudiés que d'autres, et ce, en fonction des évolutions technologiques. C'est ainsi que les techniques de synthèse ont fait l'objet de nombreux travaux dans les années 60 et 70, et si elles satisfont les musiciens aujourd'hui, leur mise en oeuvre informatique n'a parfois que très peu évolué. C'est le cas par exemple du logiciel MUSICV dont la structure permet de mettre en oeuvre toutes les techniques connues à ce jour mais qui reste difficile d'utilisation sauf pour des spécialistes. C'est pour pallier cette difficulté que j'ai effectué un travail informatique sur MUSICV tendant à faciliter le dialogue homme-machine.

Au demeurant, un aspect important concerne la mise à disposition des musiciens "d'instruments" exploitables en temps réel. La technologie permet aujourd'hui de construire de telles machines, il n'en reste pas moins vrai que la définition des algorithmes et des paramètres accessibles est d'une importance capitale. L'art du chercheur en informatique musicale consiste alors à mettre à la disposition du musicien des outils peu coûteux mais de grande qualité. C'est à cette fin que j'ai réalisé une carte de contrôle permettant d'utiliser un circuit de synthèse commercial.

L'aspect analyse a, quant à lui, fait l'objet de nombreux travaux dans tous les domaines de la physique expérimentale. Bien qu'issu de différents domaines de recherche, il est souvent classé sous la rubrique "traitement du signal". Pourtant, si les techniques peuvent sembler équivalentes dans de nombreux cas, il n'en reste pas moins vrai que les paramètres pertinents que l'on cherche à extraire doivent répondre à des critères spécifiques. C'est ainsi que l'informatique musicale n'a de sens que si les paramètres mesurés correspondent à des

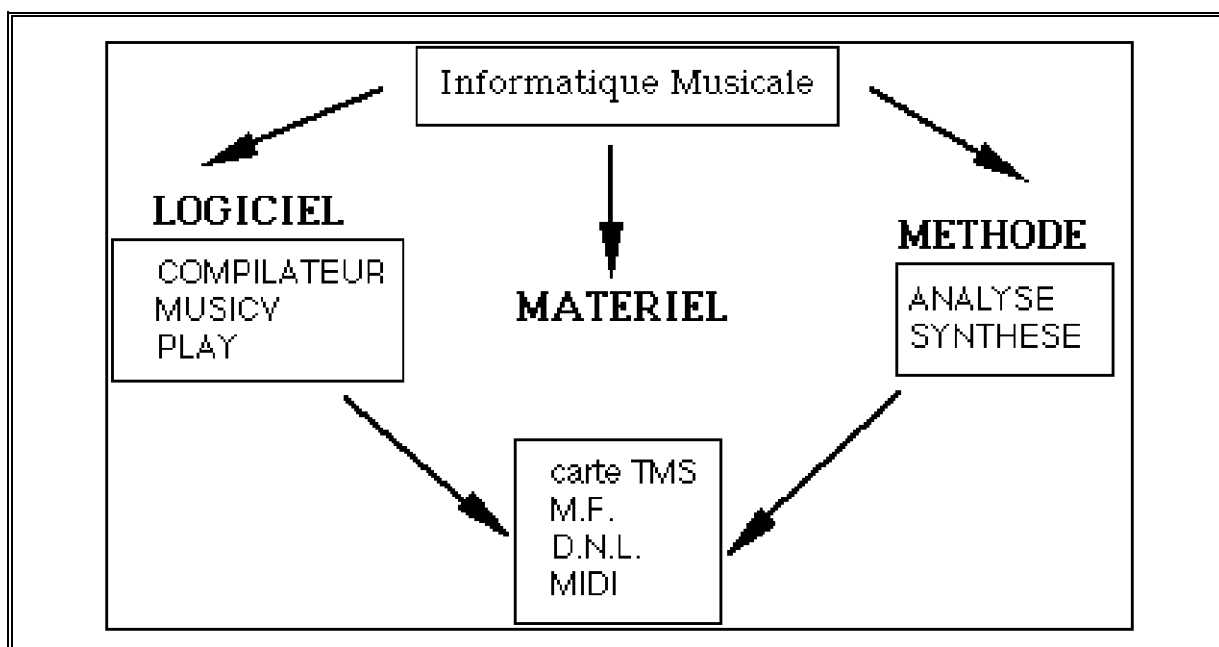
situations physiques particulières et à une réalité psychoacoustique que seule l'oreille peut apprécier.

Ces dernières années, l'intérêt porté à l'ordinateur en tant qu'instrument à part entière de l'orchestre n'a cessé de grandir. Plus qu'un générateur de sons, l'ordinateur s'est avéré être un outil potentiel de traitement des signaux sonores, permettant d'interagir sur les sons naturels en leur donnant une dimension nouvelle. Modifier un son naturel de façon intime nécessite d'en avoir une description précise, tant du point de vue physique qu'au niveau de son impact psychoacoustique. C'est alors qu'il est possible d'intervenir sur le son en agissant sur les paramètres entre les procédures d'analyse et de synthèse. Dans ce but, j'ai mis au point une technique nouvelle permettant d'extraire de sons naturels des paramètres permettant la resynthèse.

Dans ce mémoire, je présenterai mes travaux portant sur les domaines suscités (synthèse, technologie, analyse), qui correspondent à des préoccupations actuelles. Ces travaux font intervenir les aspects logiciel et matériel de l'informatique, et ont également nécessité le développement de nouvelles techniques d'analyse-synthèse de signaux sonores.

## I.1. Contenu du mémoire

L'ensemble du travail présent dans ce document porte sur les trois domaines définis ci-dessous (fig.1).



*figure 1*

### I.1.a. Synthèse: aspect logiciel

J'ai réalisé une nouvelle version de la première passe du compilateur MUSICV. Ce compilateur utilise une méthode basée sur l'autojectivité<sup>1</sup>. L'originalité de cette approche tient dans le fait que la gestion temporelle des quanta de travail du compilateur est prise en compte par le compilateur lui-même plutôt que par le système. Il en ressort que l'écriture du système incluant un tel compilateur se trouve réduite. De plus, de façon à faciliter le dialogue homme-machine, je me suis appliqué à gérer systématiquement les erreurs de syntaxe dans l'écriture de la "partition MUSICV". Ces travaux ont été menés en relation avec le LITAM (Laboratoire d'Informatique Théorique et Application de Marseille).

### I.1.b. Synthèse: aspect matériel

---

<sup>1</sup> Définition au chapitre suivant

J'ai réalisé une carte de contrôle d'un synthétiseur musical (YAMAHA SFG 01) qui accompagne l'ordinateur familial CX-5M de la même marque. Le but recherché était de pouvoir utiliser les possibilités sonores du synthétiseur sur d'autres micro-ordinateurs et de mettre à la disposition du musicien des techniques modernes de synthèse. Ces dernières sont basées sur la modulation de fréquence, ce qui permet d'utiliser les catalogues de sons correspondants. Il serait ainsi facilement possible d'associer ce module au compilateur MUSICV en passant les paramètres de synthèse au générateur et d'intervenir sur les paramètres de création des sons (modification enveloppe d'amplitude, pitch, modulation de fréquence...).

### **I.1.c. Analyse-synthèse**

Il s'agit de la partie la plus importante de mon travail. J'ai développé et mis en oeuvre une méthode d'extraction de paramètres à partir de l'enregistrement d'une gamme chromatique de sons naturels instrumentaux. La technique de synthèse utilisée correspond à un modèle source-résonance dont la source est construite par distorsion non linéaire et la résonance par filtrage linéaire. Cette méthode nouvelle est plus spécialement adaptée à l'analyse de sons d'instruments à formants fixes.

## **I.2. Un bref tour d'horizon de l'évolution des techniques**

Avant l'apparition des techniques informatiques, la production des sons électroacoustiques était essentiellement assurée par des techniques analogiques mettant en œuvre des matériels très diversifiés, donc très encombrants. A la fin des années 50, une équipe de chercheurs des laboratoires Bell développe une méthode de génération de sons à l'aide d'un ordinateur. L'apport de l'informatique permet de programmer des modules de traitement dont certains n'ont aucun équivalent analogique. Par exemple, pour engendrer un son composé de plusieurs harmoniques indépendants, il faut avec les techniques analogiques autant d'oscillateurs que d'harmoniques. Avec l'ordinateur, il suffit d'une table présente en mémoire, contenant une période de la fonction sinus. La suite des harmoniques étant déduite par une lecture plus ou moins rapide de cette table, le son final est obtenu par sommation des résultats intermédiaires dans un même emplacement mémoire. L'écoute est réalisée grâce à un convertisseur digital-analogique.

### **I.2.a. Informatique Musicale : Logiciels**

On peut donc créer des sons à l'aide d'un langage informatique. Dans les années 60 le langage le plus utilisé était FORTRAN. Comme la plupart des études faites sur les sons montraient que l'on pouvait rattacher un son à une classe de synthèse, se sont développés des langages spécialement consacrés à la génération et au traitement des sons. Cela semblait plus naturel que de créer un programme par son. Ces langages proposent tous les mêmes possibilités:

- écrire un programme à partir d'un éditeur ASCII.
- modifier des "partitions" (au sens partition MUSICV)
- produire des échantillons de sons

Ces programmes, et en particulier, MUSICV sont des compilateurs (ou interpréteurs) écrits à cette époque en FORTRAN avec des parties en langage machine. On dispose donc d'une vérification sémantique et syntaxique (qui peut être rudimentaire) et de la possibilité de composer un instrument à partir de différents modules. On n'est pas tenu à une méthode de synthèse particulière: en ce sens, ces programmes sont parfaitement adaptés à l'étude et à la resynthèse des sons. La phase de production des échantillons est en réalité une phase de déroulement qui à partir du code généré par la compilation effectuera les opérations nécessaires à la création des échantillons.

Les compilateurs sont facilement transportables sur d'autres machines, leur utilisation est simple, mais l'interaction avec le compositeur est réduite. De plus il est fréquent d'attendre plusieurs heures entre l'écriture du programme et l'écoute du résultat. C'est pourquoi d'autres types de recherches ont été effectués dans le but d'accélérer les processus de création des sons.

### **I.2.b. Informatique Musicale : Matériels (hardware)**

La phase de déroulement (création du son à partir des données présentes dans la partition) est la partie la plus longue en temps de calcul car elle dépend du nombre d'instruments et de leur complexité. On conçoit aisément que pour que les échantillons puissent être écoutés en temps réel il faut que la totalité des instructions s'effectue dans un temps inférieur à la période d'échantillonnage. Or, par définition, un programme et son temps de déroulement appartiennent au domaine du "fini illimité"<sup>2</sup> alors que la période d'échantillonnage appartient au "fini borné"<sup>3</sup>.

Pour résoudre ce problème qui consiste à faire pénétrer du "fini illimité" dans du "fini borné" il n'y a pas de solution même philosophique. On le contourne de trois manières:

- En utilisant un ordinateur plus rapide. On ne peut pas dans ce cas faire dérouler MUSICV en temps réel, car on conserve une notion des partitions "fini illimité", mais cela augmente le confort d'utilisation. Si l'ordinateur n'est pas assez rapide, certains traitements peuvent être proposés à un processeur spécialisé (ex: TMS 320, DSP 56000) qui prendra en charge tous les calculs onéreux en temps.

- En faisant correspondre les éléments minimums décrits dans un instrument (ex: oscillateur, filtre...) avec des constituants électroniques implantés sur une carte. Le nombre de modules électroniques sur une carte appartient au domaine "fini borné", donc le compositeur devra réduire la structure des instruments en fonction de la contrainte matérielle. C'est le cas de la 4X, UPIC, SYTER etc.... Cela ne présente pas que des inconvénients car on peut plus facilement introduire des commandes gestuelles.

- En forçant l'utilisation d'une méthode de synthèse particulière dans laquelle l'instrument est déjà construit mais ne possède pas certains paramètres d'entrée comme la hauteur, le

---

<sup>2</sup> Un programme doit être fini avant la compilation, par contre pendant la phase de création il peut être illimité car on peut toujours y ajouter une ligne. Le temps de déroulement en dépend.

<sup>3</sup> La durée entre deux échantillons est bornée. Cela impose une condition sur le temps de déroulement pour que le calcul des échantillons s'effectue dans un temps inférieur à la période d'échantillonnage.

timbre. C'est le cas des instruments répondant à la norme MIDI qui utilisent une méthode de synthèse par modulation de fréquence et modulation d'amplitude. C'était également le cas des anciens synthétiseurs câblés dont les connexions entre chaque module se faisaient à l'aide de fils sur la face avant de l'appareil.

Actuellement l'évolution de l'interaction homme-machine est facilitée par l'apparition d'un environnement graphique du type Macintosh. La mise en oeuvre de cet environnement est simplifiée par l'utilisation de langage de programmation objet. Des stations de travail UNIX (SUN, NEXT) proposent également cet environnement. On trouve maintenant en standard sur certaines machines NEXT la possibilité d'enregistrer et d'écouter des sons à l'aide de convertisseur analogique-numérique et des possibilités de sauvegarde énorme des échantillons sonores (disque optique). L'environnement graphique profite également à la norme MIDI. L'IRCAM a mis au point un environnement graphique sous la forme d'un programme appelé MAX (en hommage à Max Mathews) développé par Miller Puckette. Ce programme permet l'assemblage de modules permettant de piloter en temps réel tous les types d'appareils conformes à la norme MIDI. Dans le même temps YAMAHA a créé un système pour piloter un véritable piano par l'intermédiaire du MIDI. L'association avec le logiciel MAX ouvre de nouvelles possibilités créatrices (J.C. Risset) car jusqu'à présent on ne pouvait piloter que des modules de synthèse MIDI (expanders). La qualité acoustique est devenue celle du piano.

De même l'architecture des ordinateurs ne cesse d'évoluer. On trouve sur le marché des architectures de type parallèle ou RISC qui multiplient les performances de calcul par des facteurs considérables. Les processeurs de traitements de signaux se spécialisent. On trouve maintenant des processeurs qui travaillent avec une arithmétique flottante, capable d'effectuer en temps réels des opérations complexes comme du filtrage, des convolutions ou des FFTs. De nouvelles méthodes de traitement du signal temps-fréquence comme la transformée en ondelette peuvent être mise en oeuvre sur des architectures parallèles ou vectorielles, améliorant ainsi considérablement les temps de calcul par rapport à une simulation sur des machines traditionnelles.

L'étude des langages et des compilateurs a également évolué. On note l'apparition de l'intelligence artificielle et des réseaux neuronaux qui contribuent à leur manière aux recherches psychoacoustiques et notamment aux études sur l'apprentissage.

Après cet aperçu des techniques, je vais décrire en premier lieu une méthode de construction d'un compilateur du langage MUSICV. Cette méthode permet également l'écriture de n'importe quel autre type de compilateur et s'intègre dans une étude plus large des systèmes informatiques. Nous verrons que cette méthode a des implications sur l'écriture du système qui accueille l'ensemble des compilateurs écrits suivant cette méthode.





## II. ETUDE D'UNE METHODE DE COMPILATION DU LANGAGE MUSICV

MusicV est un langage de synthèse et de traitement des sons par ordinateur conçu et écrit par Max V. Mathews aux Laboratoires Bell. Il permet la conception et la réalisation d'algorithmes de calcul de sons dans une syntaxe particulière qui sera décrite plus loin.

Il existe de nombreuses versions de MusicV, dont l'analyseur syntaxique ainsi que les modules de calcul ont été implantés sur de gros ordinateurs. C'est pour faire face au développement de la micro-informatique que l'équipe d'Informatique Musicale a décidé d'implanter une version du compilateur sur un ordinateur personnel compatible IBM-PC.

Le but de cette section n'est pas de décrire une nouvelle version de MusicV, mais de proposer une méthode de compilation de ce langage qui suppose une écriture particulière du compilateur conforme à la théorie de l'"*autojectivité*". Cette méthode a été expérimentée sur une version réduite du programme, mais le principe reste valide pour une version plus complète. L'ensemble de la théorie de l'"*autojectivité*" a été développée et améliorée au Laboratoire d'Informatique Théorique et Application de Marseille (LITAM) par l'équipe du professeur Edmond BIANCO.

### II.1. Le système

Les origines de l'informatique et l'apparition de l'ordinateur sont essentiellement liées au besoin de calculer, de résoudre des problèmes arithmétiques. Pour réaliser ces opérations arithmétiques, il était nécessaire de représenter des nombres à l'intérieur de la machine. Pour cela, on a fait appel à un emplacement physique minimum, appelé bit, que l'on rassemble par groupe de huit pour former un octet. Cet octet est rapidement devenu l'élément indispensable pour créer la mémoire des ordinateurs. Cette mémoire est un ensemble de cases constituées d'octets, ensemble suffisamment grand dont chaque case est adressable individuellement. Les processeurs spécialement conçus pour l'exploitation de ce type de mémoire possèdent des registres internes dont la structure est l'image d'une case de la mémoire. Actuellement l'élément minimum représenté par l'octet s'agrandit pour devenir mot ou double mot, mais la relation entre les registres internes et les cases mémoires reste la même, à savoir qu'il est

toujours très difficile d'accéder à un sous-ensemble de l'élément minimum sans pour cela faire plusieurs opérations complexes comme des décalages, etc...

A partir de la connaissance d'une machine et de ses instructions on peut construire des programmes. En pratique, un grand nombre de tâches liées aux problèmes de l'utilisateur sont prises en compte par un programme qui s'occupe de la gestion et de la répartition du travail entre les différents compilateurs. C'est ce programme que l'on appelle communément le système.

A l'origine, ce système était très réduit: il permettait l'introduction des codes machines et des données, et contrôlait les échanges entre les différentes parties de la machine au cours du déroulement du programme. Puis, comme les services demandés étaient de plus en plus complexes, le système s'est compliqué d'autant. On peut dire qu'il s'est transformé à la demande tout en grandissant comme un énorme "patchwork".

Un système informatique est de nos jours un véritable monstre accessible seulement à quelques initiés: tous les étudiants connaissent la programmation structurée et l'insertion de procédure mais par contre, il est toujours très difficile d'introduire dans un système un nouveau compilateur. Cette opération demande quelques mois. Pour un système, un compilateur est un corps étranger, il doit apprendre à vivre avec lui grâce à des communications multiples. Chaque constructeur a résolu le problème à sa manière, il n'y a pas de cohésion dans la théorie. Ainsi face au problème insoluble de déterminer à l'avance l'arrêt d'un programme, les systèmes actuels ont résolu la question en interrompant le déroulement de manière aléatoire. Ce découpage temporel utilise actuellement les interruptions qui sont souvent la bête noire des étudiants. Une solution pourrait être l'autojectivité qui propose un découpage structurel de telle sorte que le programme puisse s'interrompre lui-même au bon moment afin de permettre au système de continuer sa gestion.

Cette propriété traduite au sein des compilateurs permet une écriture simplifiée des systèmes: dans la mesure où tous les mécanismes nouveaux que cela introduit au niveau des langages seront câblés ou microprogrammés, cela permet une rapidité d'exécution des programmes et augmente la lisibilité du système.

## **II.2. La mémoire**

Pour des raisons liées à l'adressage, la mémoire des ordinateurs est séparée fonctionnellement en deux parties: l'une interne bornée, l'autre externe illimitée.

C'est une adresse contenue dans un registre qui permet l'accès direct à chacune des cases qui composent la mémoire interne. La capacité du registre adresse est forcément bornée, c'est ce fait qui impose un traitement fini-borné dans la mémoire interne. La mémoire externe qui représente le fini-illimité est l'apport indispensable pour compléter la mémoire de l'ordinateur.

La mémoire interne ou mémoire vive est le lieu d'exécution des programmes, sa taille est fixe et les débordements de taille doivent être pris en compte par le système. La mémoire externe est souvent considérée comme une mémoire de sauvegarde des informations de grande capacité: elle est constituée généralement par des bandes magnétiques, des disques durs ou optiques. L'accès est plus lent qu'en mémoire centrale et son contrôle fait partie d'une des tâches les plus importantes du système.

On dira que le domaine du fini-borné, ou travail en mémoire interne, est le domaine réservé du compilateur. L'existence du système informatique devient indispensable pour la gestion et le contrôle du fini-illimité.

Un système pourra ainsi avoir à gérer le travail d'un ou plusieurs compilateurs différents, et également, pour chaque compilateur, d'une ou plusieurs tâches différentes.

En conséquence immédiate de ceci, le système doit être construit pour pouvoir intégrer en son sein n'importe quel compilateur - il doit pouvoir gérer le temps et l'espace mémoire pour n'importe quel type de traitement - et de façon complémentaire, le compilateur quel qu'il soit doit posséder une structure qui rende à la fois possible et simple la gestion du fini-borné (temps et mémoire).

## **II.3. Autojectivité**

C'est un modèle de structure: *l'autojectivité*, qui répond à la question. Un compilateur autojectif, donc doté d'une telle organisation [SHANNON: machine à deux états], possède les deux propriétés suivantes:

### Définition

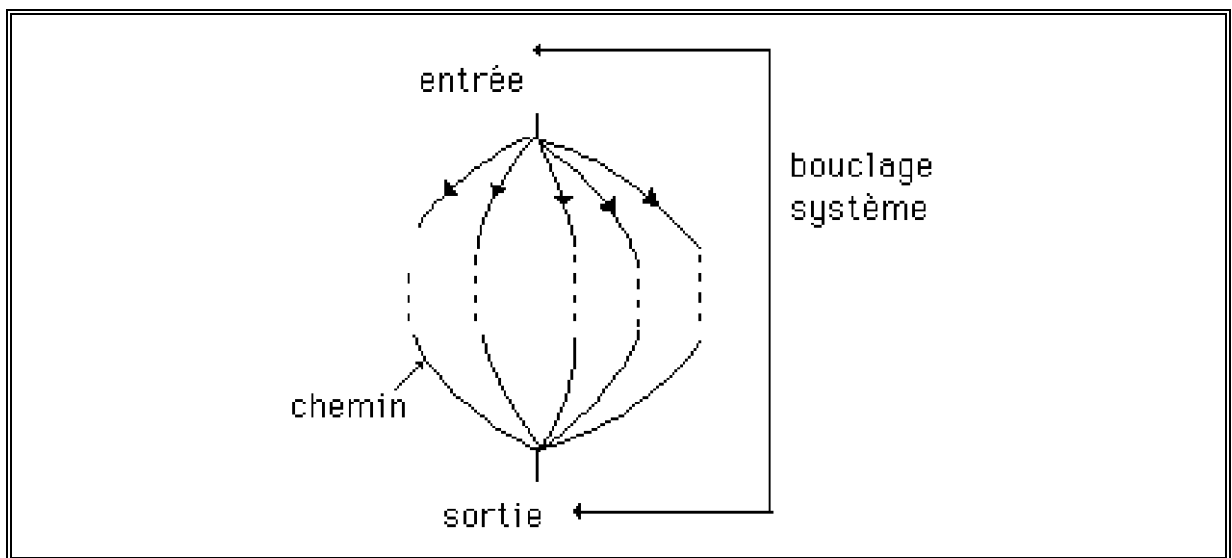
Disposant d'une seule entrée et d'une seule sortie, le compilateur est composé d'un ensemble de chemins parallèles qui assurent l'ensemble du traitement de compilation.

1 - Chacun de ces chemins est fini et borné dans le temps. Quel que soit le traitement réalisé par chacun des chemins, il demandera moins de temps qu'une borne fixée à l'avance.

2 - Chacun de ces chemins est fini et borné dans l'espace. Quel que soit le traitement réalisé, chaque chemin ne peut employer qu'un volume de mémoire inférieur ou égal à une borne de volume fixée à l'avance.

Ceci implique que le compilateur contrôle le débordement éventuel des configurations qui lui sont affectées et qu'un langage universel lui permet de communiquer avec le système qui assurera le service du fini-illimité. Je vais expliquer ci-après une structure de compilateur compact dont on montre par ailleurs qu'il est effectivement autojectif.

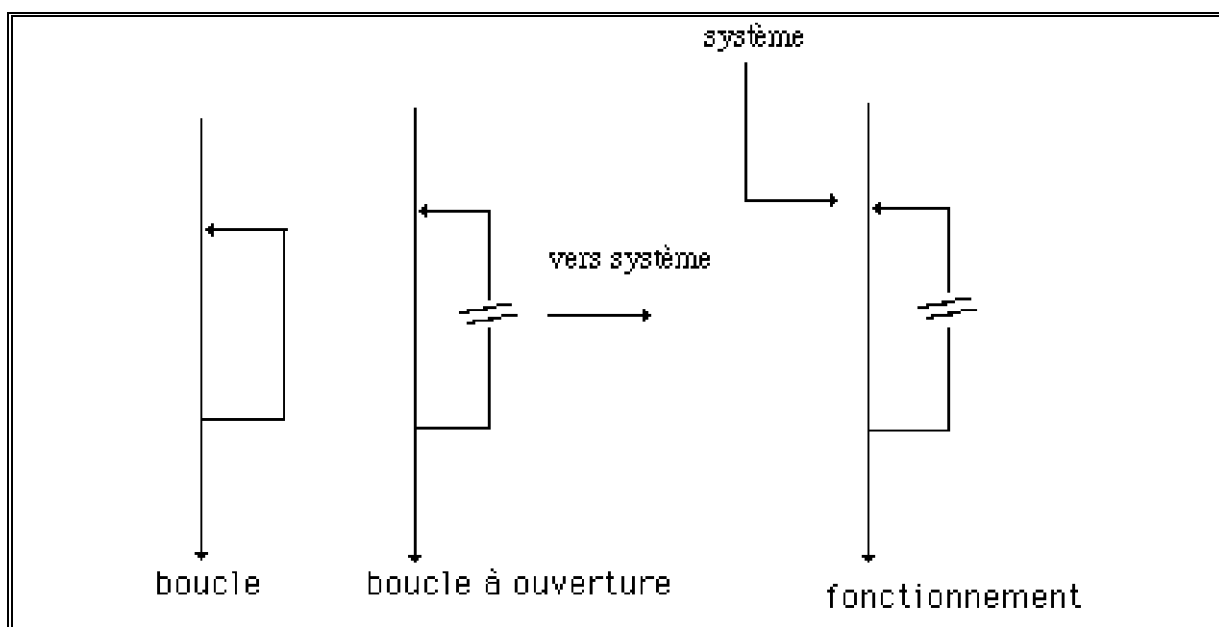
Un tel compilateur a les propriétés d'un "*compilateur compact*". Un algorithme est dit "*autojectif*" s'il possède un seul point d'entrée et un seul point de sortie, et si les chemins allant du début à la fin sont finis et bornés dans le temps et dans l'espace mémoire. On appelle chemin, la suite d'instructions prises en compte dans un parcours allant du début à la fin. Si un chemin contient une boucle, cela signifie que le nombre de tours est borné à l'avance, ou que la boucle est à ouverture.



*Schéma de principe du compilateur*

## II.4. Boucle à ouverture

Si, dans un chemin, un traitement se fait dans une boucle, celle-ci, pour ne pas gêner le partage des tâches et pour permettre au système de reprendre le contrôle, doit être à ouverture: c'est à dire permettre un retour vers le système. Le compilateur prend en charge le retour dans la boucle à l'endroit précis du départ.



*Fonctionnement d'une boucle à ouverture*

Ce mécanisme permet de dérouler le corps de la boucle en tranches finies et bornées en respectant ainsi les contraintes de "autojectivité". On associe un nombre N de tours à l'ouverture de la boucle: si au bout de N tours la boucle n'est pas terminée, on sort directement du programme, un système de commutation nous ramenant directement dans la boucle lors d'une nouvelle insertion du programme. La suite des opérations consiste à attendre la fin normale de la boucle [ISOARDI].

## II.5. Compilation

Le but de la compilation étant de transformer une phrase symbolique en une phrase codée, qui sera ensuite soumise au compilateur de déroulement, on se donne comme quantum de travail le traitement d'une lettre de la phrase symbolique. Pour réaliser l'analyse syntaxique et sémantique de la phrase et émettre un diagnostic, on se donne un jeu de variables d'états qui serviront à noter l'état de la phrase. A chaque apparition d'une nouvelle lettre, on doit pouvoir vérifier qu'elle est compatible avec ce qui précède puis transformer les valeurs des variables d'états en fonction de la nouvelle information.

Avant de décrire plus en détail le compilateur, voyons la forme des énoncés du langage.

## II.6. Partition MUSICV

L'utilisateur du programme MUSICV est, à la fois, compositeur et luthier. Il dispose d'un nombre illimité d'instruments qu'il construit lui-même à partir de modules de base et suivant une méthode de synthèse choisie. MUSICV n'impose pas de méthode de synthèse particulière. L'utilisateur donne ensuite la succession de notes qui sera jouée par chaque instrument, l'ensemble de toutes ces indications constituant la partition MUSICV.

Une partition est l'ensemble des instructions que l'on doit écrire pour créer les commandes nécessaires à la fabrication du son. De manière générale une instruction est composée d'un mot-clé suivi d'une suite de champs dont la signification est liée au type du mot-clé.

mot-clé champ1 champ2 champ3 . . .

Tout énoncé commence obligatoirement par un mot-clé et est représenté par une suite de lettres ou de chiffres commençant par une lettre et se limitant en général à trois lettres. Ex: INS, OSC, OUT . . .

mot-clé::=< lettre > < lettre >|< chiffre >

Le caractère ";" indique la fin d'une instruction et les champs peuvent être séparés par autant de blancs que l'on veut. La virgule peut également être un séparateur de champs. La syntaxe de MUSICV admet les commentaires. L'énoncé est COM suivi d'une suite de caractères ignorée jusqu'au prochain ";".

Les énoncés peuvent être de plusieurs types:

- Les affectations de variables

- Les définitions d'instruments
- Les générateurs de fonctions ou de formes d'onde
- Les notes

VARIABLES

INSTRUMENTS

GEN,FONCTIONS

NOTES

La structure d'un programme pour engendrer un son est décrite ci-dessus. Les variables que l'on définit en général au début du texte sont des paramètres utiles au déroulement, par exemple la fréquence d'échantillonnage du signal. Puis on indique dans les déclarations d'instruments les algorithmes de calcul des sons, chaque instrument correspondant à une méthode de calcul différente. Chaque définition d'instrument est constituée par l'assemblage des modules de calcul (oscillateurs, additionneurs, filtres, etc...) dont les paramètres s'écrivent d'une manière très précise. On y trouve également des énoncés de conversion qui permettent de transformer facilement les unités choisies par l'utilisateur dans celles admises par la syntaxe des modules. Certains modules de calcul utilisent des courbes qui seront définies par les générateurs de fonctions. Une fois les instruments et les courbes définis, on déclenche le calcul au moyen d'un énoncé NOT qui, en fonction du numéro de l'instrument, donnera les informations nécessaires au calcul des échantillons, dont le temps de départ, la durée de la note, et aussi d'autres paramètres comme l'amplitude et la fréquence.

Exemple de programme écrit en MUSICV:

```

SIA 0 4 25600;
COM fréquence échantillonnage ;
INS 0 1;
COM déf. de l'instrument ;
CNV P6=HTZ (P6);
CNV P7=DUR (P4);
IOS P5 P7 B3 F1 P30;
IOS B3 P6 B3 F2 P29;
OUT B3 B1;
END;
COM fin déf. de l'instrument ;
GEN 0 1 1 512 0 0 1 100 0 512;
GEN 0 2 2 1024 1 1;
NOT 0 1 2 20000 100 0;
COM fréquence 100 Hz ;
TER 2;

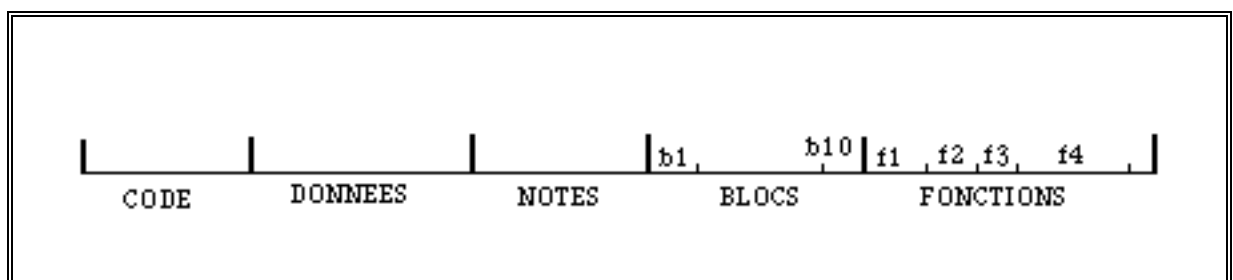
```

*Dans cet exemple le premier oscillateur détermine l'enveloppe (fonction f1), et le second la forme d'onde (fonction f2).*

Au niveau sémantique il y a une grande différence entre les déclarations d'instruments et tous les autres énoncés. En général les énoncés autres que ceux définis dans un instrument sont formés d'un mot-clé suivi par des champs qui sont des nombres évalués à la lecture de l'énoncé. Les déclarations d'instruments sont formées de nombres et de variables reliés par des opérateurs arithmétiques et des fonctions mathématiques.

## II.7. Déroulement

Au déroulement, la mémoire a la structure suivante:



*Cette structure n'est pas figée: elle varie selon le type de la machine.*

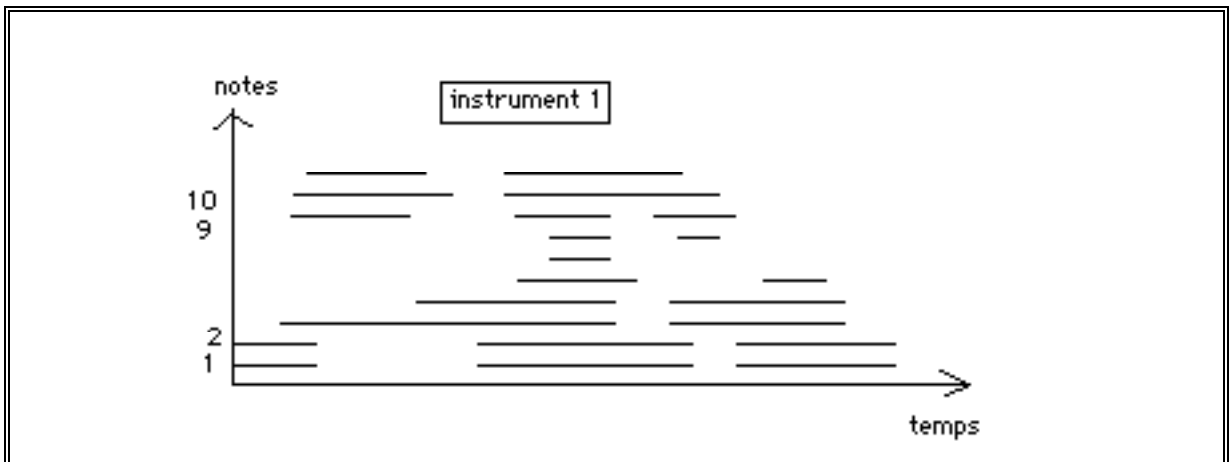
Le code contient toutes les informations sur les algorithmes des instruments, sa longueur sera connue à la fin de la compilation. La zone des données est une zone de 1000 valeurs réelles dont les 100 premières sont des indications particulières au dérouleur (sortie stéréo ou mono, fréquence d'échantillonnage, etc...) le reste étant disponible pour des affectations de variables. La zone des notes est constituée d'un tableau de 128 notes, chaque note ayant 30



paramètres. Cela veut dire qu' actuellement, on peut jouer 128 notes simultanément. La zone des blocs est celle où se transforment les échantillons sonores. Il y a 10 blocs de 256 mots et les premiers blocs B1 et B2 sont les blocs de sortie du son. La zone des fonctions est une zone allouée dynamiquement au fur et à mesure de l'apparition d'une nouvelle fonction.

Le déroulement se passe de la manière suivante: on lit une note, puis cette note renvoie à un instrument que l'on exécute en déroulant les algorithmes compris entre le début et la fin de cet instrument. Pour cela on utilise les fonctions, les paramètres des notes, les données, et le résultat est la modification des blocs de calcul. Puis on lit la note suivante jusqu'à épuisement.

Comme MUSICV peut jouer plusieurs notes en même temps soit d'un même instrument, soit d'instruments différents, on cumule dans un même bloc les événements simultanés.



Entre la phase de compilation et le déroulement, il existe une phase de préparation au déroulement qui classe l'apparition des événements en fonction du temps. Au cours de cette phase sont évaluées toutes les conversions sur les paramètres des notes.

Voici un exemple du code généré par la première phase de la compilation sur la partition écrite précédemment:

```

SIA 0 4 25600;          4 12 0 4 25600
INS 0 1;                3 2 0 1
CNV P6=HTZ(P6);        7 2 0 0 11 6 2 6
CNV P7=DUR(P4);        7 2 0 0 12 4 2 7
IOS P5 P7 B3 F1 P30;   8 2 0 13 5 7 -3 -101 30
IOS B3 P6 B3 F2 P29;   8 2 0 13 -3 6 -3 -102 29
OUT B3 B1;              5 2 0 1-3 -1
END;                    2 2 0
GEN 0 1 1 512 0 0 1 100 0 512; 11 3 0 1 1 512 0 0 1 100 0 512
GEN 0 2 2 1024 1 1;    7 3 0 2 2 1024 1 1
NOT 0 1 2 20000 100 0; 7 1 0 1 2 20000 100 0

```

La signification des paramètres est la suivante:

- Pn fait référence au champ n de l'énoncé NOT
- Vn fait référence à la variable n dans la zone des données
- Fn fait référence à la fonction n
- Bn fait référence au bloc de résultats n

On utilise le codage suivant :

Pn := n

Vn := 100+n

Fn := -(100+n)

Bn := -n

Lors du déroulement, les fonctions sont calculées et placées dans la zone réservée aux fonctions. La longueur des fonctions est variable et l'espace qu'elles occupent est géré dynamiquement. Les notes, les instruments et les données sont également installés dans leur configuration respective et le déroulement débute par la lecture de la première note. Celle-ci fait référence à un instrument, lui-même faisant référence à un générateur de fonction qui par un algorithme de calcul approprié enverra les résultats dans les blocs de sortie.

Les fonctions installées dans les tables sont lues à des vitesses variables, cela entraîne une variation de la fréquence du signal ainsi créé après une conversion numérique-analogique.

## II.8. Codes opératoires

La table suivante contient la liste des codes opératoires, leur code, leur signification. Cette liste n'est pas exhaustive. Du fait de la modularité de MUSICV, on peut créer de nouveaux codes opératoires par ajout d'une nouvelle fonction du langage.

<b>memo</b>	<b>code</b>	<b>signification</b>
CNV	0	conversion
NOT	1	joue la note
INS	2	définit un instrument
GEN	3	génère une fonction
SV3	4	passé un paramètre au déroulement
SEC	5	fin de section
TER	6	fin de la partition

SV1	7	initialisation de variables dans pass1
SV2	8	initialisation de variables dans pass2
PLF	9	exécution d'un sous-programme dans pass1
PLS	10	exécution d'un sous-programme dans pass2
SI3	11	initialisation d'entiers dans pass3
SIA	12	initialisation d'entiers dans toutes les passes
COM	13	commentaire

## II.9. Définition des modules

Chaque module fait l'objet d'un programme qui se déroule en tenant compte des paramètres d'entrée et des blocs de sortie. La modularité de MUSICV provient également de la facilité à créer de nouveaux modules.

mémo	code	signification
OUT	1	sortie mono
OSC	2	oscillateur
AD2	3	additionneur à 2 entrées
RAN	4	générateur de fonction aléatoire
ENV	5	générateur d'enveloppe
STR	6	sortie stéréo
AD3	7	additionneur à 3 entrées
AD4	8	additionneur à 4 entrées
MLT	9	multiplicateur
FLT	10	filtre
RAH	11	générateur de fonction aléatoire
SET	12	initialise une nouvelle fonction
IOS	13	oscillateur avec lissage
DNL	16	distorsion non linéaire

## II.10. Opérateurs

Les conversions se placent dans le corps de l'instrument, entre les instructions INS et END.

opérateur	code op.	signification
(	1	parenthèse ouverte

)	2	parenthèse fermée
+	3	addition
-	4	soustraction
*	5	multiplication
/	6	division
SIN	7	sinus
COS	8	cosinus
LOG	9	logarithme
EXP	10	exponentielle
HTZ	11	conversion en Hertz
DUR	12	conversion en durée

## **II.11. Compilation**

Comme on peut le constater, MUSICV est un langage proche des langages machines ou langages d'assemblages, qui possèdent une grammaire et une sémantique relativement simple. La compilation de MUSICV s'effectue en trois passes. Dans la première passe, on vérifie la syntaxe de la partition, le code est stocké dans un fichier intermédiaire (pass1.dat). La deuxième passe classe les événements par ordre chronologique: le résultat est sauvegardé dans un deuxième fichier intermédiaire (pass2.dat). La troisième passe est le déroulement du programme qui exécute les événements sonores. Dans la première et deuxième passe sont exécutés des énoncés de fonctions de composition (PLF,PLS) qui nécessitent une connaissance des particularités de MUSICV: ces énoncés seront traités à part.

### **II.11.a. Grammaire de MUSICV**

#### **II.11.a.1. Le texte.**

Le texte de MUSICV est une suite d'instructions comprenant des déclarations de variables et de constantes, des énoncés de conversion, la description des instruments et le jeu instrumental défini par l'énoncé des notes. Pour écrire la partition il suffit d'avoir un éditeur de texte ASCII; le début de la partition est donné par le début du texte et la fin par un énoncé TER qui indique le temps de fin.

#### **II.11.a.2. Les déclarations.**

Elles sont de plusieurs types: déclarations de variables et de constantes (SV1, SV2, SV3, SIA). On utilise des déclarations de type entier, réel. Il est important pour celui qui écrit la partition d'avoir à l'esprit la forme de la configuration des données qui sera présente dans la machine au moment du déroulement. C'est une des difficultés de ce langage.

#### **II.11.a.3. Les paramètres.**

MUSICV ne dispose pas d'un énoncé de déclaration des paramètres, mais une bonne connaissance de l'énoncé NOT, qui définit la note à jouer, laisse apparaître quelque chose qui ressemble au passage de paramètres en relation avec l'écriture des modules. En effet dans les champs de l'énoncé NOT certaines valeurs (à partir du champ P5) se réfèrent aux entrées des modules des instruments.

#### **II.11.a.4. Les instruments.**

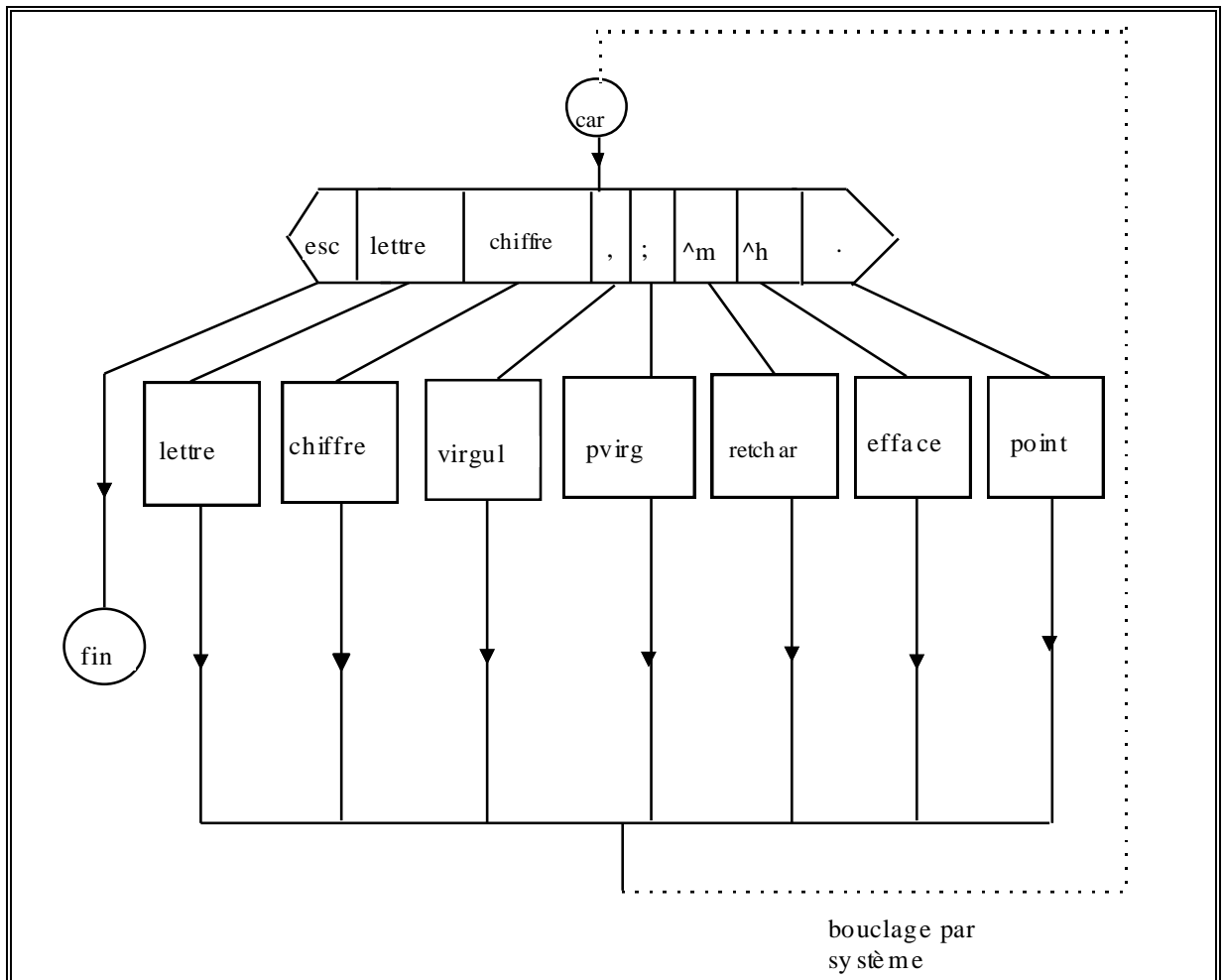
L'instrument est encadré par les énoncés INS et END. Les modules (OSC, MLT, AD2, FLT) ne font pas partie des déclarations. On doit les trouver dans le corps de l'instrument ainsi que les ordres de conversion (CNV).

Très rapidement on peut citer les propriétés que ne possède pas MUSICV.

- L'insertion de procédure.
- L'aller-à (Existe dans d'autres versions)
- La condition (Existe dans d'autres versions)
- Les boucles.

### **II.11.b. Structure du compilateur**

Un compilateur "*autojectif*" traite le fini borné, au contraire du système qui traite le fini illimité. Le point d'entrée du compilateur est un aiguillage qui traite l'élément minimum représenté par une lettre de la phrase symbolique. Chaque lettre est traitée séparément en fonction de la classe syntaxico-sémantique à laquelle elle appartient. Chaque bloc qui prendra en charge le traitement de la lettre est un chemin fini et borné dans le temps ne comportant aucune boucle illimitée. L'insertion fractionnée du compilateur pour le traitement d'une nouvelle lettre est une opération du système.



*Schéma de principe du fonctionnement du compilateur*

### II.11.c. Variables d'états

Pour construire le compilateur, on se donne un jeu de variables d'états qui permettra de réaliser l'analyseur syntaxique. La connaissance des valeurs de ces variables d'états permet à tout moment et sans ambiguïté de connaître la position et l'état syntaxique de la phrase symbolique. Cela permet de noter les informations suffisantes sur ce qui vient d'être traité, grâce à la redondance d'informations apportées par ce type de variables d'états. Ces variables sont le coeur du compilateur. Grâce à elles, on peut connaître à tout moment l'état de la phrase et prévoir la suite à venir. En cas de changement de contexte, pour insérer un autre utilisateur, il suffit de sauver l'ensemble des variables d'états plus quelques pointeurs dans le code et les données pour que le compilateur reprenne le travail au même endroit.

### II.11.d. Exemple de compilation

On se donne une représentation de l'évolution des valeurs des variables d'états en fonction des caractères (lettre, séparateur) rencontrés au cours de la compilation. Les tableaux ci-dessus donnent un exemple des différentes valeurs que peuvent prendre les variables d'états. Ce choix n'est pas unique: on peut augmenter le nombre de variables pour simplifier l'analyse de la phrase, mais un choix judicieux évite des redondances d'informations. La nature de l'algorithme placé dans un chemin déterminé par une classe d'étude associée à un jeu de variables d'états permet donc l'analyse syntaxique et sémantique de la phrase.

champ	lettre	séparateur
o	1	+1

la variable "champ" change de valeur à l'apparition d'une lettre ou d'un séparateur (blanc)

classe	type1	type2
0	1	2

Type1: INS,GEN,NOT,TER,SV1,SV2,SV3,SEC,SI3,SIA

Type2: OUT,OSC,AD2,RAN,AD3,AD4,MLT,IOS,DNL

schamp	P,B,V,F	Chiffre	Séparateur
0	1	2	0

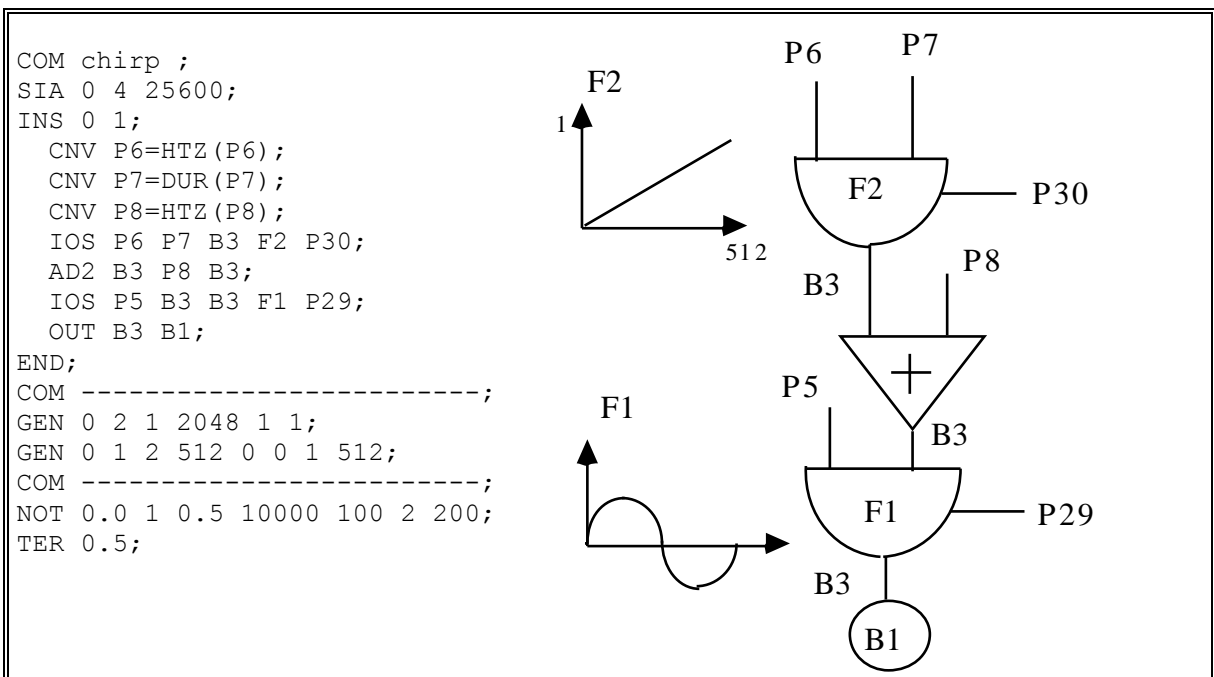


Type	Ins	End
0	1	0

0: Op code

1: Unit générateur

Pour l'exemple de programme MUSICV ci-dessous, on donne l'évolution des variables d'états au cours de la compilation.



L'évolution des variables d'états donne une indication du niveau d'analyse de la phrase à compiler. Dans l'écriture du compilateur, la partie la plus délicate reste le choix des variables d'états. On peut se donner un nombre quelconque de variables, il n'y a pas unicité de la solution et il est même conseillé de n'être pas restrictif. Les valeurs des tableaux sont engendrées par l'exécution du compilateur et leur visualisation permet de faciliter la phase de mise au point.

	_	SIA	_	0	_	4	_	25600	;
champ	0	1	2	2	3	3	4	4	0
schamp	0	0	0	0	0	0	0	0	0
classe	0	0	1	1	1	1	1	1	0
cpt_space	0	0	1	0	1	0	1	0	0

buf		INS		0		1	;
champ	0	1	2	2	3	3	0
schamp	0	0	0	0	0	0	0
classe	0	0	1	1	1	1	0
type	0	0	1	1	1	1	1
cpt_space	0	0	1	0	1	0	0
f_ins	1	1	0	0	0	0	1

buf		CNV		P	6	=	HTZ	(	P	6	)	;
champ	0	1	2	2	2	5	5	5	5	5	5	0
schamp	0	0	0	1	2	0	3	0	1	2	0	0
classe	0	0	4	4	4	4	4	4	4	4	4	0
type	1	1	1	1	1	1	1	1	1	1	1	1
pgn	0	0	0	1	1	0	0	0	1	1	1	0
debut	0	0	0	0	0	0	0	1	1	1	1	0
fegal	0	0	0	0	0	1	1	1	1	1	1	1
cpt_space	0	0	1	0	0	0	0	0	0	0	0	0

buf	_	IO S	_	P	6	_	P	7	_	B	3	_	F	2	_	P	3	0	;
champ	0	1	2	2	2	3	3	3	4	4	4	5	5	5	6	6	6	6	0
schamp	0	0	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	2	0
classe	0	0	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	0
type	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
pbvf	0	0	0	1	1	1	1	1	1	2	2	2	4	4	4	1	1	1	0
cpt_space	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	0	0

buf		AD2		B	3		P	8		B	3	;
champ	0	1	2	2	2	3	3	3	4	4	4	0
schamp	0	0	0	1	2	0	1	2	0	1	2	0
classe	0	0	2	2	2	2	2	2	2	2	2	0
type	1	1	1	1	1	1	1	1	1	1	1	1
pbvf	0	0	0	2	2	2	1	1	1	2	2	0
fegal	1	1	1	1	1	1	1	1	1	1	1	1
cpt_space	0	0	1	0	0	1	0	0	1	0	0	0
f_ins	1	1	1	1	1	1	1	1	1	1	1	1

buf		OUT		B	3		B	1	;
champ	0	1	2	2	2	3	3	3	0
schamp	0	0	0	1	2	0	1	2	0
classe	0	0	2	2	2	2	2	2	0
type	1	1	1	1	1	1	1	1	1
pbvf	0	0	0	2	2	2	2	2	0
fegal	1	1	1	1	1	1	1	1	1
cpt_space	0	0	1	0	0	1	0	0	0
f_ins	1	1	1	1	1	1	1	1	1

## II.12. Autojection et système

Un compilateur est essentiellement fait pour s'insérer dans un programme maître qui prend en charge la distribution des tâches des différents compilateurs ainsi que des échanges. Dans un tel contexte, l'autojectivité apporte un contrôle plus précis au niveau de la gestion du temps dans le système. En effet un compilateur autojectif étant auto-structuré, le retour au système se fait en des points bien précis, fonctions du chemin parcouru. Un tel compilateur peut travailler sur plusieurs tâches en même temps si les configurations sont distinctes dans la mémoire centrale et si les variables d'états sont différenciées pour chaque tâche jusqu'à la fin du travail. On voit donc disparaître la notion d'interruption qui était une gestion aléatoire du temps dans les systèmes informatiques. Le système se trouve donc considérablement allégé car il ne s'occupe plus que de l'insertion fractionnée des différents compilateurs [EL GHAZI].

A ce niveau le compilateur ressemble étrangement à un séquenceur d'un langage machine. On retrouve la même structure composée d'une boucle fermée sur un aiguillage comportant autant de chemins que le langage comporte d'instructions. Le traitement de chaque instruction est strictement borné. Ainsi une interruption au niveau le plus bas de la machine ne peut interrompre le déroulement micro programmé du micro code composant une instruction élémentaire. Celle-ci ne se fera qu'après le déroulement complet de l'instruction en cours. Si l'on s'obstine à vouloir interrompre quand même au milieu d'une instruction, c'est au prix d'une sauvegarde de contexte difficile à gérer.

Au niveau du système, on traduit cette même propriété en utilisant un compilateur auto-structuré de telle sorte que lors de l'insertion, il ne se déroule qu'une fraction finie bornée du travail, le retour au système étant automatique. Le système ne s'occupe alors plus que de la distribution des tâches des différents compilateurs.

De plus, si l'analyse syntaxique et sémantique est faite correctement, la détection d'erreurs peut se faire lettre à lettre: la compilation de la phrase au fur et à mesure de l'avancement du programme sera terminée à la fin de la phase d'écriture du programme, le résultat prêt à être soumis au compilateur de déroulement, d'où un gain de temps sur la mise au point des applications.

Il pourrait être intéressant, en utilisant les résultats obtenus dans la théorie des systèmes et en utilisant les notions qui ont été dégagées à cette occasion, de créer un système simple dans sa structure qui permettrait de mettre en parallèle plusieurs tâches. Par exemple l'utilisateur de MUSICV pourrait travailler à la compilation pendant le déroulement du calcul d'un son.

L'étude n'a pas été menée au bout pour une version complète de MUSICV car l'écriture d'un tel compilateur relève plus d'un travail de sociétés spécialisées: le but était de montrer une idée nouvelle de l'écriture d'un compilateur et de ses implications sur l'écriture des systèmes informatiques. Des travaux plus complets sur ce domaine sont effectués au LITAM (Laboratoire d'Informatique Théorique et Applications de Marseille) où l'ensemble des recherches porte sur les moyens dont il faut doter une machine pour rendre celle-ci accessible par le plus grand nombre en facilitant la compréhension et la réalisation des systèmes et des compilateurs.

### III. BIBLIOGRAPHIE

- [1] BIANCO EDMOND  
*Autojection et compilateurs*  
Bull. d'Informatique Approfondie et Application, N° 4 - Faculté des Sciences de Luminy. Dec 1982 . ISSN 0291 - 5413
- [2] BIANCO EDMOND  
*Etude et Formalisation d'une classe de systèmes*  
Thèse de doctorat d'état - Faculté des Sciences de Paris
- [3] BIANCO EDMOND  
*De la machine de Turing aux Ordinateurs modernes*  
Birkhauser Verlag - 1979
- [4] CROCUS  
*Systèmes d'exploitation des ordinateurs*  
Dunod - Paris 1975
- [5] EL GHAZI EL HOUSSAINI SOUHAIL  
*Notions Fondamentales de la théorie des Systèmes Informatiques*  
Thèse de docteur de l'Université d'Aix-Marseille II - Luminy - 10 oct. 1986
- [6] ISOARDI Patrick  
*Structures de commutation*  
Doctorat, Université d'Aix-Marseille II . Fév. 1978
- [7] ISOARDI PATRICK  
*Calculateur à mémoire champ - Nouvelles structures de mémoires, processeur associé.*  
Thèse d'état, faculté des Sciences de Luminy. Sept. 1987
- [8] NOLIN L.  
*Machines, langages, compilation*  
Bull. d'Informatique Approfondie et Application, N° 2 - Faculté des Sciences de Luminy. Marseille 1982. P. 6-36. ISSN 0291 - 5413



## **IV. CARTE DE CONTROLE DU SFG-05**

### **IV.1. Origine du standard MIDI**

Les origines du MIDI (Musical Instrument Digital Interface) remontent en 1983. A cette période plusieurs constructeurs de machines musicales (synthétiseurs) avaient déjà fait des tentatives de communications entre machines numériques. L'idée était de piloter plusieurs synthétiseurs à partir d'un seul clavier.

L'évolution des techniques, passant de l'analogique au numérique, a rendu possible cette communication. Les méthodes de synthèses numériques étant différentes des méthodes analogiques, ce sont des algorithmes informatiques qui produisent le signal sonore. De cette manière, un algorithme produisant un son reconnaissable pourra par exemple changer de fréquence par la seule variation d'une valeur propre au programme, les autres paramètres contrôlant l'aspect psychoacoustique seront internes au processus de création sous forme de modèles de synthèses, tables de valeurs etc...

On comprend aisément l'intérêt d'implanter dans une machine un algorithme de fabrication d'échantillons sonores ayant une communication avec l'extérieur par certaines variables.

Seules ces variables seront envoyées sur la liaison. Le programme s'exécute en tenant compte des variations des paramètres extérieurs.

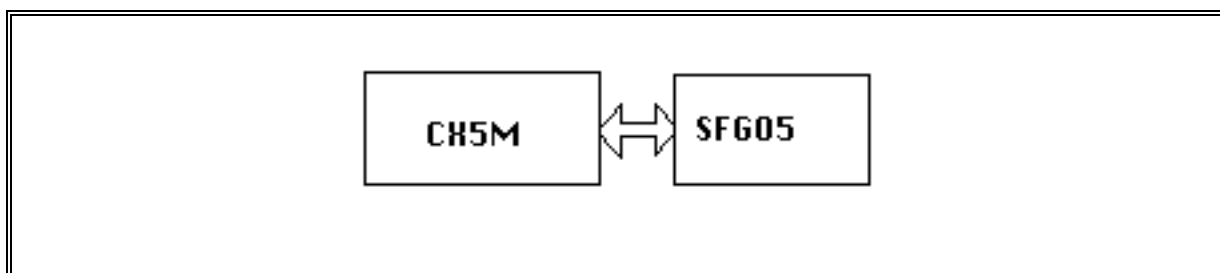
L'information de hauteur n'est évidemment pas la seule information à transmettre, et devant l'importance et le nombre de codes possibles (transmission par octet) l'ensemble des marques présentes sur le marché a décidé d'adopter un standard d'unification: la norme MIDI (Musical Interface Digital Instrument). Il se trouve que le standard de communication adopté était très proche de celui utilisé par les micro-ordinateurs dans leurs liaisons séries (RS232).

L'idée apparut immédiatement chez les constructeurs de micro-ordinateurs de développer un interface capable de lire et d'émettre les codes MIDI. Cet interface fut aussitôt adopté par les musiciens car il permettait un contrôle très précis de tous les paramètres constituant le modèle de synthèse. En moins de deux ans tous les synthétiseurs possédaient leur interface MIDI et bon nombre de marques les proposaient en option avec l'achat de l'ordinateur. Le premier ordinateur dit "musical" était le COMMODORE C64, suivi de l'APPLE II et l'ATARI. Actuellement on trouve sur tous les compatibles PC et MACINTOSH une quantité

énorme de logiciels permettant la gestion de données MIDI. L'ensemble des logiciels se présente sous diverses formes: des séquenceurs, éditeurs de partitions, éditeurs de paramètres de synthèse ...

Par rapport aux machines proposées à cette époque, le CX5M de YAMAHA était le premier ordinateur à vocation entièrement musicale. Il était accompagné d'un module de synthèse, le SFG01 (puis SFG05) qui possédait une meilleure qualité sonore que l'ATARI ou le COMMODORE, grâce à sa méthode de synthèse: la modulation de fréquence, développée par J. Chowning. Le CX5M est un micro-ordinateur qui fonctionne en association avec un module de synthèse (le SFG05). Pour un prix abordable le CX5M offre de grandes possibilités musicales. Il est basé sur une architecture de micro-ordinateur personnel adoptée par de nombreux constructeurs japonais. Le standard MSX est construit autour du microprocesseur Z80. Les logiciels MSX sont vendus sous forme de cartouche (ROM). A l'heure actuelle l'ordinateur ne dispose pas d'un logiciel de gestion de disques. Les sauvegardes se font à l'aide d'un lecteur enregistreur de cassette. Le CX5M s'attaque à un marché de musiciens possédant un synthétiseur de la série DX de YAMAHA grâce au module SFG05: celui-ci contient un programme ("Music Composer") qui transforme le CX5M en petit séquenceur ou en éditeur de partition. Pour les programmeurs, le module contient également toute une librairie de fonctions ("FM Music Macro") qui peuvent être appelées à partir d'un langage évolué comme le BASIC ou bien en langage machine.

Dans le même temps, le Laboratoire d'Informatique Musicale du L.M.A. avait acquis un savoir-faire dans la construction d'interface MIDI par l'intermédiaire de Daniel Arfib, qui avait mis au point ce type d'interface pour les ordinateurs THOMSON utilisés par l'Education Nationale. Comme il n'existait pas de module de synthèse bon marché, l'idée de se servir du petit module SFG05 comme "expand<sup>4</sup>" semblait intéressante, surtout pour des applications pédagogiques.



*Relation entre l'ordinateur et le module de synthèse*

---

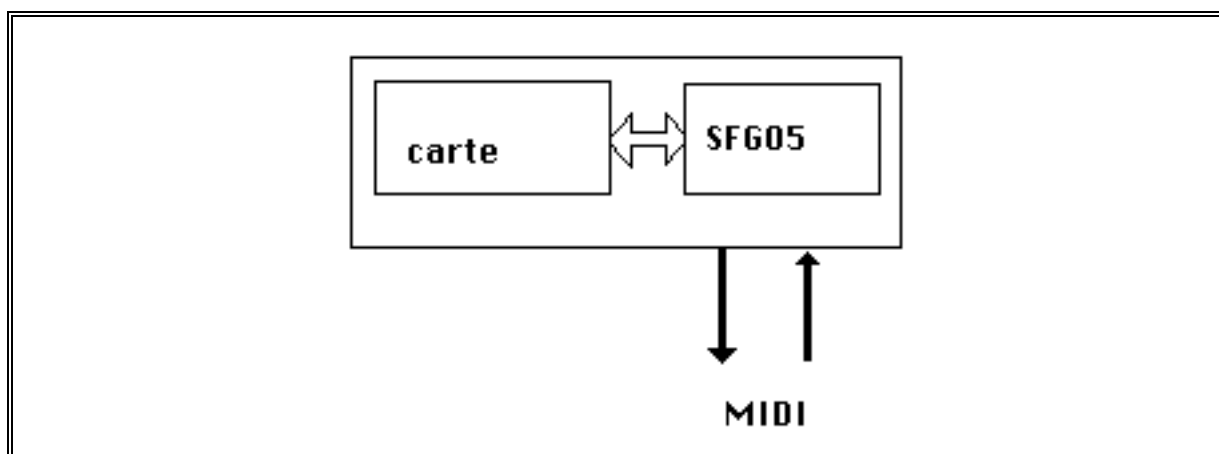
<sup>4</sup> "expand": module de synthèse qui émet un signal sonore (le corps du son) mais qui ne possède pas de clavier. L'ensemble des informations lui parvient par l'intermédiaire du canal MIDI. On peut également changer les paramètres du son par le MIDI qui est alors interprété comme une commande.



Malheureusement ce module ne pouvait pas se dissocier du micro-ordinateur CX5M (YAMAHA). D'où l'idée de chercher à le séparer grâce à la réalisation d'une carte et de le rendre autonome afin de le piloter par n'importe quel autre ordinateur possédant un interface MIDI.

Cela semblait possible car le module SFG05 possédait un moniteur d'entrée/sortie spécialement conçu pour des applications musicales pouvant contrôler les codes MIDI et les paramètres de synthèse. Le moniteur ou MBIOS (Musical Basic Input Output Service) est structuré sous la forme de fonctions pouvant être appelées à partir d'un langage machine.

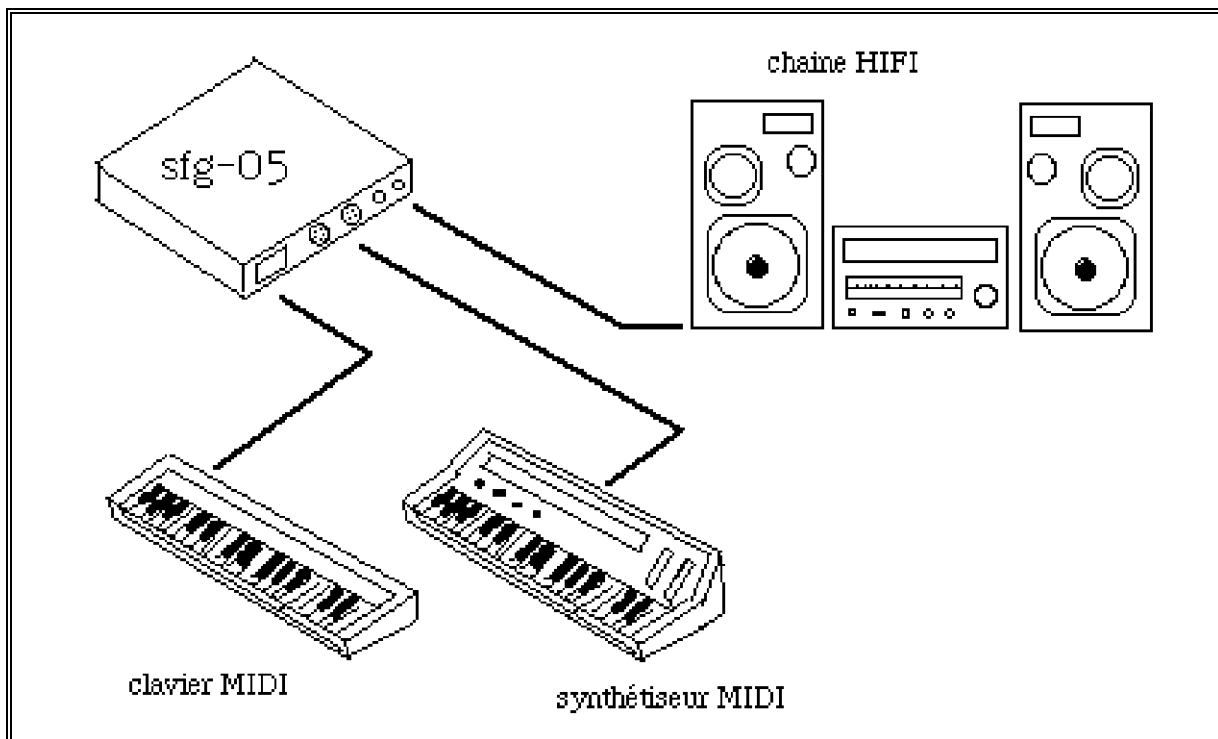
La carte de contrôle du SFG05 a pour but de transformer le SFG05 en véritable "expander" pouvant être piloté par toutes les machines pourvues d'un interface MIDI.



*Connexion vers l'extérieur par liaison MIDI*

Ce type d'appareil est maintenant couramment employé comme module de synthèse autonome. Un seul clavier peut piloter plusieurs modules grâce aux codes MIDI et notamment grâce au numéro de canal MIDI.

Le synthétiseur utilisé est le "YAMAHA FM SOUND SYNTHESIZER UNIT II SFG-05". Ce module est normalement connecté au micro-ordinateur "CX-5M" du même constructeur. Le but recherché est de pouvoir utiliser les possibilités sonores du synthétiseur sur d'autres micro-ordinateurs, par exemple sur un compatible PC, sans passer par l'achat relativement coûteux du système proposé par "YAMAHA", et comportant un ordinateur spécial.



*Utilisation familiale d'un ordinateur*

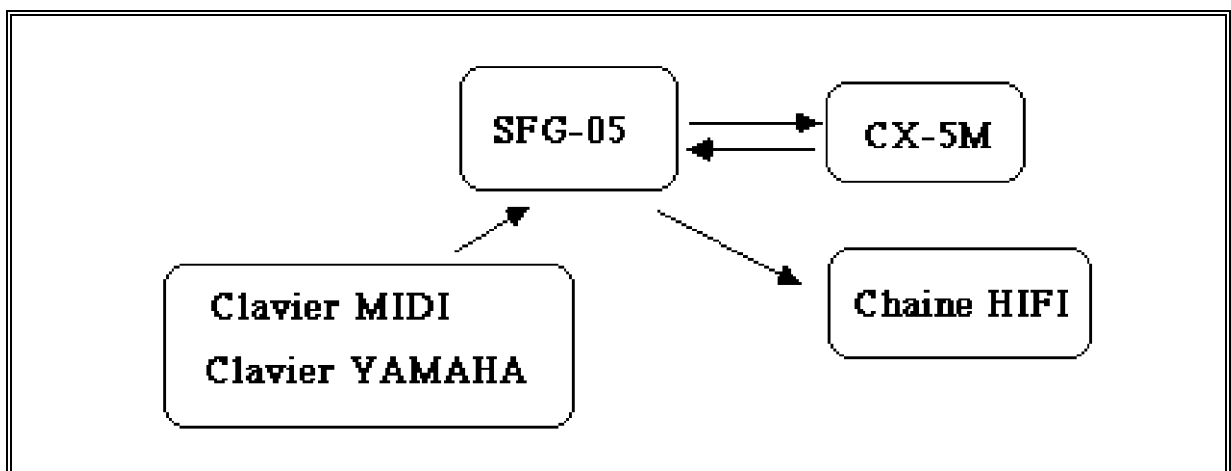
Parmi toutes les possibilités du "SFG-05", les possibilités de programmer des instruments soi-même, de modifier les valeurs des paramètres en temps réel et les connexions MIDI sont les plus intéressantes à retenir. Il faut rappeler qu'à cette période il n'existait pas de module de synthèse bon marché ayant la qualité du DX7, à part le module "SFG-05" couplé au micro-ordinateur "YAMAHA CX-5M". Il semblait évident à cette époque que le couplage de modules de ce type, soit vers des claviers MIDI, soit vers des micro-ordinateurs était une voie d'avenir dans l'informatique musicale. Cela s'est avéré exact car depuis, le marché est journellement inondé de modules de ce type couramment appelés "expandeurs". Le but de la recherche autour du "SFG-05" était de le désolidariser du micro-ordinateur afin de le rendre compatible avec tous les micro-ordinateurs du moment. Le but atteint, nous aurions eu à notre disposition un module de synthèse de bonne qualité, pouvant se connecter sur n'importe quel ordinateur. L'ordinateur peut à ce moment sélectionner les sons du module et envoyer des notes ou lire un clavier MIDI. L'ordinateur peut devenir un séquenceur, une aide à la composition musicale, un outil pédagogique pour l'apprentissage musical, tout cela en fonction du programme que l'on utilisera.

Cette recherche a duré environ six mois; car elle était rendue difficile par le secret professionnel entourant la construction du module. Néanmoins, j'ai pu proposer une solution au problème posé. Au bout du sixième mois YAMAHA a mis en vente une machine similaire, le "TX-01", qui possédait des caractéristiques techniques voisines de celles du

"SFG-05": le "TX-01" était un élément isolé pouvant se connecter à un clavier MIDI ou à un ordinateur par l'intermédiaire d'un cordon MIDI. La norme MIDI offre des possibilités de dialogue entre plusieurs synthétiseurs mais aussi entre un ordinateur et un module de synthèse.

## IV.2. Principe de fonctionnement du SFG-05

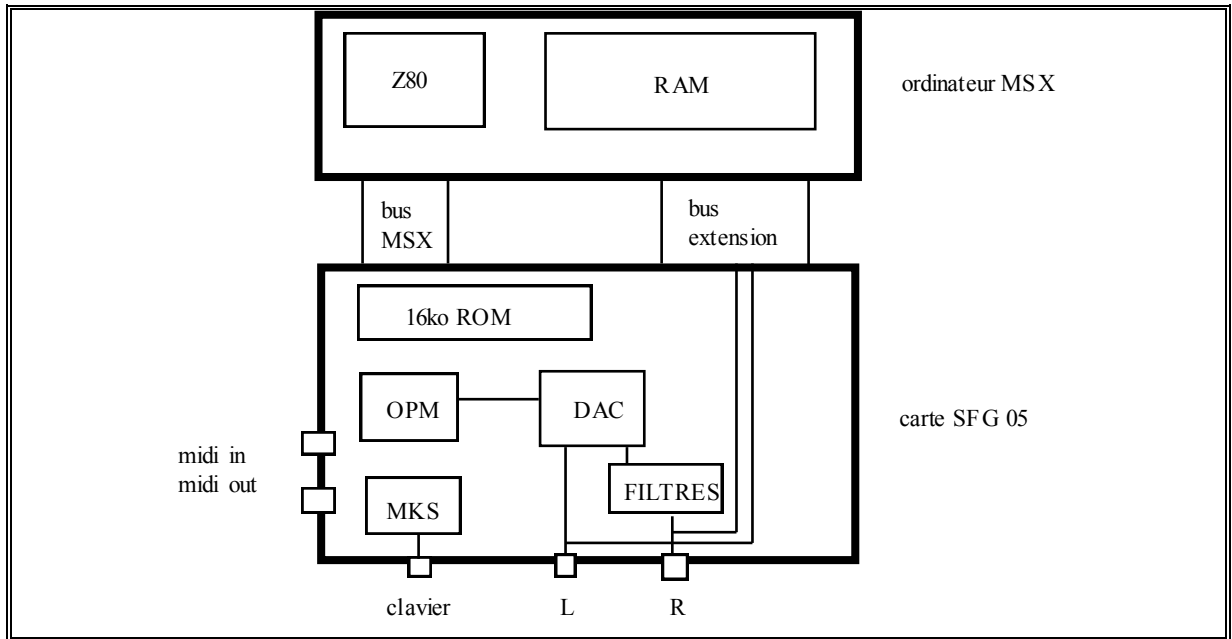
La génération des sons se fait grâce à la modulation de fréquence, méthode déjà employée par la gamme des synthétiseurs DX, qui donne des sons très réalistes et d'une qualité exceptionnelle pour le prix du module de synthèse. Le module de base s'appelle un opérateur (chez YAMAHA). Un opérateur est un générateur numérique de forme d'ondes dont l'amplitude est contrôlée par un générateur d'enveloppes. La sortie d'un opérateur est appliquée à l'entrée fréquence d'un autre opérateur créant ainsi une grande variété d'effets dus à la modulation de fréquence. Les sorties des opérateurs peuvent être associées en parallèle pour une synthèse additive ou en série pour une combinaison de synthèse FM. Les possibilités de création sont très grandes. On peut jouer simultanément huit notes, avoir 46 instruments pré-programmés et la possibilité d'en programmer soi-même, modifier les valeurs des paramètres en temps réel, s'accompagner d'un rythme pendant que le module scrute les notes d'un clavier normal ou d'un clavier muni d'un interface MIDI (Musical Instrument Digital Interface).



L'architecture interne du "CX-5M" est basée sur le processeur Z80 qui nécessite une mémoire vive de 32Ko pour stocker les notes. La mémoire morte du "SFG-05", quant à elle, est une ROM de 32Ko, divisée en deux parties:

- le M-BIOS (MUSIC-BIOS) qui contrôle le matériel

- le M-MONITOR (MUSIC-MONITOR) qui est un programme de démonstration transformant le "CX-5M" en un véritable synthétiseur ou en studio d'enregistrement.

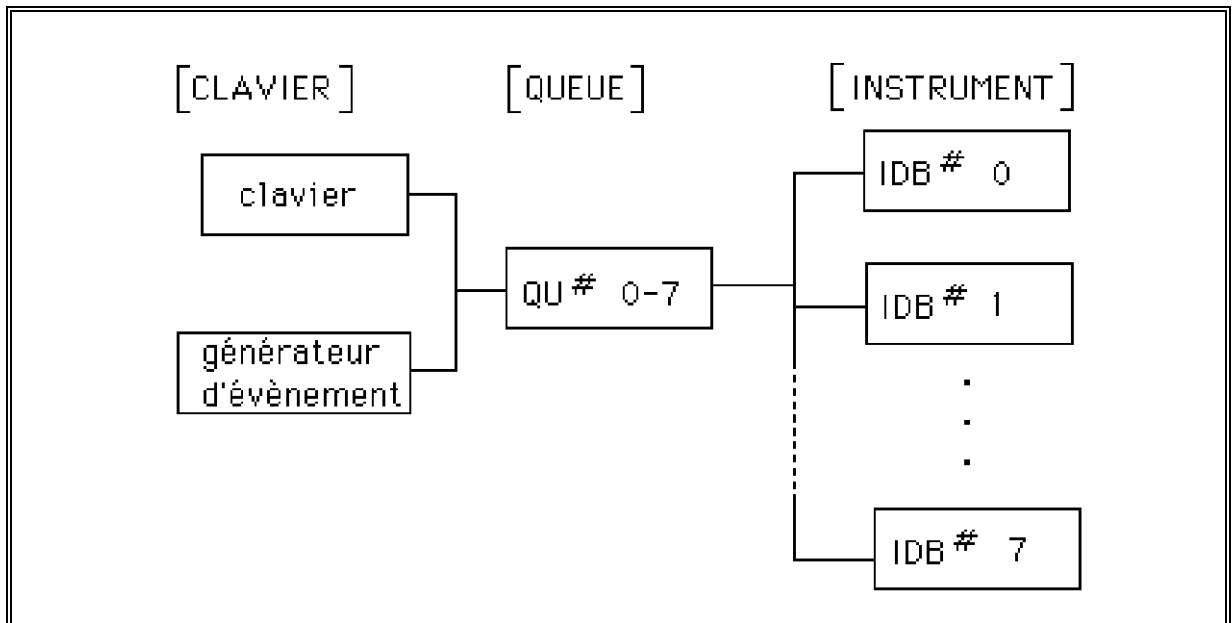


*Schéma interne de la carte de contrôle*

La qualité sonore est excellente, comparable à celle des synthétiseurs DX7.

### **IV.3. Fonctions de base**

Les entrées (claviers, générateurs d'évènements MIDI), les files de gestions ou queues et les instruments (générateurs de sons) forment la structure fondamentale du M.BIOS. La figure suivante montre les relations entre ces trois unités fonctionnelles.

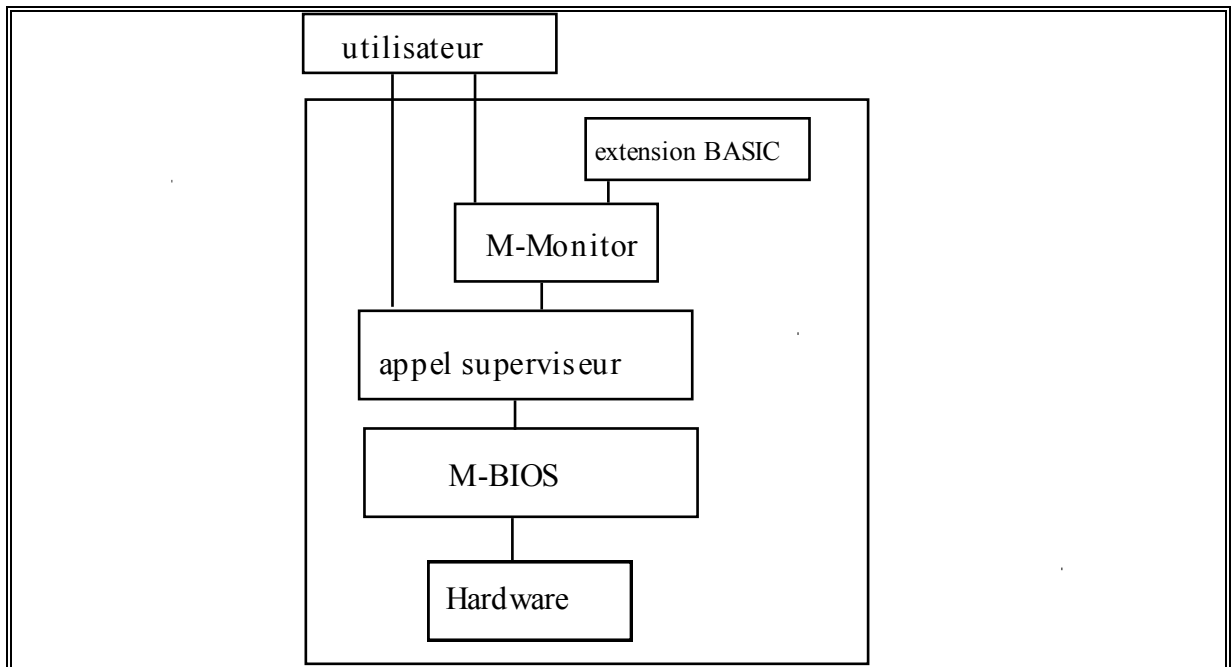


*Gestion des évènements*

Le clavier est une entrée pour le système. Il fournit des informations sur les notes (ON/OFF) auxquelles il associe la vitesse de la frappe. Lorsque la requête dépend du temps, on l'appelle événement. Les files sont l'intermédiaire entre les instruments et les événements d'entrées. L'instrument est le module qui joue les notes, il est défini par des blocs de contrôle ou IDB (Instrument Define Block). Chaque instrument peut être assigné à un ou plusieurs canaux de sortie. On a donc la possibilité de jouer de plusieurs instruments simultanément.

#### **IV.4. Configuration pour la programmation**

Sur ce graphique on peut suivre les différentes étapes pour parvenir à programmer le matériel. L'utilisateur s'adresse au superviseur, lui-même faisant appel à une fonction du moniteur d'entrée sortie.



Le transfert de données entre le superviseur et le MBIOS se fait par le biais des registres internes du Z80 ou par des tables en mémoire accessibles par le programme utilisateur.

#### **IV.5. Plan détaillé de la carte (SFG-05)**

(Voir page suivante)



## IV.6. Possibilité du montage

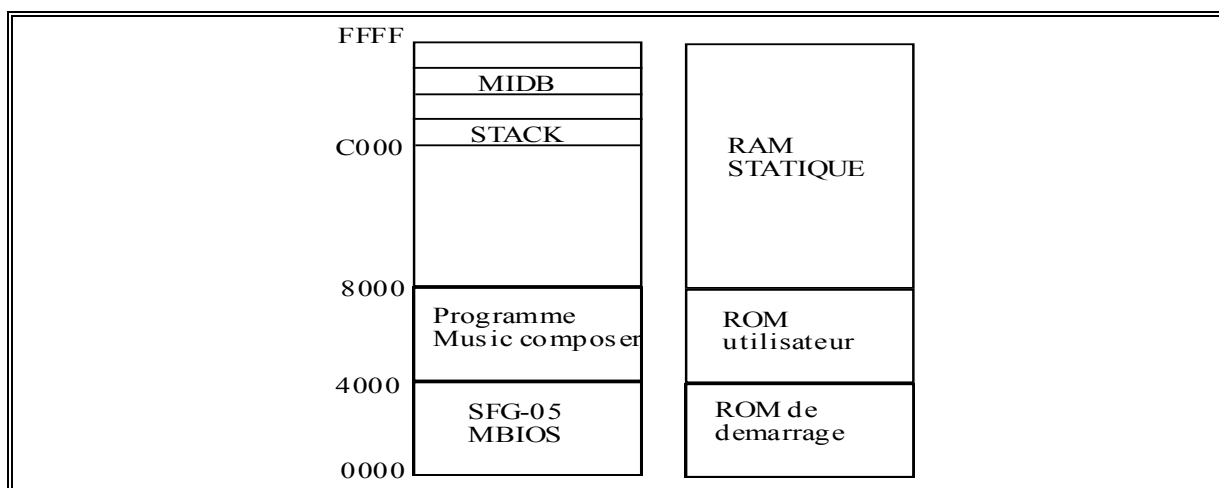
Dans l'état actuel du montage, une ROM (read only memory) placée aux adresses basses de la mémoire permet au Z80 de trouver l'amorce (boot) destinée au démarrage du programme. Le programme qui détermine le fonctionnement de la carte est transféré dans une deuxième ROM à l'aide d'un programmeur. La mise au point de ce programme se fait directement sur l'ordinateur YAMAHA grâce à un compilateur qui interprète les codes en mnémorique Z80 puis les exécute directement. Ce choix d'une ROM pour recevoir le programme définitif fige les propriétés de la carte: c'est pourquoi une version future devait permettre le téléchargement de n'importe quel code exécutable dans de la mémoire vive située à l'emplacement de la ROM.

Le programme test choisi pour vérifier le bon fonctionnement des différentes parties de la carte est le suivant:

Dans un premier temps: initialisation d'un instrument, du volume et du canal MIDI. Puis dans un deuxième temps: scrutation du clavier et jeu de la note sélectionnée.

### IV.6.a. mapping de la mémoire

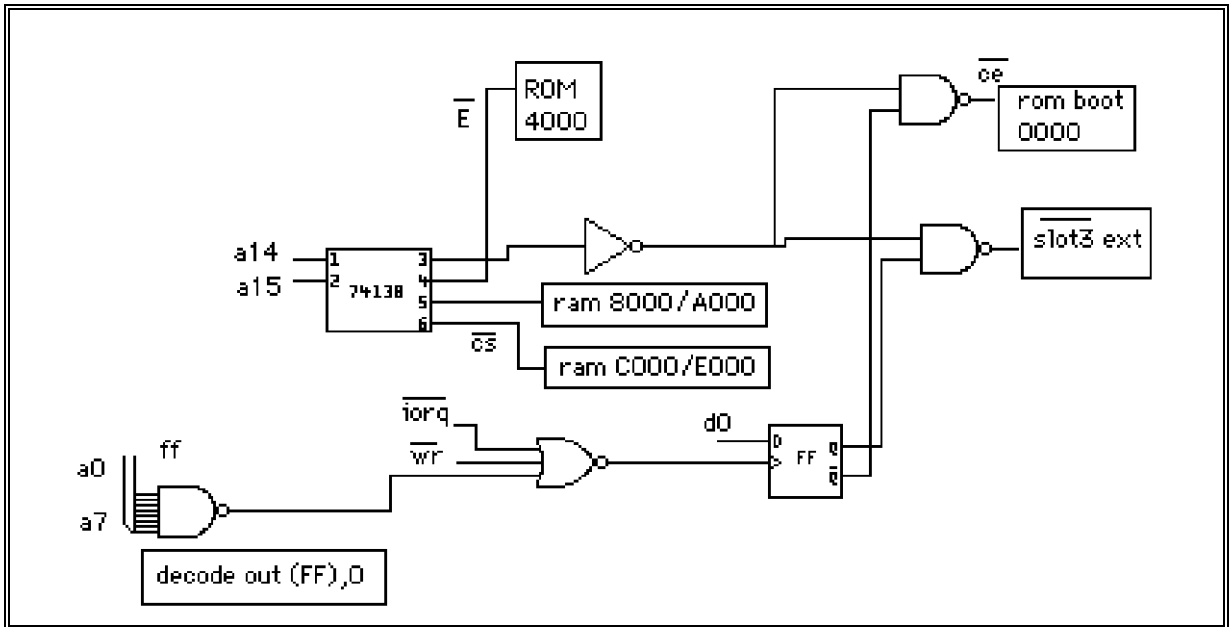
Une particularité de la mémoire provient de l'utilisation d'une zone commune, aux adresses basses. Le choix entre les différentes configurations se fait de manière logicielle par l'envoi d'un octet sur un port de sortie. L'électronique servant au décodage de cette instruction est décrite plus loin.



*Description de la mémoire interne*

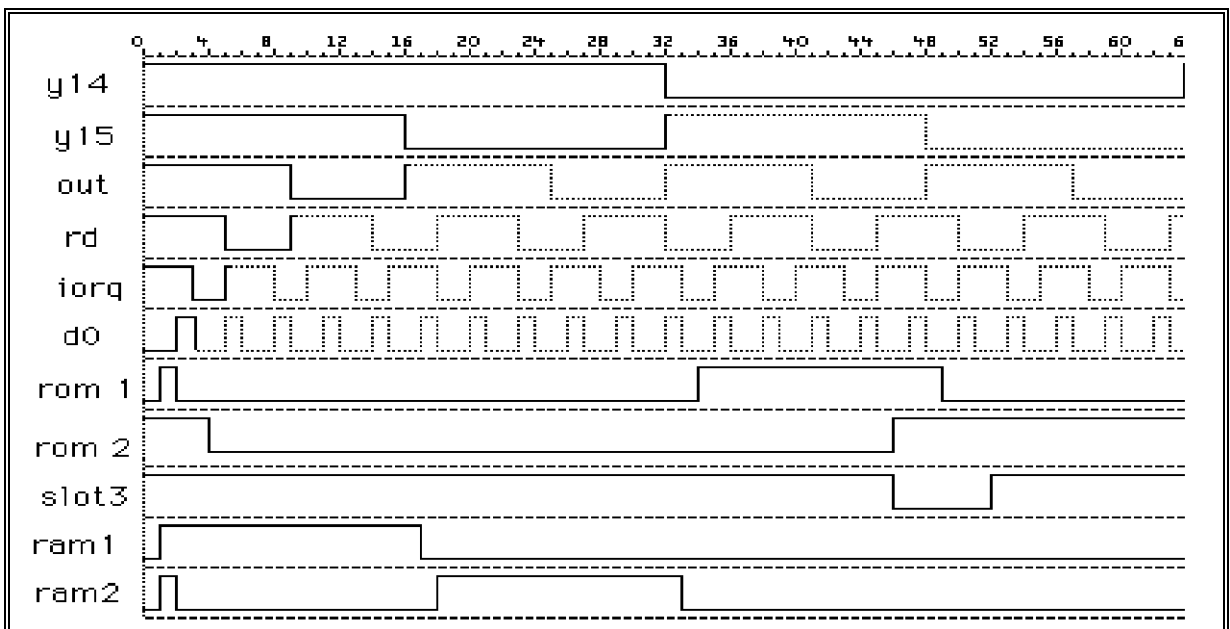
#### IV.6.a.1. décodage adresse





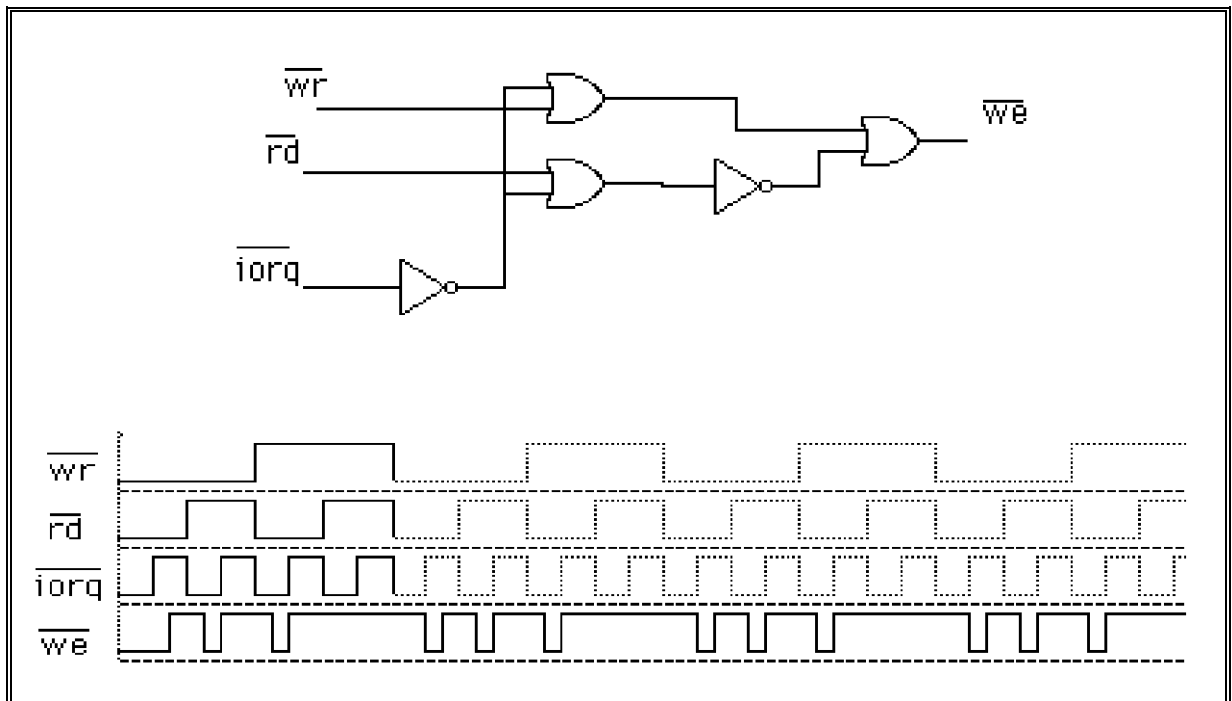
*logique du décodage d'adresses*

Le décodage des adresses est particulier pour les adresses basses. En effet le connecteur du SFG05 et la ROM de démarrage sont à la même adresse. C'est un artifice basé sur l'utilisation d'un ordre (OUT) vers un port de sortie qui permet de faire basculer à volonté la sélection de la ROM interne ou du module SFG05. Le décodage de l'instruction OUT du Z80 permet de réaliser cette sélection.



*Timing des sélections des différentes mémoires de la carte.*

#### IV.6.a.2. décodage du signal d'écriture



décodage du signal  $\overline{we}$

A partir des signaux issus du Z80, on construit le signal  $\overline{we}$  qui permet l'écriture dans les boîtiers mémoires. La sélection du boîtier est permanente.

#### IV.7. Exemples de programme

Toute l'expérimentation s'est faite sur l'ordinateur CX-5M en utilisant un clavier aux normes MIDI. Ce programme a permis le test de tous les signaux en provenance de la carte. Sa fonction était d'initialiser un instrument, de boucler en permanence sur la scrutation du clavier et d'envoyer la note vers le convertisseur.

```
ioppaequ a8h ;appel des routines superviseur
icallequ 90h
rcallequ 93h
kcallequ 96h
pcallequ 99h
scallegu 9ch
stackequ B000h
    org 9FFFh ;implantation du code
pt dw 0
buf dw 0
debutdi
    im 1
    ld sp,stack
    ld a,3
    out (ioppa),a
```

```

    call icall
;-----
    ld a,0                ;definition instrument
    ld b,1                ;define idb 1
    ld de,EA80h
    call scall
;-----
    ld a,0Ah
    ld b,1                ;assign queue 1
    ld de,8100h
    call scall
;-----
    ld a,15h
    ld bc,0104h          ;get voice (violon)
    call scall
;-----
    ld a,14h
    ld b,1                ;load voice
    call scall
;-----
    ld a,13h
    ld bc,01FFh          ;set volume (max)
    call scall
;-----
    ld a,9
    ld bc,100h           ;assign chanel
    ld de,0ABh
    call scall
;-----
bouc ld hl,table
    ld (pt),hl
bou  ld hl,(pt)
    ld a,(hl)
    inc hl
    ld (pt),hl
    cp 0FFh
    jr z,bouc
    ld b,1
    ld d,a
    ld e,80h
    ld a,2
    call rcall           ;
    ld a,0
    ld de,403h
    call pcall           ;sortie de la note
    call tempo           ;tempo validation touche
    jr bou
;-----
tabledb B0h,B1h,B2h,FFh
tempopush hl
    push af
    push bc
    push de
    ld hl,ffffh
    ld (buf),hl
temp2ld hl,(buf)        ;temporisation
    dec hl               ;pour la scrutation
    ld (buf),hl         ;du clavier
    ld a,h
    or l
    jr nz,temp2
    pop de
    pop bc
    pop af
    pop hl
    ret
end debut

```

La mise au point des différents programmes de tests m'a permis d'acquérir un savoir-faire dans le contrôle d'un composant de synthèse du commerce réalisant la synthèse de sons complexes, ainsi que dans la compréhension des phénomènes qui accompagnent le dialogue entre un ordinateur et une carte via le processus de commande MIDI. Ce savoir-faire a pu être mis à la disposition de musiciens ou de compositeurs comme cela fut le cas dans le cadre du projet TRANSGRAPH réalisé au sein de l'association Création et Réalisation en Informatique et Musique (CRIM).

















## V. BIBLIOGRAPHIE

- [1] ARFIB  
1985, "*Man-machine dialog using MIDI files*"  
Proceedings of the 1985 International Computer Music Conference, Computer Music Association, San Francisco, CA.
  
- [2] DROMAN, 1985  
"*Exploring MIDI: the Musical Instrument Digital Interface*"  
International MIDI Association Publications, North Hollywood, CA.
  
- [3] DUESENBERRY John  
"*the Yamaha CX5M Music Computer: An Evaluation*"  
Computer Music Journal, vol 9, 1985.
  
- [4] INTERNATIONAL MIDI ASSOCIATION  
1983, "*MIDI 1.0 specification*"  
International MIDI Association Publications, North Hollywood, CA.
  
- [5] LOY  
1985, "*Musicians make a standard: the MIDI phenomenon*"  
Computer Music Journal, vol 9, n°4.
  
- [6] MOOG  
1986, "*MIDI: Musical Instrument Digital Interface*"  
Journal of the Audio Engineering Society, vol 34, n°5.
  
- [7] RODNAY ZAKS  
"*Programmation du Z80*"  
Sybex

# VI. EXTRACTION DE PARAMETRES SONORES EN VUE DE LA RESYNTHESE ET SEPARATION SOURCE-RESONANCE

## VI.1. Position du problème

En Informatique Musicale, il est fréquent de constater que les musiciens qui désirent utiliser les techniques de l'informatique et du traitement de signal n'ont pas les connaissances suffisantes pour le faire. Assez régulièrement, des musiciens viennent consulter les personnes travaillant au Laboratoire d'Informatique Musicale dans le but d'apprendre plus en détail le fonctionnement des ordinateurs, afin de mieux exploiter leurs possibilités créatrices et soumettre une idée de traitement qui leur semble possible mais dont ils ignorent la faisabilité. Le dialogue qui s'installe est souvent bénéfique aux deux parties, étant donné que l'on trouve dans le laboratoire des personnes ayant une double compétence, à la fois musicale et scientifique de haut niveau, ainsi que des scientifiques dont les connaissances musicales sont suffisantes pour comprendre les désirs des musiciens. La demande des musiciens est de plus en plus importante depuis l'apparition sur le marché d'ordinateurs personnels et de cartes de conversion numérique-analogique à faible coût. Le musicien trouve à sa portée un outil adapté à la création musicale mais qui nécessite néanmoins quelques connaissances physiques du phénomène sonore. Il s'agit ici de comprendre le fonctionnement de modèles mathématiques de production d'échantillons sonores et non pas de l'étude de synthétiseurs aux normes MIDI pour lesquels une bonne lecture du manuel est suffisante pour débiter (A l'exception de certains synthétiseurs qui proposent la modification des paramètres de synthèse et qui rejoignent ainsi les problèmes posés par les ordinateurs, c'est-à-dire une bonne compréhension du phénomène physique).

Il paraît évident que l'utilisation d'un ordinateur pour la création musicale nécessite l'emploi d'un langage informatique commun permettant de partager les connaissances accumulées par de nombreux utilisateurs depuis la naissance de l'Informatique Musicale. Depuis les années 60, le langage MUSICV, qui est le langage de synthèse et de traitement des sons par ordinateur, conçu et écrit par Max V. Mathews et ses collaborateurs, aux laboratoires Bell, permet la réalisation d'algorithmes de calcul de sons dans une syntaxe particulière. Il existe beaucoup de versions de MUSICV implantées sur de gros ordinateurs, et pour faire face au développement de la micro-informatique, l'équipe d'informatique musicale, par

l'intermédiaire de Daniel Arfib, a installé une version du compilateur sur de petites machines (IBM-PC). On peut, à volonté, avec MUSICV écrire une musique de style "classique" ou de style plus contemporain. Dans la musique "classique", l'écriture des partitions est bien généralisée par l'utilisation des notes et de toutes les notations fréquemment employées à l'intérieur d'une partition. Dans la musique "contemporaine" le codage est plus difficile car l'écriture de la partition est souvent liée à la description physique du son. Par exemple les fréquences des notes ne rentrent plus nécessairement dans une gamme connue. On peut jouer sur des micro-tons. La plupart des compositeurs utilisent un langage semi-graphique pour représenter la variation du jeu et parsèment leurs partitions d'indications chronologiques, etc... D'un compositeur à l'autre, le même signe peut avoir des significations différentes. L'intérêt d'utiliser MUSICV réside dans le fait que les sons seront décrits dans un langage accessible à tous: leurs descriptions seront figées dans la partition. L'ensemble de toutes les partitions MUSICV forme une large base de données et constitue un savoir-faire commun à tous les utilisateurs. C'est pourquoi, au cours du présent ouvrage, on trouvera des partitions MUSICV qui serviront d'exemples ou qui décriront un moyen d'utiliser des travaux intermédiaires. L'ensemble de toutes ces informations et des programmes les accompagnants constitue une "boîte à outils" pour MUSICV.

Une des difficultés cruciales dans l'utilisation des programmes de synthèse consiste à mettre en correspondance les paramètres de synthèse avec des paramètres physiques correspondant à la réalité d'un son naturel donné. Ceci est pourtant primordial car il est important pour le musicien de pouvoir simuler ou se rapprocher de sons naturels et d'effectuer des modifications intimes sur ces derniers, au-delà de ce qu'a décrit Pierre Schaeffer dans son "traité des objets musicaux" [SCHAEFFER, 1966]. Pour donner de telles potentialités au musicien il est nécessaire de mettre en oeuvre des méthodes d'analyse des sons musicaux à partir d'une "expérimentation" facilement réalisable en studio, d'en extraire des paramètres physiques correspondants à une réalité psychoacoustique puis d'associer à ces derniers des paramètres de synthèse.

Bien entendu, il n'existe pas de méthode d'analyse synthèse universelle, et nous nous limiterons dans la suite de ce document à l'étude de techniques d'analyses-synthèses correspondant à une modélisation physique d'une classe particulière d'instrument de musique: les instruments à formants fixes. Un tel modèle englobe tous les instruments de musique dont la production sonore est engendrée par un système d'excitation (cordes frottées, frappées, vibration d'une anche, etc....) dont les ondes vibratoires se propagent à l'intérieur d'une structure résonante de forme fixe (table d'harmonie pour le piano, etc..). Nous verrons par la suite qu'il est ainsi possible de resynthétiser de tels signaux sonores à partir de techniques associant la distorsion non-linéaire (destinée à modéliser l'excitation) et le filtrage linéaire (destiné à modéliser la caisse de résonance).

Avant de poursuivre la description de la méthode utilisée pour la séparation source-résonance, je vais présenter de manière succincte quelques méthodes de synthèse, puis de façon plus détaillée les méthodes d'analyses employées dans la suite du travail, et notamment celles qui ont fait l'objet d'une modification comme la distorsion non linéaire ou la détection du fondamental d'une note.

## VI.2. les outils: méthodes d'analyse et de synthèse de signaux

### VI.2.a. La synthèse

L'utilisation d'un langage comme MUSICV permet de mettre en oeuvre toutes les méthodes de synthèse les plus couramment employées. Nous présentons ci-dessous les techniques les plus utilisées en insistant plus particulièrement sur la distorsion non-linéaire et les variantes que nous avons développées de façon à l'adapter à notre problème.

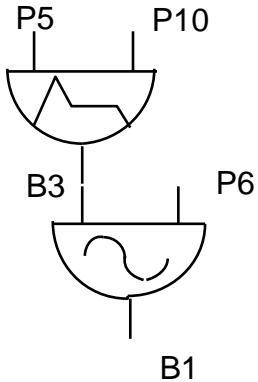
#### VI.2.a.1. synthèse simple

Cette technique consiste à moduler en amplitude un signal périodique  $x(t)$  par une enveloppe  $A(t)$  modélisant les variations globales de l'énergie au cours du temps.

$$S(t) = A(t) \cdot x(t)$$

```
SAM 25600 ;
CNV P6=HTZ(P6) ;
CNV P10=DUR(P4) ;
INS 0 1 ;
OSC P5 P10 B3 F1 P30 ;
OSC B3 P6 B1 F2 P29 ;
OUT B1 ;
END ;
GEN 0 1 1 1000 0 0 1 250 0.8 400 0.8 700
  0 1000 ;
GEN 0 2 2 1000 1 1 ;
NOT 0 1 1 5000 440 ;
TER 1 ;
```

enveloppe  
du signal



#### *Synthèse simple*

Elle correspond au modèle sonore de SCHAEFFER qui consiste à distinguer dans un son la "matière" et sa "forme".

#### VI.2.a.2. synthèse additive

Elle consiste à ajouter entre eux des éléments simples qui sont des signaux sinusoïdaux modulés en amplitude. Elle généralise la synthèse de Fourier où les fréquences sont harmoniques d'un fondamental.



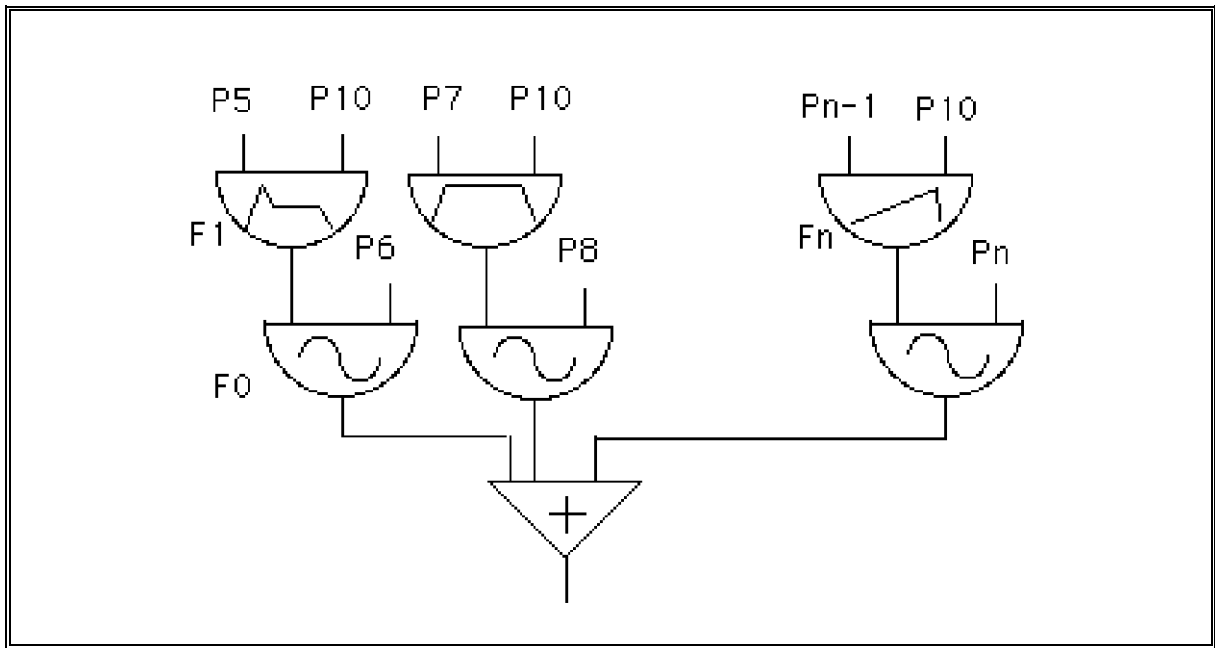


Figure 1 - La complexité du son provient de la possibilité d'ajouter un grand nombre de composantes sinusoïdales dont les paramètres d'amplitude, de fréquence et de phase peuvent varier indépendamment pour chaque composante.

### VI.2.a.3. synthèse soustractive

A partir d'une matière sonore, elle consiste à la sculpter à l'aide de filtres. De nombreux filtres peuvent être utilisés. Les plus usuels sont les filtres à réponse impulsionnelle limitée (FIR) qui permettent de prendre en compte la phase de sortie et les filtres à réponse impulsionnelle infinie (IIR): filtre de Tchebychev, Butterworth.

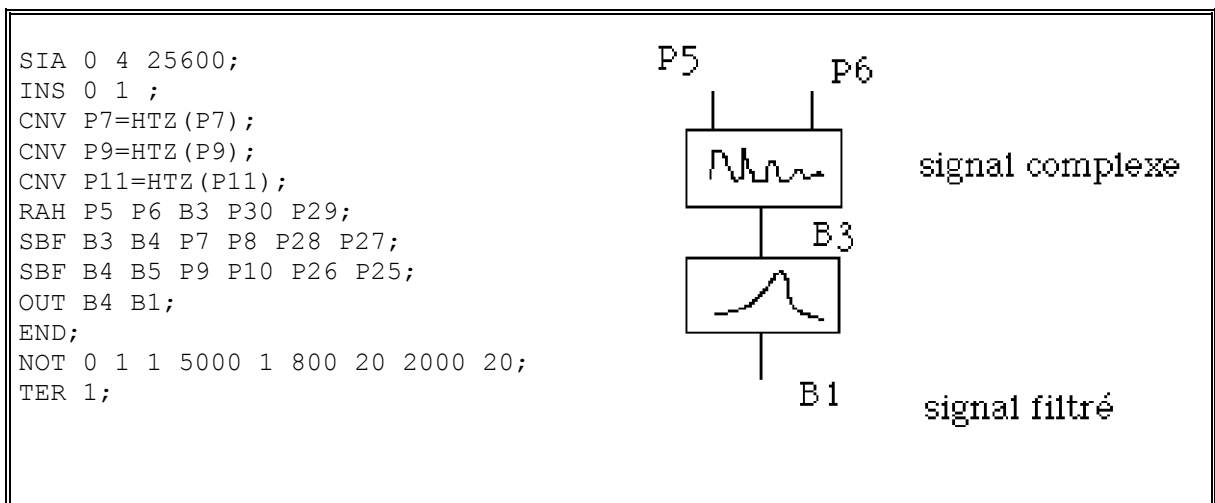


Figure 2 - La source est généralement un son riche en harmoniques (bruit blanc): le filtre possède un ensemble de paramètres qui détermine l'allure du spectre du signal de sortie.

## VI.2.b.Synthèse globale ou non-linéaire

### VI.2.b.1. modulation d'amplitude

Elle consiste à moduler en amplitude un signal généralement simple par un autre signal dont le rapport en fréquence est proche de 1. Le spectre du signal résultant est la convolution des spectres des 2 signaux d'entrée.[DODGE].

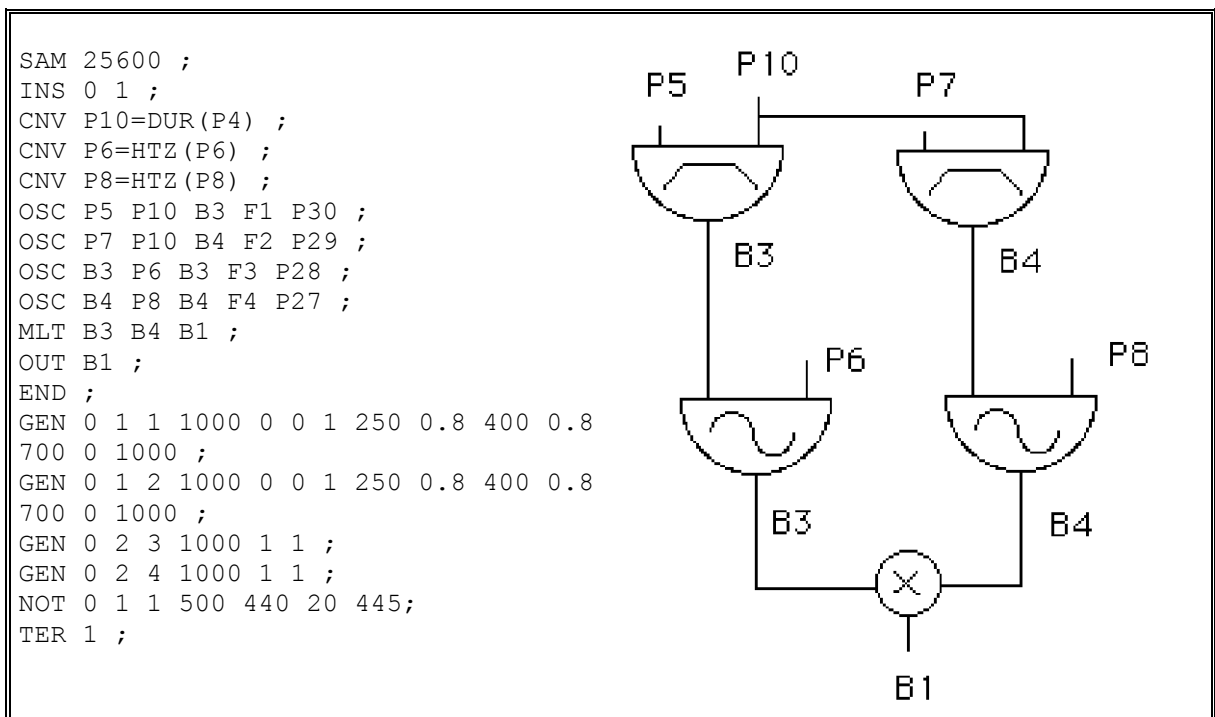


Figure 3 - Cette modulation d'amplitude est souvent employée par les musiciens pour créer un trémolo.

### VI.2.b.2. modulation de fréquence

Elle consiste à moduler en fréquence un signal, la fréquence de modulation étant une audiofréquence.

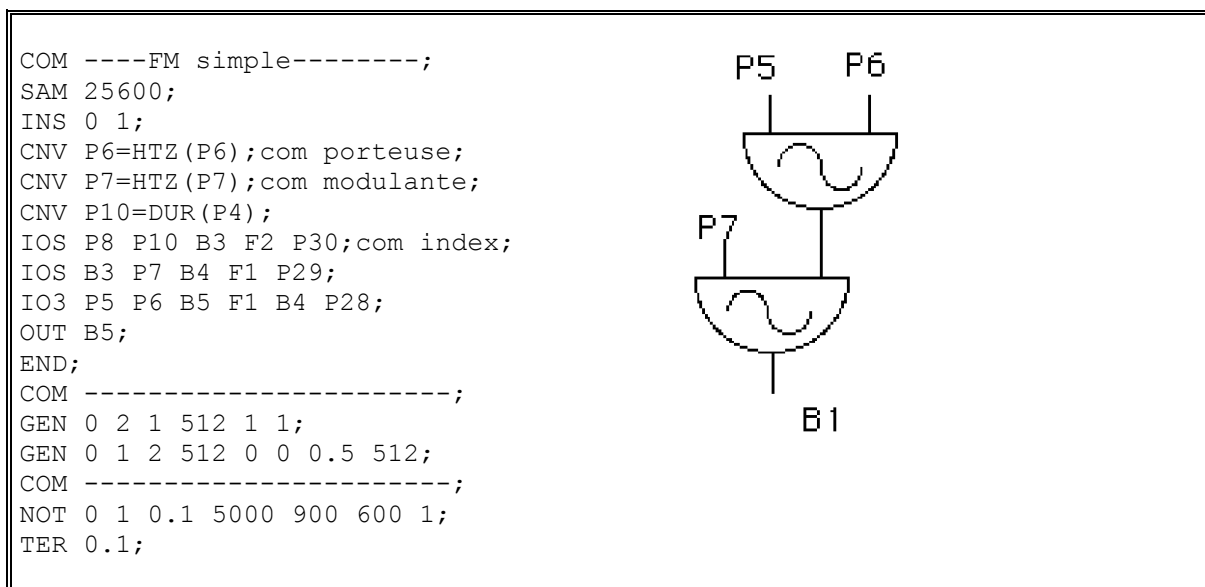


figure 4 - Cette technique est utilisé par la gamme des synthétiseur DX7.

Cette technique a été développée avec succès par John CHOWNING à partir des techniques de radio-communication. Elle permet la génération d'une grande quantité de spectres avec des timbres distincts qui peuvent être contrôlés facilement. Cette technique est utilisée par YAMAHA pour toute sa gamme de synthétiseurs (le plus connu: DX7).

### VI.2.b.3. distorsion non-linéaire

Développée par ARFIB et LE BRUN [1979], la distorsion non linéaire est une technique de synthèse permettant la création de spectres complexes pouvant évoluer dynamiquement au cours du temps. Elle consiste à distordre des sinusoides à l'aide de fonctions non linéaires. Cette méthode est issue de l'électronique ou à l'aide d'un mauvais amplificateur on pouvait créer des harmoniques dont les amplitudes dépendaient du signal sinusoidal d'entrée. Cette technique a été remise au goût du jour en Informatique Musicale par D. Arfib<sup>7</sup>. C'est une technique peu coûteuse en temps de calcul et très facile à mettre en oeuvre sur un ordinateur.

---

<sup>7</sup> Doctorat d'état, "L'usage musical de la synthèse de spectres complexes par distorsion non linéaire et modulation d'amplitude".

L'essentiel de la méthode consiste à créer une table contenant une fonction de distorsion et à lire cette table à l'aide d'une sinusoïde d'entrée ayant une amplitude donnée.

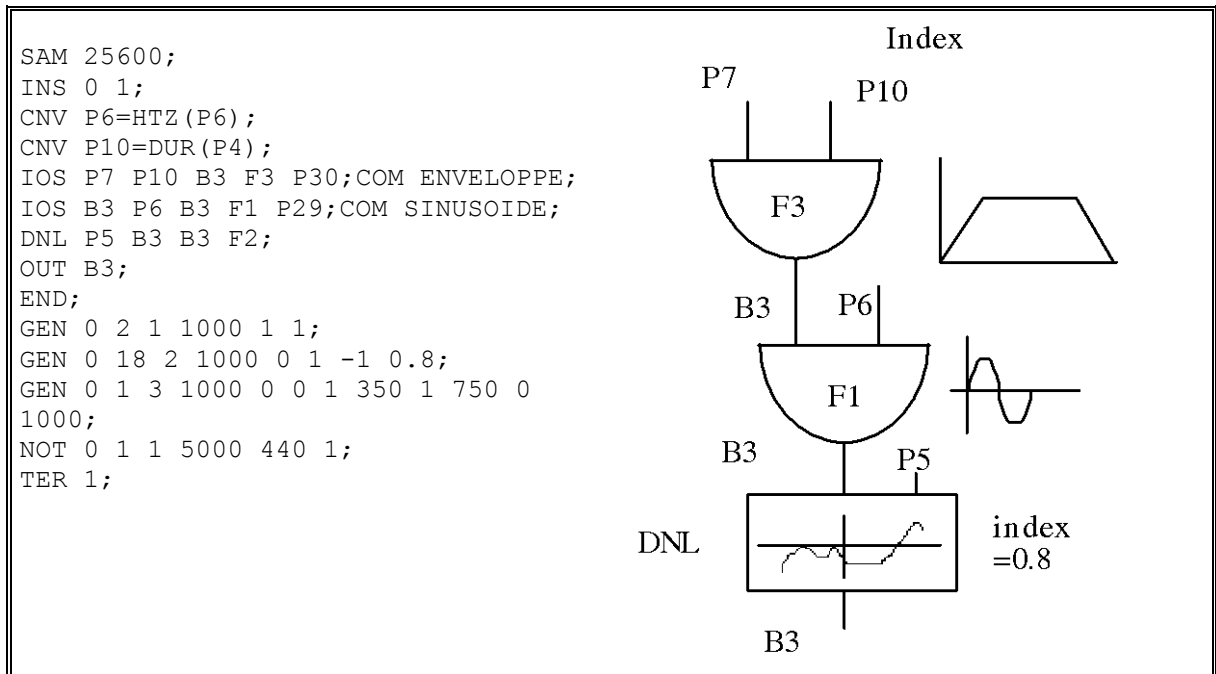


figure 5 - La distorsion calculée pour un index égal à 0.8 permet d'enrichir le spectre du signal de sortie en harmoniques lorsque l'index dépasse cette valeur.

On peut calculer à l'avance l'allure du spectre de sortie pour une fonction de distorsion et une valeur de l'amplitude en entrée. Inversement pour obtenir un spectre donné il est possible de calculer une fonction de distorsion pour une valeur de l'amplitude d'entrée.

Le paramètre d'amplitude est généralement appelé index de distorsion, cette appellation provient de son utilisation par John Chowning pour décrire un paramètre influant sur le timbre. En effet cette méthode s'apparente par certains aspects à la modulation de fréquence.

Lorsque l'index évolue au cours du temps cela provoque une variation du spectre en sortie. Cette propriété a été utilisée par J.C. Risset [1969] dans son catalogue de sons pour définir un instrument semblable à la clarinette ainsi que dans la pièce numérique de D. Arfib [1979].

Une des propriétés de la distorsion non linéaire qui ont fait son succès est qu'une fonction polynomiale de degré  $n$  engendre un spectre ayant au maximum  $n$  harmoniques. On peut donc engendrer  $n$  harmoniques non pas avec  $n$  oscillateurs (synthèse additive) mais avec un seul module de distorsion. La réciproque est vraie: à un spectre possédant  $n$  harmoniques on peut associer une fonction polynomiale de degré  $n$ . Par la suite on n'utilisera que des fonctions de distorsion provenant d'une combinaison linéaire de polynômes de Chebyshev de degré  $n$  qui forment une base orthogonale.

Soit  $e(t)$  un signal d'entrée et  $f$  une fonction de distorsion, la sortie  $s(t)$  s'exprime par:

$$s(t) = f[e(t)]$$

Dans le cas qui nous intéresse  $e(t)$  est de la forme :

$$e(t) = I(t) \cdot \cos(\omega t)$$

où  $I(t)$  est l'index de distorsion en fonction du temps.

Pour un index égal à 1 si on écrit  $s(t)$  sous la forme:

$$\begin{aligned} s(t) &= 1/2h_0 + h_1\cos(\omega t) + h_2\cos(2\omega t) + \dots + h_n \cos(n\omega t) \\ &= 1/2h_0 + \sum_{k=1}^n h_k \cos(k\omega t) \end{aligned}$$

Les termes en  $\sin(k\omega t)$  sont nuls pour ce choix de  $e(t)$ .

les  $h_0, \dots, h_n$  sont les coefficients de l'analyse de Fourier du signal de sortie, en utilisant les polynômes de Chebyshev définis par la relation :

$$T_n(\cos(\omega t)) = \cos(n\omega t) \text{ et en posant } \cos(\omega t) = x:$$

il vient

$$s(t) = 1/2 h_0 T_0(x) + h_1 T_1(x) + h_2 T_2(x) + \dots + h_n T_n(x)$$

Or il existe une relation de récurrence permettant de calculer  $T_n$  en fonction de  $T_{n-1}$  et  $T_{n-2}$  qui est la suivante:

$$T_n(x) = 2x T_{n-1} - T_{n-2}$$

On a donc un moyen simple de calculer par récurrence la somme des polynômes de Chebyshev pondérée par les valeurs des coefficients des harmoniques du signal de sortie, de manière à obtenir un polynôme  $f(x)$  de degré  $n$  en  $x$ , tel que si l'on introduit un signal d'entrée  $e(t)$  égal à  $\cos(\omega t)$ , on aura en sortie un signal  $e(t) = f(\cos(\omega t))$  ayant un spectre dont les composantes seront les coefficients  $h_n$  du départ.

$$T_0(x) = 1$$

$$T_1(x) = x$$

$$T_2(x) = 2x^2 - 1$$

$$T_3(x) = 4x^3 - 3x$$

$$T_4(x) = 8x^4 - 8x^2 + 1$$

$$T_5(x) = 16x^5 - 20x^3 + 5x$$

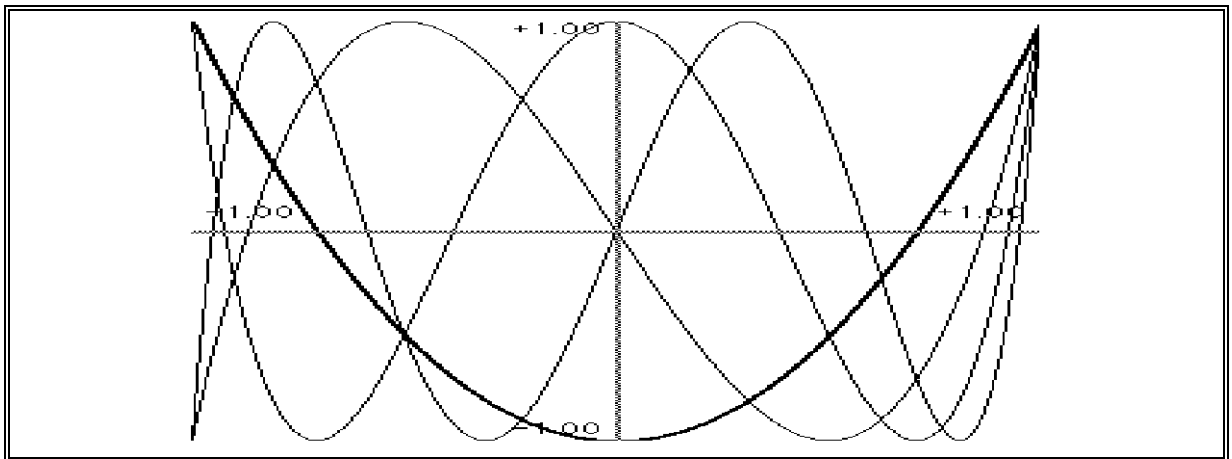


figure 6 - fonctions de Chebyshev du premier ordre de rang 2,3,4,,5.

La résolution numérique consiste à combiner le calcul des polynômes de Chebyshev avec leur somme pondérée pour obtenir directement les valeurs de la fonction de transfert. Ces valeurs sont stockées dans une table.

Si maintenant on fait varier l'index de distorsion entre 0 et 1 et si l'on observe l'allure du spectre de sortie, on constate que son évolution est lente au début puis brutale vers la fin. Cette évolution dépend beaucoup des signes des harmoniques.

Les valeurs des harmoniques  $h_n$  peuvent être affectées d'un signe + ou - correspondant à une rotation de phase de  $180^\circ$ . D. Arfib [1983] a montré de manière empirique qu'une alternance ++,-- conduisait à une variation plus régulière de l'évolution du spectre en fonction de l'index.

J'ai généralisé l'approche de Daniel Arfib en tenant compte de la phase du signal. Cette méthode a fait l'objet d'un nouveau module dans MUSICV. L'intérêt est double: on constate contrairement à la méthode empirique de l'alternance des signes une variation plus régulière de l'évolution du spectre en fonction de l'index, et de plus le signal obtenu est visuellement plus proche du signal de départ (si l'on utilise les valeurs des harmoniques provenant de l'analyse d'un signal réel). En effet l'alternance des signes introduit des déphasages sur la reconstruction du signal qui ne sont pas gênants pour l'écoute mais qui peuvent l'être pour un "micro-chirurgien" du son qui a l'habitude d'observer sur un écran d'ordinateur les résultats de ses manipulations physiques sur le signal.

Si on écrit  $s(t)$  sous la forme:

$$s(t) = 1/2h_0 + \sum_{k=1}^n H_k \cos(k\omega t - \phi_k)$$

$$= 1/2h_0 + \sum_{k=1}^n H_k [\cos(k\omega t) \cos(\phi_k) + \sin(k\omega t) \sin(\phi_k)]$$

alors, de même que l'on a introduit  $T_k$  pour  $\cos(k\omega t)$ , on introduit une nouvelle fonction  $S_k$  pour  $\sin(k\omega t)$  :

si on pose  $h_k = H_k \cos(\phi_k)$  et  $\overline{h_k} = H_k \sin(\phi_k)$

$$s(t) = \sum_{k=1}^n [h_k T_k + \overline{h_k} S_k]$$

Si on pose  $y = \overline{x} = \pm\sqrt{(1-x^2)} = \sin(k\omega t)$  on peut de la même manière que pour les polynômes de Chebyshev définis précédemment calculer les  $S_n$  tels que  $S_n = \sin(n\omega t)$

$$S_0 = 0$$

$$S_1 = y$$

Les fonctions  $S_n$  satisfont à la même relation de récurrence que les  $T_n$ :

$$S_{n+1} = 2xS_n - S_{n-1}$$

Ces fonctions  $S_n$  sont appelées les polynômes de Chebyshev du second ordre.

$$S_0 = 0$$

$$S_1 = y$$

$$S_2 = y \cdot (2x)$$

$$S_3 = y \cdot (4x^2 - 1)$$

$$S_4 = y \cdot (8x^3 - 4x)$$

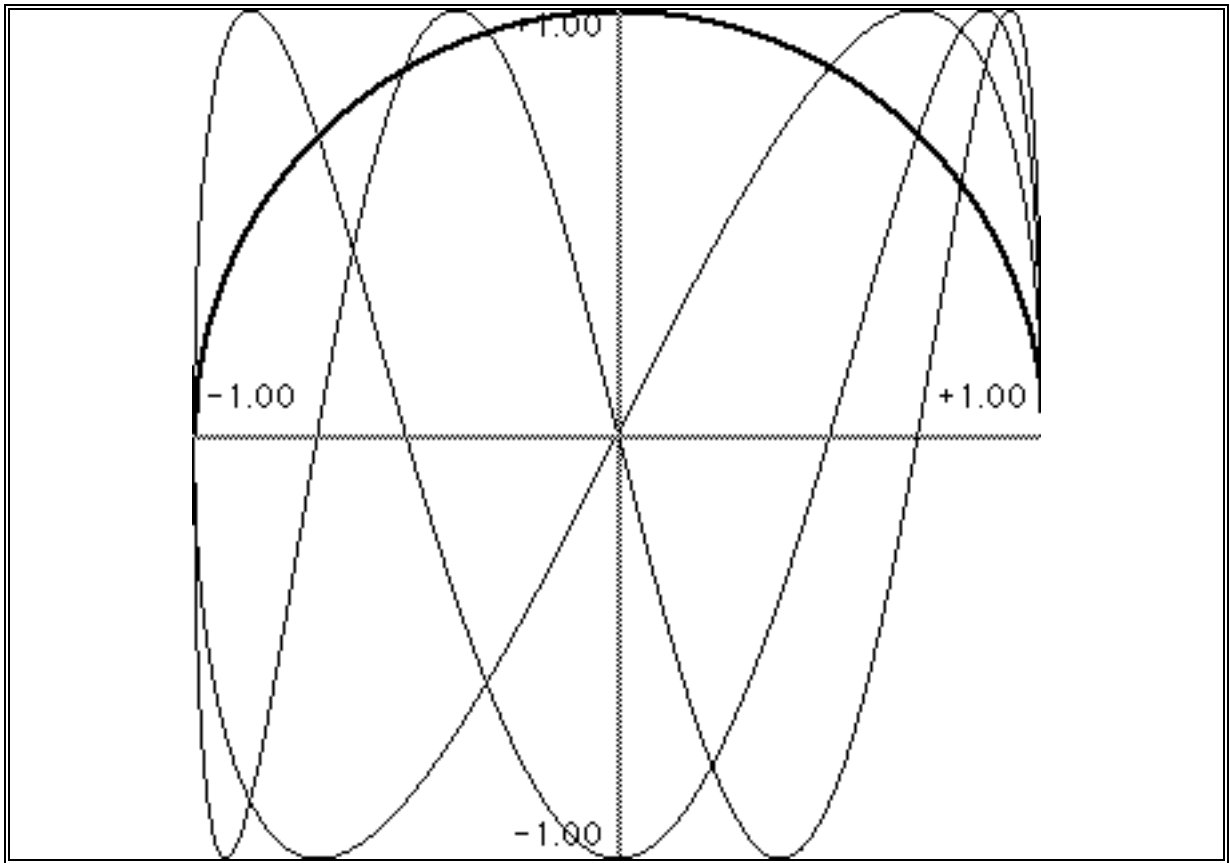


figure 7 - fonctions de Chebyshev du second ordre de rang 1,2,3,4.

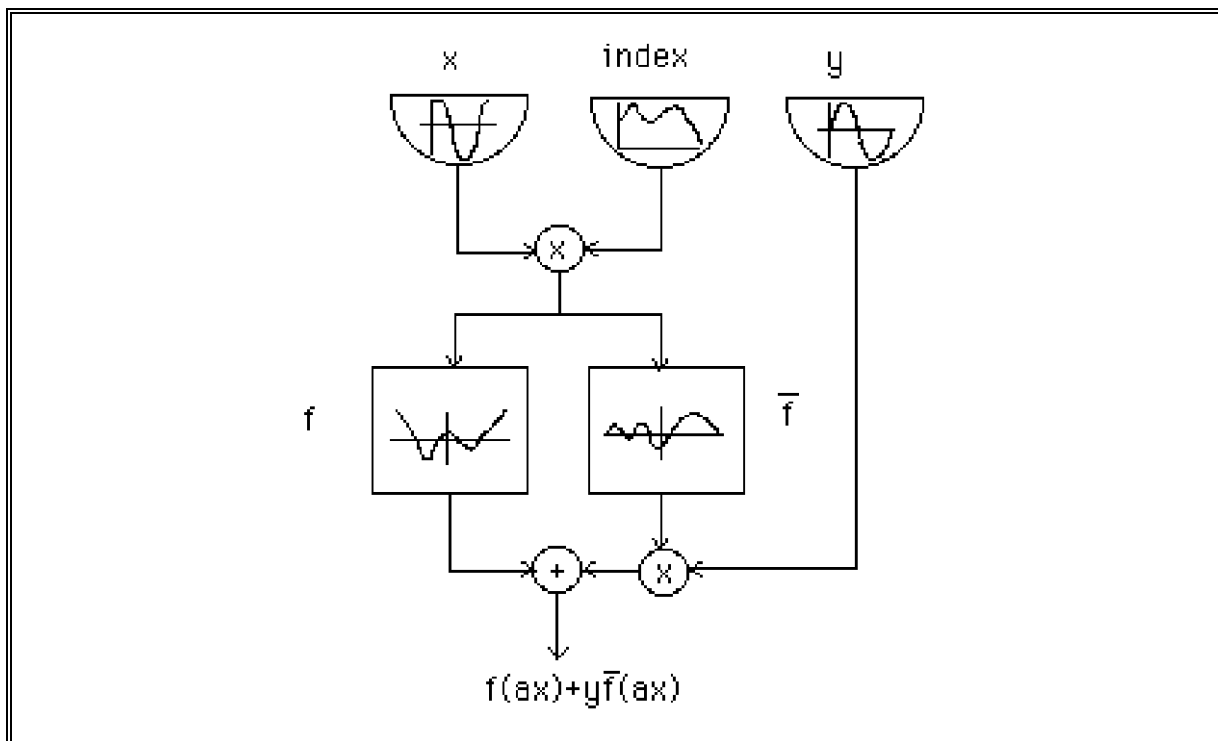
La mise en oeuvre numérique est identique à celle des polynômes de Chebyshev du premier ordre, mais cette fois on tient compte de la phase de chaque harmonique. Si les coefficients  $h_k$  proviennent de l'analyse de Fourier d'un son réel, alors la forme du signal reconstruit sera identique à celle du signal d'origine, ce qui n'était pas le cas précédemment.

Si  $f$  est la fonction de distorsion obtenue avec les polynômes de Chebyshev du premier ordre et  $\overline{f}$  la fonction de distorsion obtenue avec les polynômes de Chebyshev du second ordre,  $a$  étant l'index, le signal  $s(t)$  s'exprime par:

$$s(t) = f(ax) + y \overline{f}(ax)$$

L'instrument MUSICV qui réalise cette synthèse est le suivant:





*figure 8 - instrument pour synthèse par distorsion non linéaire*

Pour reconstruire le signal dans sa totalité, on pourrait imaginer un découpage par tranche et calculer à l'aide de FFTs sur chaque morceau autant de fonctions de distorsion et de valeurs d'index. Cette solution est trop fastidieuse et trop coûteuse en place mémoire: on lui préfère une solution plus douce qui consiste à calculer l'évolution du barycentre des fréquences au cours du temps. On déduit à partir de cette connaissance une table d'index qui restitue au mieux le barycentre du spectre original.

Cette technique crée un bon critère psycho-acoustique qui met en relation le son réel et le son synthétisé. Ce critère a été proposé par Beauchamp et Arfib. Il consiste à mettre en relation l'index de distorsion avec le barycentre du spectre qui évolue au cours du temps et cela mesure d'une certaine façon la brillance du son. Cette possibilité sera examinée plus loin.

### **VI.2.c. L'analyse**

J. C. Risset a développé une technique très probante qui consiste à effectuer l'analyse par la synthèse. Il s'agit d'affiner les paramètres de synthèse de façon à extraire des paramètres pertinents d'un point de vue auditif.

Ces méthodes font l'objet de modules spéciaux à l'intérieur de MUSICV, mais elles peuvent être également mises en oeuvre par l'agencement de modules de base.

L'analyse consiste en l'identification des paramètres de synthèse de façon à les utiliser pour des synthèses de type algorithmique et éventuellement, faire varier ces paramètres pour des transformations intimes sur le son.

L'analyse doit permettre d'effectuer une réduction des données tenant compte des particularités de la perception auditive. Dans certains cas, il devient intéressant de distinguer ce qui provient de l'excitation de ce qui provient de l'aspect résonant.

Il n'est pas question ici de faire un tour d'horizon exhaustif des techniques d'analyse de signaux, c'est pourquoi nous nous limiterons aux méthodes nécessaires à notre étude, à savoir:

- analyse spectrale (FFT synchrone), sur-échantillonnage
- Estimation des coefficients d'un filtre linéaire (méthode de Marquardt) et condition de stabilité.
- Convolution et filtrage linéaire

L'ensemble des méthodes utilisées par la suite ont fait l'objet de création de programmes appropriés. Pour plusieurs raisons:

- il n'en existait pas dans la littérature
- ils étaient beaucoup trop compliqués et il semblait plus simple de les réécrire pour une application simple
- il s'agissait de modules spéciaux pour MUSICV faisant intervenir des passages de paramètres.

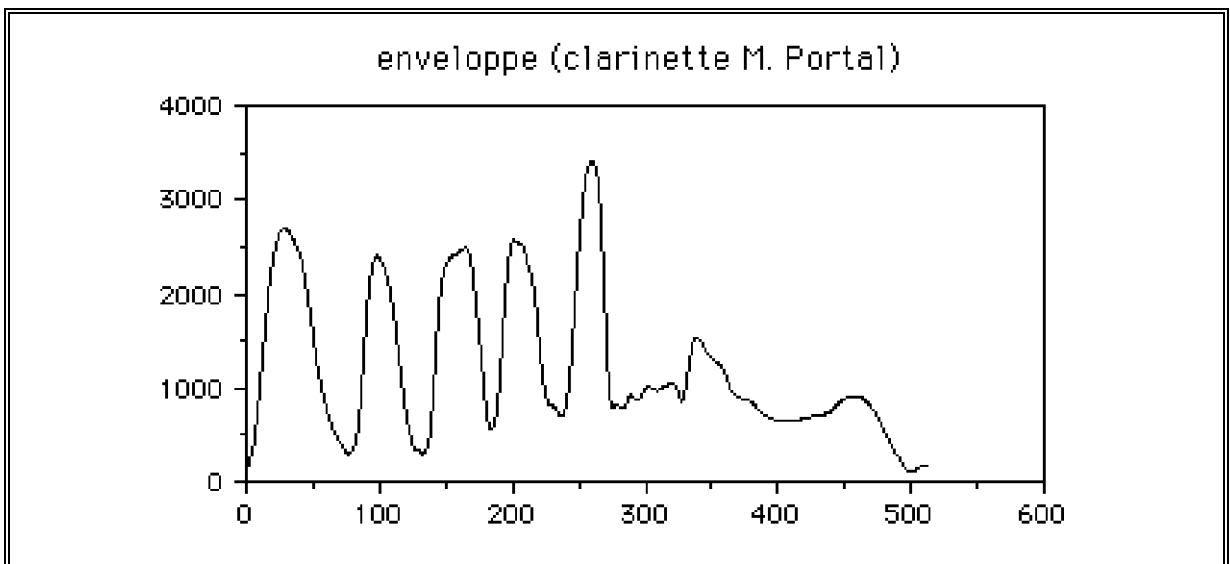
#### **VI.2.c.1. énergie du signal**

Dans la mesure où une note se décompose en quatre parties (une attaque, une décroissance, un plateau et une chute) et que l'on désire extraire des périodes dans le plateau (son filé), une bonne visualisation de ce plateau consiste à dessiner à l'écran l'énergie du signal qui est représentative de l'enveloppe du signal dans le cas d'une synthèse simple.

Pour des signaux numériques l'énergie du signal se caractérise par la somme des puissances de ses harmoniques:

$$\text{Energie} = \frac{1}{N} \sum_{n=0}^{N-1} |x(n)|^2 = \sum_{k=0}^{N-1} |X(k)|^2 \quad (\text{Egalité de Parseval})$$

Le programme ENVELOPPE.PAS permet de calculer l'énergie du signal et de représenter l'enveloppe ainsi obtenue par FFTs successives en se déplaçant d'un nombre quelconque de points à l'intérieur du fichier sonore. L'enveloppe est sauvegardée dans un fichier et peut être récupérée par un instrument MUSICV. Ce calcul d'enveloppe sera ensuite utilisé dans la partie expérimentale pour se positionner dans la zone stationnaire ou pseudo-périodique du signal obtenu à partir du jeu d'une note filée. Cette zone se traduit graphiquement par une partie plane de l'enveloppe spectrale. Au contraire de l'exemple ci-dessous qui représente l'enveloppe d'une série de notes très rapides jouées à la clarinette par Michel Portal.



*figure 9 - Extraction de l'enveloppe d'une succession de notes de clarinette de M.Portal*

C'est dans la partie stationnaire que l'on choisira une période du signal qui sera utile pour poursuivre l'étude. Nous allons voir comment pratiquer cette extraction de période, toujours dans l'optique de l'utilisation du programme par un musicien.

## **VI.2.c.2. analyse spectrale synchrone**

### *VI.2.c.2.1. extraction des périodes*

L'extraction des périodes se fait à partir d'un fichier sonore contenant une note filée d'un instrument de musique. L'extraction d'une période n'a alors de sens que dans la partie périodique ou pseudo-périodique du signal. Le musicien repère facilement sur un écran une période du signal. Il sera aidé pour un découpage plus précis de la période autour de la valeur zéro en recherchant autour de cette valeur le passage par zéro.

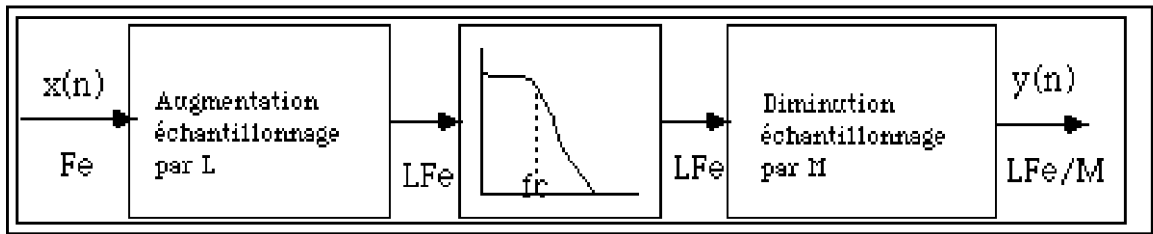
Les étapes pour obtenir une période du signal sont les suivantes:

- Enregistrement du son
- repérage visuel de 2 zéros: donne un morceau de signal ayant un nombre quelconque de points
- interpolation pour avoir la période sur 1024 points en vue de la FFT.
- FFT, donne directement les harmoniques, sans artefact dû à la coupure ou à l'introduction d'une fenêtre.

Le programme (COUPEPER.PAS) propose le positionnement automatique d'un curseur au début de chaque note et la possibilité de couper à vue les périodes mises en évidence sur l'écran de contrôle. Si la fréquence est trop élevée, il est possible de sur-échantillonner le signal en utilisant la méthode proposée par R. E. Crochiere [IEEE] et qui sera évoquée plus loin.

### *VI.2.c.2.2. sur-échantillonnage*

La fréquence d'échantillonnage d'entrée  $F_e$  est augmentée par un facteur  $L$  en insérant  $L-1$  zéros entre chaque échantillon. Le signal ainsi obtenu est filtré par un filtre passe-bas à réponse impulsionnelle finie dont la fréquence de coupure est la plus petite des 2 valeurs  $\frac{F_e}{2}$  et  $\frac{L \cdot F_e}{2 \cdot M}$ . La fréquence d'échantillonnage du signal de sortie est réduite par un facteur  $M$  en prenant un point tous les  $M$  échantillons.



On applique ensuite sur le signal de sortie la même méthode de recherche des passages par zéro pour couper la nouvelle période. A ce niveau des opérations, on peut en déduire la valeur approchée du fondamental, car on connaît la fréquence d'échantillonnage  $F_e$  et le nombre de points  $n$  par période du signal:

$$f_0 = \frac{F_e}{n}$$

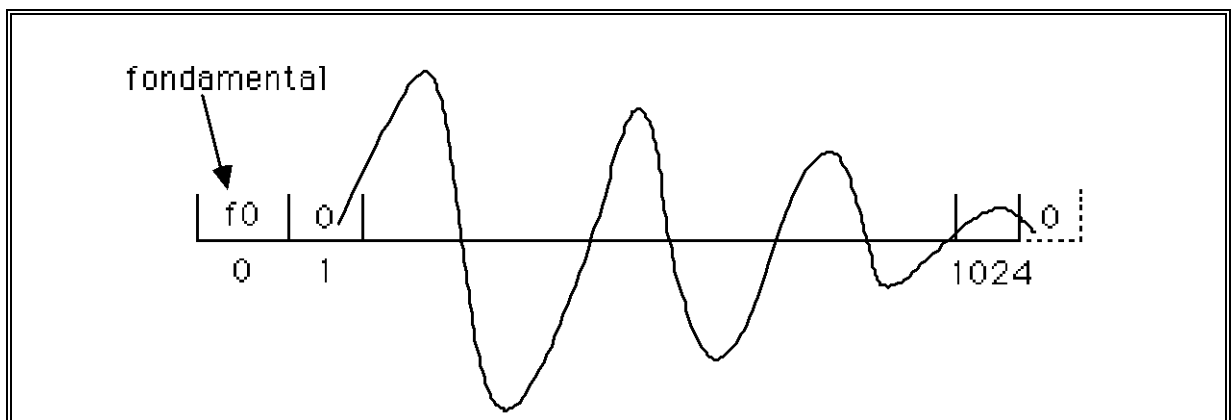
Le nombre  $n$  de points par période est une donnée variable. Pour pouvoir effectuer une FFT, on interpole la période de façon à l'obtenir sur 1024 points. La méthode employée est soit une simple interpolation linéaire, soit une interpolation par les polynômes de Lagrange de degré 2 ou 3 [Press & al., 1987].

$$P(x) = \frac{(x-x_2)(x-x_3)\dots(x-x_N)}{(x_1-x_2)(x_1-x_3)\dots(x_1-x_N)} Y_1 +$$

$$\frac{(x-x_2)(x-x_3)\dots(x-x_N)}{(x_2-x_1)(x_2-x_3)\dots(x_2-x_N)} Y_2 +$$

$$\dots + \frac{(x-x_2)(x-x_3)\dots(x-x_N)}{(x_N-x_1)(x_N-x_2)\dots(x_N-x_{N-1})} Y_N$$

Les périodes sont stockées dans des fichiers de réels ayant la structure suivante:



La période peut être utilisée directement comme forme d'onde dans un instrument MUSICV. Nous verrons plus loin qu'il était indispensable de pouvoir extraire des périodes à partir d'une position quelconque du fichier sonore, notamment dans l'étude de la variation du barycentre de l'énergie du spectre pour effectuer une resynthèse par distorsion non linéaire.

Le calcul de la FFT donne directement les valeurs des harmoniques car c'est une analyse synchrone à la période, le signal étant pseudo-périodique. Ce calcul de FFT sur une période permet de faire abstraction de la fenêtre temporelle nécessaire dans toute autre analyse sur une longueur quelconque de signal. Chaque calcul donne un spectre de Fourier qui est celui de la fonction périodique qui répéterait indéfiniment cette période.

Il existe une possibilité d'extraction automatique des périodes à partir d'un fichier sonore contenant les notes d'une gamme d'une durée égale. En effet, le musicien est capable de retrouver les valeurs des fréquences des notes qu'il joue. A partir de cette connaissance préalable approximative de la fréquence, on peut par autocorrélation déterminer le nombre de points par période et se positionner automatiquement aux bornes de la période. Cela permet d'automatiser le travail d'expérimentation qui est très laborieux pour un grand nombre de notes.

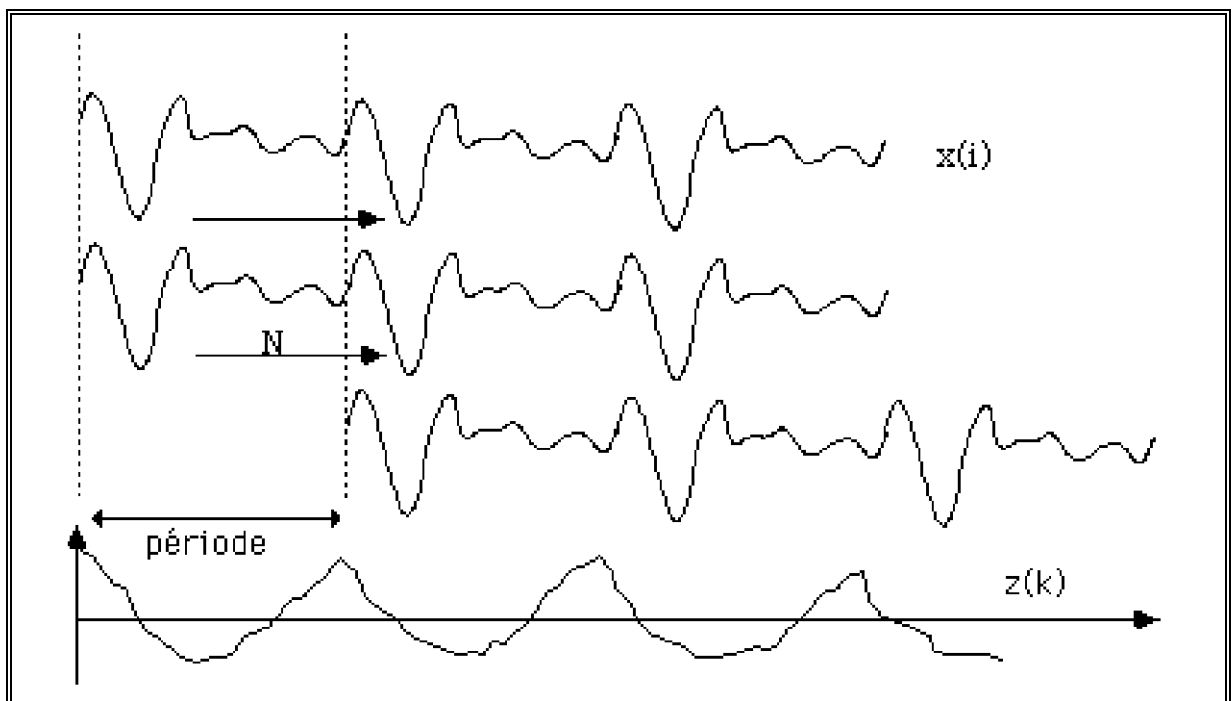
### VI.2.c.3. estimation de la fréquence fondamentale

#### VI.2.c.3.1. cas du signal périodique

La connaissance au préalable de la fréquence permet de donner une estimation de la longueur en nombre de points de la période et de chercher le maximum aux alentours de ce point dans le tableau contenant le résultat de l'autocorrélation. Cette méthode n'est valable que si le tableau contient plus d'une période.

Soit  $x(i)$  une suite de valeurs de longueur  $N$ : la séquence  $z(k)$  est donnée comme la corrélation périodique de  $x(i)$  avec elle-même de la manière suivante:

$$z(k) = \sum_{i=0}^{N-1} x[i] x[(k+i) \bmod N] \quad , \quad k=0, \dots, N-1$$



Dans le cas d'un signal périodique, la fonction d'autocorrélation présente des maxima locaux à chaque multiple de la période. Dans le cas d'un signal pseudo-périodique, on note une légère décroissance des pics du maximum de la fonction d'autocorrélation à chaque période. Cela est suffisant pour déterminer la période à un échantillon près. Pour améliorer le traitement, on peut filtrer au préalable le signal à l'aide d'un filtre passe-bas pour éliminer les harmoniques du signal qui pourraient interférer avec le fondamental. Pour la mise en oeuvre rapide de filtres de tout gabarit, j'ai utilisé un logiciel du commerce (ILS) qui permet, à partir

de gabarits déterminés à l'avance, la création de filtres à réponse impulsionnelle finie par la méthode de Remez, et qui permet de faire la convolution avec le signal source pour obtenir le filtrage désiré. Le programme s'appelle CONVILS.PAS et nécessite une bonne connaissance du logiciel ILS.

Application au fondu-enchaîné d'une voix humaine se transformant peu à peu en clarinette. On extrait une période d'une voyelle chantée ("a" par exemple) et une période d'un son de clarinette. Voici la partition MUSICV effectuant l'interpolation entre les deux périodes:

```

SAM 25600; COM fréquence
d'échantillonnage ;
COM définition instrument ;
INS 0 1;
CNV P10=DUR(P4);
CNV P6=HTZ(P6);
CNV P8=HTZ(p8);
ios p5 p10 B3 F1 p28;
ios p7 p10 B4 F2 p27;
IOS B3 P6 B3 F1 P30;
IOS B4 P8 B4 F2 P29;
AD2 B3 B4 B1;
OUT B1;
END;
COM lecture des fichiers période ;
FIC 0 1 a.per ; COM vpoix;
FIC 0 2 clar.per ; COM clarinette ;
GEN 0 20 3 1024 1;
GEN 0 20 4 1024 2;
GEN 0 1 1 1000 0 0 1 1000;
GEN 0 1 2 1000 1 0 0 1000;
COM -----;
NOT 0.00 1 3 5000 176.04 5000 257.52;
COM fréquence voix, fréquence clarinette ;
TER 3;

```

### VI.2.c.3.2. cas où la fréquence du fondamental est une fonction du temps

Dans le but de parfaire le réalisme du modèle, on a besoin de connaître la variation de fréquence du fondamental au cours du temps. Cela est possible car on se limite à des exemples sonores issus d'une gamme chromatique dans laquelle chaque note est un son filé, donc quasi-périodique. Si on utilise une méthode basée sur la détection de la périodicité par autocorrélation, on obtient une précision insuffisante. Pour pallier cette erreur, on utilise une propriété de la transformée de Fourier.



Soit  $x(n)$  une suite de nombres complexes, représentant les échantillons du signal périodique à analyser, et  $X(k)$  la transformée de Fourier des  $x(n)$ , on établit entre ces deux suites la relation suivante:

$$X(k) = \frac{1}{N} \sum_{n=0}^{N-1} x(n) e^{-j2\pi \frac{nk}{N}}$$

Si on calcule une transformée de Fourier après une translation de  $n_0$  échantillons cela entraîne une rotation de phase des  $X(k)$  d'un angle égal à  $2\pi \frac{n_0 k}{N}$  :

$$X_{n_0}(k) = \frac{1}{N} \sum_{n=0}^{N-1} x(n-n_0) e^{-j2\pi \frac{nk}{N}} = X(k) e^{-j2\pi \frac{n_0 k}{N}}$$

On se place en un point précis du signal où l'on calcule une FFT, puis on se décale de  $n_0$  points dans le signal et l'on calcule une nouvelle FFT. A partir de ces deux FFT on calcule un déphasage pour une valeur  $k$  d'une raie (on prendra la raie correspondant au fondamental déterminé par autocorrélation), puis connaissant le déphasage théorique on peut déterminer la variation de fréquence entre les deux FFT.

$$\omega = 2\pi f \quad \delta\omega = \frac{\delta\phi}{\delta t} \quad \text{d'ou} \quad f = \frac{\delta\phi}{2\pi\delta t} = \frac{\phi_{\text{réel}} - \phi_{\text{théo}}}{2\pi(t_2 - t_1)}$$

$N$  = nb. point de la FFT

$k$  = indice de la raie

$f_e$  = fréquence échantillonnage

$$f = \frac{k f_e}{N}$$

pour un décalage de  $n_0$  points entre  $t_1$  et  $t_2$ :

$$t_2 - t_1 = \frac{n_0}{f_e}$$

$f_{\text{théo}} = 2\pi \frac{n_0 k}{N}$  à  $k = \frac{N}{2}$  la raie correspondante tourne de  $\pi n_0$  pour une fréquence égale à  $\frac{f_e}{2}$ . Cela signifie que pour un décalage d'un échantillon, à  $\frac{f_e}{2}$  la phase tourne de  $\pi$ . Si le décalage est suffisamment petit pour que la phase d'une raie de la FFT tourne d'un angle inférieur à  $2\pi$ , on peut calculer la fréquence réelle d'après la formule:

$$f = \frac{\delta\phi}{2\pi\delta t} = \frac{\phi_{\text{réel}} - \phi_{\text{théo}}}{2\pi(t_2 - t_1)} = f_e \frac{\phi_{\text{réel}} - \phi_{\text{théo}}}{2\pi n_0}$$

Pour améliorer la recherche du maximum de l'autocorrélation, il est préférable de filtrer le signal par un filtre passe-bas. La trajectoire du fondamental peut être utilisée par MUSICV, qui mettra les valeurs lues dans un oscillateur afin d'obtenir des sons plus réalistes.

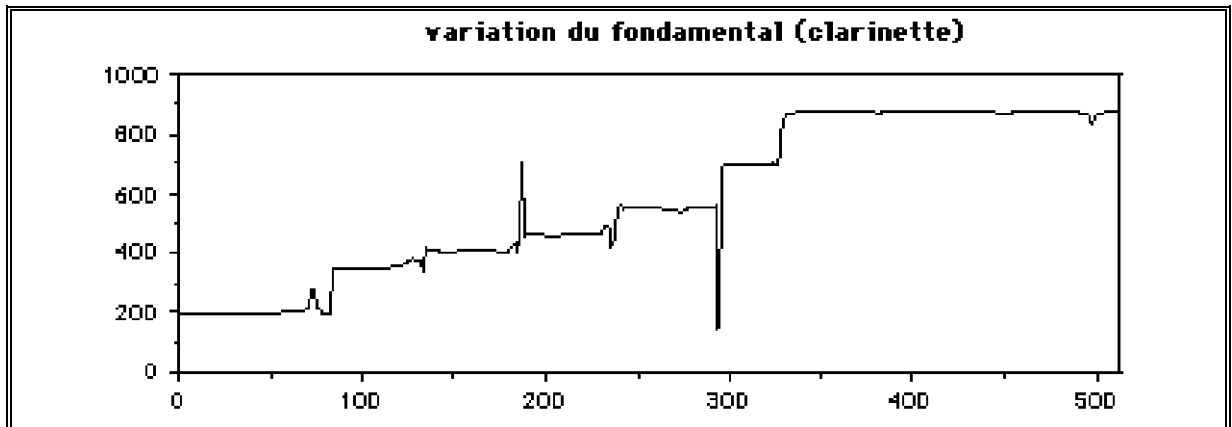


Figure 9- Suivi du fondamental d'une succession de notes de clarinette jouées par Michel Portal. Les pics sont essentiellement dus au bruit des clefs de l'instrument.

Dans l'exemple suivant, effet

on observe la trajectoire d'un son de clarinette de Michel Portal. La figure suivante montre un instrument MUSICV permettant d'attribuer les fréquences entendues à un simple générateur sinusoïdal.

```

SAM 25600;
INS 0 1;
CNV P10=DUR(P4);
IOS P6 P10 B4 F1 P30 ;com pitch ;
IOS P5 B4 B3 F2 P29 ; com sinus ;
OUT B3 ;
END;
FIC 0 1 pitch.dat ;
GEN 0 22 1 512 1;
GEN 0 2 2 1000 1 1;
NOT 0 1 1 10000 1 ;
TER 1 ;

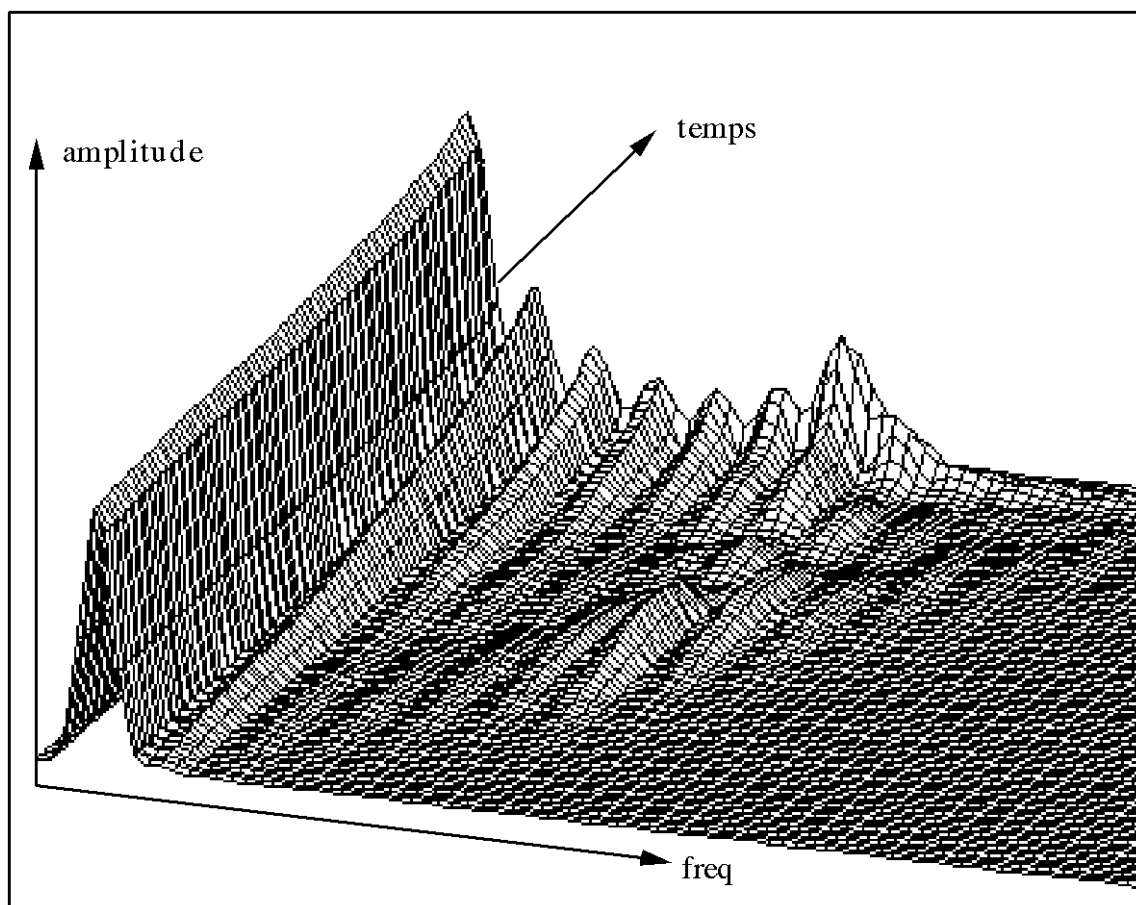
```

Exemple de l'utilisation des résultats de l'analyse de recherche du fondamental pour restituer la mélodie jouée par l'instrumentiste à un instrument MUSICV.

## VI.3. séparation source-résonance

### VI.3.a. estimation des formants

Avant de décrire les méthodes mises en oeuvre, il convient d'expliquer succinctement la raison pour laquelle il est important de supprimer l'influence des formants dans le cadre d'une synthèse par distorsion non linéaire. En effet, on pourrait définir une fonction de distorsion telle que l'évolution de l'index au cours du temps dans une simulation conduise au résultat attendu. Mais on se rend compte que si pour un index égal à 1 le spectre obtenu présente des formants, il n'en est pas de même lorsqu'on fait varier cet index. Les formants peuvent se déplacer en fonction des valeurs de l'index, ce qui est contraire à notre modèle de départ: à savoir des instruments à formants fixes.



*figure 10 - Exemple de variation au cours du temps du spectre d'un son synthétisé par distorsion non-linéaire et index de distorsion évolutif. Tiré de la partition VAR\_IND.X.PAR, LA à 110 Hz comportant 2 formants visibles en fin de jeu.*

A titre d'exemple, la figure ci-dessus représente la variation au cours du temps du spectre d'un son synthétisé par distorsion non-linéaire et index de distorsion évolutif (évolution

linéaire entre 0 et 1). La fonction de distorsion a été choisie de façon à ce que le spectre engendré pour un index égal à 1 corresponde au spectre d'un signal réel à formants. On voit clairement que lors de l'évolution de l'index de distorsion, les formants disparaissent pour n'apparaître clairement qu'en fin de jeu (index égal 1).

Dans un modèle source-résonance il convient donc de supprimer l'influence des formants du signal, afin de modéliser l'excitation par distorsion non linéaire.

La détection des formants en vue de la synthèse peut se faire de diverses manières.

### **VI.3.a.1. méthodes classiques**

#### ***VI.3.a.1.1. maxima locaux du spectre***

Cette méthode consiste à reconnaître dans le spectre les maximum d'énergie. Elle semble la plus naturelle car, par définition, les formants sont des zones du spectre qui ont une énergie maximale.

Dans cet ordre d'idée, on peut chercher l'harmonique qui a la plus grande énergie. Cette méthode ne donne qu'une estimation de la position des formants, d'autant meilleure que la fréquence du fondamental est faible par rapport à celle des formants. Elle présente un risque, celui de prendre pour des formants des maxima locaux du spectre. Dans la littérature [Flanagan, 1972] les fréquences des formants sont connues à 200 Hz près. Pour les voix de femmes, dont le fondamental est très élevé, l'espacement des harmoniques est plus important, ce qui empêche une bonne extraction des formants par cette technique.

#### ***VI.3.a.1.2. représentation homomorphique: cepstrale***

Le signal est donné par la convolution de l'excitation par la réponse impulsionnelle du filtre, ce qui correspond au produit des transformées de Fourier.

$$s(t) = (e * r)(t)$$

$$s(\omega) = e(\omega) \cdot r(\omega)$$

la multiplication devient une addition si on prend le logarithme

$$\text{Log}(s(\omega)) = \text{Log}(e(\omega)) + \text{Log}(r(\omega))$$

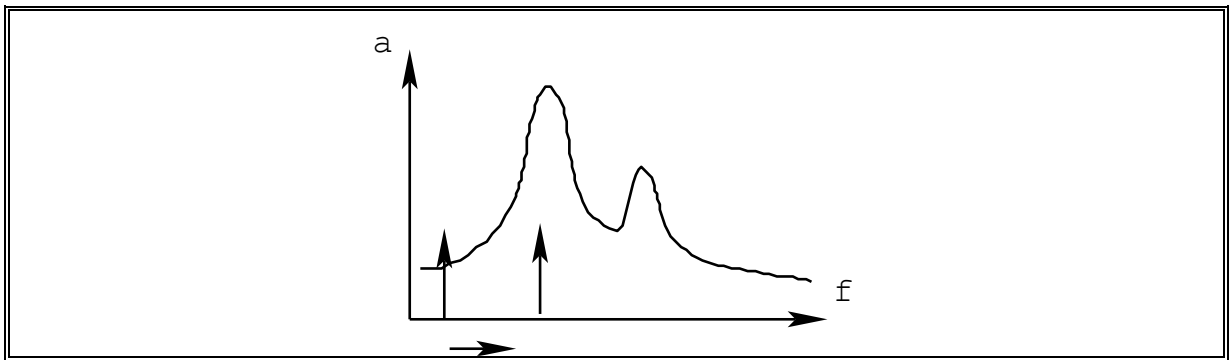
Si  $r(\omega)$  est de basse fréquence par rapport à  $e(\omega)$  (c'est le cas de formant "mou") on peut alors séparer aisément  $e(\omega)$  et  $r(\omega)$  par filtrage passe-bas du logarithme de  $s(\omega)$ .

On appelle cepstre la transformée de Fourier inverse du logarithme du module de la transformée de Fourier d'un signal.

$$C(\tau) = F^{-1}(\log |F(f)|)$$

où  $F$  et  $F^{-1}$  représentent respectivement les transformées directe et inverse de Fourier.

Je n'ai pas utilisé cette technique car au cours du jeu d'une gamme chromatique, le fondamental peut se déplacer vers les formants, rendant difficile la séparation.



*Déplacement du fondamental vers les formants*

### VI.3.a.1.3. Modèle auto-régressif: prédiction linéaire

Cette technique [ATAL & al., 1971] a été largement développée dans le cadre de l'analyse et synthèse de la parole. Elle consiste à modéliser le système par un filtre récursif à résonances multiples d'ordre P (fig 11) de fonction de transfert en Z:

$$H(z) = \frac{X(z)}{\delta(z)} = \frac{1}{1 - \sum_{k=0}^P a_k \cdot z^{-k}}$$

$\delta(z)$  représente le signal d'entrée que l'on peut modéliser par exemple pour les signaux vocaux par:

$\delta(z)$  = Train d'impulsions périodiques (sons voisés)

ou  $\delta(z)$  = Bruit blanc Gaussien (sons non voisés)

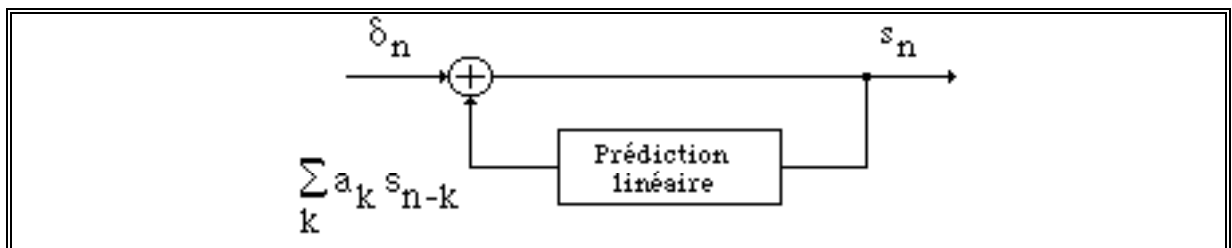


figure 11

L'analyse d'un signal revient alors à déterminer les coefficients  $a_k$ , ce qui peut être effectué de la façon suivante:

Le système est caractérisé par:

$$s(n) = \sum_{k=1}^P a_k \cdot s(n-k) + \delta(n)$$

Soit  $\hat{s}(n)$ , l'estimé associé à  $s(n)$ . On cherche  $\hat{s}(n)$  sous la forme:

$$\hat{s}(n) = \sum \hat{a}_k \cdot \hat{s}(n-k)$$

L'erreur de prédiction est donnée par:

$$E(n) = s(n) - \hat{s}(n) = s(n) - \sum \hat{a}_k \cdot \hat{s}(n-k)$$

Les  $\hat{a}_k$  sont choisis de façon à minimiser l'erreur quadratique moyenne:

$$(1) \quad \langle E(n) \rangle^2 = \sum_n \left( s(n) - \sum_k \hat{a}_k \cdot \hat{s}(n-k) \right)^2$$

En dérivant (1) par rapport aux  $\hat{a}_j, j=1,p$  et en annulant les dérivées, on obtient un système de  $p$  équations à  $p$  inconnues:

$$(2) \quad \sum_k \hat{a}_k \sum_n s(n-k) \cdot s(n-j) = \sum_n s(n) \cdot s(n-j)$$

$j=1, p$

On peut écrire (2) sous la forme matricielle suivante:

$$\Phi[\hat{a}] = \psi$$

Avec:  $\Phi$ : Matrice d'autocorrélation du signal (symétrique et définie positive)

$$\Phi_{i,j} = \sum_n s(n-i) \cdot s(n-j)$$

$\psi$ : Vecteur d'autocorrélation du signal,  $\psi_j = \Phi_{i,j}$

$[a]$ : vecteur de composantes  $a_k$

En pratique, il s'agit donc de calculer pour chaque intervalle de temps compatible avec l'évolution du signal la matrice  $\Phi$  et résoudre le système d'équations correspondant, de manière à extraire les coefficients de prédiction caractérisant l'aspect résonant du système.

La connaissance des coefficients  $a_k$  permet d'accéder facilement à l'enveloppe spectrale, car celle-ci correspond au module au carré (densité d'énergie) de la réponse du filtre lorsque le signal d'entrée est une distribution de Dirac (spectre constant).

$$R(\omega) = \frac{1}{\left| 1 - \sum_{k=0}^p a_k \cdot \exp(-2i\pi\omega) \right|^2}$$

Le défaut majeur de cette méthode est la simplification de l'excitation par un train périodique d'impulsions. Pour des enveloppes spectrales contenant de nombreux formants (violon), l'identification n'est possible que si l'on recherche un grand nombre de coefficients. Dans ce cas, on aboutit généralement à des filtres instables.



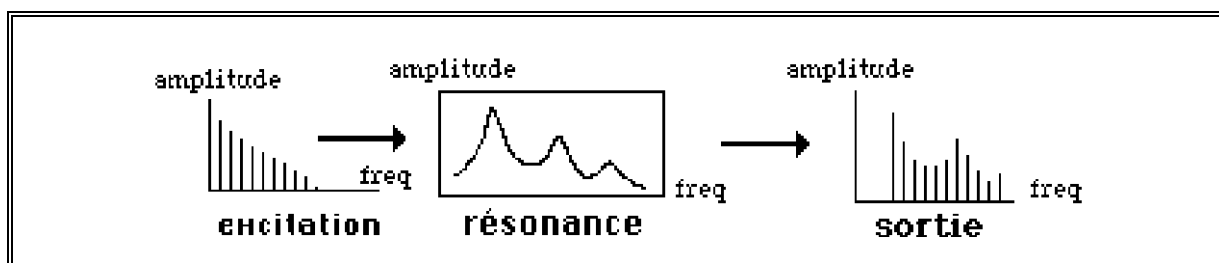
## VI.4. approche originale du problème

Le but de la recherche est double. Premièrement, séparer les paramètres physiques et psychoacoustiques provenant de l'excitation de ceux provenant de la résonance, tout en vérifiant avec MUSICV l'exactitude du modèle proposé, deuxièmement proposer une méthode originale basée sur le jeu instrumental.

En effet, par le passé, l'étude des instruments a souvent été menée comme on le faisait pour des composants électroacoustiques. L'instrument était considéré comme une boîte noire pour laquelle on substituait au système excitateur un générateur, soit de bruit blanc, soit d'impulsions, et dont on regardait la sortie en relevant des courbes de réponses. Cette façon de voir les choses s'écarte totalement des conditions normales de jeu, un instrument ne réagissant pas de la même manière à des excitations académiques qu'il ne le fait dans un contexte plus compliqué impliquant le musicien.

La méthode de séparation source-résonance qui sera décrite au cours de ce chapitre ne nécessite que l'enregistrement d'une gamme chromatique. L'analyse se fait sans intervention d'un quelconque appareil de mesure, cela dans le but de simplifier l'étude de l'instrument et de rendre celle-ci accessible au musicien ayant à sa disposition un micro-ordinateur et une carte d'acquisition. L'instrument MUSICV final qui permet la resynthèse du signal d'origine n'est soumis qu'au seul jugement de l'oreille.

Il existe plusieurs méthodes pour modéliser un instrument de musique ainsi que différentes manières d'arriver au résultat. Ces techniques sont souvent l'oeuvre de spécialistes en traitement du signal et nécessitent des appareillages complexes. Parmi tous les modèles, on retiendra essentiellement un modèle simple d'excitation et de résonance, que l'on complétera en modélisant l'excitation par distorsion non linéaire.



*modèle simple excitation résonance*

L'utilisateur de MUSICV ne possède en général pas les moyens techniques d'une équipe de recherche, mais il possède au minimum un ordinateur, une carte de conversion numérique analogique (qui doit être également analogique numérique) et l'instrument à étudier. S'il ne possède pas cet instrument il peut en avoir un enregistrement sur bande magnétique. Il peut également avoir l'enregistrement de toutes les notes de la gamme. En effet, sur une seule note, on ne peut pas séparer source et résonance pour plusieurs raisons:

- Premièrement car la recherche de la résonance revient à un problème de déconvolution qui s'avérera être "mal posé" si l'on ne connaît pas la réponse de l'instrument à toutes les fréquences, ce qui correspondrait à avoir un signal d'excitation de type bruit blanc. On peut néanmoins se rapprocher de cette situation si l'on dispose des réponses à de nombreuses fréquences (cas de la gamme chromatique).

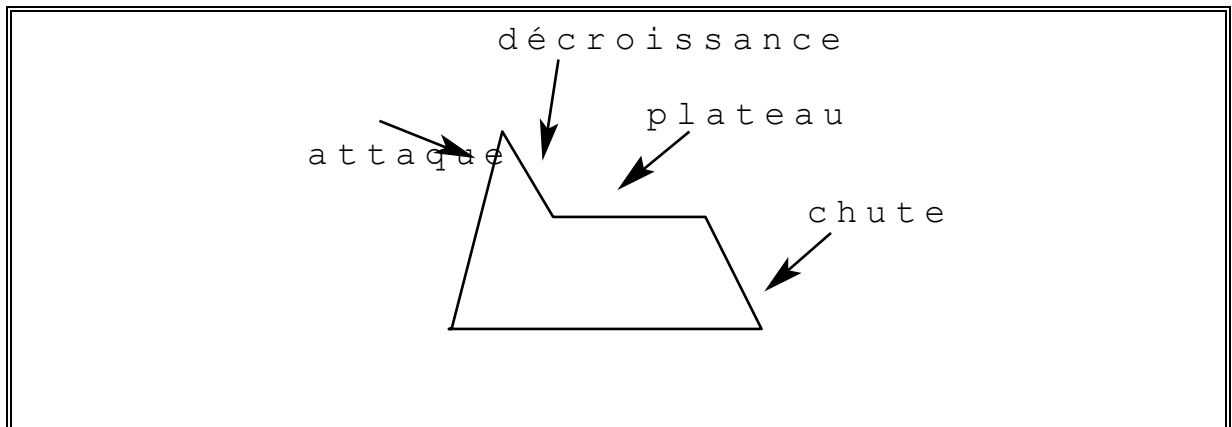
- Deuxièmement, le spectre d'excitation décroît généralement à haute fréquence. Celui-ci n'étant pas connu à priori, il est impossible d'effectuer la correction correspondante sur le spectre total. Nous verrons qu'il est possible de pallier cet inconvénient à condition de disposer de signaux provenant des notes d'une gamme chromatique jouées dans des conditions similaires. Dans la plupart des cas la séparation source-résonance se fait de manière aveugle. La condition imposée sur le jeu de l'instrumentiste revient à avoir une idée précise de la forme du spectre de l'excitation appliquée à l'instrument. C'est cette connaissance au préalable de l'excitation qui permettra la séparation.

Ces raisons font que les méthodes classiques décrites ci-dessus ne se sont pas suffisamment bien adaptées au problème que l'on se pose, ce qui m'a incité à développer une méthode originale plus appropriée.

La méthode consiste à montrer qu'avec le seul enregistrement d'une série de notes jouées suivant une gamme chromatique sur certains instruments, on peut avoir une information suffisante pour retrouver les paramètres d'un modèle de synthèse source-résonance. A partir de ces paramètres il doit alors être possible de simuler l'instrument. Intuitivement cela semble réaliste, car dans le cas d'une voyelle chantée sur une gamme chromatique, l'oreille perçoit les zones fréquentielles fixes dues aux formants, et il en est de même pour certains instruments à résonances fixes.

Le travail du musicien consiste à jouer des notes suivant une gamme chromatique avec comme premier postulat une condition sur le contrôle de l'excitation. Nous verrons plus loin comment se justifie cette condition. Auparavant je présente une explication du protocole expérimental ainsi qu'un descriptif graphique de la méthode pour permettre des raccourcis dans le développement.

Comme il a été dit, le musicien joue une gamme chromatique sur toute la tessiture de son instrument. S'il s'agit d'un instrument à cordes, il le fera avec son archet et cela dans des conditions bien particulières afin d'imposer une condition sur l'excitation. Il devra positionner son archet de telle manière que le rapport de la distance de l'archet au chevalet sur la longueur de la corde soit constant. Nous verrons plus loin que cela impose une condition sur le spectre de l'excitation qui reste connu et quasi identique au cours du jeu (ou du moins on peut supposer qu'il varie très peu). Le spectre d'une corde frottée est bien connu dans la littérature et cette connaissance est indispensable pour en déduire la courbe de résonance de l'instrument. On procède ensuite à une analyse pour chacune des notes. Pour cela on se place dans une partie stationnaire du signal. Cela est rendu possible car on demande à l'instrumentiste de jouer des sons filés. On peut souvent considérer qu'une note est constituée par une attaque, une décroissance, un plateau et une chute, chacune de ces quatre parties étant plus ou moins longue suivant les instruments.



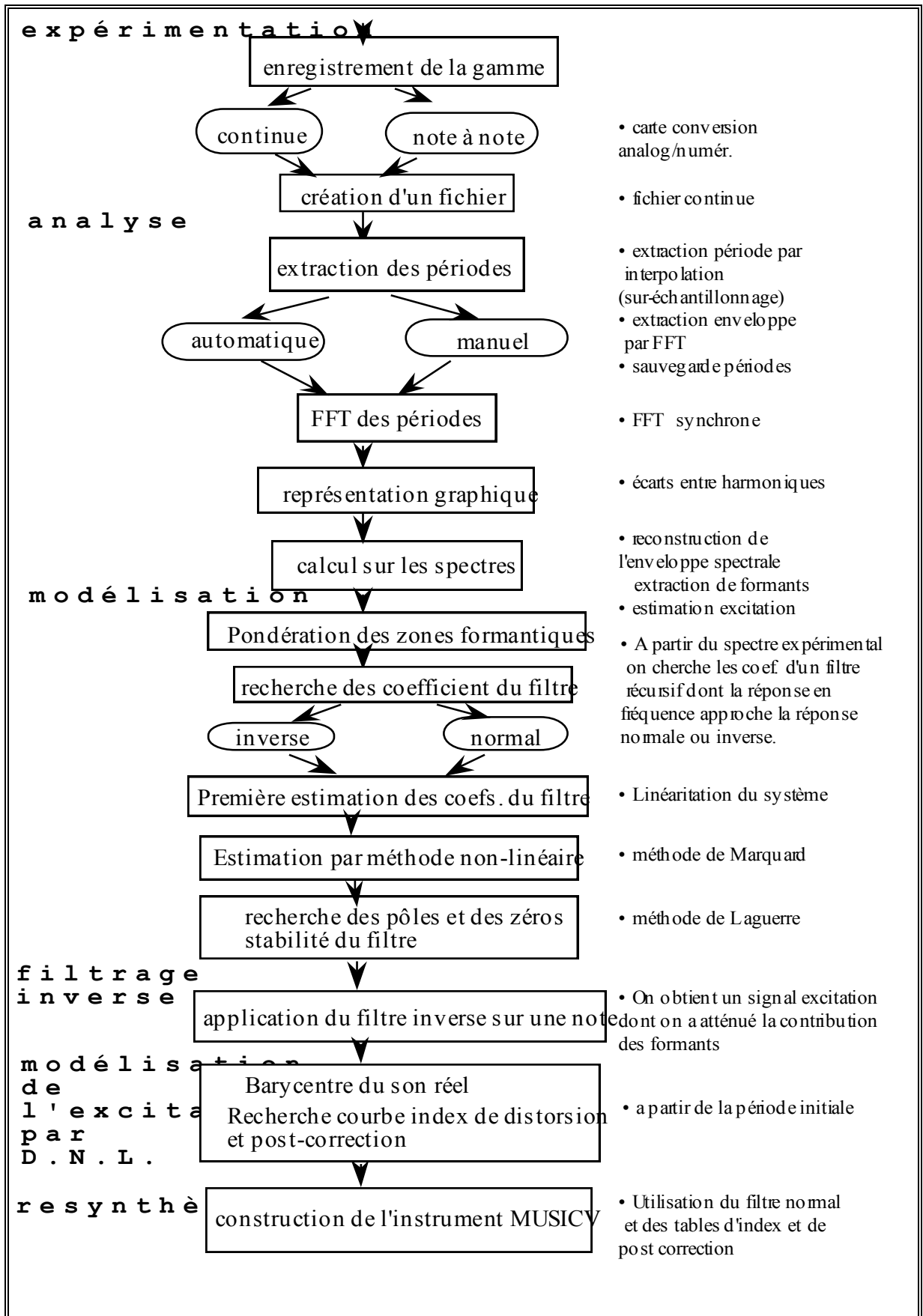
On se place donc sur le plateau, endroit où le son est quasi-périodique, et l'on extrait une période. On essaie dans cette zone d'avoir une représentation de la résonance. La méthode est décrite plus loin. A partir de l'enveloppe spectrale ainsi obtenue, on trouve les coefficients d'un filtre récursif qui approche au mieux cette enveloppe. On inverse le filtre numérique ainsi obtenu et on filtre le signal d'origine de façon à enlever au maximum la participation de la résonance. On applique ce filtre inverse sur toutes les notes. Première constatation possible: si l'hypothèse de départ est juste, c'est-à-dire que l'excitation est à peu près constante, on doit retrouver, après analyse du signal à l'endroit où on a prélevé les périodes, un spectre identique pour chacune des notes. Cela sera la première vérification. On pourra également comparer la forme de la réponse en fréquence de la résonance avec celle obtenue par d'autres méthodes.

Lors de cette première approche de modélisation d'un instrument par combinaison d'un système excitateur et d'un système amplificateur, nous nous sommes placés dans la partie stationnaire ou périodique du signal. Or en réalité, au cours de l'attaque ou de la chute d'une

note, il se passe bien souvent des phénomènes transitoires qui font apparaître des fréquences élevées. On essayera donc de modéliser l'excitation par un modèle de distorsion non linéaire qui pourra représenter de façon continue l'évolution du spectre de l'excitation au cours du jeu. L'ensemble de tous ces résultats sera introduit dans un instrument MUSICV qui permettra de juger la validité de la méthode.

Pour illustrer l'ensemble de la méthodologie qui sera utilisée, il convient de présenter un exemple à partir d'un résultat connu. Pour cela on va créer un signal composé d'une excitation dont le spectre sera engendré par une suite d'harmoniques ayant entre elles un rapport déterminé et une résonance formée d'un ou plusieurs filtres du second ordre dont les paramètres seront donnés à l'avance. On construira un signal musical composé d'une suite de notes séparées d'un demi-ton correspondant à une translation en fréquence d'un spectre de la première note. Cela sera notre signal de départ. Pour le moment on ne fait pas d'hypothèse sur une excitation provenant d'une simulation par distorsion non-linéaire. La théorie s'appuiera ainsi sur des exemples académiques qui seront une simplification de la réalité.

Le schéma suivant explique en détail la succession d'étapes nécessaires au déroulement de la méthode.



#### VI.4.a. cas du spectre d'excitation constant

Le cas de figure qui sera examiné en premier est le suivant: le spectre de l'excitation de chaque note est identique au cours du jeu. On se ramène à une translation en fréquence de l'excitation de la première note.

Nous allons nous intéresser à la répartition sur un axe fréquentiel des harmoniques obtenus en parcourant la gamme demi-ton par demi-ton ainsi qu'à l'information fréquentielle.

Si on considère le fondamental de la première note comme étant un Do grave (65.40 Hz) et si l'on étudie pour chacune des notes les positions des 20 premiers harmoniques (ce nombre pouvant varier suivant les cas) on arrive pour le dernier harmonique de la dernière note à des fréquences de l'ordre de 10000 Hz. Le tableau ci-dessous fait apparaître le fondamental et le 20<sup>ième</sup> harmonique pour trois octaves de la gamme chromatique. Les formants recherchés seront dans la plupart des cas compris dans cet ensemble de fréquences.

note	fondamental	20 <sup>ième</sup> harmo
DO	65.4064	1308.1278
DO#	69.2957	1385.9132
RE	73.4162	1468.3238
RE#	77.7817	1555.6349
MI	82.4069	1648.1378
FA	87.3071	1746.1412
FA#	92.4986	1849.9721
SOL	97.9989	1959.9772
SOL#	103.8262	2076.5235
LA	110.0000	2200.0000
LA#	116.5410	2330.8200
SI	123.4700	2469.4000
DO	130.8127	2616.2556
DO#	138.5977	2771.9544
RE	146.8344	2936.6888
RE#	155.5623	3111.2460
MI	164.8000	3296.0000
FA	174.6019	3492.0376
FA#	184.9917	3699.8264
SOL	195.9977	3919.9544
SOL#	207.6523	4153.0470
LA	220.0000	4400.0000
LA#	233.0819	4661.6376
SI	246.9417	4938.8330
DO	261.6256	5232.5113
DO#	277.1826	5543.6526
RE	293.6648	5873.2954
RE#	311.1270	6222.5397
MI	329.6276	6592.5511
FA	349.2282	6984.5646
FA#	369.9944	7399.8885
SOL	391.9954	7839.9087
SOL#	415.3047	8306.0940
LA	440.0000	8800.0000
LA#	466.1638	9323.2752
SI	493.8833	9877.6660

Si on représente sur un même graphique les positions fréquentielles des harmoniques de chacune des notes et si on projette celles-ci sur l'axe des fréquences, on obtient la figure suivante qui a été réalisée pour 6 harmoniques.

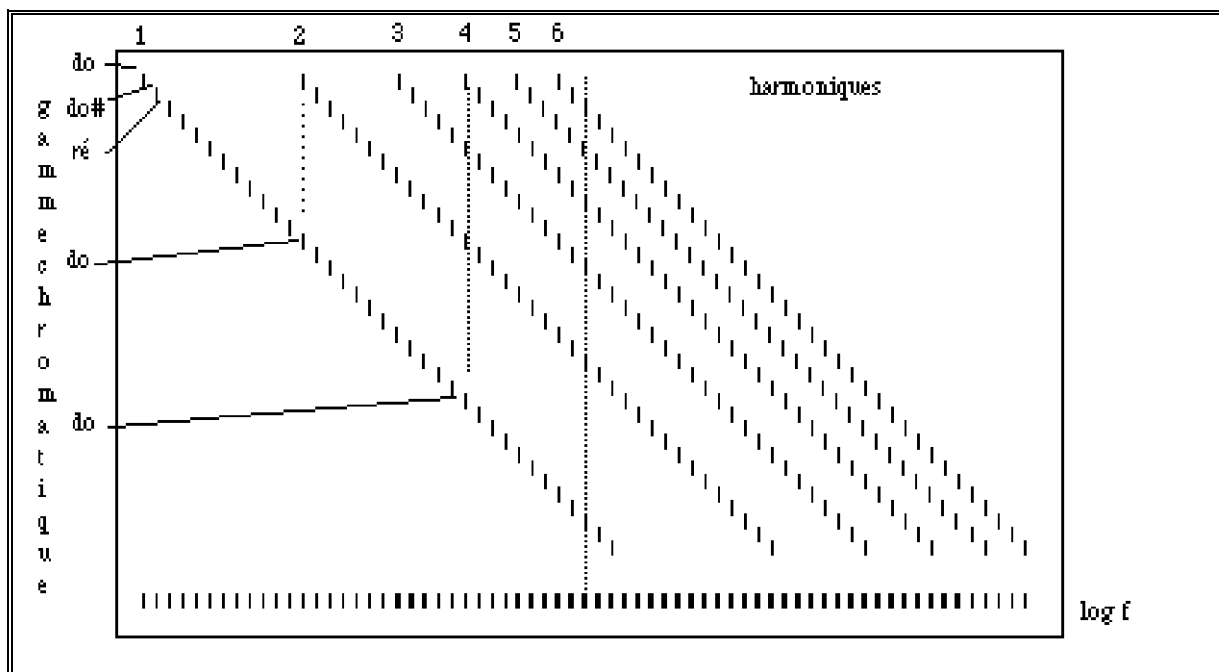


figure 11 - Représentation des 6 premières harmoniques de la gamme chromatique

On remarque que le "do" de la 2<sup>ème</sup> octave correspond au 2<sup>ème</sup> harmonique du "do" de la 1<sup>ère</sup> octave, ainsi que le "do" de la 3<sup>ème</sup> octave coïncide avec le 4<sup>ème</sup> harmonique du "do" de la 1<sup>ère</sup> octave. Il en sera de même pour d'autres fréquences, mais les correspondances sont en nombre réduit. L'axe des fréquences ainsi obtenu n'est pas échantillonné de manière régulière: par la suite pour une meilleure présentation des résultats et pour faciliter certains calculs, on sera amené à échantillonner l'axe des fréquences en huitième de tons. Une octave sera divisée en 48 parties égales logarithmiquement.

Sur les figures suivantes on observe la réponse en fréquence d'un filtre du second ordre, échantillonnée à partir des fréquences déduites précédemment par la gamme chromatique.

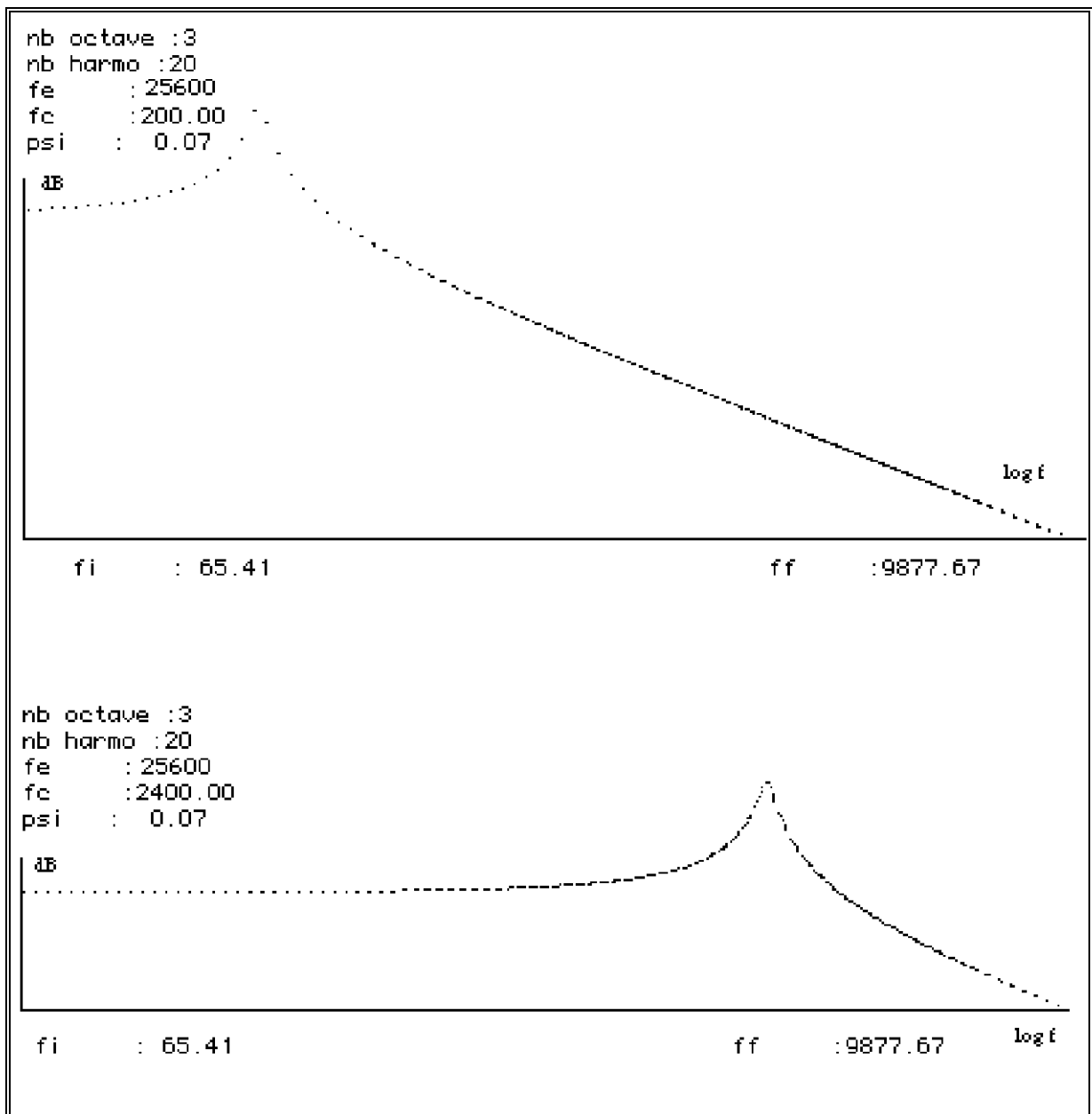


figure 12 - Echantillonnage d'un filtre du second ordre pour deux valeurs de la fréquence de coupure à l'aide des fréquences issues du jeu de la gamme chromatique en tenant compte des 20 premiers harmoniques de chaque note.

On constate que vers les très basses fréquences et vers  $\frac{f_e}{2}$  les valeurs du spectre sont beaucoup plus espacées. L'écart entre les fréquences se réduit dans la zone centrale du spectre où l'on rencontre un même point de la courbe défini plusieurs fois selon que des harmoniques différentes coïncident.

L'oreille est bien adaptée pour reconnaître un son constitué d'un spectre à enveloppe fixe. On peut donc supposer que l'information détenue par l'ensemble des fréquences ci-dessus est suffisante pour percevoir des pics spectraux fixes ou des formants, car de toute manière, si on



respecte les conditions de jeu, on n'aura à notre disposition que cette information et elle seule. Dans un exemple sonore, on peut faire la différence entre 2 gammes produites par un instrument dont le spectre est fixe et un instrument dont le spectre se translate avec la fréquence du fondamental.

Dans un son naturel, l'amplitude de chacun des harmoniques est généralement différente. Il est alors important de traiter indépendamment chacun d'entre eux. Cette étude permettra d'une part de "remonter" au spectre d'excitation et d'autre part de "renormaliser" les amplitudes de façon à décrire l'enveloppe spectrale de la résonance.

On se propose alors d'observer sur un même graphique successivement l'amplitude du premier harmonique de chaque note, puis le deuxième (etc...). La grille de fréquences décrite plus loin dans l'étude permet de positionner sur le même axe les différentes données ainsi que la courbe d'interpolation correspondante et ce pour chaque harmonique.

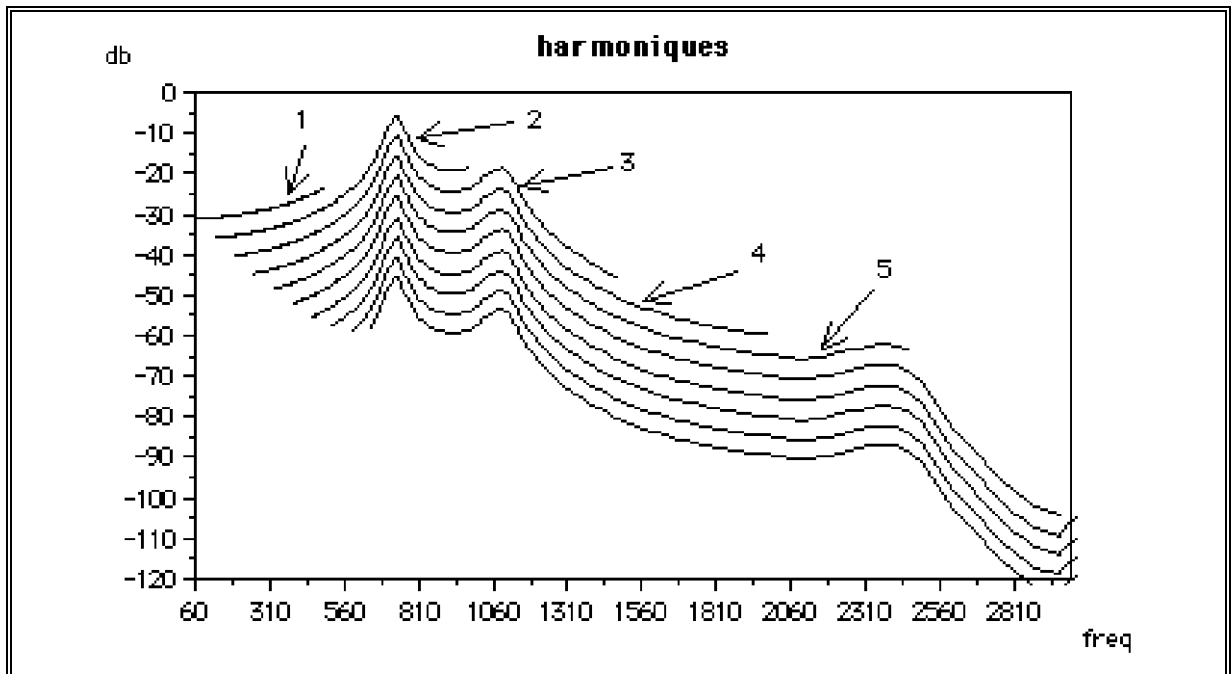


figure 13 - Représentation sur une même grille de fréquences le l'interpolation de tous les harmoniques de chaque note:

### VI.4.a.1. Calcul d'une grille de fréquence commune

Pour des raisons de commodité on présentera les résultats sur un axe de fréquence commun que l'on obtient arbitrairement en divisant l'échelle des fréquences en 8<sup>ième</sup> de ton. L'octave est divisée en 48 intervalles égaux logarithmiquement. La fréquence la plus basse sera le do (C1) de l'octave du la à 55 Hz (LA1). Les expérimentations ont été faites à l'aide d'une carte d'acquisition ayant une fréquence d'échantillonnage de 25600 Hz ainsi que les simulations avec MUSICV. Compte tenu de cet impératif, la fréquence la plus élevée que l'on peut observer est de 12800 Hz (fréquence d'échantillonnage divisée par deux). On calcule le nombre de points disponibles sur cet axe.

$f_0$ : fréquence du do (C1) = 32.7032 Hz

$f$  : fréquence finale

$f_e$ : fréquence d'échantillonnage = 25600 Hz

$f = f_0 * 2^{n/48}$ ,  $f \leq f_e/2$  soit  $n \leq 48 * \ln(f_e/2f_0) / \ln(2)$

On obtient avec ces valeurs 413 points sur l'axe des fréquences. A partir des données expérimentales ou simulées, il faudra interpoler les valeurs obtenues pour chaque harmonique de manière à exprimer tous les résultats sur l'axe discret de fréquence ainsi défini.

On peut alors se demander si l'interpolation du spectre n'est pas biaisée, car entre deux valeurs obtenues par FFT on peut "rater" un maximum du spectre. Cela serait le cas pour un instrument dont la résonance aurait un aspect spectral comportant énormément de pics, très rapprochés et très pointus - ce qui n'est pas le cas des instruments étudiés. La voix chantée possède des zones formantiques bien connues qui dépendent des voyelles vocalisées, et les formants peuvent être définis à 200 Hz près. Pour les autres instruments, le but de la méthode n'est pas de modéliser complètement l'instrument par une source parfaite et une résonance comportant des pics jusqu'aux fréquences élevées, mais de rechercher l'existence d'une résonance basse fréquence puis de l'atténuer par anti-filtrage sur le signal d'origine afin de modéliser l'excitation par distorsion non linéaire. On élimine par filtrage passe-bas la contribution dans le timbre des fréquences élevées pouvant provenir de l'excitation, bruit de l'archet pour les cordes ou souffle pour les vents.

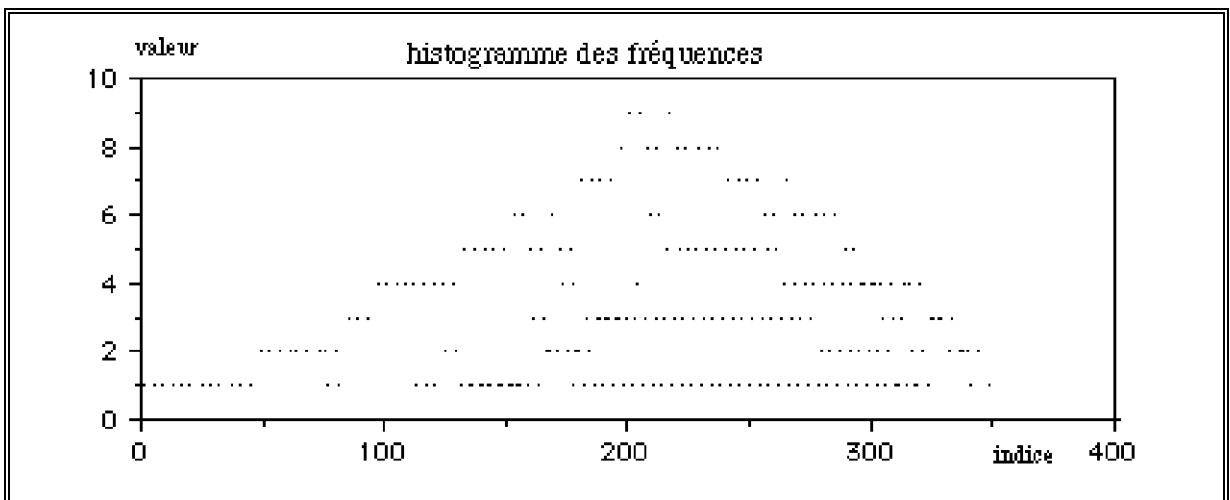
Sur l'axe des fréquences, constitué d'un tableau de 413 valeurs indicées de 0 à 412, la fréquence de rang  $n$  sera déduite à partir de la fréquence de rang 0 par la relation:

$$f = f_0 * 2^{n/48} \quad (1)$$

L'ensemble des fréquences engendré par la gamme chromatique et ses harmoniques sera approché par la formule (1). A chaque valeur correspondra un indice dans le tableau, indice qui sera calculé selon la formule:

$$n = \text{INT} (48 \cdot \ln (f / 2f_0) / \ln (2))$$

Dans ces conditions certaines valeurs de l'axe des fréquences seront atteintes plusieurs fois (fig 12). L'erreur entre la valeur réelle et la valeur approchée est très faible, elle n'intervient pas dans une estimation des fréquences des formants à quelques dizaines de Hertz près. Le tableau ci-dessous représente l'apparition des valeurs des fréquences.



*figure 14 - Cette figure représente l'histogramme des fréquences atteintes. Cela peut être interprété comme une zone de confiance: au centre de la figure certains harmoniques atteindront 9 fois la même information, d'où une certaine redondance. Si un formant se trouve dans cette zone on lui accordera un poids plus important que s'il se trouve aux extrémités.*

Cette fonction sera utilisée comme une fonction poids dans la recherche d'un modèle de filtre approchant le plus possible la réponse en fréquence obtenue par l'analyse en demi-tons.

```

nb octave : 3
nb harmo : 20
fe      : 25600.00
fc      : 2400.00
psi     : 0.07

```

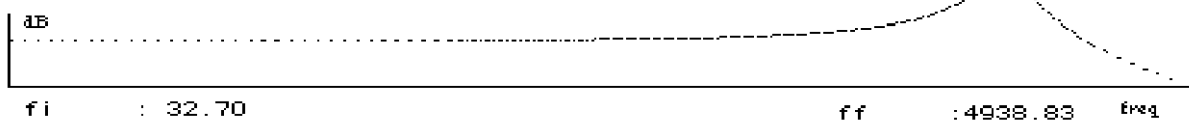
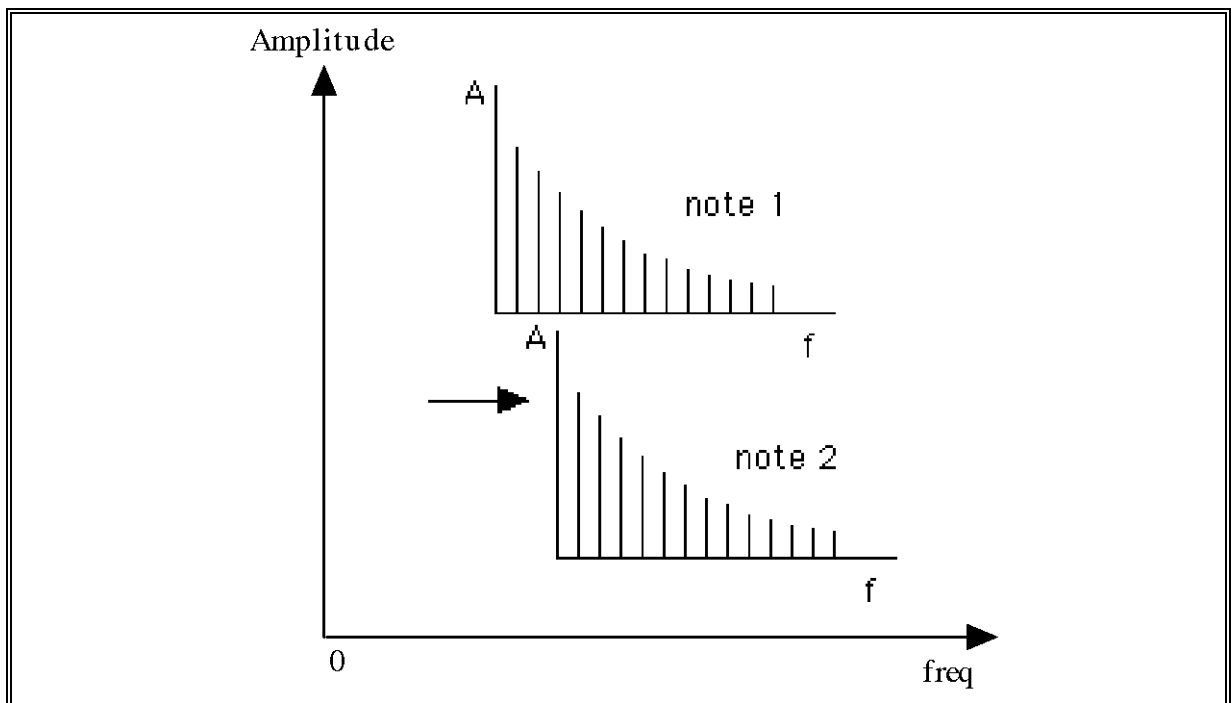


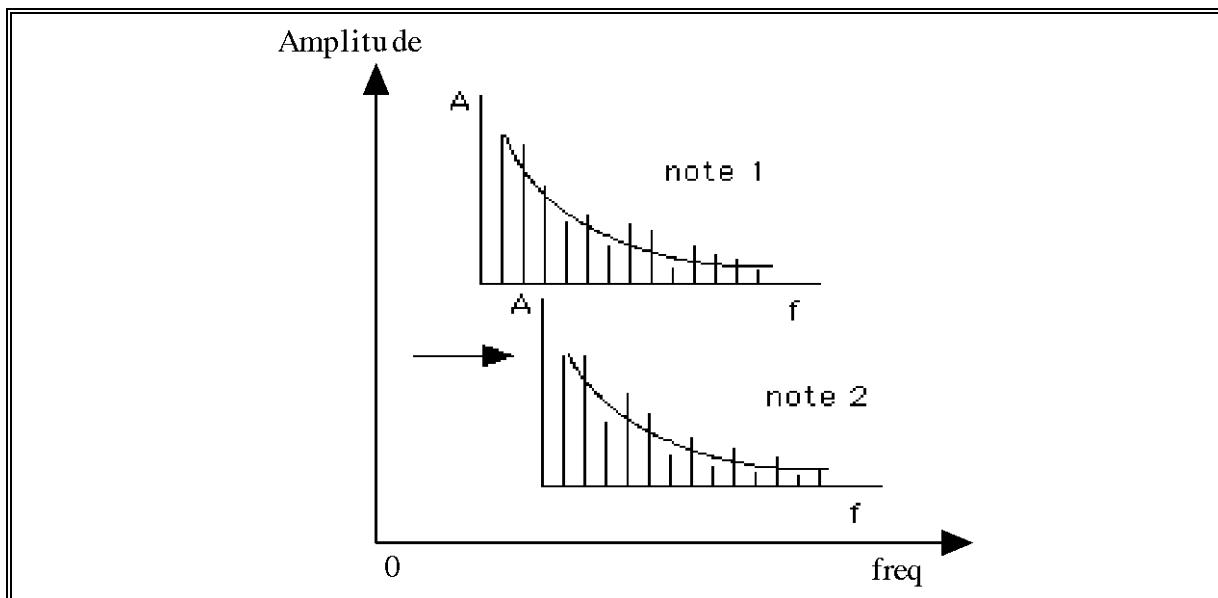
figure 15 - Spectre échantillonné suivant l'axe discret de fréquences atteintes par les harmoniques.

Si on échantillonne la réponse en fréquence d'un filtre du second ordre avec les fréquences obtenues à partir de cette construction pour différentes valeurs de la fréquence de résonance et du facteur de qualité, on observe visuellement l'enveloppe spectrale (fig.13). La question que l'on traitera par la suite est de savoir si cet ensemble de valeurs est suffisant pour déterminer les coefficients d'un filtre numérique récursif approchant cette réponse spectrale.

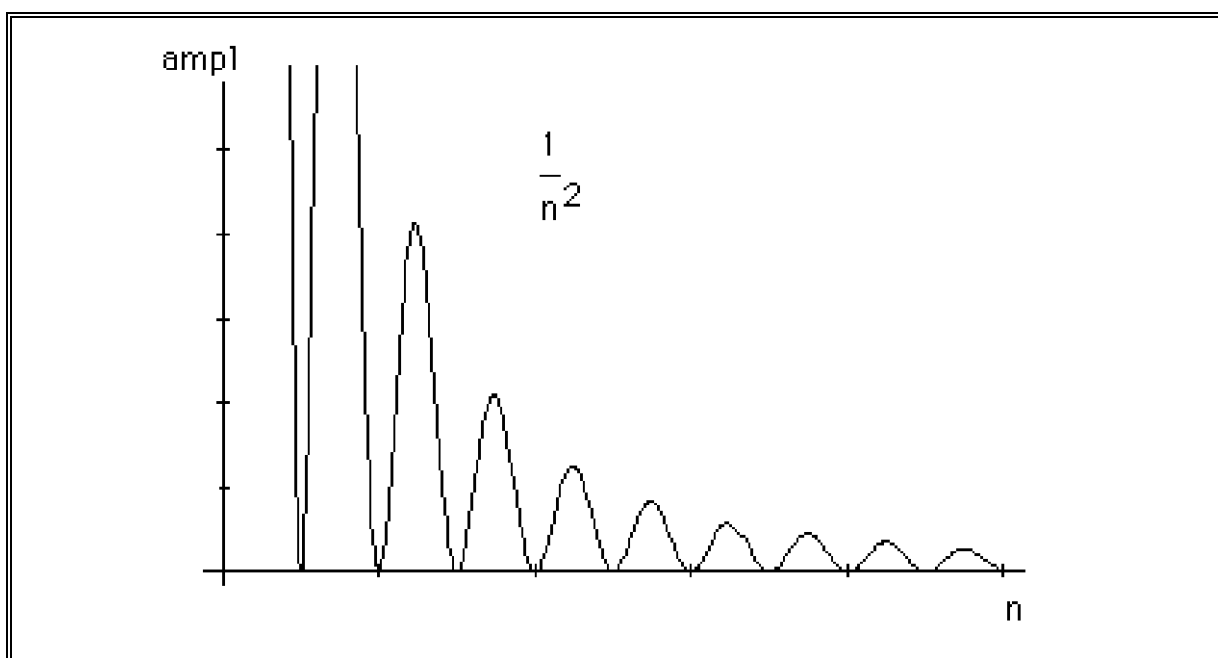
On abordera ce problème suivant différentes étapes. On supposera en premier lieu pour l'excitation, un spectre constant qui sera identique pour chaque note, mais décalé en fréquence.



Puis pour se rapprocher des conditions expérimentales on choisira un spectre dont l'amplitude varie légèrement autour d'une moyenne. En pratique on rajoutera une valeur aléatoire sur chaque harmonique de l'excitation.



Enfin, on étudiera les signaux réels, en imposant expérimentalement une condition sur le jeu pour se retrouver dans les deux cas précédents. Exemple: dans le cas d'un instrument à cordes on placera toujours l'archet à une distance correspondant à un rapport constant entre la longueur de la corde et la distance archet chevalet. Cela afin de créer une excitation dont on saurait retrouver l'allure par construction mathématique.

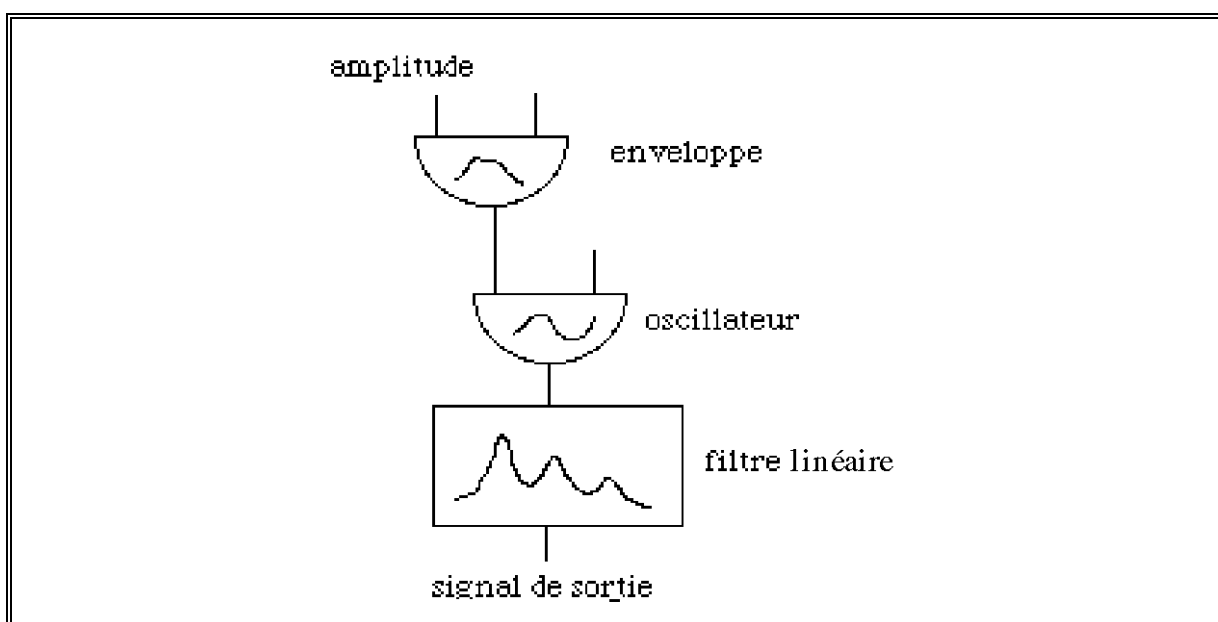


*Les harmoniques engendrées par une corde frottée suivent en première approximation une loi en  $1/n^2$  (modèle correspondant à une condition initiale de déplacement de la corde suivant une fonction triangle).*

### VI.4.a.2. Estimation de l'enveloppe spectrale

Il s'agit en premier lieu de créer un signal musical ayant les caractéristiques désirées, c'est-à-dire un spectre à résonance fixe (pics spectraux fixes) et une excitation à spectre constant au cours du jeu de l'instrument. On définit à l'aide de MUSICV un instrument comportant une excitation ayant le nombre d'harmoniques voulu, que l'on utilisera comme signal d'entrée d'une série de filtres mis en série, chacun des filtres présentant des caractéristiques de résonances différentes.

L'instrument sera simulé par une série de filtres résonants dont on connaît la fréquence d'accord.



L'énergie du signal de sortie sera modulée en amplitude par une enveloppe ayant au moins une zone plane (sustain), de manière à placer l'étude dans cette partie stationnaire ou pseudo-périodique du signal. On se rapproche ainsi du son filé ou soutenu d'un jeu instrumental. On réalise cet instrument à l'aide de MUSICV et on l'applique sur une gamme chromatique comprenant trois octaves, le premier octave de la gamme étant celui comprenant le "la" à 110 Hz et la première note étant un "do".

Le spectre initial est formé d'une suite d'harmoniques en  $\frac{1}{n^2}$  assimilable au spectre engendré par une corde frottée.

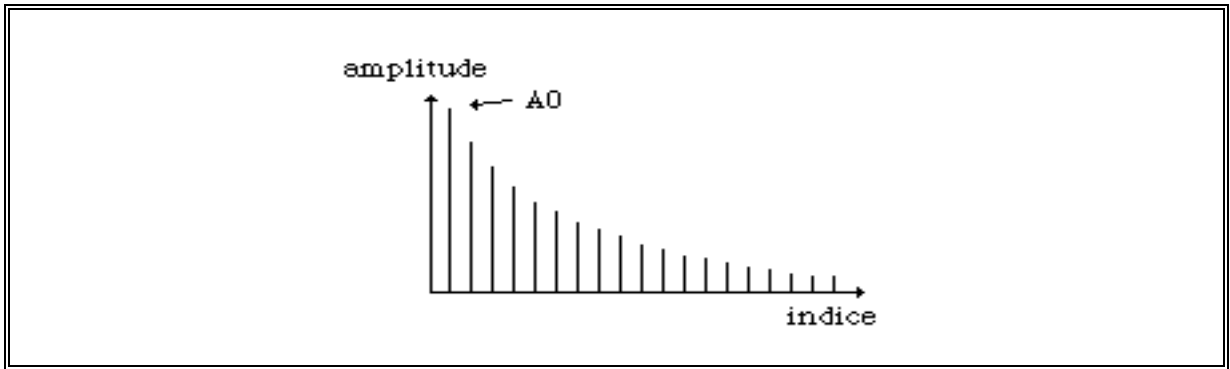


figure 16

De manière générale on peut prendre comme excitation une suite d'harmoniques telle que le rapport d'amplitude de chaque harmonique au fondamental dépende du rang de l'harmonique .

$$\frac{A_n}{A_0} = \alpha_n$$

Dans l'exemple ci-dessus (fig 16) on a  $\alpha_n = \frac{1}{n^2}$

Le signal  $e(t)$  correspondant à cette suite d'harmoniques est:

$$e(t) = \sum_0^N a_n \cos(n\omega_0 t) + b_n \sin(n\omega_0 t) = \sum_0^N A_n e^{in\omega_0 t}$$

$h(t)$  étant la réponse impulsionnelle de la fonction de transfert, on obtient par transformée de Fourier:

$$s(t) = (e * h)(t) \Leftrightarrow S(\omega) = E(\omega) \cdot H(\omega) = \sum_0^N A_n \delta(\omega - n\omega_0) \cdot H(\omega)$$

soit pour un harmonique  $n$  fixé:

$$S(n\omega_0) = S_n(\omega_0) = A_n \cdot H(n\omega_0)$$

Ce qui montre que pour une fréquence donnée, la valeur de la transformée de Fourier du signal de sortie est égale au produit de la valeur de la transformée de Fourier du signal d'entrée et de la réponse en fréquence du filtre de résonance.

On observe donc la réponse de  $e(t)$  à un filtre de réponse en fréquence  $H(\omega)$ . On se donne une notation pour représenter le  $k^{\text{ème}}$  harmonique de la  $i^{\text{ème}}$  note:

$A_k^i$  : k<sup>ème</sup> harmonique de la note i

$\omega_k^i$  : fréquence associée

on regarde l'harmonique 1 de la réponse à toutes les notes:

$$S_1(\omega_1^i) = E(\omega_1^i) H(\omega_1^i) = A_0 H(\omega_1^i)$$

Pour l'harmonique 2 on a également

$$S_2(\omega_2^i) = E(\omega_2^i) H(\omega_2^i) = A_0 \alpha_2 H(\omega_2^i)$$

et de manière générale pour l'harmonique n

$$S_n(\omega_n^i) = E(\omega_n^i) H(\omega_n^i) = A_0 \alpha_n H(\omega_n^i)$$

Le problème réside à identifier les  $\alpha_k$  et trouver  $S(\omega)$ .

On se donne une nouvelle notation:  $20\text{Log}(S) = X$

On écrit le logarithme de la sortie pour le rang k et k+1

$$\left. \begin{aligned} X_k(\omega_k^i) &= 20 \text{Log}(A_0 \alpha_k) + X(\omega_k^i) \text{ harm. } k \\ X_{k+1}(\omega_{k+1}^i) &= 20 \text{Log}(A_0 \alpha_{k+1}) + X(\omega_{k+1}^i) \text{ harm. } k+1 \end{aligned} \right\} (1)$$

La figure suivante représente sur une grille de fréquence commune les valeurs des harmoniques de même rang pour chaque note.



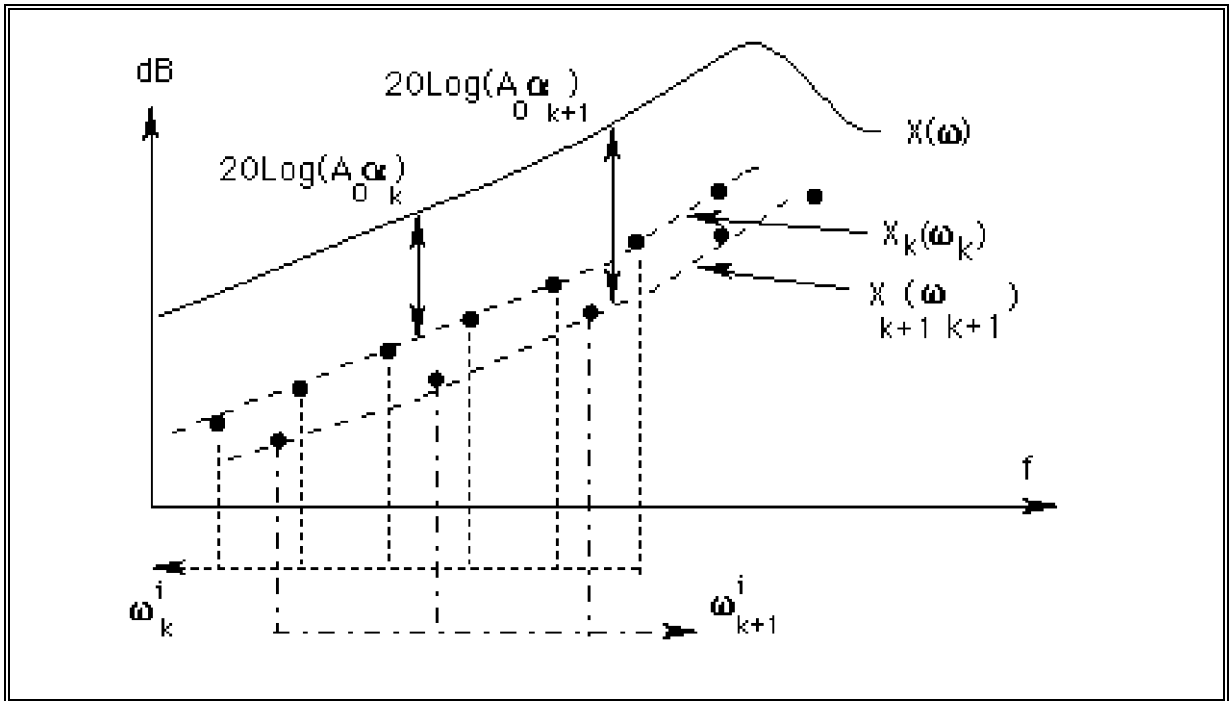


fig 17

Les équations (1) et la figure 17 montrent que si l'on veut calculer l'écart entre les courbes il est nécessaire d'interpoler les  $X_k(\omega_k^i)$  pour les fréquences  $\omega_{k+1}^i$ . Cette interpolation a été définie précédemment: on utilise une simple interpolation linéaire ou bien par polynômes de Lagrange. On obtient ainsi une écriture des  $X_k$  en fonction des  $\omega_{k+1}^i$ . Le système (1) s'écrit alors:

$$\left. \begin{aligned} X_k(\omega_{k+1}^i) &= 20 \text{Log}(A_0 \alpha_k) + X(\omega_{k+1}^i) \text{ harm. } k \text{ (a)} \\ X_{k+1}(\omega_{k+1}^i) &= 20 \text{Log}(A_0 \alpha_{k+1}) + X(\omega_{k+1}^i) \text{ harm. } k+1 \text{ (b)} \end{aligned} \right\} (2)$$

On effectue alors (a) - (b) pour réaliser la disparition des  $X(\omega_{k+1}^i)$

$$\begin{aligned} X_k(\omega_{k+1}^i) - X_{k+1}(\omega_{k+1}^i) &= 20(\text{Log}(A_0 \alpha_k) - \text{Log}(A_0 \alpha_{k+1})) \\ &= 20 \text{Log}\left(\frac{A_0 \alpha_k}{A_0 \alpha_{k+1}}\right) \end{aligned} \quad (3)$$

Les formules (1) montrent que le spectre  $X(\omega)$  s'obtient par assemblage des  $X_k(\omega)$  après renormalisation par les  $\alpha_k$  obtenus auparavant. Comment cela se réalise-t-il en pratique?.

De manière schématique on fait apparaître deux étapes:

1- On considère en sortie la réponse à tous les harmoniques de même rang et on interpole les morceaux de courbe pour toutes les fréquences.

2- On mesure les écarts entre les courbes: on obtient la source (formule 3). Le parallélisme des courbes que l'on observe est dû aux coefficients  $\alpha_k$ , identiques au rang k. Pour reconstituer le spectre complet, on remonte chaque courbe sur la précédente. L'écart entre chaque courbe donne une approximation de l'excitation, laquelle doit correspondre aux  $\alpha_k$ .

On replace par normalisation les morceaux, grâce aux valeurs précédentes, sur la courbe provenant de l'harmonique 1: on obtient la résonance.

Pour l'ensemble des courbes obtenues expérimentalement (fig 13), on calcule une estimation de l'écart entre l'amplitude de chaque raie de l'excitation. Pour réaliser cette approximation on recalcule les valeurs sur l'axe de fréquences défini précédemment, en calculant par interpolation les nouvelles valeurs manquantes de la courbe. On effectue alors les différences entre chaque point de 2 courbes successives dans leur zone de recouvrement (fig 19).

La moyenne de l'écart est donnée par:

$$\bar{e} = \sum_{i=i_1}^{i=i_2} \frac{X_k(\omega_{k+1}^i) - X_{k+1}(\omega_{k+1}^i)}{i_2 - i_1 + 1}$$

$i_1$ : indice de début de recouvrement

$i_2$ : indice de fin de recouvrement

la variance sera calculée par la formule suivante:

$$v = \frac{\sum_{i=i_1}^{i=i_2} e^2 - N \cdot \bar{e}^2}{N-1} \quad \text{avec } N = i_2 - i_1 + 1$$

et l'écart type par la relation :

$$\sigma = \sqrt{v}$$

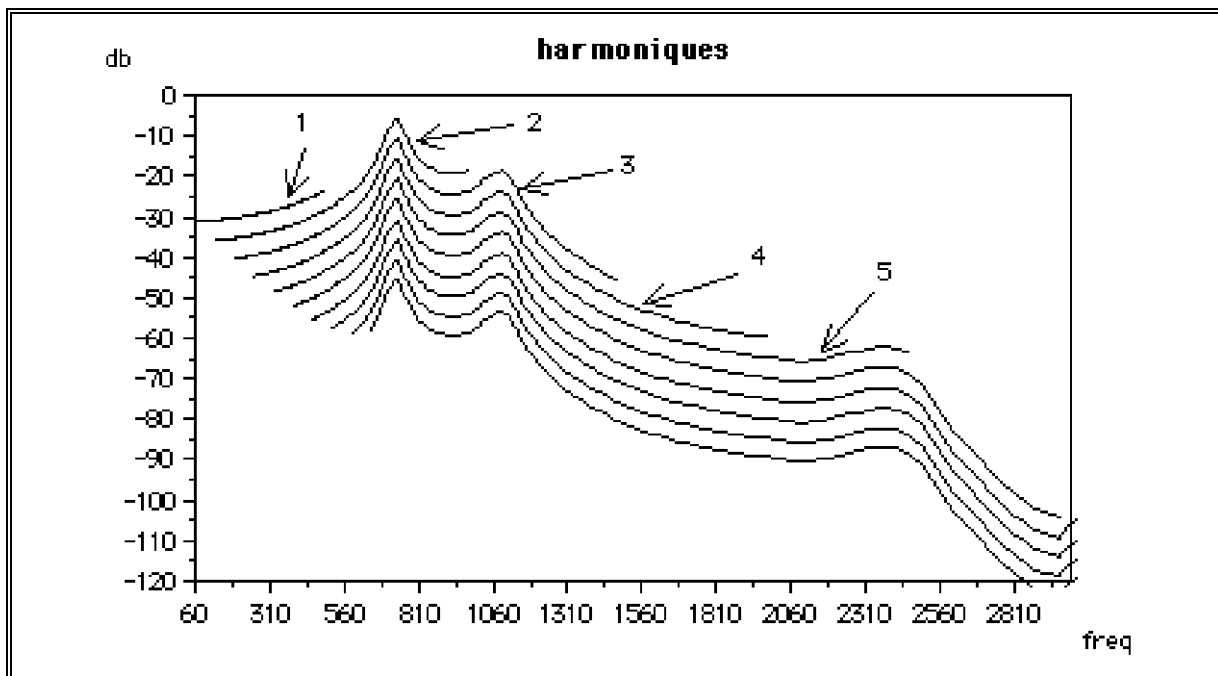


figure 18 - Représentation de l'amplitude de chaque raie du spectre du signal de sortie en calculant une FFT dans la partie stationnaire du signal. On calcule toutes les FFTs pour toutes les notes et on porte l'ensemble des valeurs des spectres ainsi échantillonnées sur un même axe fréquentiel, l'amplitude étant en décibels.

On relie graphiquement les valeurs correspondants aux harmoniques de même rang pour chacune des notes. Cette façon de faire revient à interpoler les points voisins de même rang. Cela permet dans les zones de recouvrement de calculer l'écart entre deux morceaux de courbe.

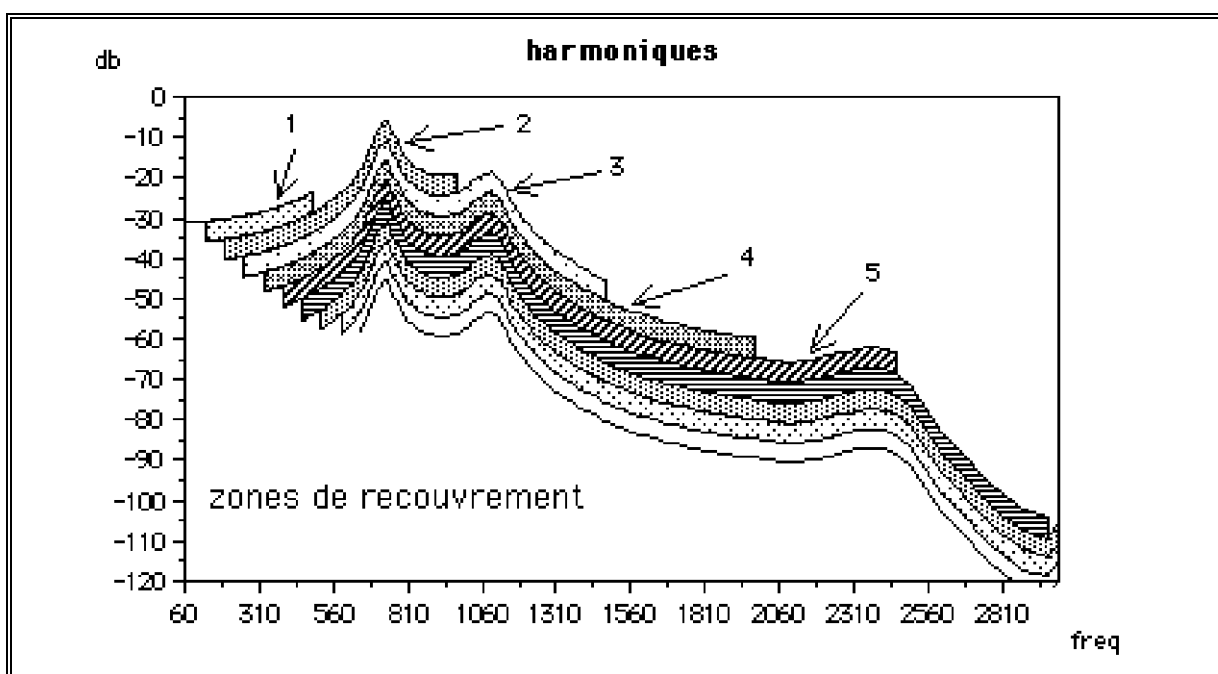
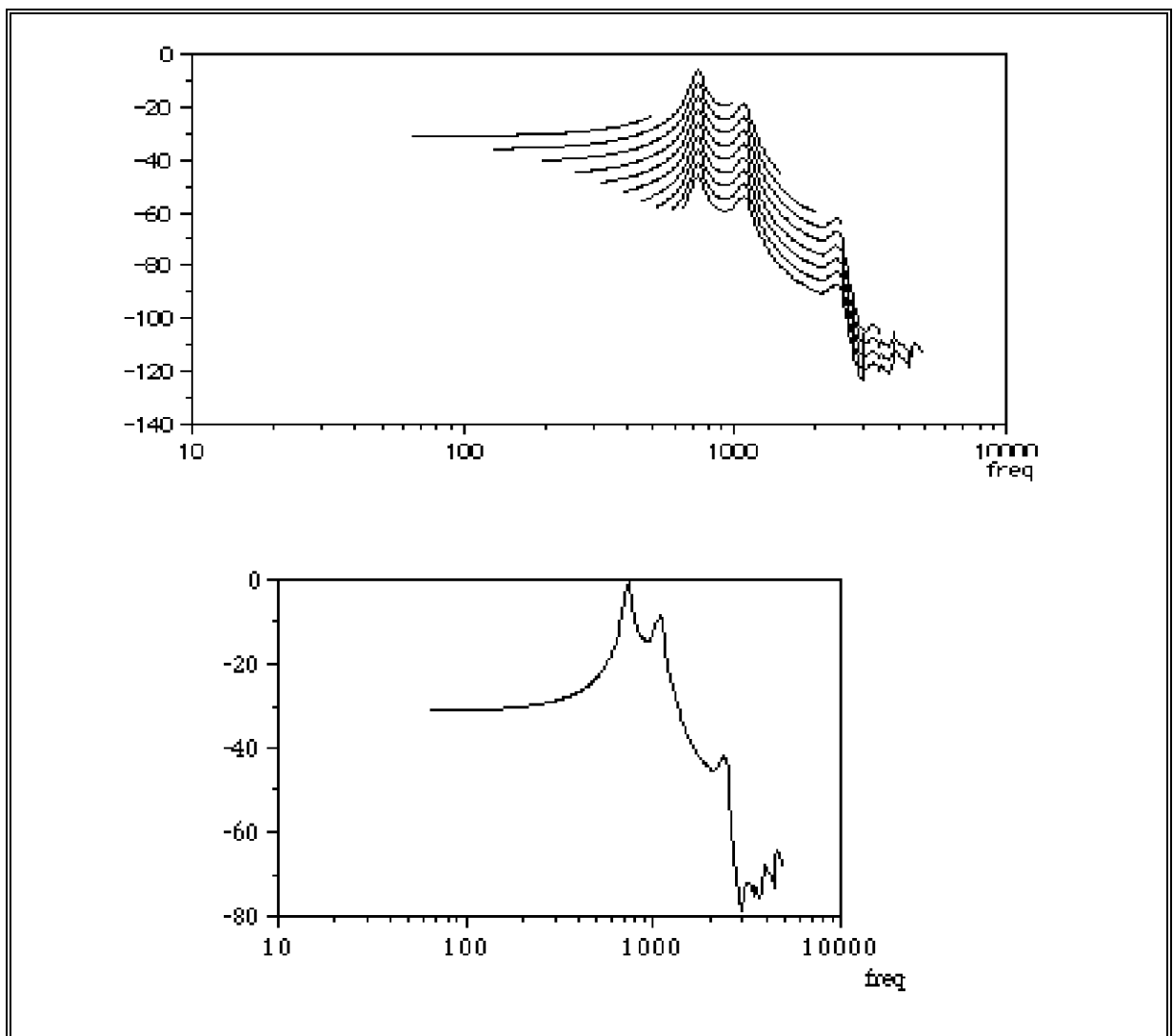


figure 19 - représentation des zones de recouvrement.

La condition initiale étant une excitation identique pour chaque note, l'ensemble des valeurs est obtenu par translation fréquentielle. Cette translation, due à l'utilisation de la gamme chromatique, se fait avec des valeurs bien connues qui correspondent aux valeurs du fondamental et de ses harmoniques.

Si tous les harmoniques de rang  $n$  ont pour chaque note la même amplitude en entrée (spectre constant), le spectre de sortie est une version échantillonnée de la réponse en fréquence du filtre.

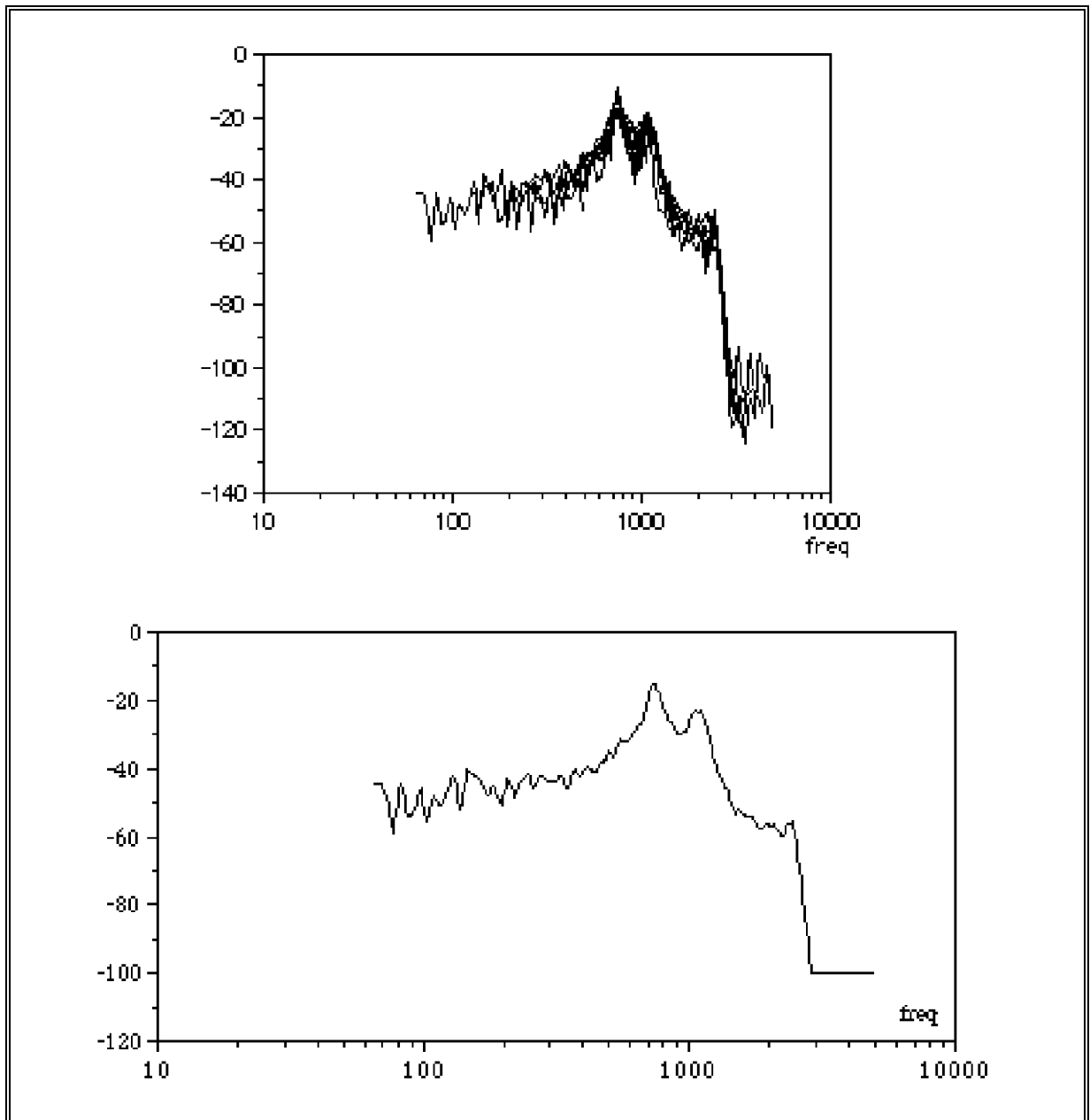
Cette façon de procéder expérimentale permet d'obtenir un échantillonnage du spectre qui servira par la suite de valeurs initiales pour calculer les coefficients du filtre récursif employé par le modèle. Cette méthode conduit à la connaissance du module d'un filtre. La méthode de modélisation utilisée nécessite également la connaissance de la phase du filtre.



*figure 20 - Module du spectre reconstitué à partir d'une excitation fixe translatée suivant l'axe des fréquences (gamme chromatique)*

Cette courbe correspond au module de la réponse d'un filtre caractéristique des résonances du système.

Dans le cas d'un spectre dont le module de chaque harmonique varie peu autour d'une moyenne on obtient des résultats analogues mais d'autant plus biaisés que l'on dispose de moins de mesures. Biais minimal dans la zone de recouvrement maximal (fig 21). Le lissage de l'enveloppe permet d'obtenir un gabarit spectral plus réaliste.



*figure 21 - spectre reconstitué à partir d'un excitation dont les harmoniques varient autour d'une moyenne fixe. Apparition de bruit dans les basses fréquences.*

### VI.4.a.3. modélisation de l'enveloppe spectrale

#### VI.4.a.3.1. modélisation de la phase: phase minimale

L'indétermination sur le déphasage peut être levée en caractérisant le filtre par un temps de propagation minimal, ce qui conduit à rechercher une caractéristique de déphasage minimal. La réponse d'un signal au filtre ne pouvant précéder son application, le filtre est nécessairement causal. Si  $h(n)$  est la réponse impulsionnelle à coefficients réels:

$$h(n) = 0 \text{ pour } n < 0$$

la réponse peut se décomposer en partie réelle et partie imaginaire:

$$H(f) = H_r(f) + H_i(f)$$

Les propriétés de la FFT de suites réelles et causales conduisent aux relations:

$$H_r(f) = h(0) + \sum_{n=1}^{\infty} h(n) \cos(2\pi n f)$$

$$H_i(f) = - \sum_{n=1}^{\infty} h(n) \sin(2\pi n f)$$

En introduisant la FFT au sens des distributions de la fonction échelon, on constate que partie réelle et partie imaginaire d'une suite causale sont liées par la transformée de Hilbert des signaux continus [BELLANGER, 1984]:

$$H_r(f) = h(0) + \int_{-1/2}^{1/2} [H_i(f) - H_i(f')] \cotg[\pi(f-f')] df$$

$$H_i(f) = \int_{-1/2}^{1/2} [H_r(f) - H_r(f')] \cotg[\pi(f-f')] df$$

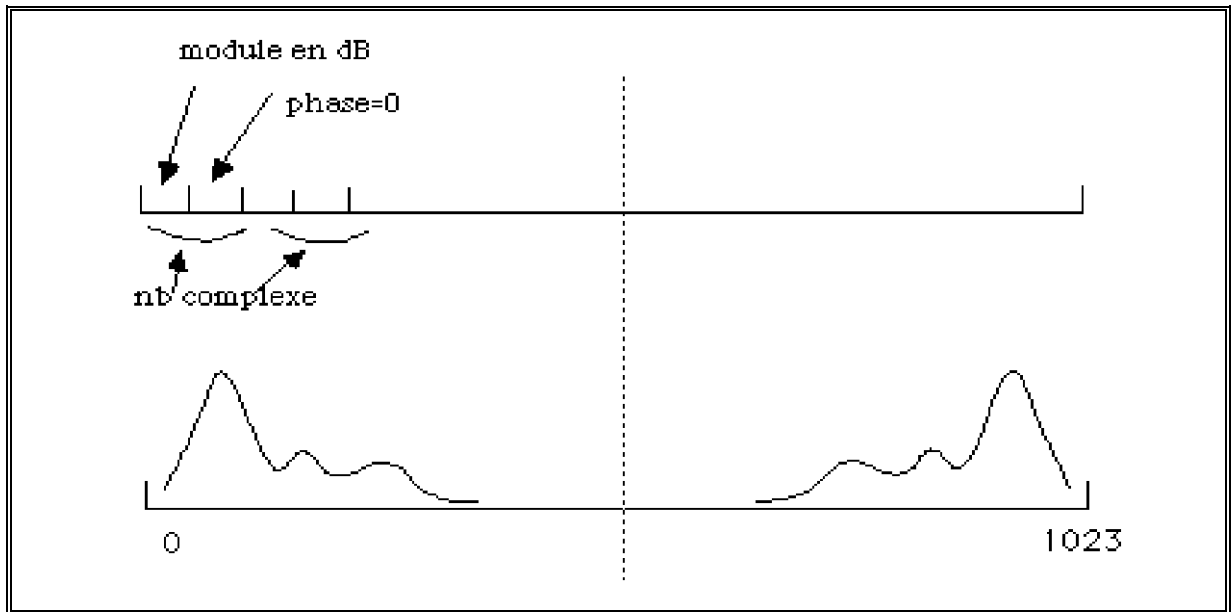
Un filtre stable et réalisable possède une fonction de transfert en  $Z$  dont les pôles sont à l'intérieur du cercle unité, les zéros pouvant être à l'extérieur. La condition de phase minimale est que la fonction:

$$\text{Log}[H(Z)] = \text{Log}[A(Z)] - j\phi(Z)$$

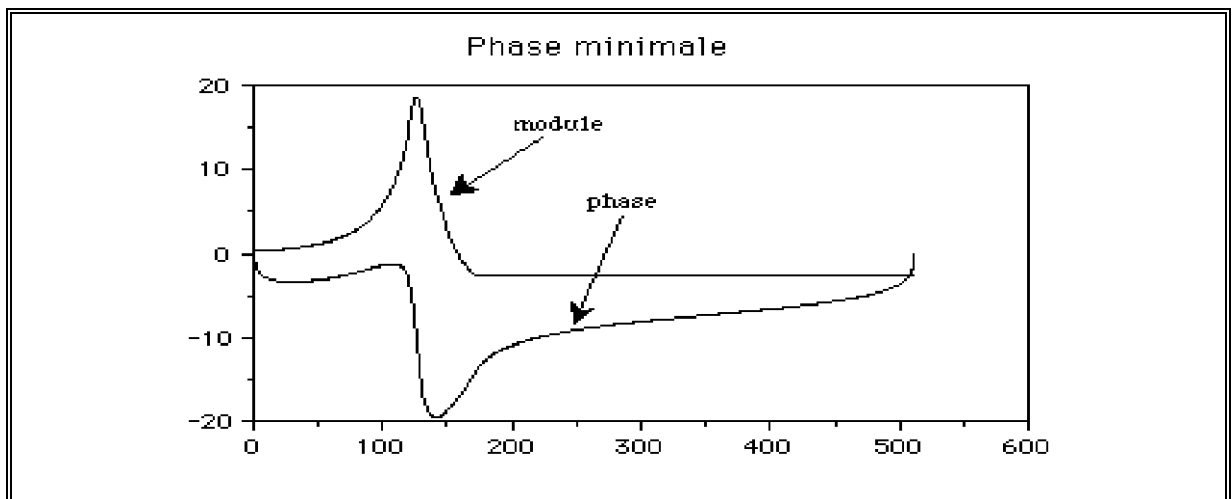
n'ait pas de pôle à l'extérieur du cercle unités [Bellanger, 1984].

Pratiquement le programme on calcule la phase minimale à partir de la connaissance du module du spectre de la manière suivante:

On place dans un tableau de nombres complexes le Logarithme du module et on positionne les phases de chaque harmonique à zéro.



On symétrise la courbe de réponse, on calcule la transformée de Fourier inverse, on élimine les fréquences négatives. Puis on fait de nouveau une transformée de Fourier. Si on a pris la précaution de multiplier par deux les valeurs de la première moitié du tableau et d'annuler la valeur initiale, on retrouve le module inchangé dans la partie réelle et une phase dans la partie imaginaire. La connaissance de ces 2 valeurs permet de connaître le spectre complexe qui sera employé par la suite dans la recherche des coefficients du filtre.





Phase minimale reconstituée à partir de la courbe de réponse d'un filtre du second ordre.

#### VI.4.a.3.2. estimation des coefficients du filtre

La forme la plus générale d'un filtre à réponse impulsionnelle infinie peut être donnée par sa transformée en Z telle que :

$$H(z) = \sum_{n=0}^N h(n) \cdot z^{-n} = \frac{\sum_{m=0}^M a_m \cdot z^{-m}}{1 + \sum_{n=1}^N b_n \cdot z^{-n}}$$

avec  $z = e^{j\omega}$

Les données expérimentales sont constituées par un échantillonnage de la fonction de transfert précédemment reconstituée. Ces valeurs sont des nombres complexes obtenus grâce à la connaissance de la phase minimale, obtenue dans une étape antérieure (PHASE.PAS).

La fonction poids peut être obtenue à l'aide d'un programme qui permet d'affecter visuellement les coefficients de pondération aux zones formantiques (POIDS.PAS) ou grâce à la fonction poids définie lors de la création de la grille des fréquences expérimentales.

Le problème est donc de trouver les  $\{a_m\}_{m=0,M}$  et les  $\{b_n\}_{n=1,N}$  qui minimisent au sens des moindres carrés l'erreur quadratique moyenne donnée par l'expression :

$$E(a_m, b_n) = \sum_{k=0}^K P_k \left| H_{ex}(z_k) - \frac{\sum_{m=0}^M a_m \cdot z_k^{-m}}{1 + \sum_{n=1}^N b_n \cdot z_k^{-n}} \right|^2$$

$E(a_m ; b_n)$  est la somme pondérée pour tous les  $z_k$  du module de la différence entre la réponse expérimentale et la réponse théorique du filtre. Les nombres  $\{P_k\}_{k=0,K}$  sont des poids positifs destinés à augmenter l'importance de certaines parties du spectre.

$$E(a_m ; b_n) = \sum_{k=0}^K P_k \left| H_{\text{ex}}(z_k) - \frac{N(z_k)}{D(z_k)} \right|^2$$

$$= \sum_{k=0}^K P_k \frac{|H_{\text{ex}}(z_k) \cdot D(z_k) - N(z_k)|^2}{|D(z_k)|^2}$$

L'erreur quadratique  $E(a_m ; b_n)$  sera minimale si les dérivées partielles par rapport aux coefficients  $a_m$  et  $b_n$  sont nulles.

$$\frac{\delta E(a_i ; b_j)}{\delta a_i} = 0 \quad i = 0, M$$

$$\frac{\delta E(a_i ; b_j)}{\delta b_j} = 0 \quad j = 1, N$$

On se ramène à la résolution du système suivant <sup>4</sup>:

$$\sum_{k=0}^K P_k \cdot \left[ \frac{2}{|D(z_k)|^2} \cdot \text{Re} \left\{ z_k^{*-i} \cdot (N(z_k) - H_{\text{ex}}(z_k) \cdot D(z_k)) \right\} \right] = 0$$

,  $i = 0, M$

$$\sum_{k=0}^K P_k \cdot \left[ \frac{2}{|D(z_k)|^2} \cdot \text{Re} \left\{ (H_{\text{ex}}(z_k) \cdot z_k^{-j})^* \cdot (H_{\text{ex}}(z_k) \cdot D(z_k) - N(z_k)) \right\} \right]$$

$$- \sum_{k=0}^K P_k \left[ \frac{|H_{\text{ex}}(z_k) \cdot D(z_k) - N(z_k)|^2}{|D(z_k)|^4} \cdot 2 \cdot \text{Re} \left\{ D(z_k) \cdot z_k^{*-j} \right\} \right] = 0$$

,  $j = 1, N$

---

<sup>4</sup>  $|z|^2 = z \cdot z^*$  ,  $\frac{|z|^2}{x} = 2 \cdot \text{Re} \left\{ z \cdot \left( \frac{z}{x} \right)^* \right\}$

### VI.4.a.3.3. Réduction à un problème linéaire

Ce système d'équations est fortement non linéaire . Pour ne pas utiliser dans un premier temps des méthodes de résolution de système non linéaire, on transforme ce système en un système linéaire par rapport aux  $a_i$  et  $b_j$  en utilisant la simplification proposée par Christian CARME [CARME 86]. Cette "linéarisation" est réalisée en remplaçant le terme  $\left| D(z_k) \right|^2$  par 1 chaque fois qu'il se présente au dénominateur et en négligeant le second terme de la relation établie précédemment.

On obtient donc le système suivant :

$$\sum_{k=0}^K P_k \cdot \left[ \operatorname{Re} \left\{ z_k^{*-i} \cdot (N(z_k) - H_{\text{ex}}(z_k) \cdot D(z_k)) \right\} \right] = 0 ,$$

$$i = 0 , M$$

$$\sum_{k=0}^K P_k \cdot \left[ \operatorname{Re} \left\{ (H_{\text{ex}}(z_k) \cdot z_k^{-j})^* \cdot (H_{\text{ex}}(z_k) \cdot D(z_k) - N(z_k)) \right\} \right] = 0$$

,

$$j = 1 , N$$

soit

$$\sum_{k=0}^K P_k \cdot \left[ \operatorname{Re} \left\{ z_k^{*-i} \cdot \left( \sum_{m=0}^M a_m z_k^{-m} - H_{\text{ex}}(z_k) \cdot \left( 1 + \sum_{n=1}^N b_n \cdot z_k^{-n} \right) \right) \right\} \right] = 0$$

$$i = 0 , M$$

$$\sum_{k=0}^K P_k \cdot \left[ \operatorname{Re} \left\{ (H_{\text{ex}}(z_k) \cdot z_k^j)^* \cdot \left( H_{\text{ex}}(z_k) \cdot \left( 1 + \sum_{n=1}^N b_n \cdot z_k^{-n} \right) - \sum_{m=0}^M a_m \cdot z_k^{-m} \right) \right\} \right] = 0$$

$$j = 1 , N$$

en développant les équations précédentes on obtient le système matriciel suivant:

$$\sum_{m=0}^M \sum_{k=0}^K P_k a_m \operatorname{Re}\{Z_k^{i-m}\} - \sum_{n=1}^N \sum_{k=0}^K P_k b_n \operatorname{Re}\{H_{\text{ex}}(Z_k) Z_k^{i-n}\}$$

$$= \sum_{k=0}^K P_k \operatorname{Re}\{H_{\text{ex}}(Z_k) Z_k^i\}$$

$$i = 0, M$$

$$\sum_{m=0}^M \sum_{k=0}^K P_k a_m \operatorname{Re}\{H_{\text{ex}}^*(Z_k) Z_k^{j-m}\} - \sum_{m=0}^M \sum_{k=0}^K P_k b_n |H_{\text{ex}}(Z_k)|^2$$

$$\operatorname{Re}\{Z_k^{j-n}\}$$

$$= \sum_{k=0}^K P_k |H_{\text{ex}}(Z_k)|^2 \operatorname{Re}\{Z_k^j\}$$

$$j = 1, N$$

Le vecteur  $X$  solution des  $a_m$  et  $b_n$  est obtenu en résolvant le système linéaire suivant

:

$$A \cdot X = B$$

où  $A$  est une matrice carré de la forme :

$$\begin{pmatrix} \sum_{k=0}^K P_k \operatorname{Re}\{Z_k^{i-m}\} & \sum_{k=0}^K P_k \operatorname{Re}\{H_{\text{ex}}(Z_k) Z_k^{i-n}\} \\ \dots & \dots \\ \sum_{k=0}^K P_k \operatorname{Re}\{H_{\text{ex}}^*(Z_k) Z_k^{j-m}\} & \sum_{k=0}^K P_k |H_{\text{ex}}(Z_k)|^2 \operatorname{Re}\{Z_k^{j-n}\} \end{pmatrix} \begin{matrix} i=0 \\ \dots \\ j=1 \\ \dots \\ N \end{matrix}$$

$B$  le vecteur colonne suivant:

$$\begin{pmatrix} \sum_{k=0}^K P_k \operatorname{Re}\{H_{\text{ex}}(Z_k) Z_k^i\} \\ \dots \\ \sum_{k=0}^K P_k |H_{\text{ex}}(Z_k)|^2 \operatorname{Re}\{Z_k^j\} \end{pmatrix}_{\substack{i=0 \\ \dots \\ j=1 \\ \dots \\ N}}^M$$

et X le vecteur solution :

$$\begin{pmatrix} a_i \\ \dots \\ b_j \end{pmatrix}_{\substack{i=0 \\ \dots \\ j=1 \\ \dots \\ N}}^M$$

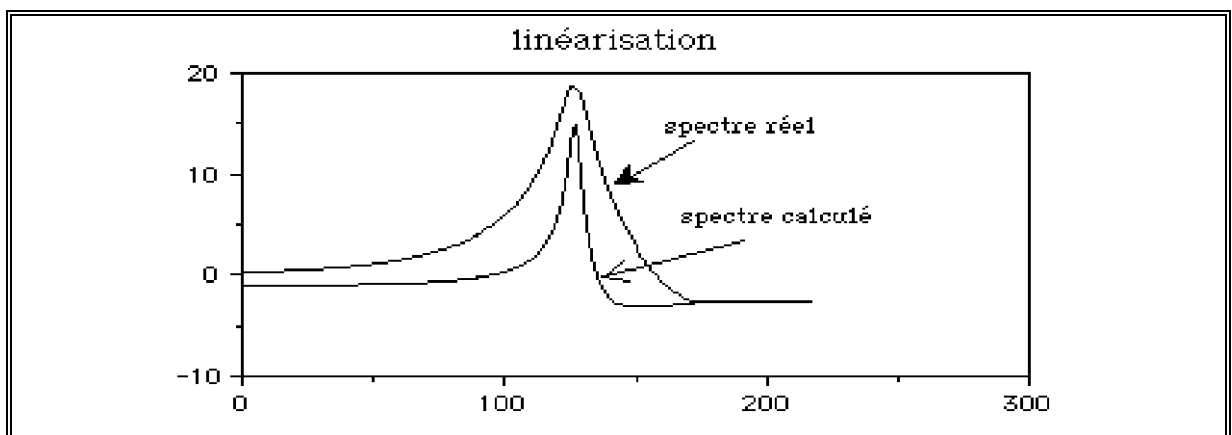
On est donc amené à résoudre le système matriciel suivant:

$$X = A^{-1}B$$

Pour inverser la matrice A et résoudre le système linéaire nous allons employer la méthode de Gauss-Jordan. L'intérêt principal de cette méthode réside dans le fait qu'elle est très stable surtout si on l'associe à la méthode du pivot. [Press & al, 1987].

Le programme LINEAR.PAS calcule à partir de l'échantillonnage de la fonction de transfert et de la connaissance de la fonction poids, un vecteur solution comprenant les coefficients  $\{a_i\}_{i=0, M}$ ,  $\{b_j\}_{j=1, N}$  que l'on peut introduire dans l'équation générale du filtre afin de visualiser les deux courbes (expérimentale et modélisée) sur un même axe de fréquences.

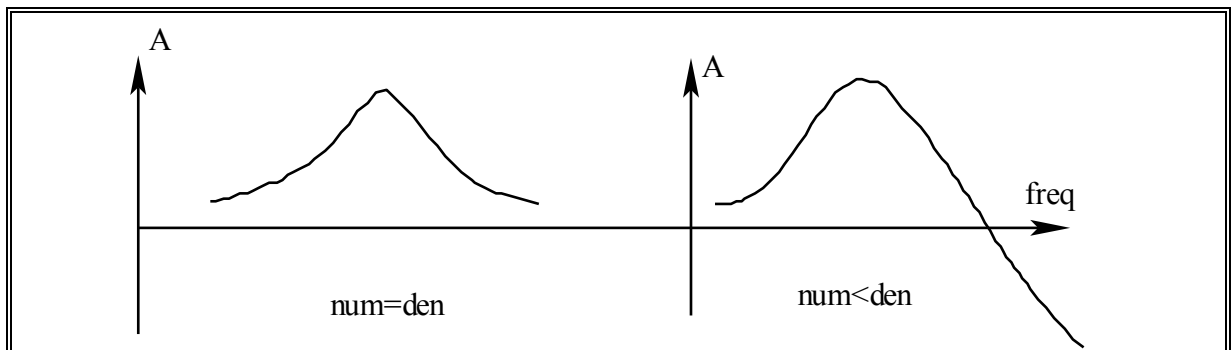
Avec un exemple de filtre du second ordre, on obtient les résultats suivants:



*Exemple d'approximation que l'on obtient en traçant le module de la fonction de transfert obtenu avec les valeurs des coefficients après linéarisation*

On constate une bonne approximation de la courbe expérimentale mais la correspondance des formants n'est pas maximale.

Pendant l'utilisation du programme LINEAR.PAS, il faut introduire les degrés des polynômes au numérateur et au dénominateur. Cela revient à fixer le nombre de pôles et de zéros de la fonction de transfert qui assureront la stabilité du filtre solution. Une méthode consiste à associer de manière empirique à un formant un filtre correspondant à 2 pôles et 2 zéros. Cela se traduit par la correspondance suivante: si la fonction de transfert présente à l'écran  $n$  bosses (formants) on prendra comme degré du numérateur et du dénominateur  $2n$ . Cette conduite est également dictée par le fait qu'un filtre ayant le même nombre de pôles et de zéros présente un module de sa fonction de transfert constant en particulier lorsque la fréquence approche  $f_c/2$ .



Ce type de filtre est plus facilement inversible à l'aide d'un filtre passe-tout (NOCH).

#### VI.4.a.3.4. méthode de Marquardt

Pour affiner le calcul des coefficients de la fonction de transfert modélisée, on utilise le vecteur solution comme initialisation d'une méthode itérative basée sur la méthode de Levenberg-Marquardt [Cf Press & al., 1987].

Pour réaliser l'optimisation des coefficients à l'aide d'une méthode de modélisation non-linéaire on reprend la forme générale de la transformée en  $Z$  du filtre à réponse impulsionnelle infinie définie précédemment.

$$H(z) = \sum_{n=0}^N h(n) \cdot z^{-n} = \frac{\sum_{m=0}^M a_m \cdot z^{-m}}{1 + \sum_{n=1}^N b_n \cdot z^{-n}}$$

$$z = e^{j\omega}$$

On calcule le gradient de la fonction  $\chi^2$  (chi 2) défini par:

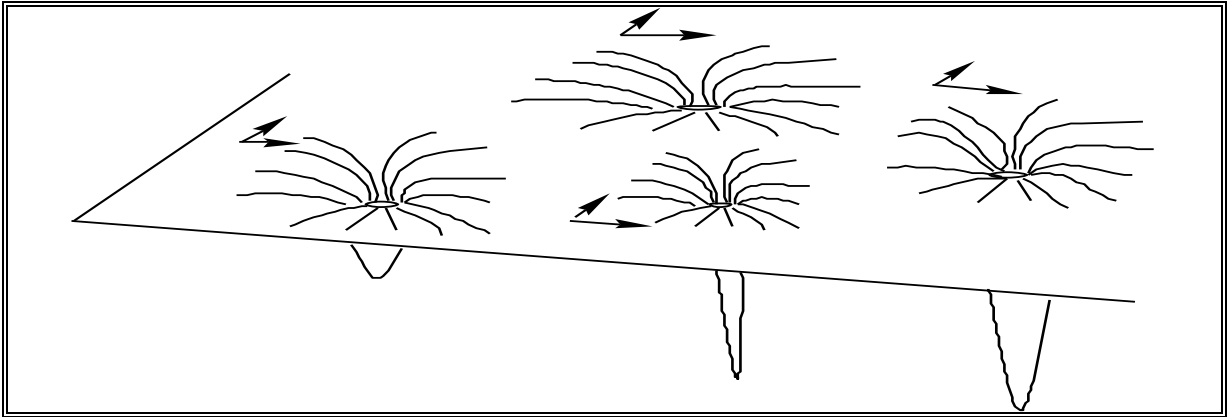
$$\chi^2(z, a, b) = \sum_{i=1}^{i=N} \left( \frac{H_{ex}(z) - H(z, a, b)}{\sigma_i} \right)^2$$

La fonction  $\chi^2$  sera minimale si son gradient est nul:

$$\frac{\delta \chi^2(z, a, b)}{\delta a} = 0$$

$$\frac{\delta \chi^2(z, a, b)}{\delta b} = 0$$

De manière poétique on peut dire que les solutions qui minimisent le  $\chi^2$  sont des valeurs du plan complexe qui sont visualisées par des creux sur le plan. Par la méthode du gradient, on plonge par itérations successives à l'intérieur de ces creux en étant guidé par la pente. La simplification apportée par la linéarisation du problème a permis de se positionner aux alentours des solutions et bénéficier ainsi d'une convergence plus rapide vers la solution.

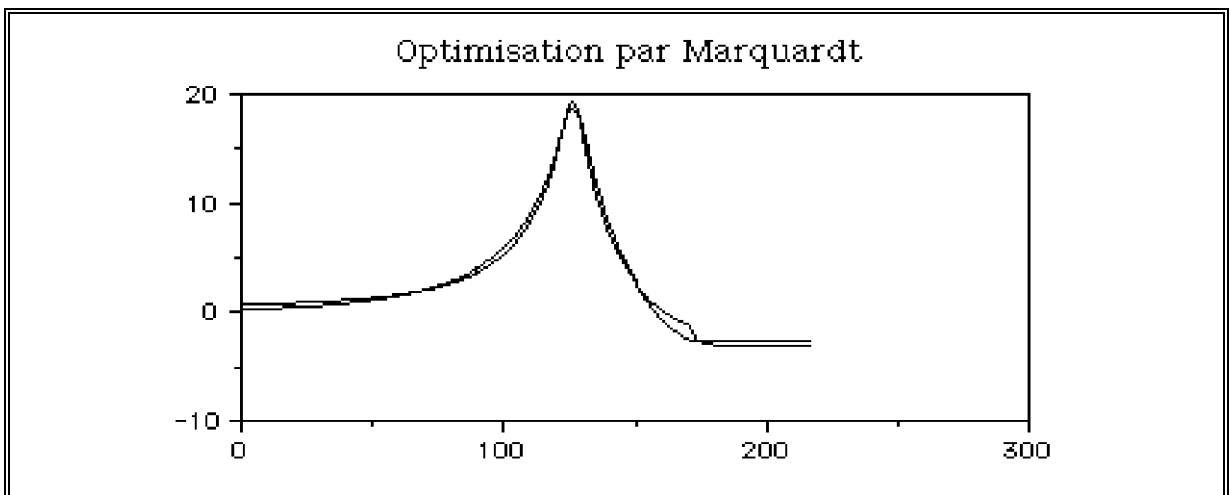


*Les valeurs initiales sont disposées sur le plan. Si les solutions ne sont pas trop éloignées, il y aura convergence.*

On espère que les valeurs trouvées par la résolution après simplification du système linéaire sont des points du plan, pas trop éloignés des solutions.

Une solution consiste à calculer un nouveau gradient dans une direction à partir de l'ensemble de valeurs initiales. Si le  $\chi^2$  diminue on continue dans cette direction vers la solution, sinon on change de direction. Cette méthode est décrite dans Numerical Recipes [Press & al., 1987].

Le programme associé à cette méthode nécessite pour sa mise en oeuvre un ensemble de valeurs initiales, la fonction de transfert ainsi que ses dérivées partielles par rapport aux coefficients (COEFP.PAS).



*L'algorithme de Marquardt appliqué au résultat précédent. On constate une nette amélioration par rapport à la linéarisation du cas précédent.*



#### VI.4.a.3.5. méthode de Laguerre

La fonction de transfert  $H(z)$  doit être mise sous une forme différente pour une étude de la stabilité du filtre. Cela signifie qu'il faut trouver les pôles et les zéros du numérateur et du dénominateur afin de mettre  $H(z)$  sous la forme:

$$H(z) = \frac{N(z)}{D(z)} = \frac{\prod_{i=1}^{i=N} z - z_i}{\prod_{j=1}^{j=M} z - p_j}$$

Pour trouver les zéros il faut résoudre:

$$\sum_{m=0}^M a_m z^{-m} = 0$$

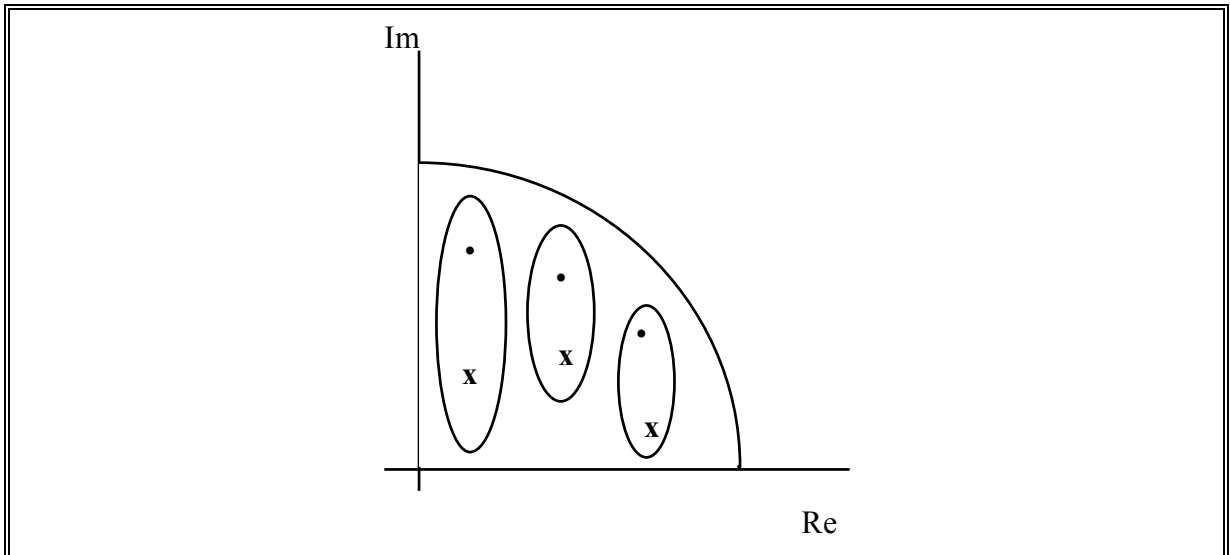
Pour trouver les pôles:

$$1 + \sum_{n=1}^N a_n z^{-n} = 0$$

La méthode employée est la méthode de Laguerre qui permet de trouver par itérations successives les racines complexes d'un polynôme. [Press & al. (1987) p. 263-265].

Cette décomposition en produits permet de réaliser une étude de la stabilité. Après avoir rendu le filtre stable (si celui-ci ne l'était pas), cette méthode permet de regrouper les pôles et les zéros en filtre élémentaire du second ordre pour réaliser une implémentation sur le processeur audiophonique temps réel (SYTER) qui demande une structure en cascade des filtres. La cellule du second ordre est la plus utilisée en raison de la modularité qu'elle apporte dans la réalisation de filtre complexe.

Pour cela on réalise un appariement des pôles et des zéros. Cela se traduit simplement [Cf. Bellanger, 1984] par la procédure qui consiste à associer au pôle le plus proche du cercle unité le zéro le plus voisin.



Cette procédure est réalisée dans le programme SYTERE.PAS.

#### VI.4.a.3.6. Stabilité

On étudie la stabilité du filtre pour vérifier que les pôles se trouvent bien à l'intérieur du cercle unité. Si un pôle se trouve à l'extérieur du cercle unité, une solution pour rendre le filtre stable consiste à conserver l'argument du nombre complexe et changer le module en son inverse.

Cette opération semble suffisante pour obtenir une réponse stable du filtre.

#### VI.4.a.3.7. Synthèse du filtre

La fonction de transfert  $H(z)$  ainsi obtenue peut être mise sous la forme:

$$H(z) = \frac{\sum_{m=0}^M a_m \cdot z^{-m}}{1 + \sum_{n=1}^N b_n \cdot z^{-n}} = \frac{S(z)}{E(z)}$$

$$z = e^{j\omega}$$

On obtient donc l'équation aux différences suivante :

$$x(t-k) \square z^{-k} X(z)$$

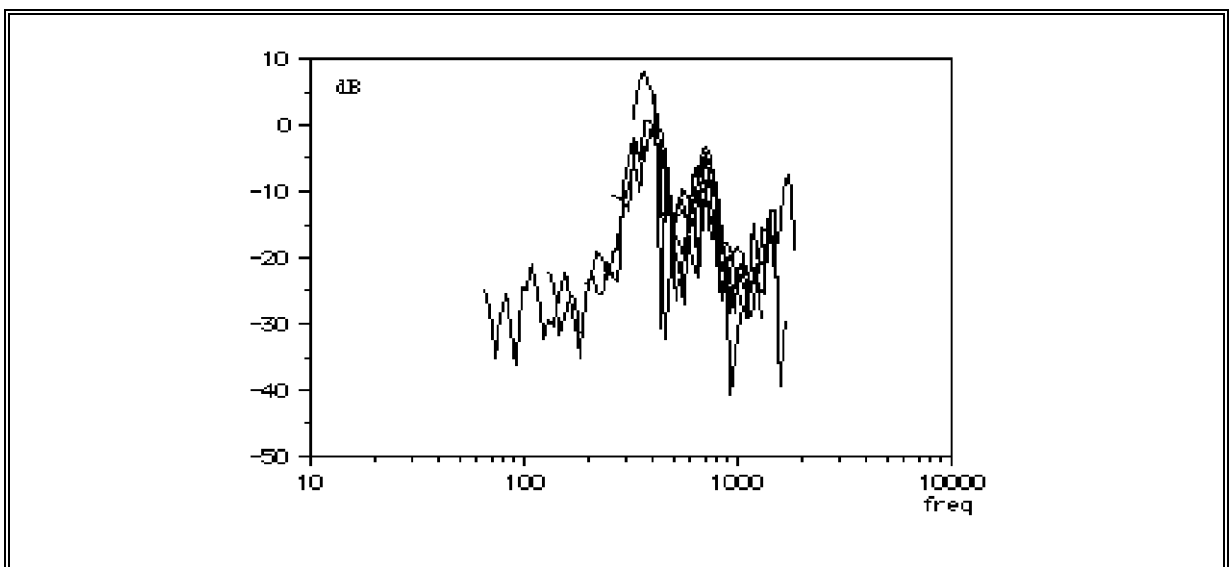
$$S(z) \cdot \left(1 + \sum_{n=1}^N b_n \cdot z^{-n}\right) = E(z) \cdot \sum_{m=0}^M a_m \cdot z^{-m}$$

$$s(t) = \sum_{m=0}^M a_m \cdot e(t-m) - \sum_{n=1}^N b_n \cdot s(t-n)$$

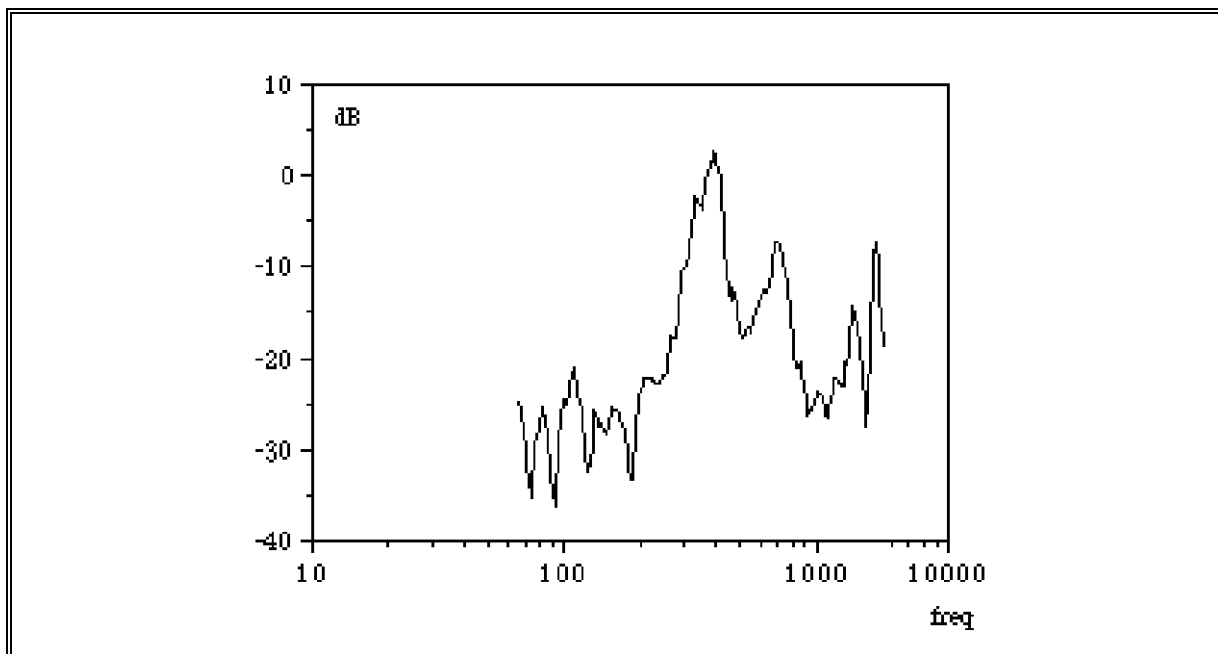
Cette écriture montre que la sortie dépend des échantillons définis aux instants précédents ainsi que des valeurs de l'entrée. La programmation de ce type d'équation temporelle est relativement simple.

On constate à ce niveau des opérations que l'on a réussi, étape par étape, à modéliser l'enveloppe spectrale obtenue expérimentalement à partir du jeu d'une gamme chromatique sur un instrument à formant fixe en imposant une condition sur l'excitation. Cette enveloppe spectrale n'a pas été obtenue n'importe où dans le signal, mais uniquement dans la zone stationnaire ou pseudo-périodique du signal. C'est dans cette zone que l'on a extrait les périodes du signal qui par FFT ont donné l'ensemble de valeurs défini précédemment (morceau de spectre parallèle). Pour réaliser le filtrage inverse, il nous faut approximer l'inverse de l'enveloppe spectrale. Cette opération se déduit de l'opération normale par l'inversion du dénominateur et du numérateur dans l'écriture générale du filtre. Cela ne présente pas de difficulté informatique: on peut considérer que les programmes sont identiques exception faite de l'inversion de la fraction donnée par l'équation du filtre.

#### VI.4.b.Exemple de résultat issu de signaux réels: voix chantée.

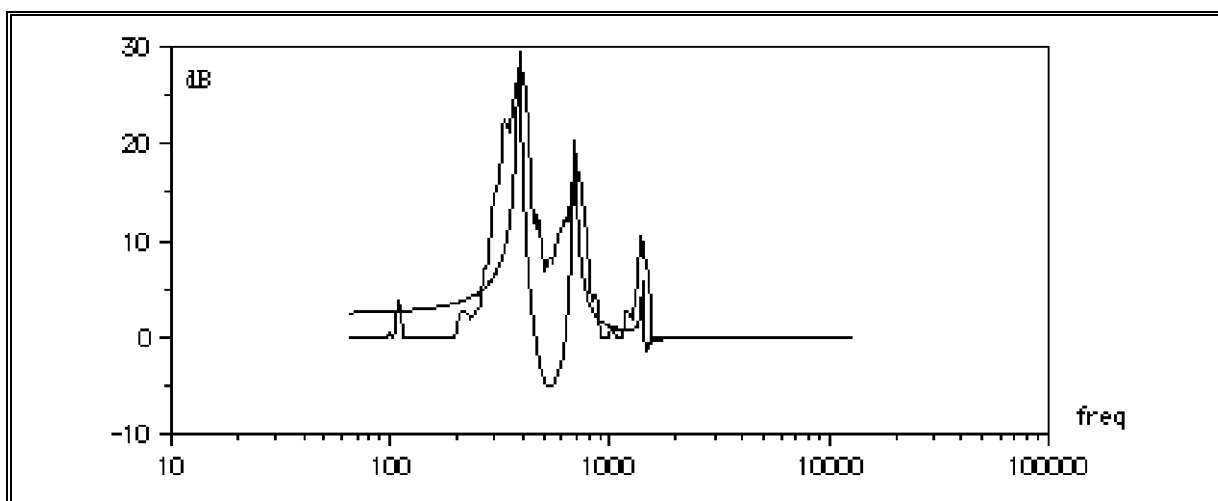


*Données brutes obtenues par la représentation sur un même axe de fréquence des valeurs des 15 premiers harmoniques d'une voyelle "a" chantée suivant la gamme chromatique.*

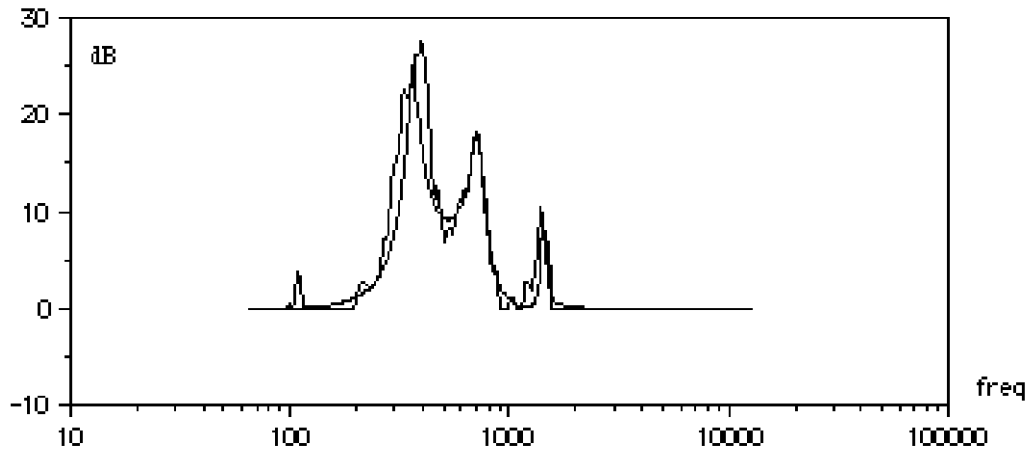


*Spectre reconstitué à partir de l'analyse de la voyelle "a" chantée par le musicien Jacques Dudon.*

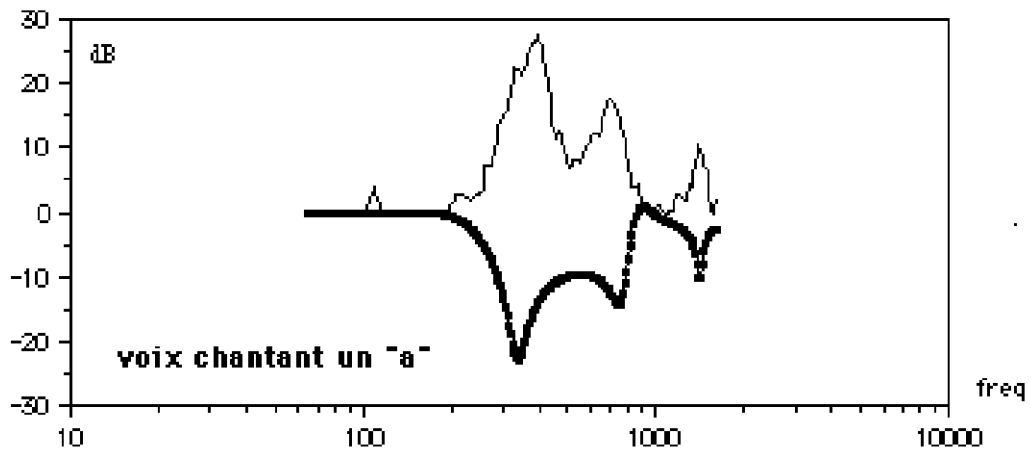
Jacques Dudon a chanté sur la gamme chromatique cette voyelle en effectuant un contrôle du conduit vocal à l'aide d'un miroir. La pratique du chant Tibétain l'a beaucoup aidé pour réaliser cette expérience.



*Après linéarisation on obtient une première approche de l'enveloppe spectrale*



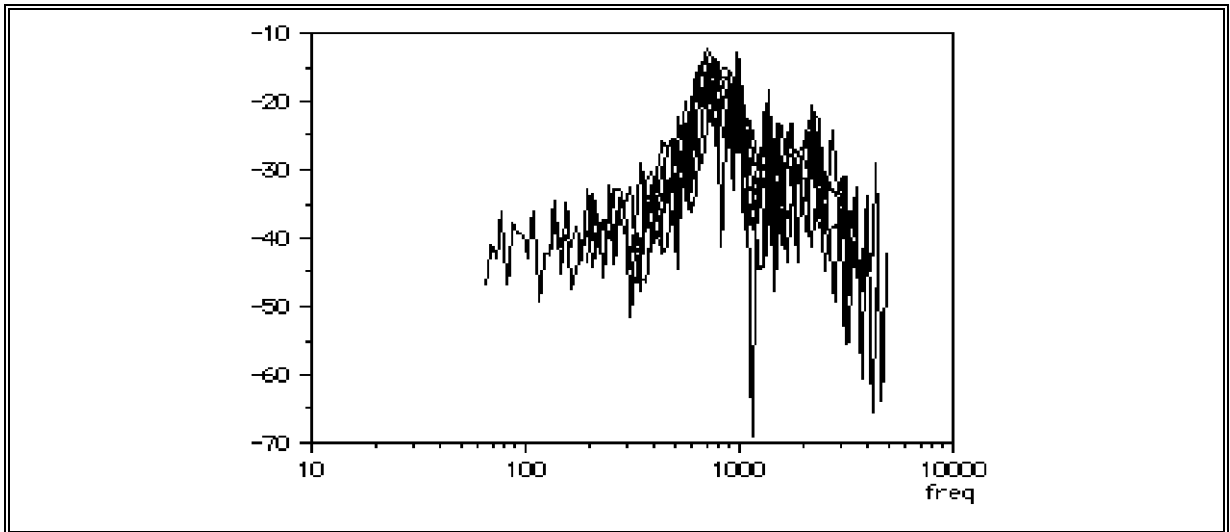
*On affine le modèle par la méthode de Marquard*



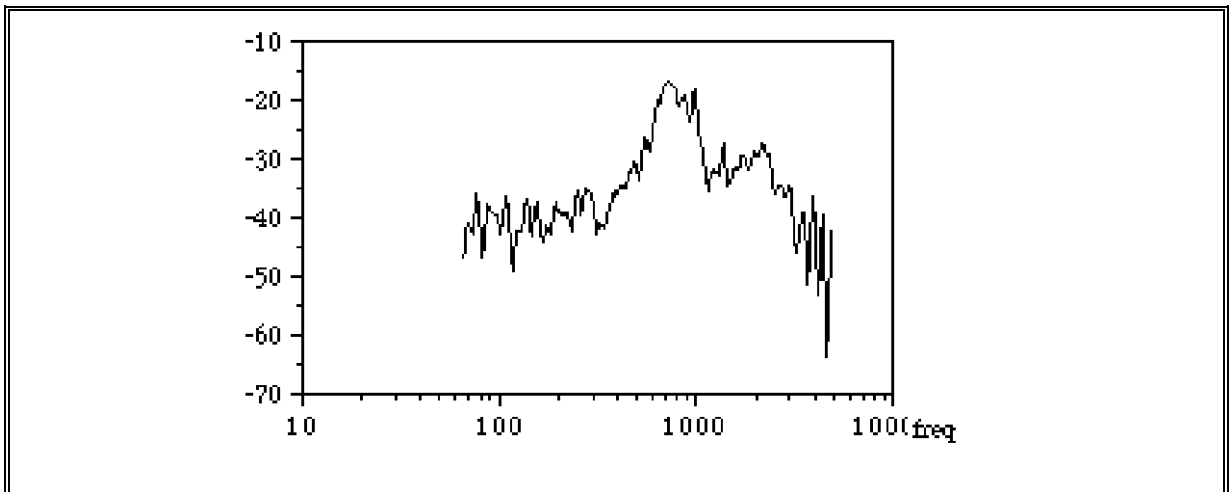
*Calcul du filtre inverse.*

On obtient une bonne approximation du filtre inverse. Les valeurs des coefficients du numérateur et du dénominateur sont conservées dans les deux cas (filtre normal et filtre inverse) afin de réaliser l'antifiltrage et plus tard de constituer le modèle complet: distorsion plus filtrage.

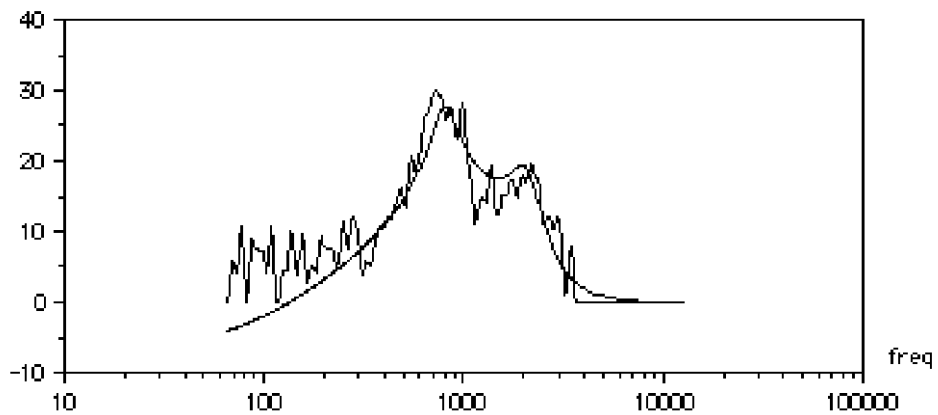
Dans l'exemple suivant on étudie un instrument de fabrication artisanale: le violon courge (réalisé à l'aide d'une courge séchée et d'un manche de mandoline). Cet instrument assez rudimentaire possède une résonance très marquée.



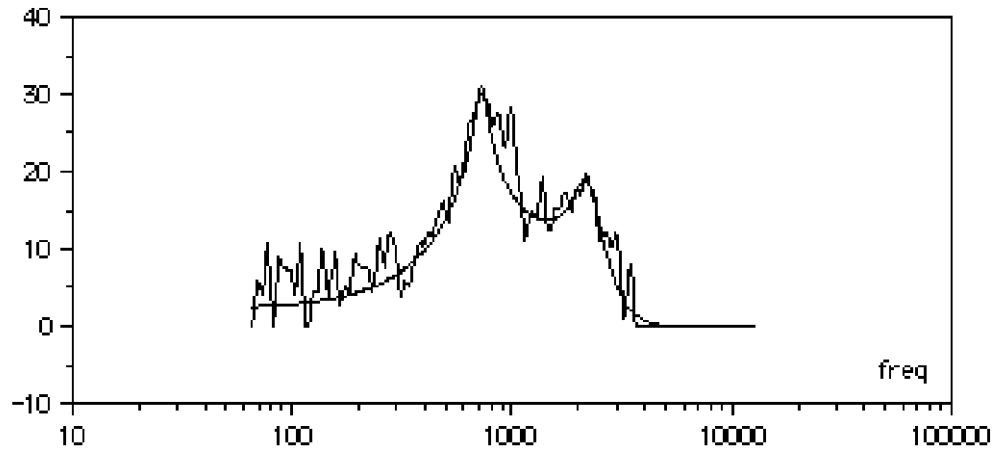
*Donnée brute extraite d'un violon courge. Fabrication de Jacques Dudon. L'archet était constamment à un rapport fixe environ 1/4 de la longueur de la corde.*



*Spectre reconstitué à partir des données précédentes. On note deux zones formantiques plus accentuées.*



*Première approximation avec la simplification apportée par la linéarisation du problème.*



*Calcul du filtre approchant avec l'erreur la plus faible l'enveloppe spectrale obtenue expérimentalement.*

### VI.4.c. Inversion du filtre résonant

Le filtre inverse de la résonance que l'on a déterminée pour une partie stationnaire peut être appliqué maintenant à l'ensemble du signal en tenant compte de l'attaque et de la chute. On suppose qu'au cours du jeu, la résonance ne se déforme pas. On supprime ainsi l'influence des formants pour ne s'intéresser qu'à l'aspect excitation. Le signal obtenu sera modélisé par la distorsion d'une sinusoïde à travers une fonction de transfert non linéaire.

### VI.4.d. modélisation du signal d'excitation par distorsion non-linéaire

On filtre par l'"anti-filtre" résonant le signal d'origine de manière à supprimer au maximum l'influence des formants, ceci afin de pouvoir modéliser l'excitation par distorsion non linéaire. Pour effectuer cette opération, on représente la variation du barycentre du signal "anti-filtré" au cours du temps (cela correspond à un paramètre psycho-acoustique lié au timbre de l'instrument) ainsi que l'énergie qui correspond à l'enveloppe du signal. On se donne une période du signal que l'on prélève au maximum des valeurs du barycentre. Une FFT sur cette période permet la construction par les polynômes de Chebyshev de deux tables de distorsion. A partir de ces deux tables, on reconstruit un signal d'excitation théorique en faisant varier un index de distorsion et on trace de la même manière le barycentre et l'énergie du signal. A partir de ces deux représentations, on peut construire une table d'index et une table de post-correction qui, placées dans l'instrument MUSICV décrit ci-dessous, permettront la resynthèse du signal étudié.

#### VI.4.d.1. moments spectraux

Si le spectre est échantillonné en un certain nombre de points de fréquence  $f_1, f_2, \dots, f_n$  d'amplitude  $A(f_i)$ , le moment spectral d'ordre  $n$  est défini par la relation:

$$M_n = \sum_{i=1}^{i=N} f_i^n A(f_i)$$

Si on considère une zone convenablement choisie du spectre définie par les positions  $i_1$  et  $i_2$  (indice dans le spectre), on peut calculer la fréquence du formant situé dans cette zone à l'aide de :



$$F = \frac{M_1}{M_0} = \frac{\sum_{i=i_1}^{i=i_2} f_i A(f_i)}{\sum_{i=i_1}^{i=i_2} A(f_i)}$$

Cette formule est équivalente au calcul du barycentre du spectre, qui sera utilisé plus loin dans la resynthèse par distorsion non linéaire. On peut pour chaque formant déterminer les limites  $f_1$  et  $f_2$  évitant au maximum les recouvrements de zones formantiques, en utilisant les moments d'ordre 2.

#### VI.4.d.2. barycentre du spectre

L'évolution du barycentre d'un son au cours du temps peut être un critère psychoacoustique de l'aspect plus ou moins "timbré" ou "brillant". A chaque instant, on peut représenter la décomposition spectrale d'un son en ses harmoniques  $h_i$ . On calcule le barycentre de la manière suivante:

$$B = \frac{\sum_{i=1}^N i \cdot h_i}{\sum_{i=1}^N h_i}$$

Il ne faut pas oublier que l'on travaille sur un signal auquel on a enlevé par filtrage inverse l'influence des formants. Il reste donc essentiellement l'excitation et du bruit, notamment dans l'attaque du son. Il ne faut pas perdre de vue également le but de la méthode qui consiste à approximer des sons naturels à l'aide du filtrage et de la distorsion non-linéaire. Le son réel se décompose en deux parties: la source et la résonance. La résonance précédemment modélisée n'intervient plus dans la suite des traitements. Elle sera réintroduite à la sortie du modèle de distorsion par filtrage pour restituer les formants. De plus, on se bornera dans cette étude à la création de sons harmoniques car les problèmes posés par la décomposition spectrale du son réel au cours du temps sont loin d'être résolus, surtout lors de l'attaque d'une note. Il apparaît alors des formes d'inharmonicité ou du bruit difficile à modéliser.

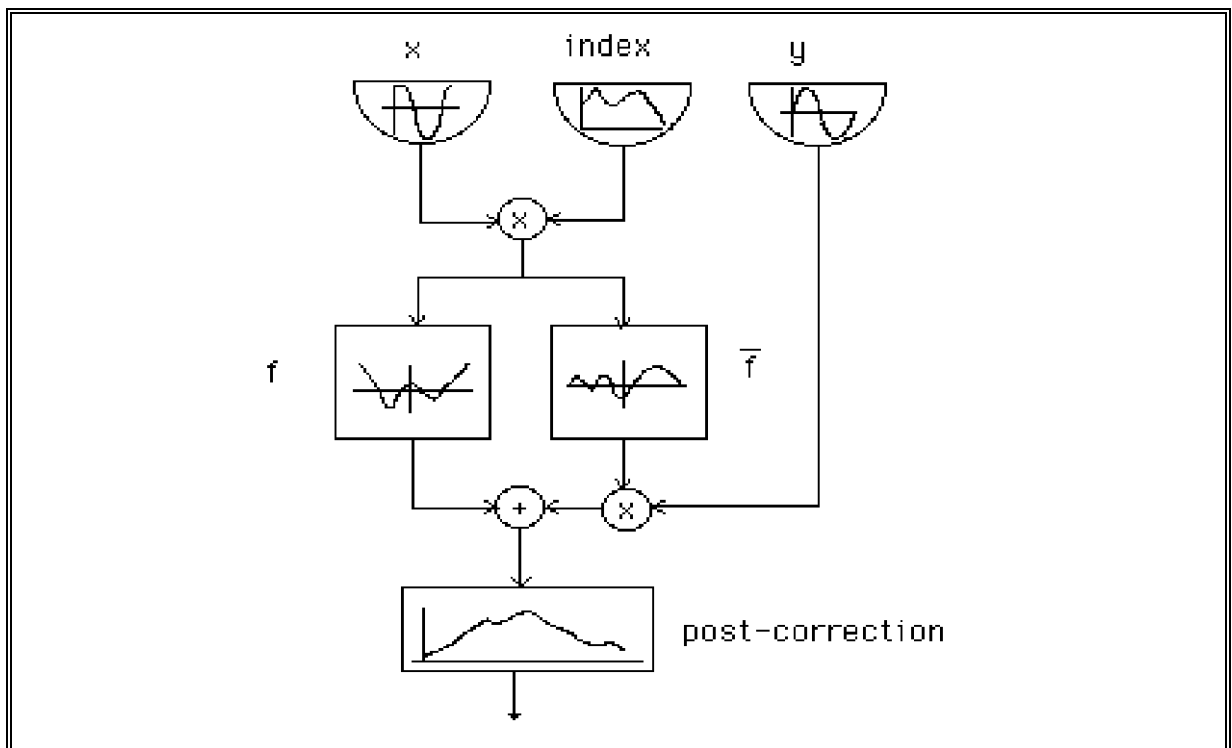
L'expérimentation a été faite principalement sur des sons filés ou chantés avec une recherche plus poussée des formants car c'est un paramètre important auquel s'accroche le système de reconnaissance auditif.

Pour une bonne synthèse après analyse, il est important de reproduire fidèlement les variations du fondamental d'une note. Cela a été tenté dans la partie stationnaire du signal

mais conduit à de moins bons résultats dans les parties transitoires. Pour y remédier, il faudrait coupler certaines parties de l'analyse avec des méthodes temps-fréquences et trouver un autre critère différent du barycentre pour mesurer la "grosueur" du son. La "grosueur" du son est le paramètre qui nous intéresse le plus pour restituer l'évolution des harmoniques au cours du temps.

Une conséquence directe de l'utilisation de la distorsion est de lier le timbre à l'amplitude du signal d'entrée. Cela entraîne une variation de l'amplitude du signal de sortie. Pour y remédier, on corrige le signal de sortie à l'aide d'une fonction de post-correction qui module en amplitude la sortie normale du son.

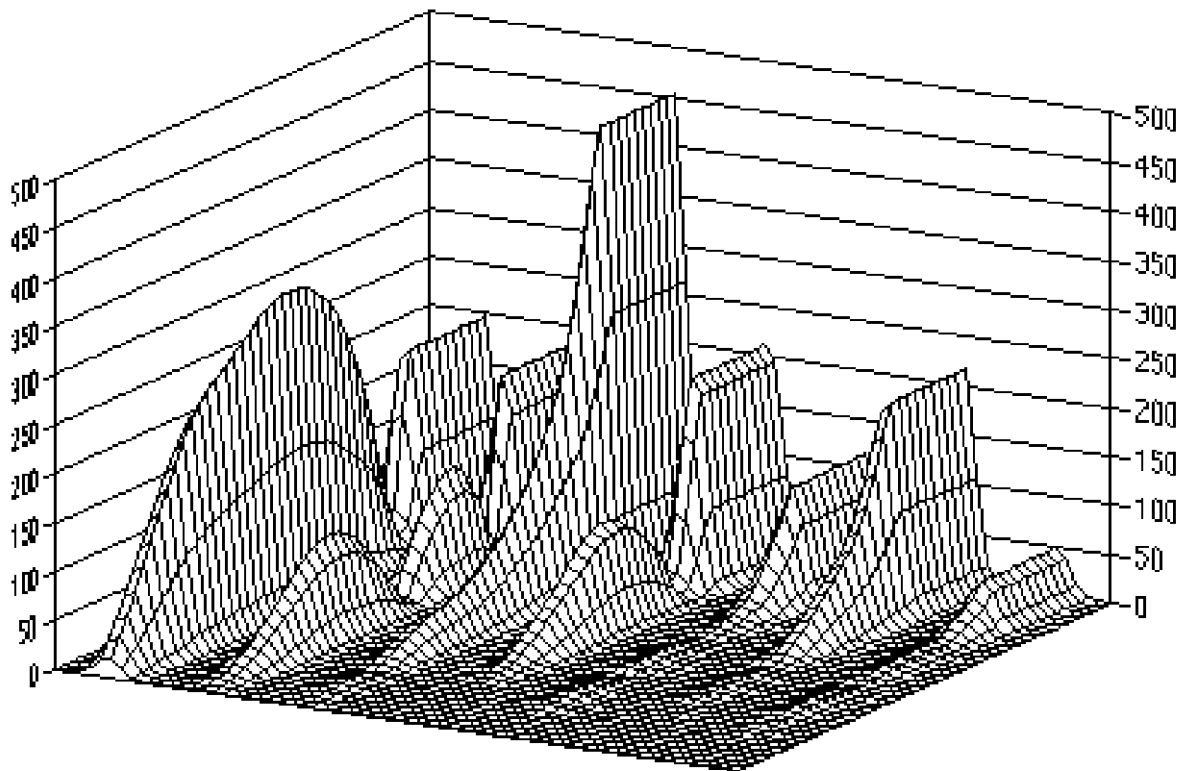
L'instrument MUSICV complet devient:



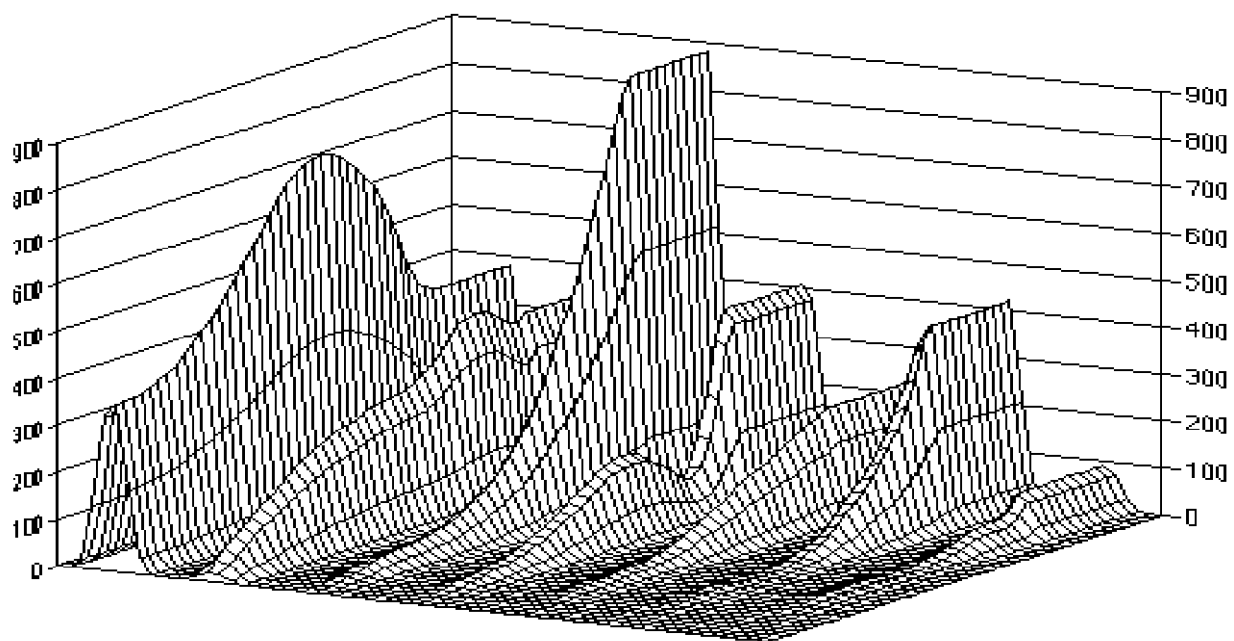
*Instrument MUSICV pour resynthèse par distorsion non-linéaire*

Dans la discussion préliminaire sur la distorsion [cf. §VI.2.b.3], lors de la définition des outils de synthèse, on a vu que le modèle par distorsion non linéaire était sensible, notamment dans l'évolution du barycentre en fonction de l'index, aux phases de chaque harmonique. Daniel Arfib a montré de manière empirique qu'une alternance des signes sur les harmoniques dans le calcul de la fonction de distorsion améliorerait sensiblement l'évolution du spectre et son barycentre. Les trois dessins ci-dessous ont été réalisés pour générer le même spectre pour un index égal à 1, mais suivant trois méthodes différentes. Tout d'abord en

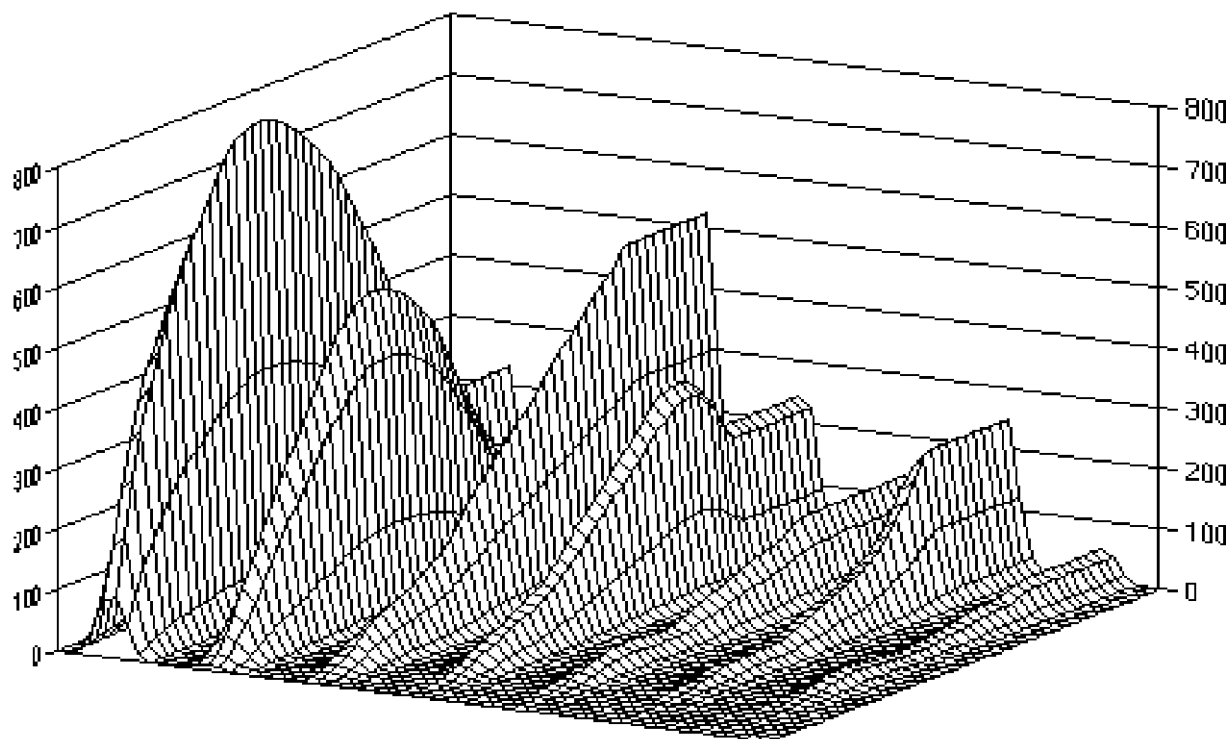
calculant la fonction de distorsion sans alternance des signes des harmoniques, puis avec alternance et enfin en tenant compte de la phase réelle de chaque harmonique. On constate dans l'évolution du spectre une régularité uniquement dans les deux derniers cas. Cela se traduit au niveau du barycentre par une fonction monotone continue, ce qui n'est pas le cas dans le premier exemple. Cette monotonie de la fonction barycentre est importante car on verra plus loin que nous serons amenés à déterminer la valeur de l'index qui donne un barycentre connu. Si la fonction n'est pas monotone on pourra avoir plusieurs valeurs possibles, sans possibilité de lever l'indétermination.



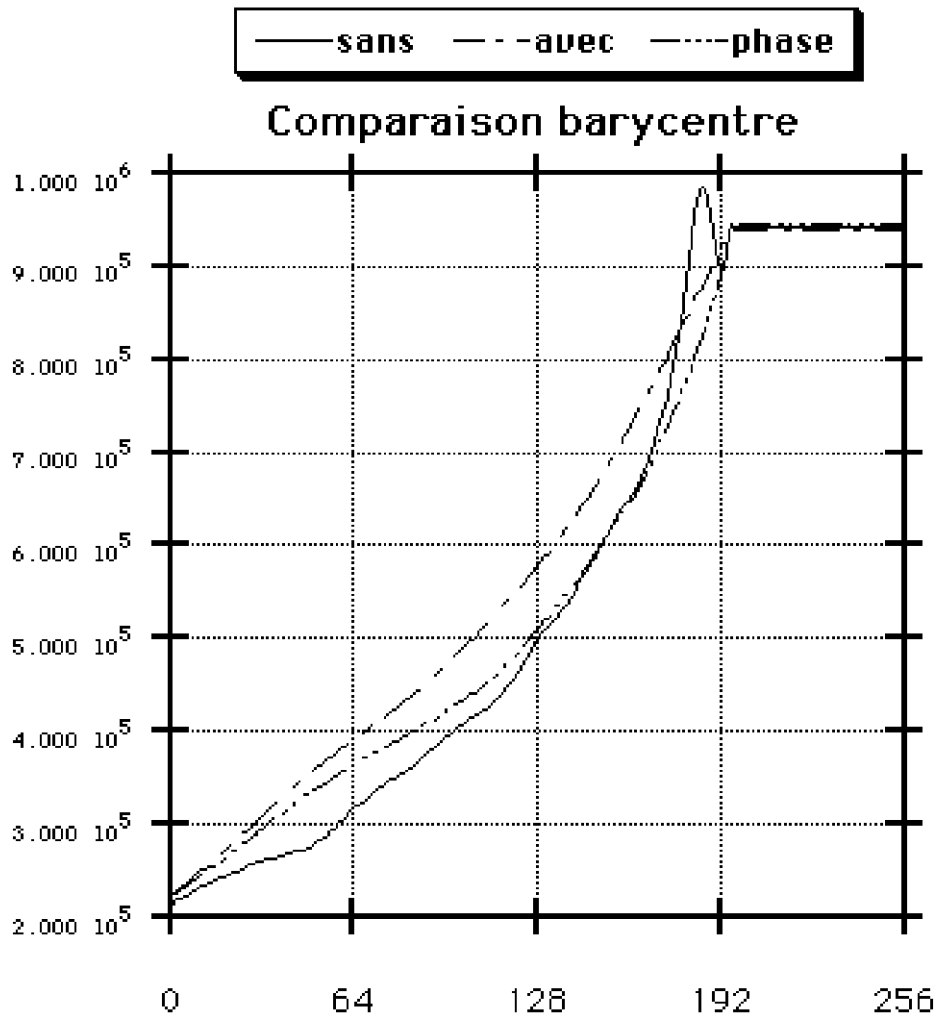
*Variation du spectre pour un index allant de 0 à 1. La fonction de distorsion a été calculée pour un index égal à 1 sur les 15 premiers harmoniques d'un "a" chanté sans alternance de phase.*



*même cas que précédemment, mais avec alternance ++-- sur les phases.*



*Avec conservation des phases du signal réel.*



*L'évolution du barycentre dans le premier cas n'est pas satisfaisante. Ne permet pas de trouver une valeur de l'index unique qui restitue un barycentre donné.*

Lorsque la valeur du barycentre ne permet pas de trouver une valeur unique de l'index on cherche la valeur de l'index assurant la meilleure continuité de l'évolution du barycentre.

#### VI.4.e. estimation de la fonction de distorsion et de l'index de distorsion

Nous allons voir comment déterminer les valeurs des tables de distorsion et de post-correction à partir d'une analyse de sons réels.

Etapes

1- A partir du signal complet (attaque, chute) on calcule deux tables.

a- La variation de l'énergie du signal au cours du temps

b- La variation du barycentre du spectre au cours du temps

On obtient ainsi une idée de l'enveloppe du son et de la brillance. L'enveloppe permet de vérifier la stationnarité du signal et le barycentre permet de situer les parties les plus riches en fréquences.

2- On extrait une période du signal dans la partie stationnaire qui semble la plus riche en harmoniques. Cela est indispensable car lors d'une synthèse par distorsion le spectre obtenu possède au maximum N harmoniques si le polynôme de Chebyshev est d'ordre N. On ne pourra donc pas dépasser le nombre d'harmoniques fixé par l'analyse de la période (pour un calcul avec un index égal à 1).

En réalité on peut très bien faire l'analyse pour un index qui ne soit pas maximal puis au moment de la synthèse, dépasser cette valeur. Cela entraîne une explosion du nombre d'harmoniques, mais le contrôle du spectre est moins rigoureux.

3- On fait une analyse de Fourier à partir de cette période de manière à déterminer réellement le nombre d'harmoniques significatifs que l'on voudra faire intervenir dans le calcul des fonctions de distorsion. On décidera que tous les harmoniques à partir d'un certain rang inférieurs de 60 dB par rapport à un harmonique maximal ne seront pas pris en compte. En général cette restriction réduit aux alentours de 30 les harmoniques significatifs.

4- On calcule par récurrence les deux fonctions de distorsion à partir des valeurs de la FFT en tenant compte des phases de chaque raie.

5- A l'aide de l'instrument MUSICV de la figure 20 on réalise le signal

$$s(t) = f(ax) + y \overline{f}(ax)$$

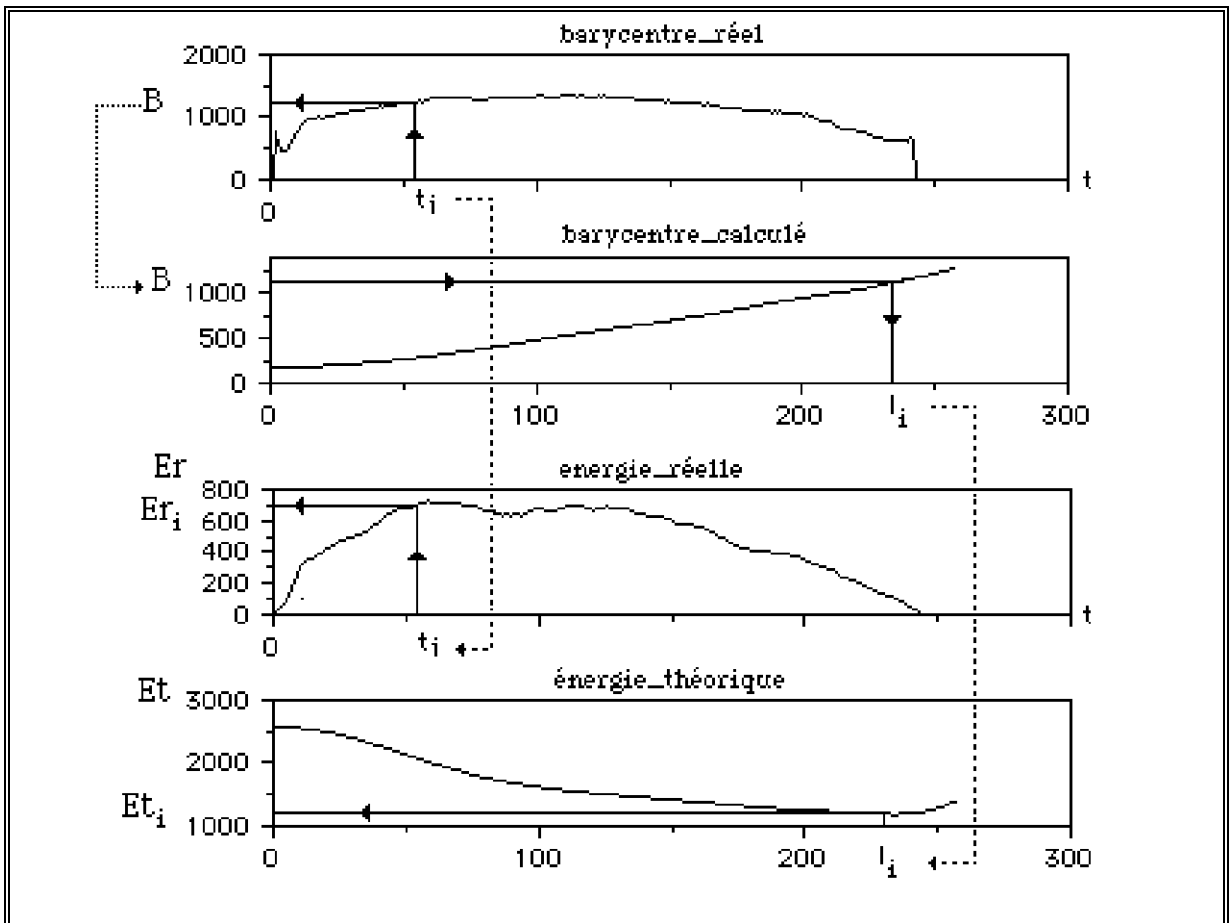
pour une durée égale à celle du signal réel en faisant varier l'index de distorsion entre 0 et 1.

6- On calcule deux nouvelles tables à partir de ce signal synthétique.

a- La variation de l'énergie du signal en fonction de l'index de distorsion.

b- La variation du barycentre du signal en fonction de l'index de distorsion.

La connaissance des quatre tables nous permet de calculer une table d'index et une table de post-correction de la manière suivante:



Pour un temps  $t_i$  on détermine une valeur du barycentre  $B$  dans la table 1.

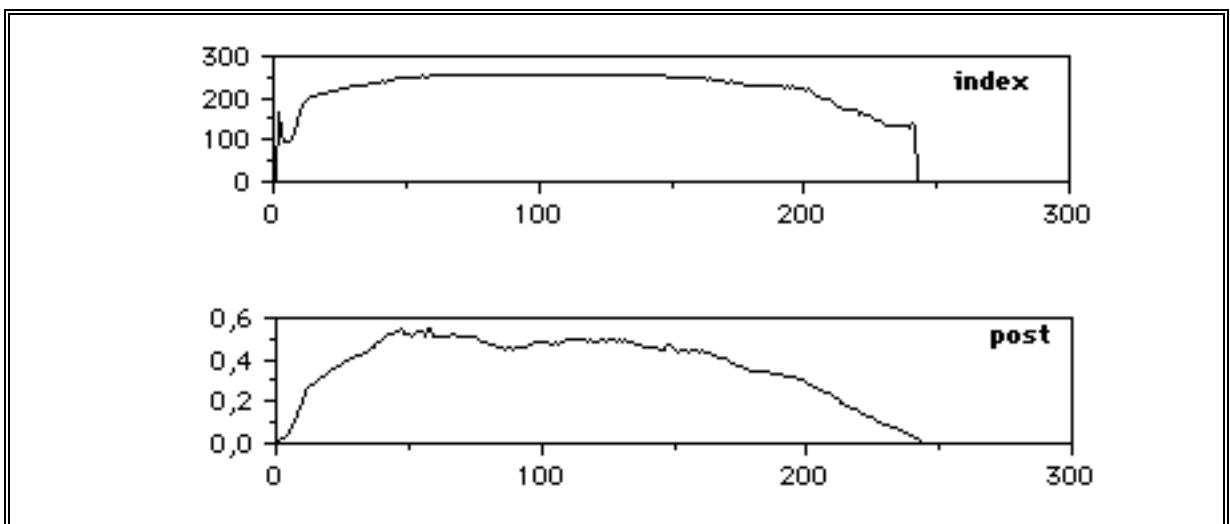
On porte cette valeur en ordonnée dans la table 2 et on cherche la valeur de l'index correspondant  $I_i$ . On porte cette valeur dans la table des variations de l'index au cours du temps.

$$I_i = \text{barycentre\_calculé}^{-1}[\text{barycentre\_réel}(t_i)]$$

$$\text{index}[t_i] = I_i$$

Pour une valeur  $t_i$  correspond une énergie réelle  $E_{r_i}$ . Pour la valeur de l'index  $I_i$  au temps  $t_i$  correspond une énergie théorique  $E_{t_i}$ . On calcule la valeur de la post-correction en  $t_i$  par:

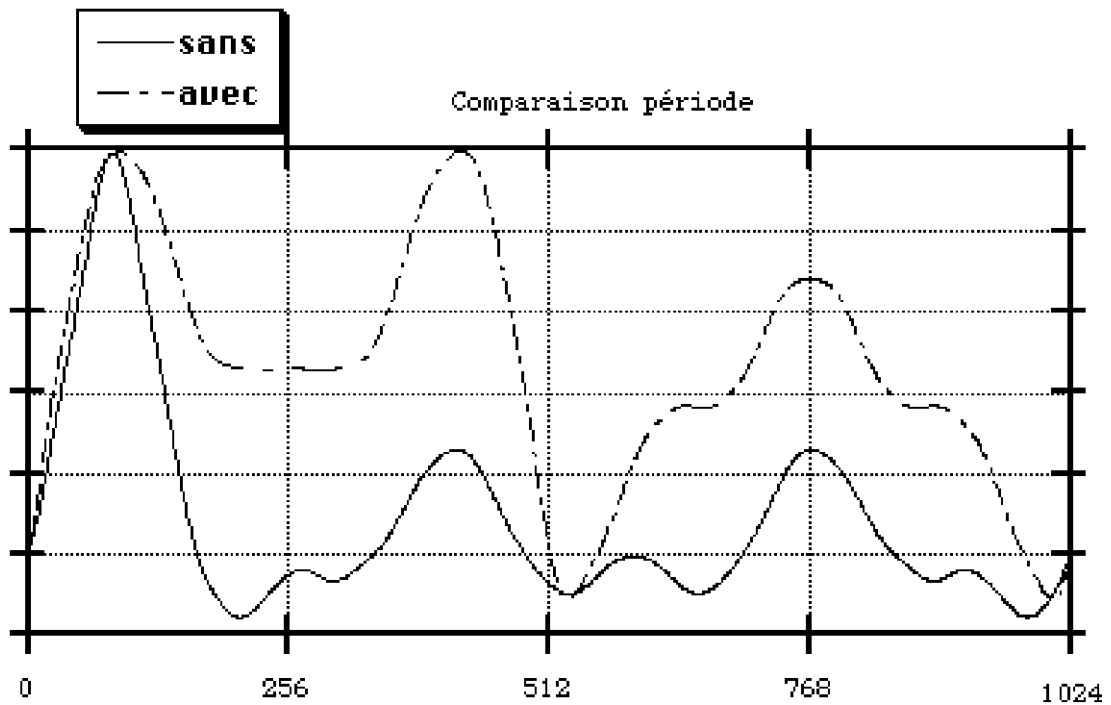
$$\text{Post\_correction}[t_i] = \frac{E_{r_i}}{E_{t_i}}$$



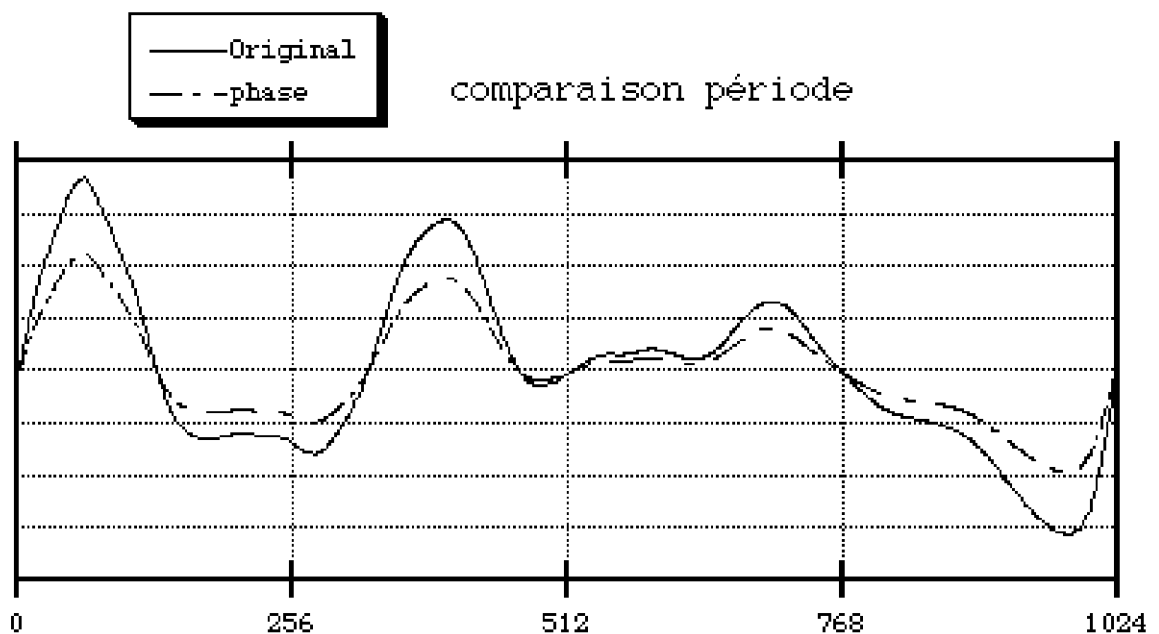
Certains problèmes peuvent se poser lors du calcul du barycentre du son théorique. La courbe obtenue peut ne pas être monotone. Dans ces conditions on peut avoir pour une valeur du barycentre plusieurs valeurs d'index possibles. Une solution consiste à lisser la courbe pour la rendre monotone ou bien choisir une autre période comportant plus d'harmoniques.

On compare pour l'exemple de la voyelle "a" chantée une période du signal de sortie modélisé par distorsion non linéaire. On remarque que la période la plus fidèle graphiquement est celle obtenue en conservant les phases. Cela ne présente d'intérêt que pour la forme du signal car l'oreille n'est pas sensible aux phases des harmoniques dans la reconstruction du signal. Par contre dans l'optique d'un musicien qui ne connaît pas la physique, il est agréable de constater que l'analyse d'un son réel, donne par resynthèse une forme d'onde qui se rapproche de l'original: Cela rassure et peut permet une classification des périodes.





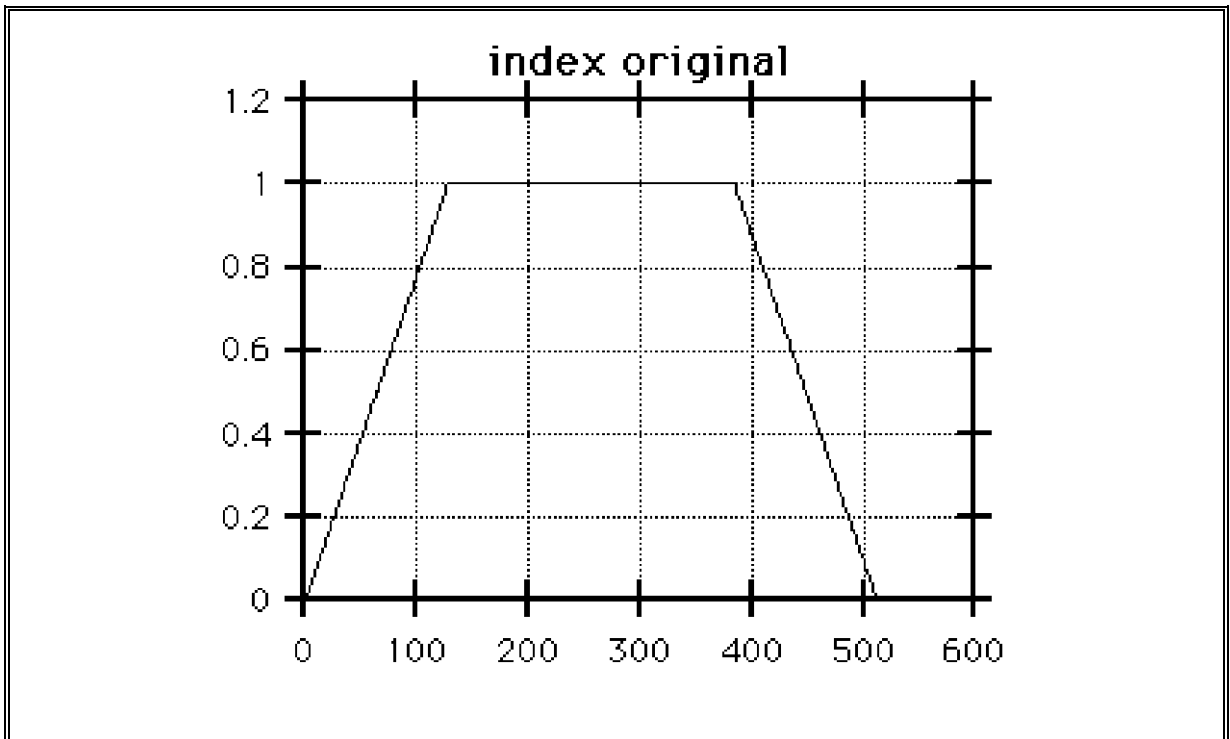
*Comparaison des périodes obtenues par l'utilisation dans le modèle de distorsion non linéaire d'une fonction construite à partir des harmoniques avec alternance des phases et sans alternance.*



*Comparaison de la période originale avec celle obtenue par distorsion non linéaire avec conservation des phases.*

Dans l'exemple MUSICV suivant on vérifie que l'on retrouve bien grâce à la méthode du barycentre décrite plus haut, une courbe d'index équivalente à celle du départ.

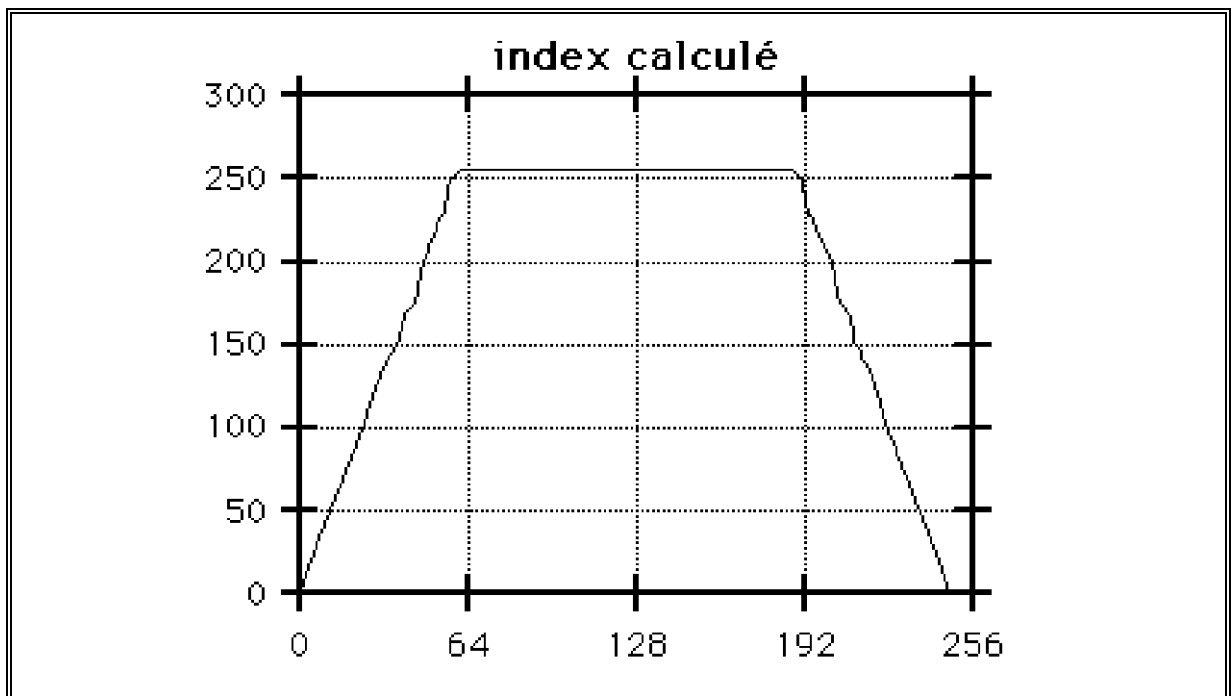
On construit un signal excitation comportant 15 harmoniques et dont l'index varie de la manière suivante:



```

SAM 25600;
INS 0 1;
CNV P6=HTZ(P6);
CNV P10=DUR(P4);
IOS P5 P10 B3 F1 P29 ; com enveloppe;
IOS P7 P10 B4 F2 P28 ; com index ;
IOS B4 P6 B4 F3 P27 ; com cosinus ;
DNL B3 B4 B5 F4 P26 ; com dist cos ;
DNL B3 B4 B6 F5 P25 ; com dist sin ;
IOS B6 P6 B6 F6 P24 ; com sinus ;
AD2 B5 B6 B5 ;
OUT B6 ;
END;
GEN 0 1 1 1000 0 0 1 125 0.7 250 0.7 750 0 1000;
GEN 0 1 2 1000 0 0 1 250 1 750 0 1000;
GEN 0 2 3 1000 0 1 0 ;
GEN 0 18 4 1000 0.0
  0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1;
com h*cos(a) l=index max;
GEN 0 19 5 1000 0.0
  0.999995 0.499667 0.332735 0.249193 0.198968 0.165444
  0.141404 0.123246 0.109079 0.097680 0.088269 0.080308
  0.073473 0.067354 0.061543 1 ;
com h*sin(a) l=index max ;
GEN 0 2 6 1000 1 1;
NOT 0 1 1 5000 110.000000 1;
TER 1.1 ;

```



La fonction index n'est pas normalisée, mais l'allure générale correspond exactement à la fonction initiale.

## VI.5. applications

### VI.5.a. modélisation des familles d'instruments acoustiques: lutherie informatique

De manière générale un instrument de musique est une machine à fabriquer des sons composée d'un système "excitateur" et d'un système "résonateur". Le système "résonateur" est constitué d'une ou plusieurs résonances qui jouent le rôle d'un filtre sur le système "excitateur". Ce dernier est contrôlé par le musicien lors du jeu (dynamique). On distingue plusieurs types d'instruments:

- les instruments à cordes

Caisse de résonance généralement fixe. Différents systèmes d'excitation (cordes frottées, frappées, pincées).

- les instruments à vent

Caisse de résonance évolutive (trombone) ou faiblement évolutive (saxophone, clarinette, flûte). Pour ces derniers, l'influence prépondérante du pavillon font qu'en première approximation ils peuvent être considérés comme instrument à formant fixe, ou simplement modélisés par un système excitateur. (pas de formant marqué).

- les instruments à percussion

Les modes de résonance sont essentiels (généralement inharmoniques): l'excitation n'est généralement pas un signal périodique mais du type impulsif.

La voix étant également considérée comme un instrument, on ne s'intéressera qu'à la voix chantée. Les formants sont produits par les résonances du conduit vocal (différentes pour chaque voyelle), l'excitation est produite par les vibrations des cordes vocales (signal périodique).

Dans la suite du travail on ne s'intéressera qu'aux instruments ayant des résonances fixes: c'est le cas des instruments à cordes ou de la voix. Pour la voix on parle de formant mais par analogie on utilisera le même mot pour les instruments de musique.

MUSICV permet de réaliser des instruments par assemblage de modules de base. L'élément de base le plus employé est l'oscillateur. L'oscillateur est programmé pour engendrer des formes d'ondes particulières préalablement stockées dans des tables. La lecture plus ou moins rapide de ces tables permet de faire varier la fréquence du signal. D'autres

composants de base comme l'additionneur, le multiplieur ( etc..) sont associés aux oscillateurs afin de construire des blocs de plus en plus complexes en reliant les sorties aux entrées d'autres modules. Dans le cas d'un instrument composé d'un système excitateur et d'une résonance, on peut construire en premier lieu l'excitation par une méthode choisie, puis connecter la sortie du modèle à l'entrée d'un modèle de résonance. MUSICV offre un grand choix de modules de filtrage qui peuvent par adjonction constituer le filtre complexe qui simulera au mieux l'aspect résonant de l'instrument. MUSICV se distingue des autres moyens de synthèse (notamment électroniques) par son aspect modulaire et par le fait que son utilisation n'est pas liée à un modèle de synthèse donné. On peut intervenir sur tous les paramètres de l'instrument et cela très facilement grâce à de nombreuses fonctions supplémentaires.

La fonction que j'ai le plus utilisée est la création d'une forme d'onde à l'intérieur d'un oscillateur par la seule connaissance de son spectre. Cela permet d'engendrer une excitation dont le spectre est connu. Cette excitation servira de signal d'entrée du modèle de résonance dont on pourra étudier les caractéristiques du signal de sortie.

L'exemple suivant montre un instrument constitué d'une excitation dont le spectre est donné par la fonction GEN (spectre en 1/n) et d'une résonance constituée par un filtre dont on connaît la fréquence d'accord et la largeur de bande à -3 dB (fonction FLT).

```

COM FLT1.PAR;
SAM 25600;
SV3 0 10 1.98549 0.99510 1.89054 0.97576
1.08420 0.95210;
INS 0 1;
CNV P15=DUR(P4);
CNV P6=HTZ(P6);
IOS P5 P15 B3 F2 P30;
IOS B3 P6 B3 F1 P29;
FLT B3 B4 V10 V11 P28 P27;
FLT B4 B5 V12 V13 P26 P25;
FLT B5 B6 V14 V15 P24 P23;
OUT B6;
END;
COM -----;
GEN 0 2 1 1000 1.00 0.50 0.3333 0.25 0.2
0.1667 0.1429 0.125
0.1111 0.1 0.0909 0.0833 0.0769 0.0714
0.0667 15;
GEN 0 1 2 1000 0 0 1 150 1 750 0 1000;
COM -----;
NOT 0.00 1 1 10 110 ;
TER 1;

```

*exemple académique*

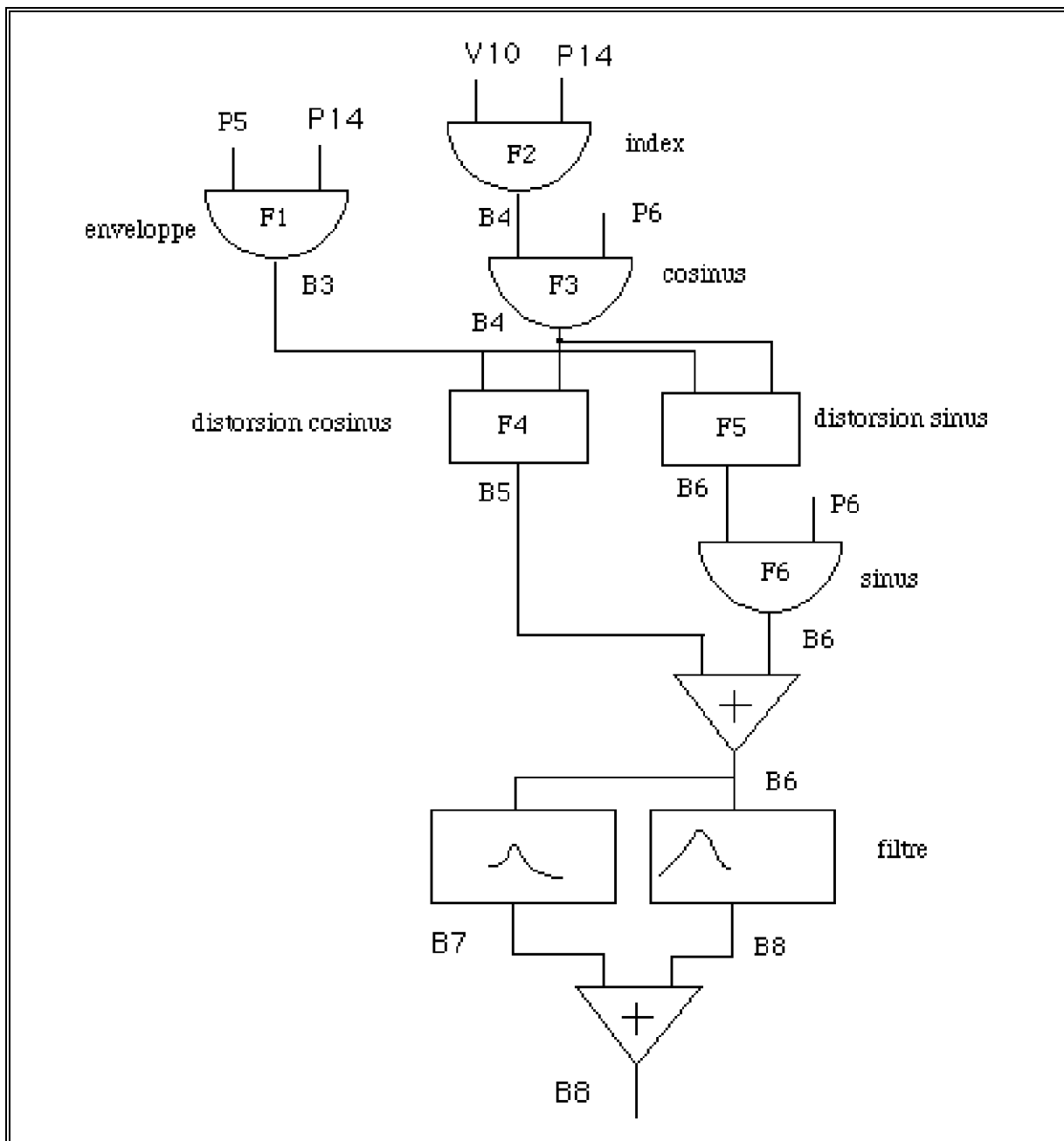
## **VI.5.b.Expérimentations**

### **VI.5.c. construction d'un exemple académique et recherche des paramètres**

Le modèle académique complet réalisé à partir d'un instrument MUSICV schématisé ci-dessous, est constitué d'un signal excitation modélisé par distorsion non linéaire et d'une résonance créée par deux filtres dont les fréquences d'accords sont connues. La méthode d'extraction de paramètres, appliquée sur le signal de sortie provenant du jeu d'une gamme chromatique grâce à cet instrument, doit permettre de retrouver les paramètres de l'excitation et de la résonance, connus par construction. Nous allons suivre pas à pas les différentes étapes pour réaliser l'extraction des paramètres. Le schéma ci-dessous représente l'instrument MUSICV qui a été utilisé.

La fonction F1 décide de la variation de l'amplitude du signal de sortie ou enveloppe du signal. Cette enveloppe a la forme traditionnelle, c'est à dire qu'elle possède une attaque, une décroissance, un plateau et une chute.

La fonction F2 représente la variation de l'index de distorsion au cours du temps. Au milieu du plateau, l'index aura la valeur 1. Cette condition permet de créer à l'aide des fonctions de distorsion F4 et F5 un signal dont le spectre sera connu (pour un index égal à 1) dans la partie stationnaire de la note. Les harmoniques donnés pour un index égal à 1 suivent une loi en  $1/n$  ( $n$  étant le rang de l'harmonique). C'est dans cette partie périodique ou pseudo-périodique que l'on extraira les périodes afin de modéliser la résonance. A la sortie de l'additionneur, nous avons constitué le signal excitation. Celui-ci sera filtré par les deux filtres mis en parallèle. Les fréquences de résonances des filtres sont respectivement 800 Hz et 1600 Hz pour un gain de 10.



### *Instrument académique*

On associe l'excitation et le filtrage dans un seul exemple MUSICV. L'instrument ainsi réalisé permet de jouer une gamme chromatique dans des conditions idéales. Au cours du jeu on réalise une translation logarithmique en fréquence de l'excitation: il n'y a donc pas variation. Par contre du fait de l'introduction de la distorsion (variation de l'index) le spectre de l'excitation n'est pas invariant au cours du jeu d'une note. Il se trouve comme souhaité au milieu de la note en correspondance avec la zone stationnaire: zone où se feront tous les prélèvements de période.

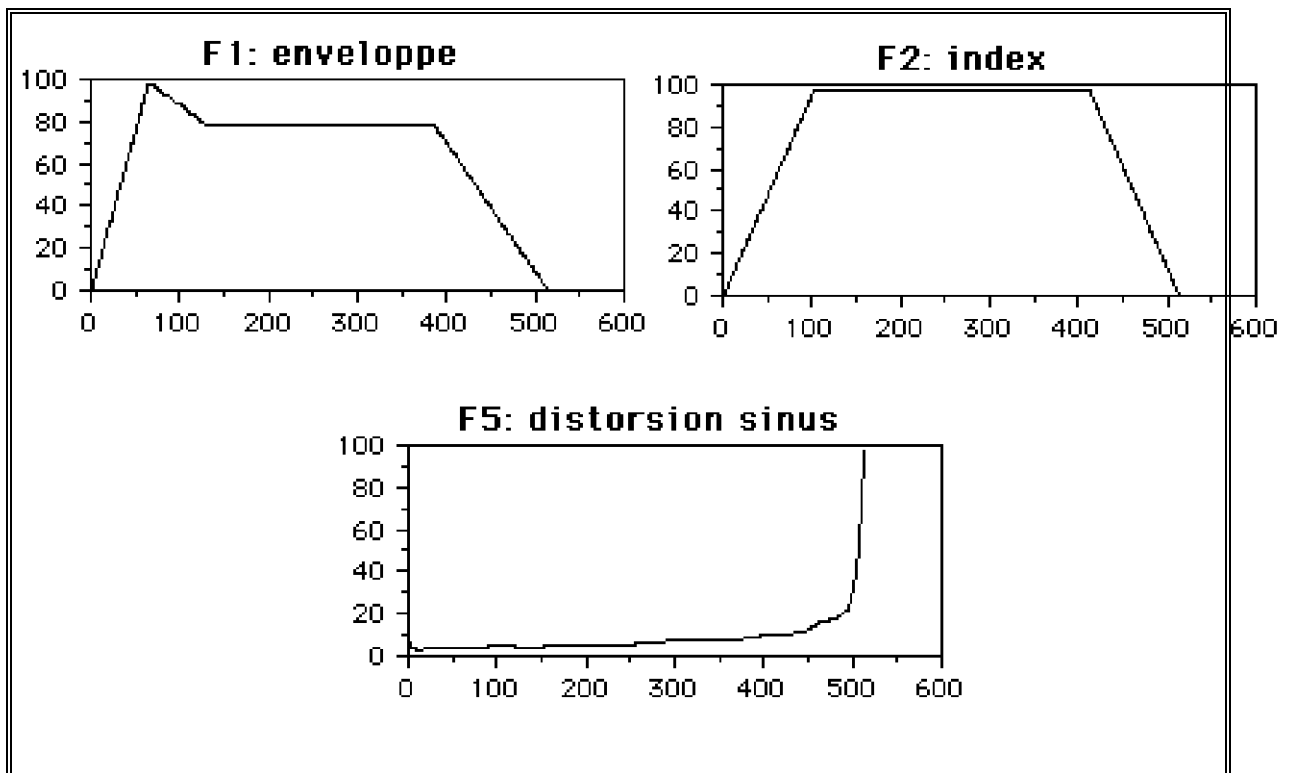
On crée une partition musicale à partir de 2 octaves de la gamme chromatique, soit 24 notes, la première étant le DO de l'octave contenant le LA2 (110 Hz). La partition MUSICV est la suivante.

```

SAM 25600;
SV3 0 10 1;
INS 0 1;
CNV P6=HTZ(P6);
CNV P7=HTZ(P7);
CNV P9=HTZ(P9);
CNV P14=DUR(P4);
IOS P5 P14 B3 F1 P29 ; com enveloppe;
IOS V10 P14 B4 F2 P28 ; com index ;
IOS B4 P6 B4 F3 P27 ; com cosinus ;
DNL B3 B4 B5 F4 P26 ; com dist cos ;
DNL B3 B4 B6 F5 P25 ; com dist sin ;
IOS B6 P6 B6 F6 P24 ; com sinus ;
AD2 B5 B6 B6 ;
SBF B6 B7 P7 P8 P23 P22;
SBF B6 B8 P9 P10 P21 P20;
AD2 B7 B8 B8;
OUT B8 ;
END;
GEN 0 1 1 1000 0 0 1 125 0.8 250 0.8 750 0 1000;
GEN 0 1 2 1000 0 0 1 200 1 800 0 1000;
GEN 0 2 3 1000 0 1 0 ;
GEN 0 18 4 1000 0.0
  0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1;
com h*cos(a) l=index max;
GEN 0 19 5 1000 0.0
  1.0000 0.5000 0.3333 0.2500 0.2000 0.1667 0.1429 0.1250
  0.1111 0.1000 0.0909 0.0833 0.0769 0.0714 0.0667 1;
com h*sin(a) l=index max ;
GEN 0 2 6 1000 1 1;
NOT 0.00 1 0.50 5000 65.406391 800 10 1600 10 ;
NOT 0.50 1 0.50 5000 69.295658 800 10 1600 10 ;
NOT 1.00 1 0.50 5000 73.416192 800 10 1600 10 ;
NOT 1.50 1 0.50 5000 77.781746 800 10 1600 10 ;
NOT 2.00 1 0.50 5000 82.406889 800 10 1600 10 ;
NOT 2.50 1 0.50 5000 87.307058 800 10 1600 10 ;
NOT 3.00 1 0.50 5000 92.498606 800 10 1600 10 ;
NOT 3.50 1 0.50 5000 97.998859 800 10 1600 10 ;
NOT 4.00 1 0.50 5000 103.826174 800 10 1600 10 ;
NOT 4.50 1 0.50 5000 110.000000 800 10 1600 10 ;
NOT 5.00 1 0.50 5000 116.540940 800 10 1600 10 ;
NOT 5.50 1 0.50 5000 123.470825 800 10 1600 10 ;
NOT 6.00 1 0.50 5000 130.812783 800 10 1600 10 ;
NOT 6.50 1 0.50 5000 138.591315 800 10 1600 10 ;
NOT 7.00 1 0.50 5000 146.832384 800 10 1600 10 ;
NOT 7.50 1 0.50 5000 155.563492 800 10 1600 10 ;
NOT 8.00 1 0.50 5000 164.813778 800 10 1600 10 ;
NOT 8.50 1 0.50 5000 174.614116 800 10 1600 10 ;
NOT 9.00 1 0.50 5000 184.997211 800 10 1600 10 ;
NOT 9.50 1 0.50 5000 195.997718 800 10 1600 10 ;
NOT 10.00 1 0.50 5000 207.652349 800 10 1600 10 ;
NOT 10.50 1 0.50 5000 220.000000 800 10 1600 10 ;
NOT 11.00 1 0.50 5000 233.081881 800 10 1600 10 ;
NOT 11.50 1 0.50 5000 246.941651 800 10 1600 10 ;
TER 12.00 ;

```





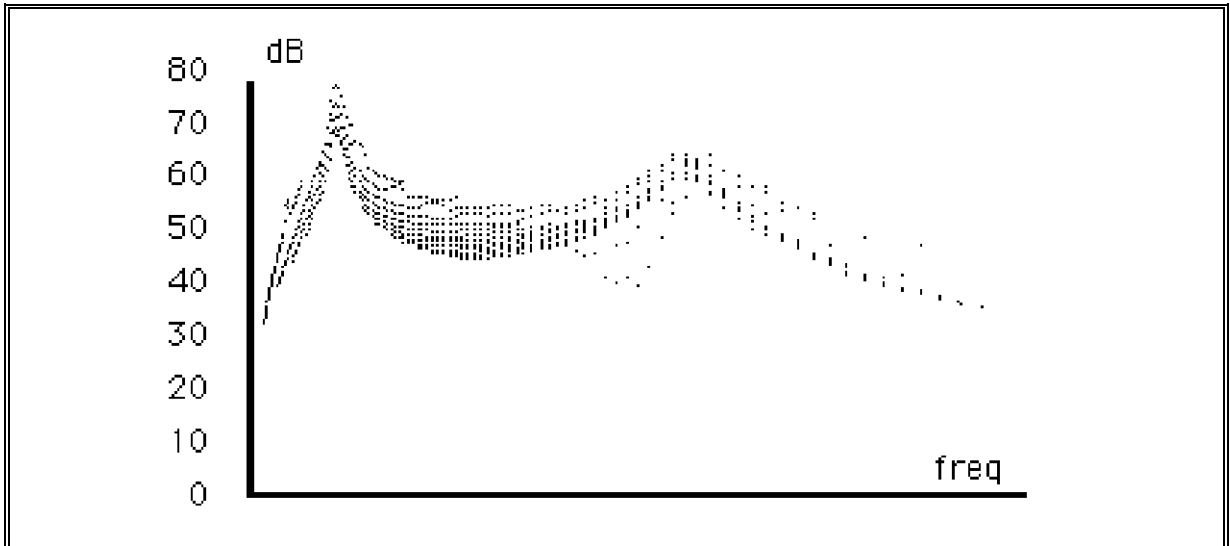
*Les différentes fonctions utilisées par les oscillateurs*

L'extraction des périodes se fait normalement à vue. Dans ce cas il est facile de se servir d'un passage par zéro pour repérer le début et la fin de la période. Dans le cas présent l'extraction se fait automatiquement, pour des raisons de commodité. La connaissance au préalable de la fréquence de la note permet de fixer le nombre de points de la période, et donc de rechercher un passage par zéro au alentour de cette valeur. L'extraction des périodes se fait visuellement à partir des signaux échantillonnés. Le fait de travailler sur des sons filés permet une bonne visualisation de la périodicité du signal. Si la fréquence devient trop importante (plus de 10 périodes visibles à l'écran) on peut sur-échantillonner le signal pour une meilleure visibilité de la période.

Chaque période est sauvegardée dans un fichier après interpolation. L'interpolation se fait sur 1024 points. Pour un même instrument on conserve le nom et on ajoute le numéro de la période. Le fichier contient en entête une estimation de la fréquence calculée à partir du nombre de points sélectionné et de la fréquence d'échantillonnage.

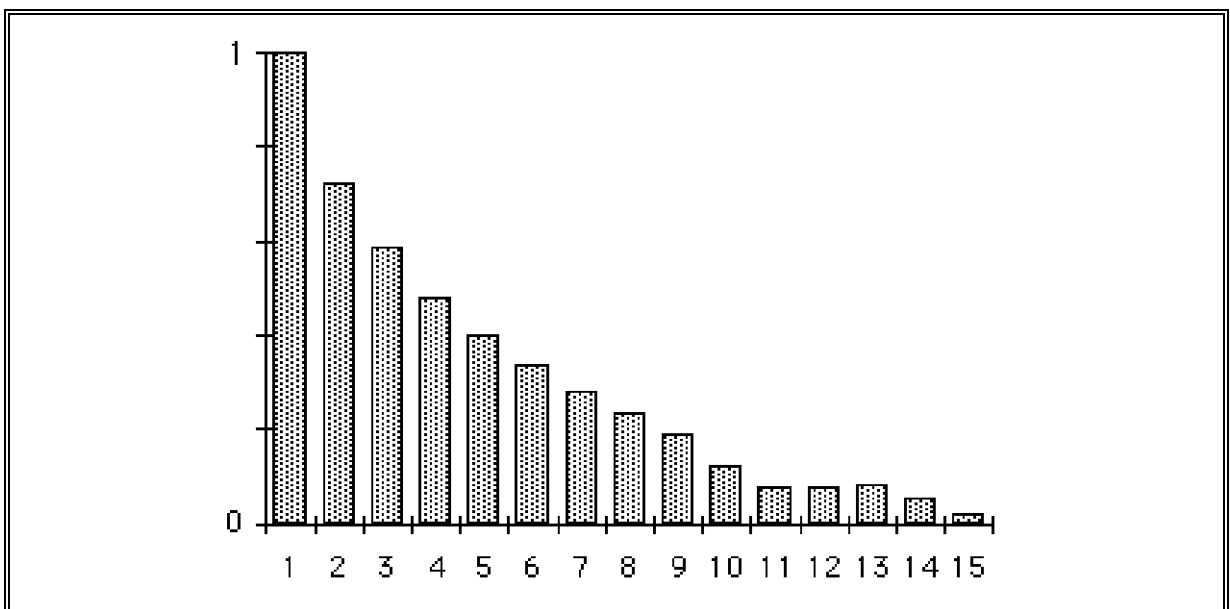
Cette extraction des périodes peut se faire automatiquement à partir du fichier provenant de l'enregistrement. On se place automatiquement au milieu de la plage de 0.5 seconde allouée à chaque note, le programme détermine par autocorrélation la longueur de la période, et comme précédemment la note est stockée dans un fichier.

On calcule ensuite autant de FFTs qu'il y a de périodes. La figure suivante montre la trajectoire de chaque harmonique, et l'on observe le parallélisme dû aux harmoniques de même rang.

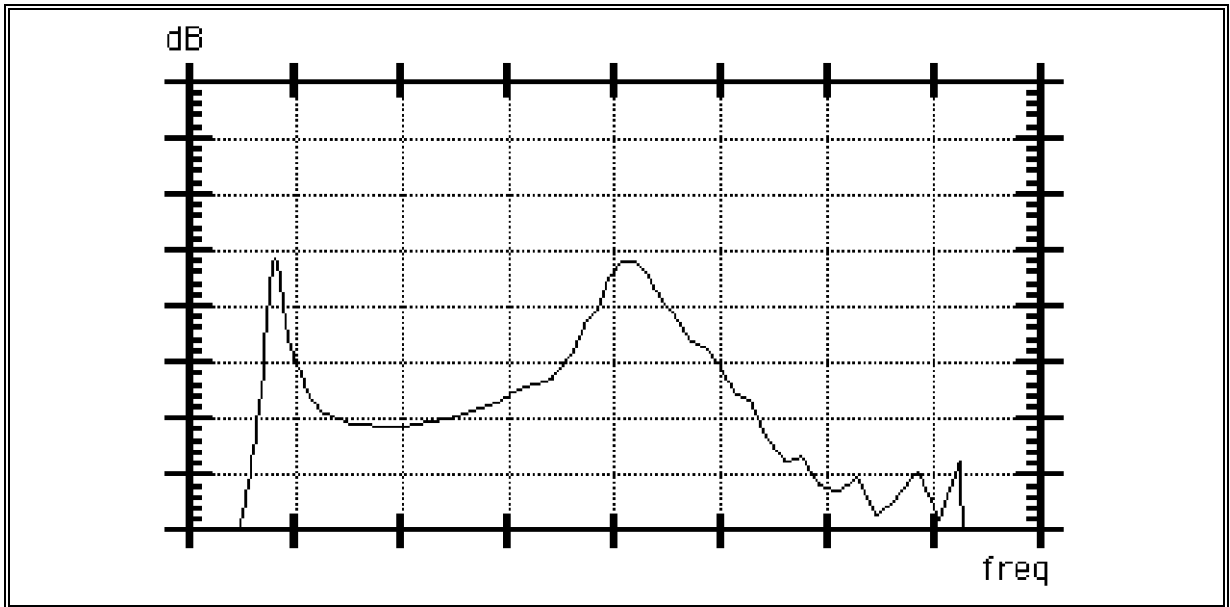


*Visualisation des résultats expérimentaux obtenus à l'aide de l'exemple académique*

En remontant chaque courbe sur la précédente comme expliqué précédemment [cf § VI.4.a.2] on reconstitue une enveloppe spectrale, les écarts entre les courbes étant représentatifs du spectre de l'excitation. Les figures suivantes montrent l'allure de l'excitation et du spectre de la résonance reconstituée.

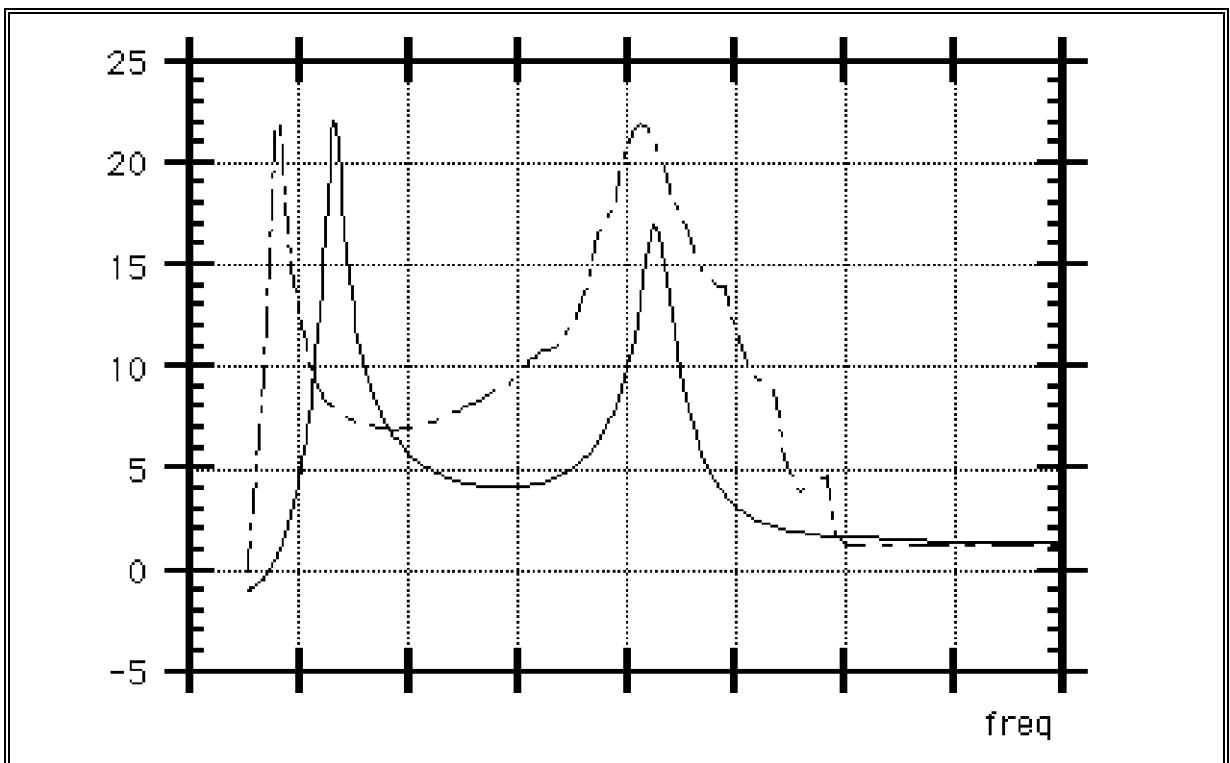


*Excitation reconstitué. On reconnaît une décroissance des harmoniques en  $1/n$*

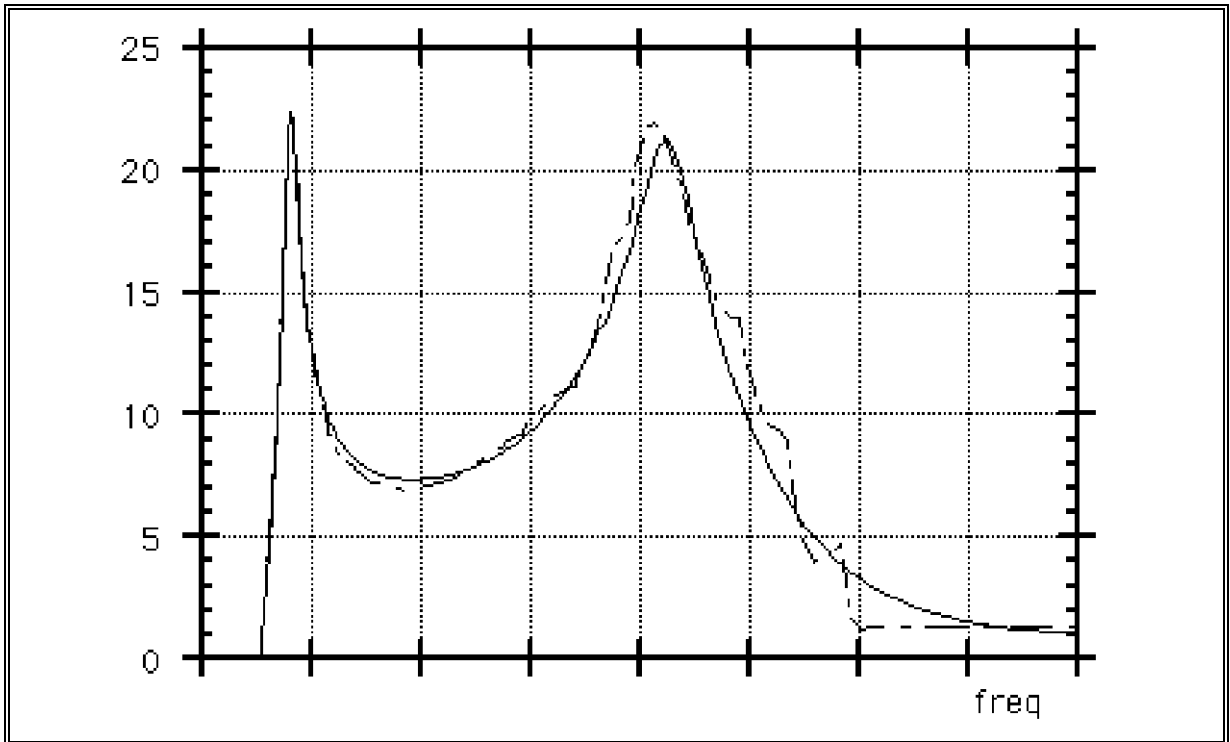


*figure 30 - Spectre de résonance reconstitué*

On recherche maintenant les coefficients du filtre dont l'enveloppe spectrale approche au mieux cette courbe expérimentale (fig 30). La méthode de Marquardt conduit au résultat suivant:

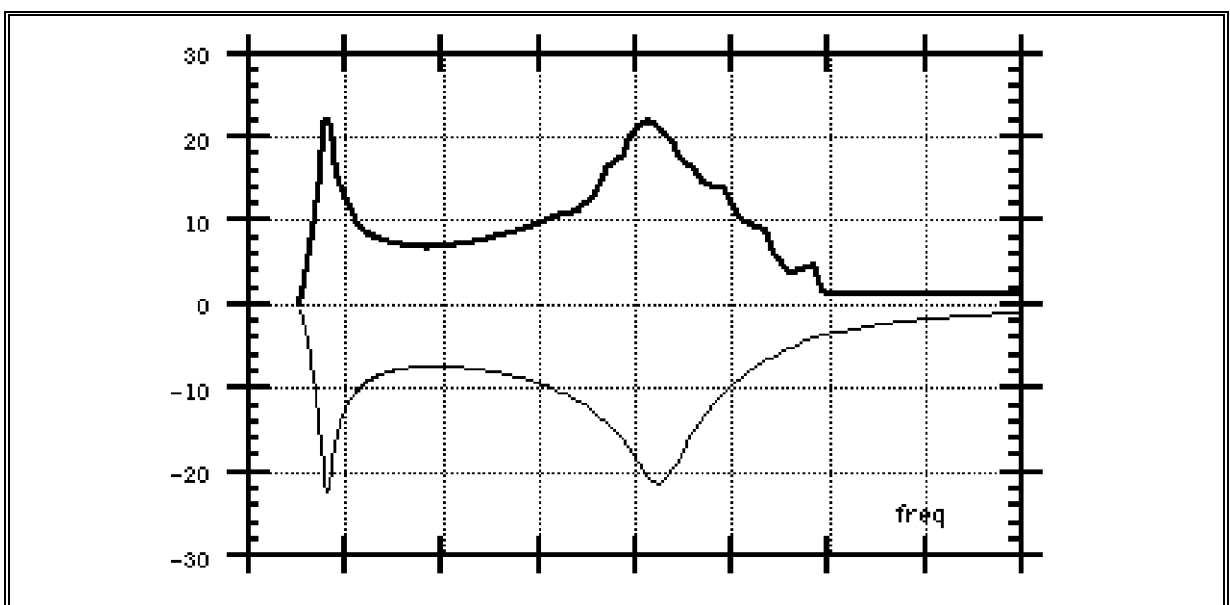


*Première approximation, après simplification en système d'équations linéaires*

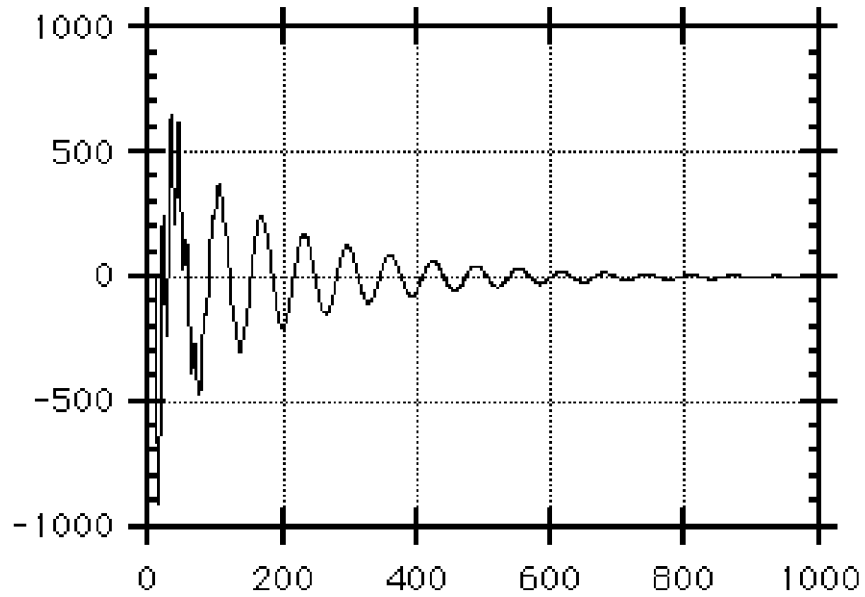


*Après linéarisation, on utilise une méthode du gradient pour affiner la modélisation et l'estimation des paramètres.*

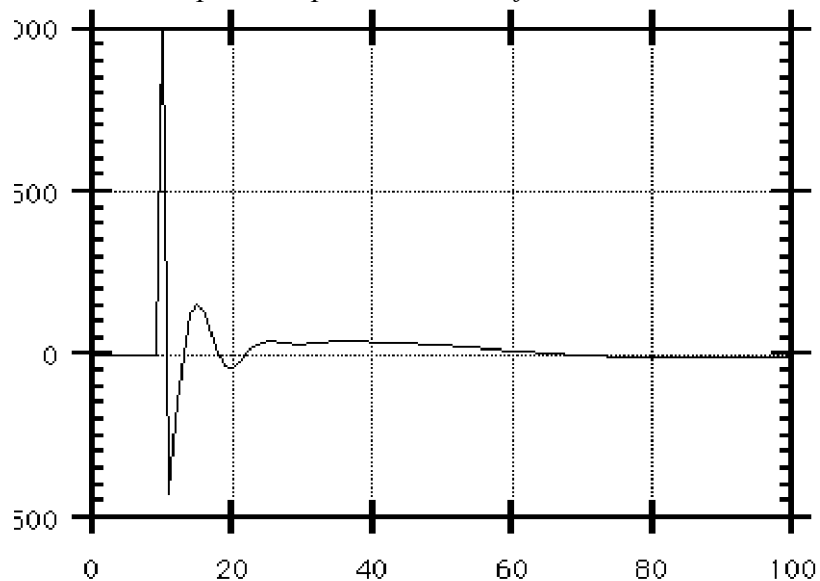
L'étape suivante consiste à faire la même chose pour le filtre inverse. Une partie des algorithmes pour trouver les coefficients du filtre inverse sont repris à partir du cas du filtre normal, mais en inversant numérateur et dénominateur. Le résultat est le suivant:



*calcul du filtre inverse.*

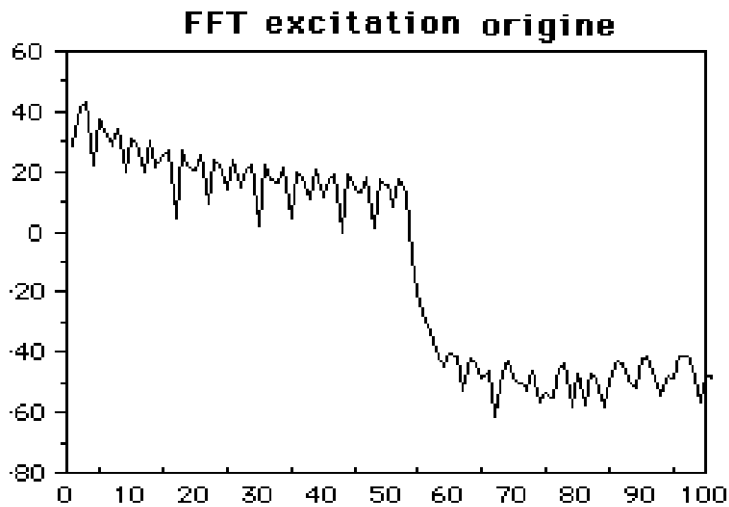


*réponse impulsionnelle du filtre résonant.*

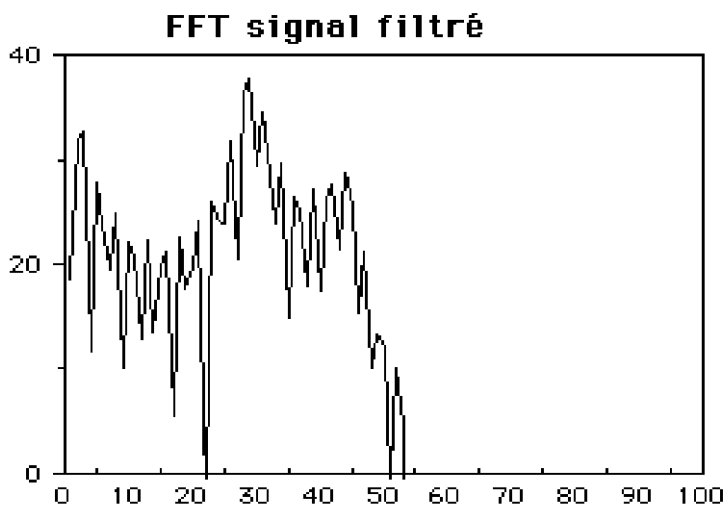


*réponse impulsionnelle du filtre résonant inverse.*

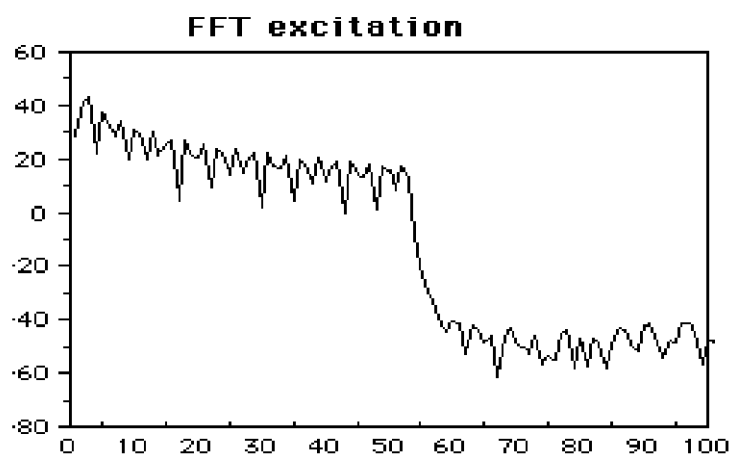
La stabilité des deux filtres a été observée par la position des pôles à l'intérieur du cercle unité. Pour vérifier les filtres, on construit un signal excitation identique au précédent: on filtre avec le filtre amplificateur puis on anti filtre avec le filtre inverse. On doit retrouver un signal dont le spectre est égal au spectre du signal d'origine.



*FFT d'un signal excitation composé d'harmoniques qui suivent une loi en  $1/n$ .*

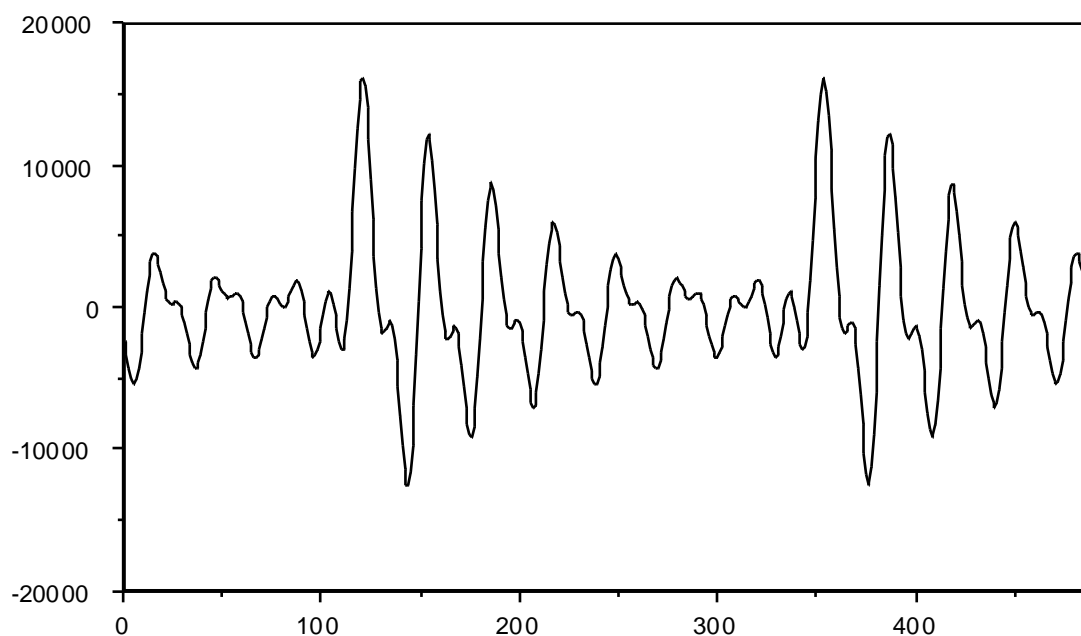


*Après filtrage par 3 filtres du second ordre mis en série de façon à constituer des formants identifiables.*

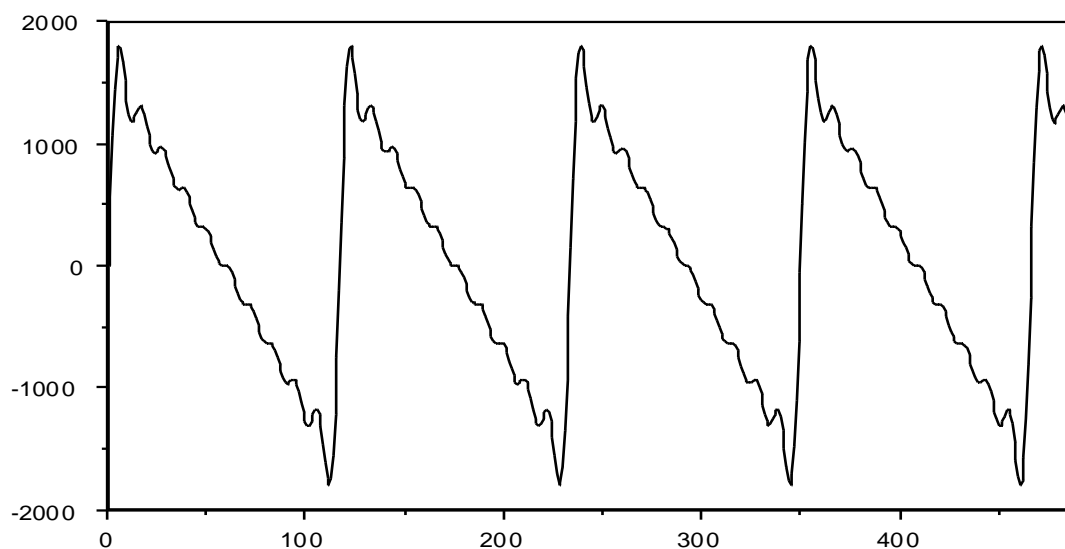


*Puis antifiltrage par le filtre inverse. On retrouve l'allure du spectre de départ.*

A ce point de l'expérimentation, on connaît parfaitement la résonance, et on est capable d'anti-filtrer le signal pour atténuer l'influence des formants. Ceci afin de passer à l'étape suivante qui consiste à modéliser l'excitation par distorsion non linéaire. On commence par appliquer le filtre inverse sur une note, depuis le début de la note jusqu'à la fin. Pour cela on prend par exemple le LA2 (110 Hz): la figure ci-dessous montre l'allure du signal anti-filtré.

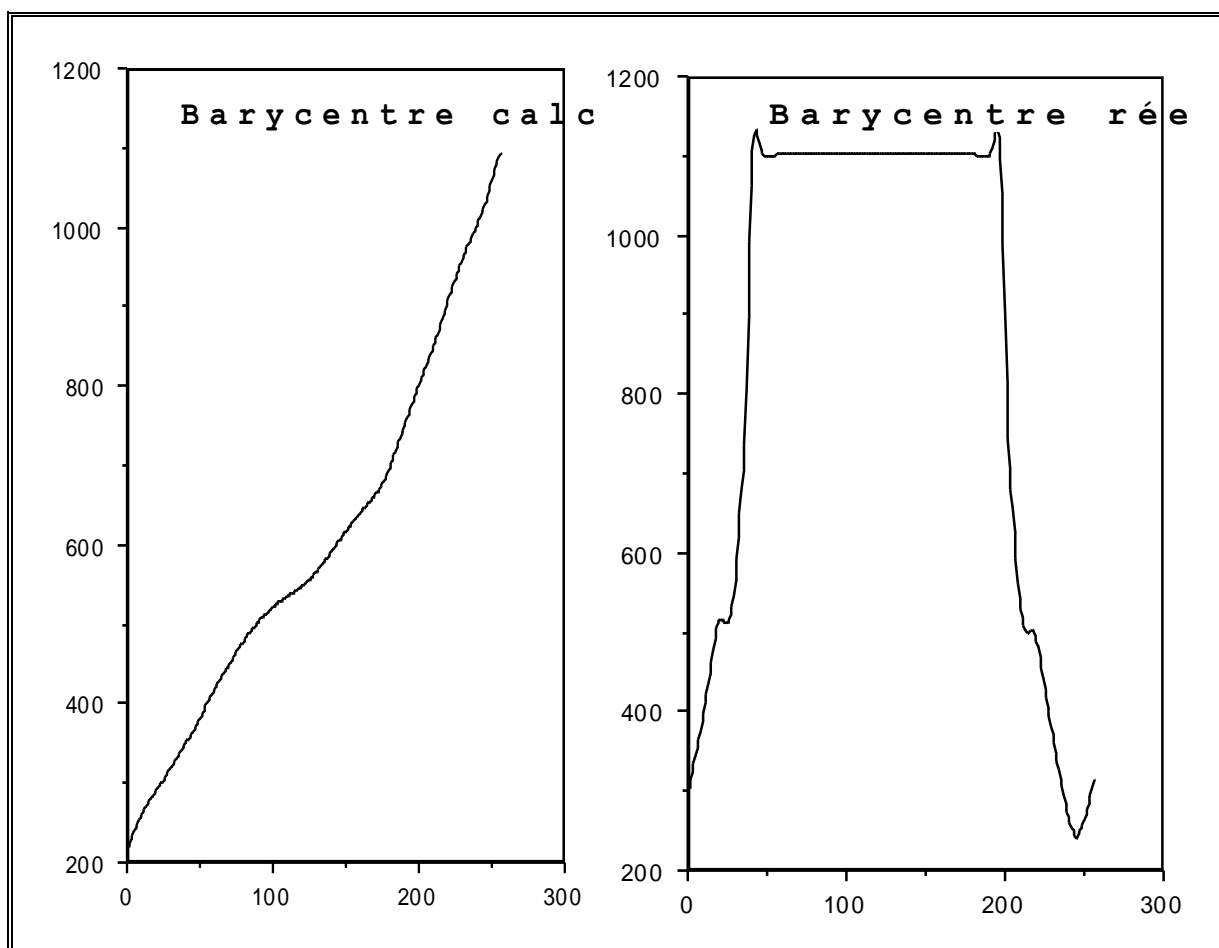


*signal de départ : note (LA2) jouée avec l'instrument académique.*



*Partie du signal anti-filtrée, à l'aide du filtre inverse obtenu expérimentalement. Cette image de la période a été obtenue en mettant les phases des harmoniques à zéro. Initialement MUSICV initialise les phases à zéro, et la conservation de la phase n'est pas garantie.*

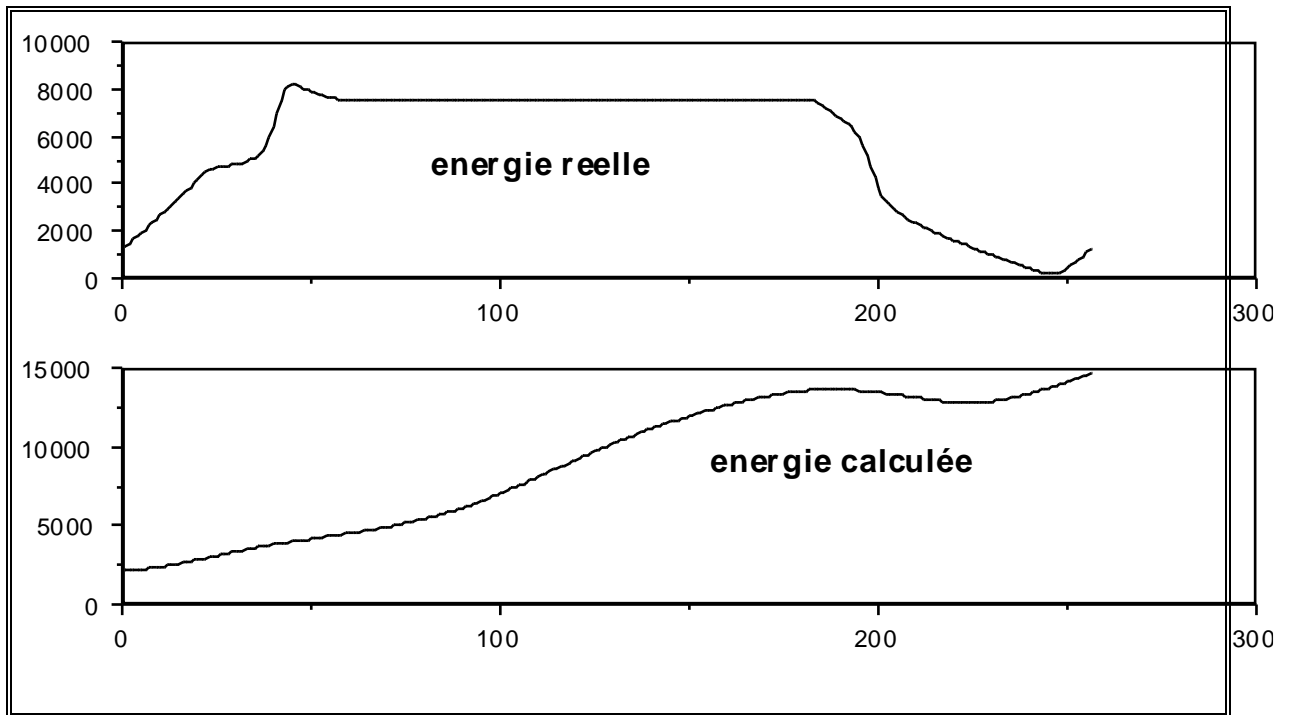
On travaille sur la totalité du signal créée par une note. Pour retrouver l'index de distorsion, il faut extraire une période, calculer le barycentre du son réel et du son expérimental. Le son expérimental est obtenu par la variation de 0 à 1 de l'index pour une fonction de distorsion calculée pour la période extraite.



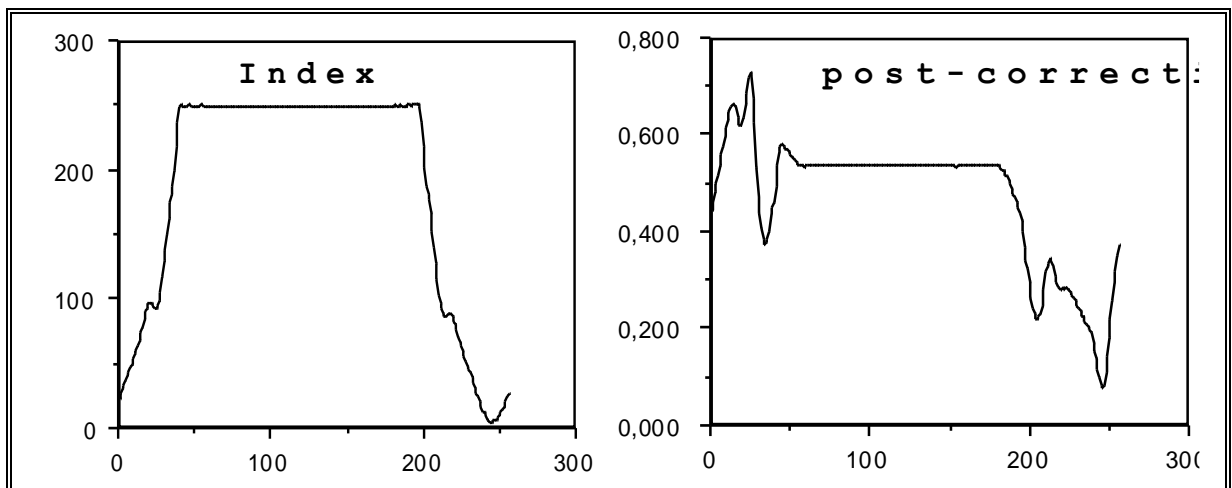
*Sur la figure de gauche on observe l'évolution du barycentre en fonction de la variation de l'index: sur la figure de droite on observe l'évolution du barycentre du son réel au cours du temps.*

A partir de la connaissance de ces deux tables, on construit la courbe de l'évolution de l'index au cours du temps. On fait de même avec l'énergie du signal.

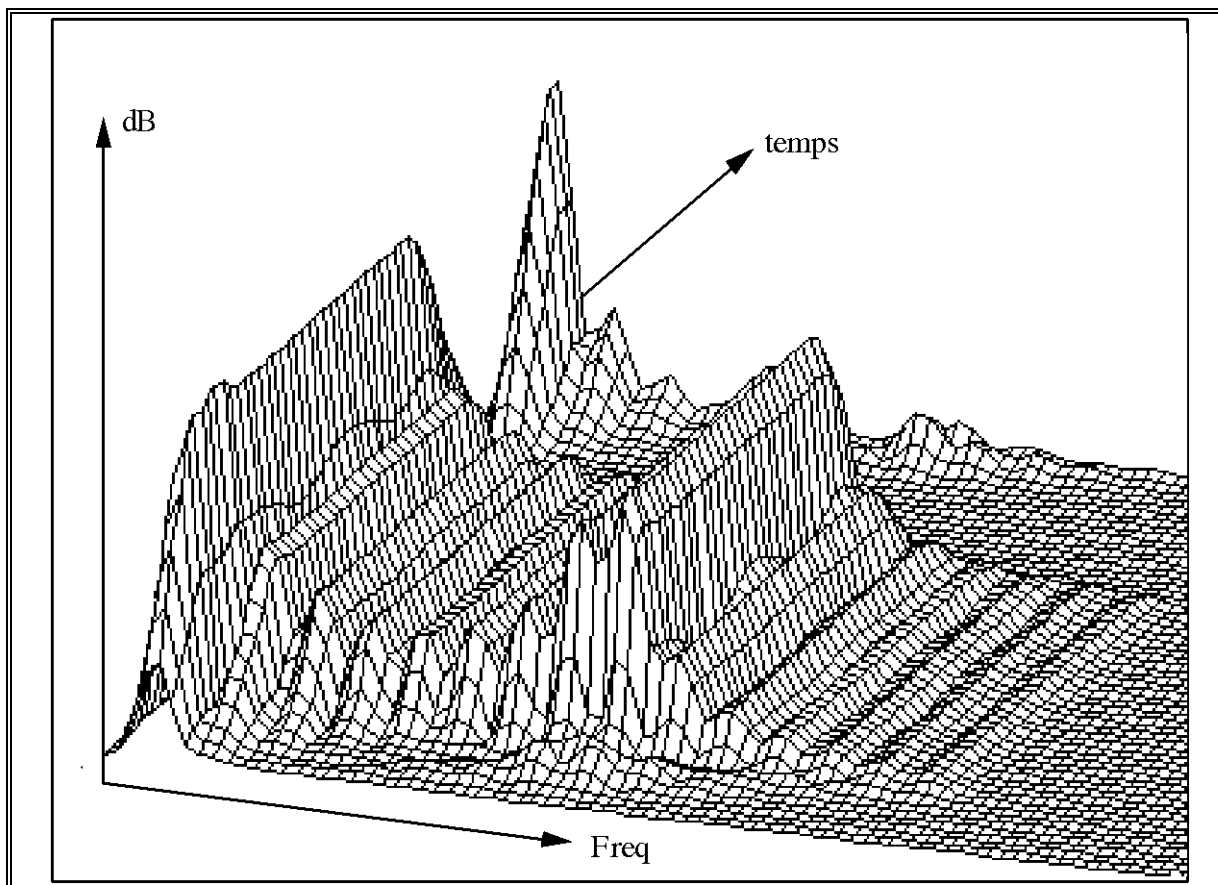




La courbe d'index approche assez bien la courbe initiale.



*Courbe de variation de l'index de distorsion et de post-correction que l'on retrouve à partir de l'expérimentation.*



*représentation 3D de la variation de l'enveloppe spectrale pour une note de la gamme*

#### **VI.5.d.signaux réels**

On réalise l'enregistrement d'une gamme chromatique. Cet enregistrement expérimental se fait de deux manières:

- soit en temps réel à l'aide d'une carte de conversion. Les données sont immédiatement stockées dans la machine. Le programme DSPAC.PAS permet l'acquisition d'une plage de 0.5 seconde choisie aléatoirement pendant le jeu d'une note tenue au moment où le signal semble le plus périodique. Cela demande un peu d'habileté de la part de l'instrumentiste et de l'expérimentateur. Cette méthode est rapide mais ne permet pas de choix.

- soit sur une bande magnétique. Dans ce cas il faudra tout de même repasser par le cas précédent pour faire l'acquisition des notes. L'avantage est que l'on dispose de tout son temps pour choisir l'endroit du signal présentant les meilleures dispositions de stationnarité et de périodicité. On visualise l'enveloppe du signal et on détermine à l'aide d'un curseur graphique l'instant convenable. Cette indication temporelle permettra de retrouver une période dans le signal dans un voisinage immédiat.

Dans les deux cas l'instrumentiste monte une gamme chromatique dans les conditions qui lui ont été fixées à l'avance: essayer de conserver au mieux une excitation identique pour chacune des notes. On a vu que cela consistait par exemple pour le violoniste à frotter la corde toujours au même rapport de longueur.

#### ***VI.5.d.1. Contrôle de l'excitation***

##### ***VI.5.d.1.1. pour l'archet***

Pour les instruments à cordes on effectue un contrôle sur la position de l'archet par rapport au bas du chevalet. Ceci afin de produire au mieux un signal dont le spectre pourra être connu à l'avance. Exemple, le spectre d'une corde frottée au quart de sa longueur peut être déterminé mathématiquement.

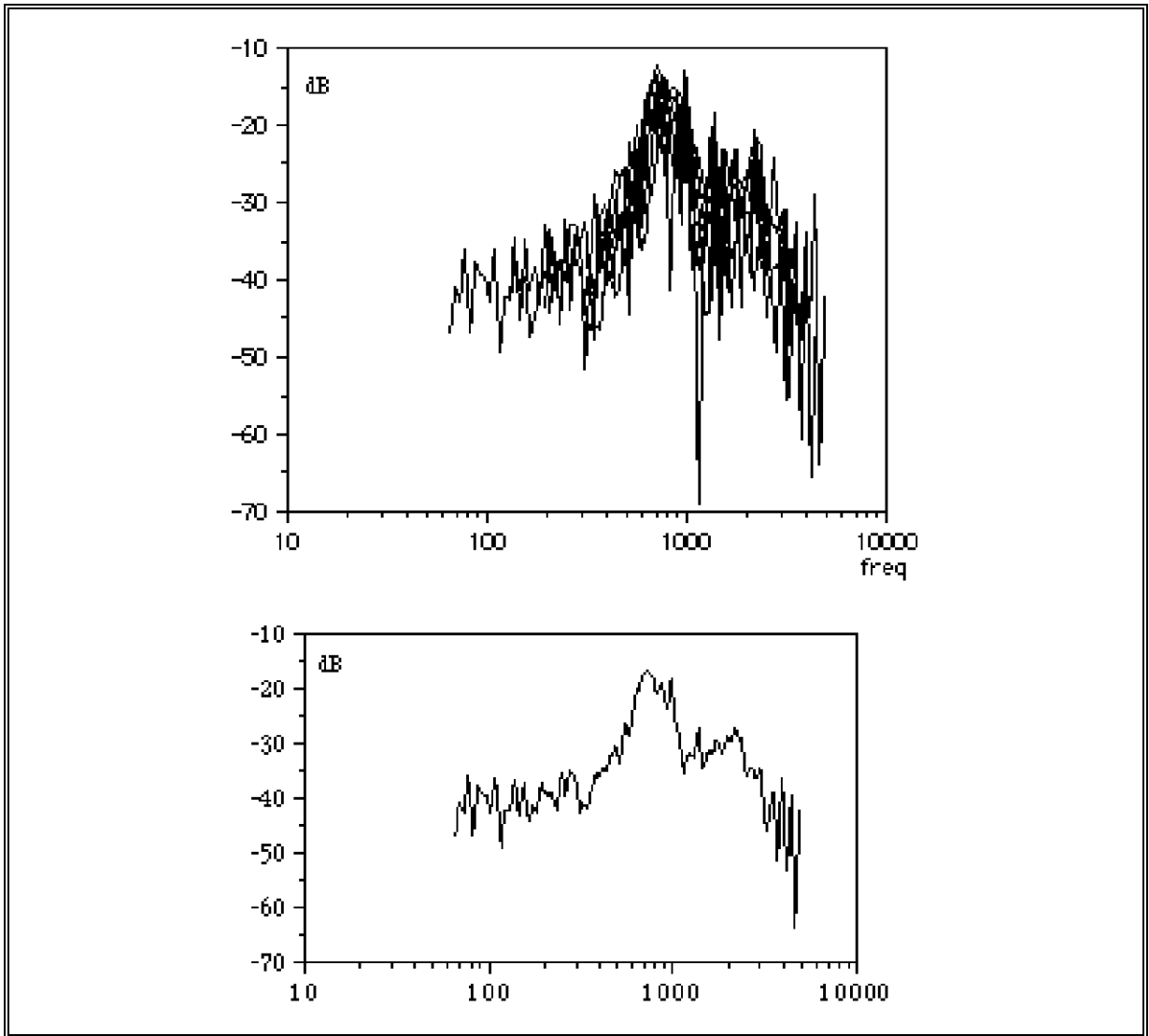
##### ***VI.5.d.1.1.1. pour la voix chantée***

Il a été demandé à la personne qui prêtait sa voix pour l'expérience de chanter une voyelle en montant une gamme chromatique. Chaque note était filée et le chanteur essayait d'imposer un contrôle sur la forme du conduit vocal à l'aide d'un miroir. Cette technique a été possible car le chanteur possédait une technique employée dans un certain style de chant tibétain.

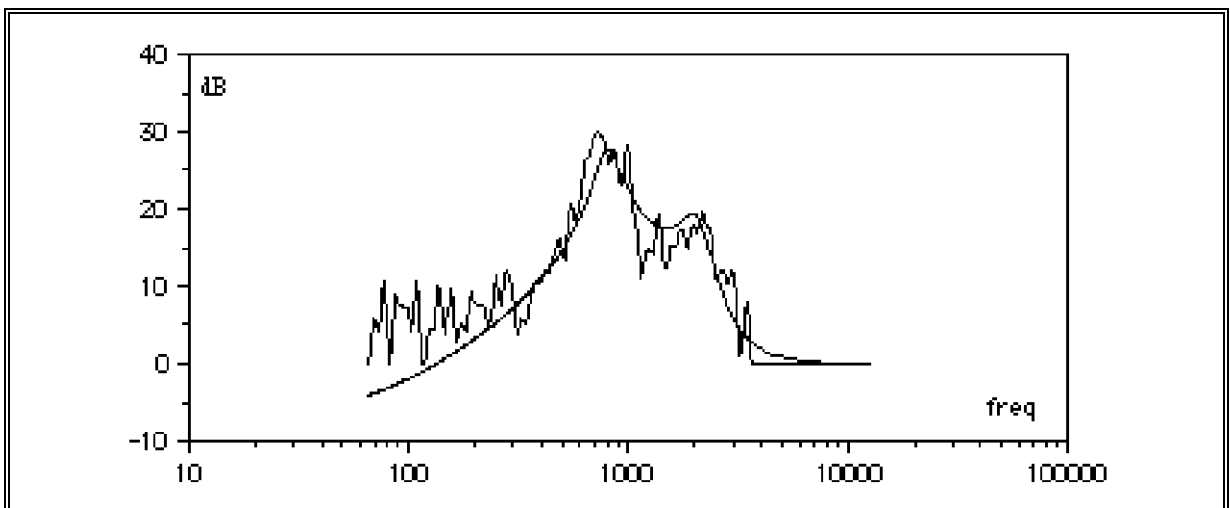
#### ***VI.5.d.1.2. Type d'instrument***

##### ***VI.5.d.1.2.1. Instruments à cordes***

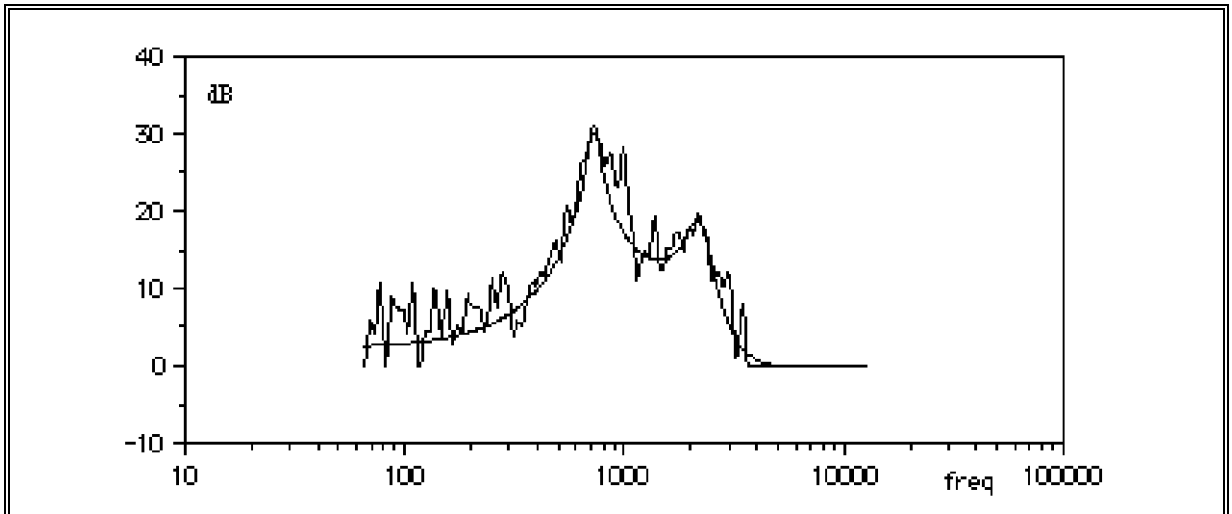
Le premier utilisé est un violon rudimentaire, constitué d'une courge séchée qui sert de caisse de résonance sur laquelle sont montés le manche et les cordes d'un violon traditionnel.



*Reconstitution du spectre du violon-courge de Jacques Dudon à partir de la connaissance des trajectoires de tous les harmoniques*

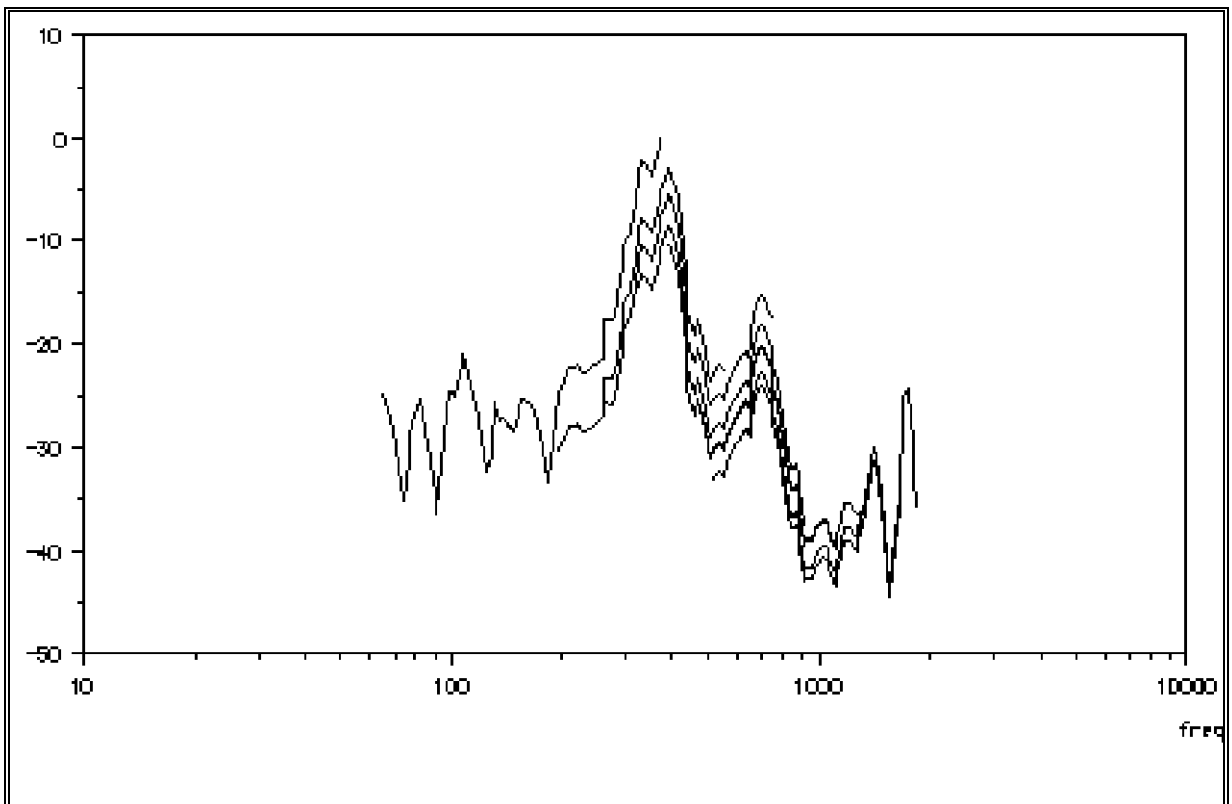


*Après la linéarisation on obtient une première approche de l'enveloppe du spectre*

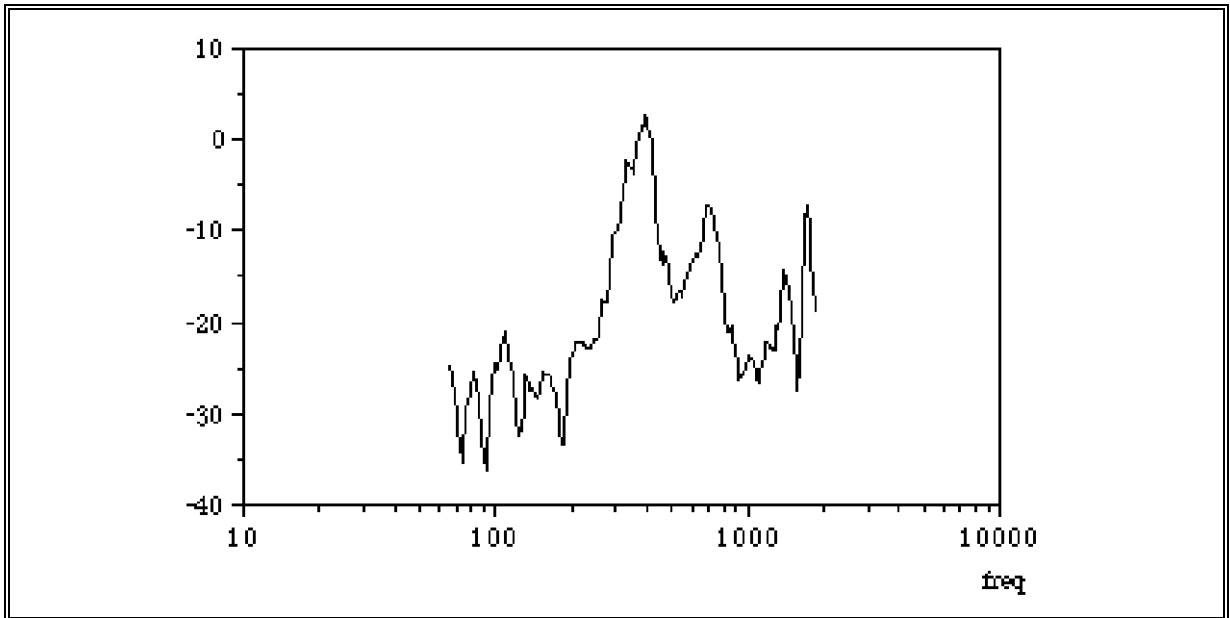


*Après la modélisation par Marquardt on obtient une détection des formants de plus en plus exacte.*

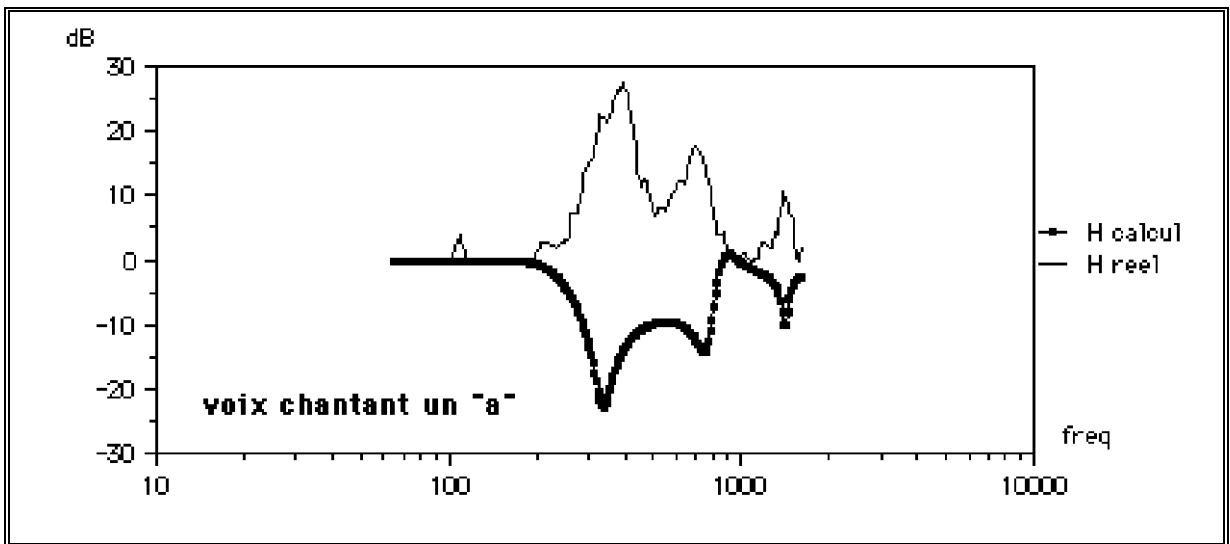
**VI.5.d.1.2.2. Voix chantée**



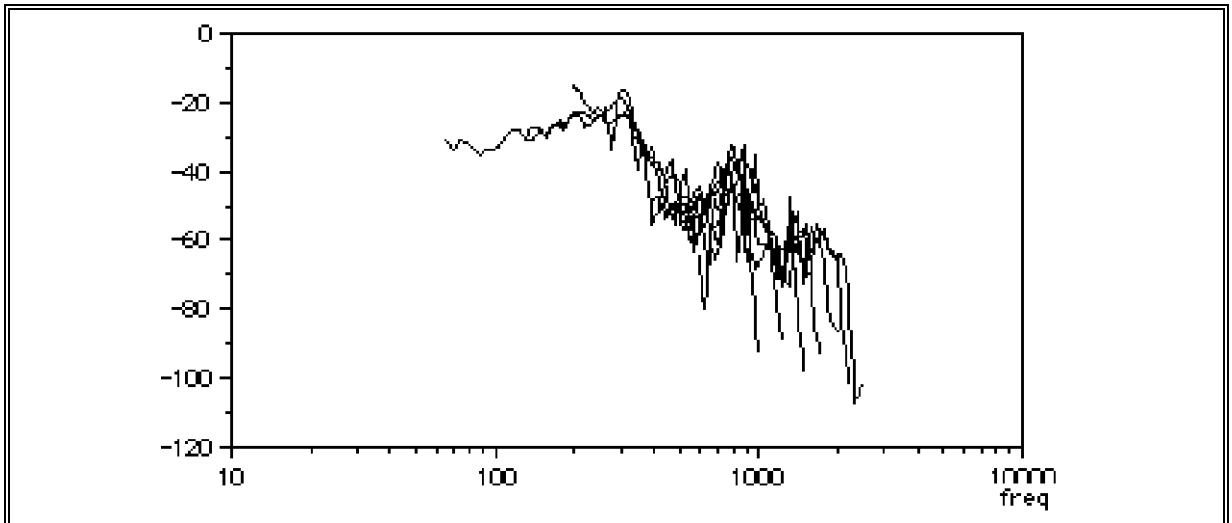
*Ensemble des valeurs obtenues pour la voyelle "a" chanté par Jacques Dudon.*



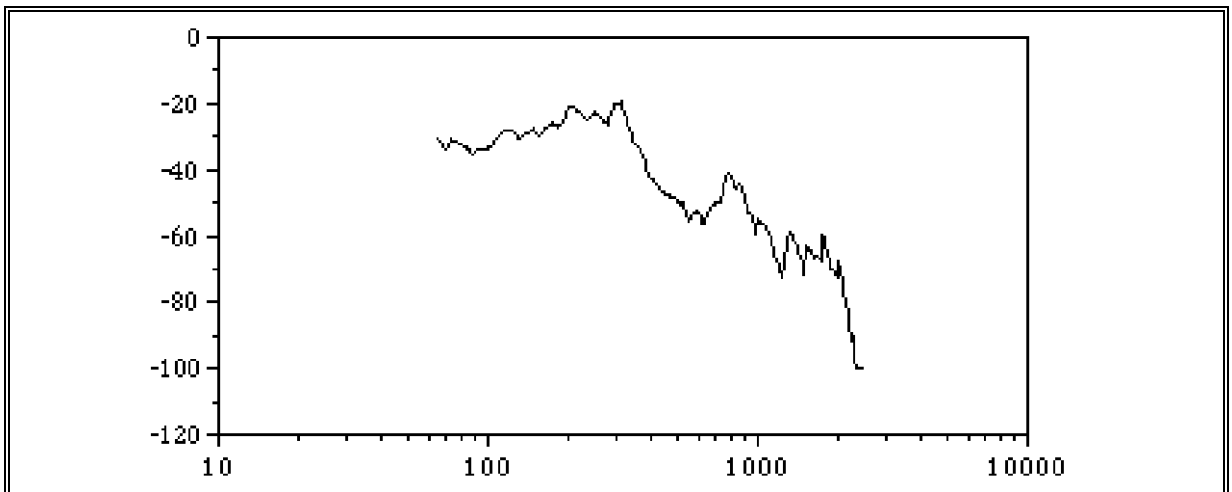
*Spectre reconstitué*



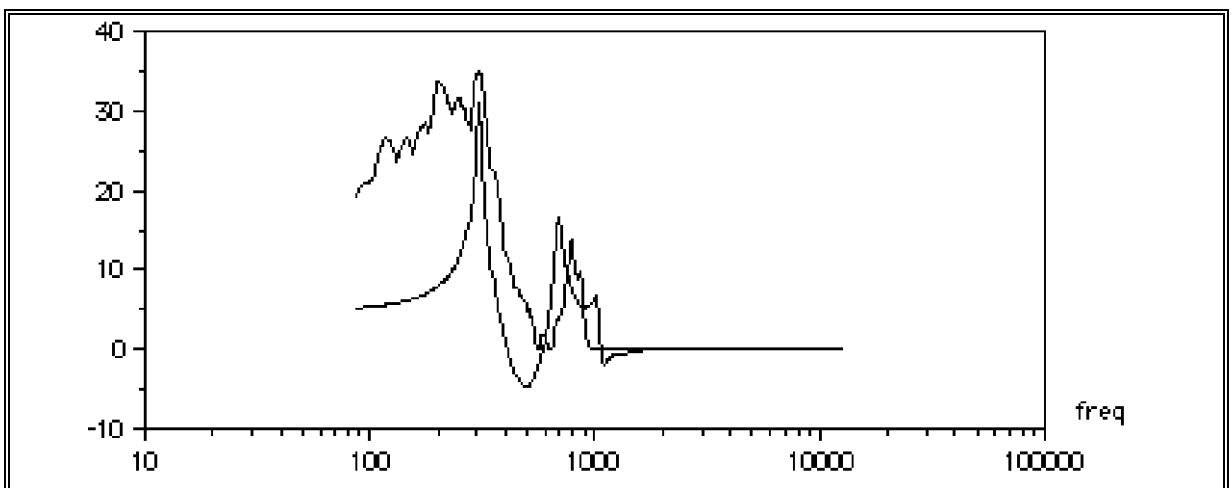
*Estimation de l'enveloppe spectrale du filtre inverse.*



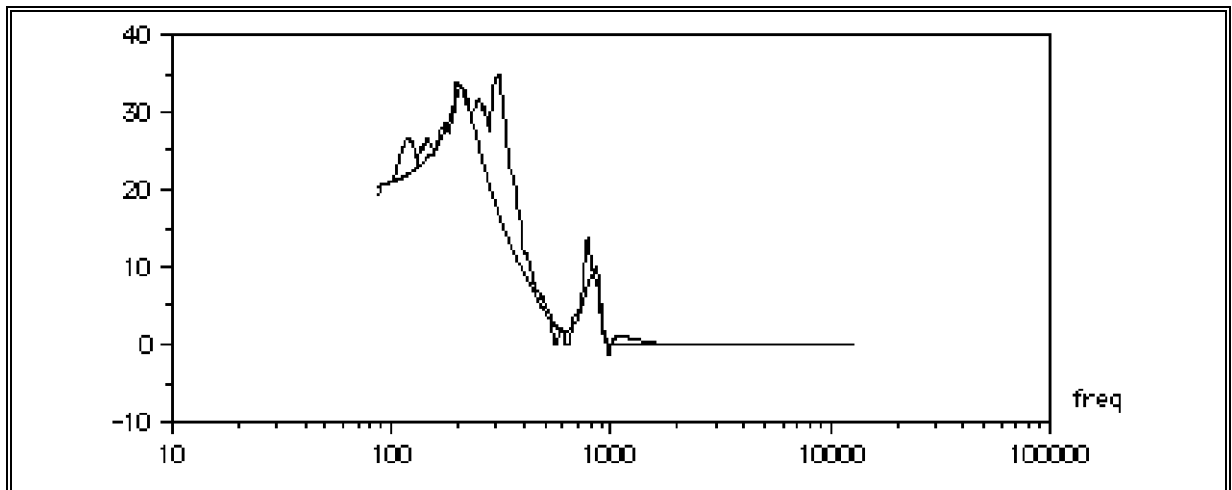
*Ensemble des harmoniques de même rang pour la voyelle "a" chantée par une femme*



*spectre reconstitué*



*Estimation de l'enveloppe après linéarisation*



*On affine par Marquardt*

### **VI.5.e. applications sonores**

Le procédé fournit des données utiles pour la resynthèse. Il a été effectivement utilisé musicalement, notamment dans une pièce de J.C. Risset écrite pour le clarinettiste Michel Portal. Les possibilités de "microchirurgie sonore" développées dans le cadre de ce travail y ont été exploitées, notamment la possibilité de resynthétiser le son de clarinette par distorsion non-linéaire.



## VII. CONCLUSION

Dans l'ensemble des travaux décrits dans ce mémoire, la plus grande place a été donnée à l'extraction de paramètres sonores en vue d'une resynthèse. Néanmoins mon étude sur la méthode de compilation du langage MUSICV a mis en oeuvre une nouvelle façon d'écrire des compilateurs. L'amélioration apportée au compilateur existant de MUSICV consiste en une meilleure détection des erreurs au niveau de la syntaxe et de la sémantique du langage. Mais de façon plus globale, l'écriture d'un compilateur autojectif apporte une simplification de l'écriture du système informatique qui l'accueille.

Cette simplification passe par la suppression de la notion d'interruption dans les systèmes. Elle est remplacée par une auto-structuration du compilateur qui conduit à une gestion plus simple par le système des quanta de travail. L'intérêt de cette méthode tient à l'insuffisance de réflexion théorique accompagnant la création de systèmes informatiques. Depuis l'apparition du premier système qui incluait un compilateur MUSICV, celui-ci n'a été qu'une succession de modifications à partir d'un système initial. Ce système initial était conçu pour des ordinateurs ayant très peu de mémoire centrale, d'où une gestion plus difficile. Il serait bon de se poser de nouvelles questions sur l'écriture d'un compilateur (musical ou non) qui ne serait plus tributaire du système, mais dont l'écriture impliquerait une modification de celle du système, allant dans le sens d'une simplification.

L'avantage d'une méthode qui simplifie la compréhension d'un système informatique doit être de mettre le contrôle du système à la portée de tous les utilisateurs: celui-ci n'étant pas réservé au seul privilège de quelques initiés. Il semblerait qu'avec cette méthode, l'écriture du compilateur soit un peu plus compliquée, tandis que l'écriture du système se trouve ramenée au niveau de difficulté d'une simple procédure en PASCAL (pour citer un langage connu de tous les étudiants).

Sur le plan musical, on observe sur le marché l'apparition de nombreux logiciels qui sont souvent livrés avec un sous-ensemble du système: notamment tous ceux dont le "package" contient un éditeur, un compilateur et un dérouleur. Cela montre la difficulté pour les créateurs de logiciel de faire abstraction du système sur lequel sera implanté leur programme. L'apparition de nouvelles interfaces de type graphique tend à faire oublier la mise en oeuvre de compilateurs ou d'interpréteurs. Cette face cachée de l'informatique est souvent délaissée, car par habitude on se satisfait de ce qui fonctionne tant bien que mal. C'est pourquoi une

étude sur les compilateurs n'est pas négligeable surtout quand elle entraîne une unification et une simplification de la théorie des systèmes.

L'étude électronique de la carte de contrôle du synthétiseur du commerce (SFG-05) m'a permis d'acquérir et de mettre en oeuvre des connaissances pratiques liées à l'Informatique Musicale. Une grande part théorique de l'Informatique Musicale se pratique dans les laboratoires de recherche et le grand public profite des réalisations techniques, souvent par le biais de l'industrialisation et de la commercialisation de brevets. C'est le cas en particulier de la synthèse FM [CHOWNING] développée à l'université STANFORD et dont la réalisation pratique équipe tous les synthétiseurs DX fabriqués par YAMAHA. Cette invasion d'appareils dotés de l'interface MIDI, qui permet le dialogue entre ordinateurs et synthétiseurs, a créé également un besoin de techniciens et d'informaticiens spécialistes dans ce domaine. Les studios de musique s'équipent de plus en plus en appareillages numériques et en commandes MIDI. Des recherches dans ce domaine ont été réalisées notamment par Frédéric BOYER qui a réalisé un séquenceur MIDI doté d'une passerelle vers MUSICV. L'électronique liée à la fabrication des synthétiseurs ne cesse d'évoluer: on est loin, à l'heure des "transputers", des premiers systèmes analogiques. L'Informatique Musicale se doit, dans ce contexte, de parvenir à un équilibre entre les études théoriques et la réalisation pratique, et la connaissance des interfaces MIDI est le passage quasi-obligé entre le monde de la recherche et le monde du studio.

Mon étude de la séparation source-résonance à partir d'une gamme jouée sur des instruments à formants fixes m'a permis de réaliser l'un des premiers exemples d'extraction de paramètres sonores en vue d'une resynthèse. On peut ainsi arriver à l'imitation de sons naturels par synthèse, et la modélisation permet des modifications intimes sur les sons. Cette étude a été menée dans un cadre réduit, avec comme contrainte la mise en oeuvre de très peu de matériel. Ce qui correspond souvent au cas d'un compositeur ayant à sa disposition un ordinateur et une carte d'acquisition. Il est évident que des laboratoires bien équipés ont à leur disposition d'autres types de matériels, ce qui montre par ailleurs l'intérêt porté à ce type de recherche. L'apparition de la théorie des ondelettes a stimulé une nouvelle vision des phénomènes temps-fréquence. Les travaux de Richard KRONLAND portent également sur le problème de l'extraction automatique de paramètres, rendue plus difficile par la représentation du phénomène avec une dimension supplémentaire. Il serait envisageable d'améliorer certaines parties de mon étude, notamment dans les parties transitoires de l'attaque et de la chute d'une note, la distorsion non-linéaire donnant dans ces zones des sons qui éclatent comme l'attaque d'un son de trompette ou de trombone.

L'étude a été menée sur des instruments à formants fixes. Néanmoins si on fait abstraction de la première partie qui consiste à modéliser un spectre, on peut facilement conserver la

modélisation par distorsion non-linéaire pour tous les sons ne possédant pas de formants comme les cloches ou la clarinette. J'ai également utilisé des méthodes classiques employées dans certains analyseurs pour extraire des paramètres spectraux d'un signal quelconque. L'enveloppe spectrale peut être modélisée sans passer par le jeu d'une gamme chromatique et le filtre est inversé par la même méthode. Cela donne des résultats convenables. La méthode utilisée est basée sur l'accumulation des puissances spectrales (PSACC). Tout cela montre que la méthode d'extraction de paramètres à l'aide d'une gamme chromatique n'est pas figée: les différentes parties qui la constituent peuvent être employées par les utilisateurs comme les éléments d'une "boîte à outils" pour MUSICV. Cela était l'un des objectifs de l'étude.

Le problème d'extraction de paramètres sonores comporte un enjeu musical: conférer au traitement des sons "naturels" la ductilité propre à la synthèse. Son étude est poursuivie actuellement par les travaux de Nathalie DELPRAT et de Philippe GUILLEMAIN au sein de l'équipe d'Informatique Musicale du LMA (Laboratoire de Mécanique et d'Acoustique). Nathalie DELPRAT s'attache à l'extraction de paramètres pour un modèle de synthèse non linéaire (modulation de fréquence) à partir d'une analyse temps-fréquence (ondelette et GABOR). Philippe GUILLEMAIN a réalisé des resynthèses satisfaisantes à partir de l'extraction de données (temps fréquences: ondelette) pour un modèle de synthèse additive.

Les perspectives d'extraction de paramètres pour les sons réels sont en plein développement et l'ensemble des méthodes de l'équipe donnent dans l'ensemble de bons résultats pour les signaux académiques, mais elles doivent encore faire l'objet d'améliorations pour les signaux réels. L'apparition de nombreuses méthodes développées à partir de la théorie des ondelettes (paquet d'ondelettes) permet de penser que le problème de l'extraction des paramètres, abordé dans mon travail, connaîtra des développements considérables dans les prochaines années.



## VIII. BIBLIOGRAPHIE

- [1] D'ALESSANDRO CHRISTOPHE  
*Représentation de signal de parole par somme de fonctions élémentaires*  
these soutenue le 1er avril 1989, université Paris Sud
- [2] ARFIB DANIEL.  
*L'usage musical de la synthèse de spectres complexes par distorsion non linéaire et modulation d'amplitude.*  
1983, Thèse d'Etat, Université Aix-Marseille II.
- [3] ARFIB DANIEL.  
*Transformations exotiques de sons naturels sur ordinateur.*  
1990, Colloque de physique, colloque C2, supplément n°2, Tome 51, Février, 1er Congrès Français d'Acoustique, pp. C2-853-856.
- [4] ATAL B.S. AND S.L. HANAUER  
*Speech Analysis and synthesis by Linear Prediction of the speech wave*  
JASA, 50, 637-655 (1971)
- [5] BEAUCHAMP JAMES W.  
*Synthesis by Spectral Amplitude and "Brightness" Matching of Analysed Musical Instrument Tones.*  
1982, J. Audio Eng. Soc., Vol. 30, No. 6, June, pp. 396-406.
- [6] BEAUCHAMP JAMES W.  
*A Computer System for Time-Variant Harmonic Analysis and Synthesis of Musical Tones.*  
1969, Music by Computers, FOERSTER H. Von et BEAUCHAMPS James W. Eds., Wiley, New-York, pp. 19-62.
- [7] BEAUCHAMP JAMES W., FORNANGO JAMES.  
*Transient analysis of musical tones with digital filters.*  
1967, J. Audio Eng. Soc., 15, 1, 96.
- [8] BEAUFILS P. M., RAMI M.  
*Le Filtrage Numérique.*  
SYBEX, Paris, 1986.
- [9] BELLANGER M.  
*Traitement Numérique du Signal.*  
Masson, Paris, 1984.

- [10] BELLANGER M.  
*Analyse des signaux et filtrage numérique adaptatif.*  
1989, Masson.

- [11] BELLANGER MAURICE G., DAGUET JACQUES L., LEPAGNOL GUY P.  
*Interpolation, Extrapolation, and Reduction of Computation Speed in Digital Filters.*  
1974, IEEE Trans. ASSP, Vol. ASSP-22, No. 4, Aug., pp. 231-235.
- [12] BENADE ARTHUR H.  
*Fundamentals of Musical Acoustics.*  
1976, Oxford Univ. Press.
- [13] BENNETT GERALD, RODET XAVIER.  
*Synthesis of the Singing Voice.*  
1989, Current Directions in Computer Music Research, Max V. Mathews and John R. Pierce Eds, The MIT Press.
- [14] BOITE R. , LEICH H.  
*Les Filtres Numériques.*  
Masson, Paris, 1980.
- [15] CALBERG LUC, LEFEBVRE PATRICK.  
*L'indispensable pour la musique assistée par ordinateur.*  
1988, Marabout, Alleur, Belgique.
- [16] CHAMBERLIN H.  
*Musical Applications of Microprocessors.*  
Rochelle Park, New Jersey: Hayden Book Company, Inc., 1980. Digital Filtering.
- [17] CHOWNING J. M.  
*The Simulation of Moving Sound Sources.*  
J. Audio Eng. Soc., Vol 19, pp 2-6, and Computer Music Journal, 1977 June.
- [18] COULON F.  
*Théorie et Traitement des signaux.*  
Dunod, 1984.
- [19] DESCOUT R.  
*Méthodes d'étude portant sur la détermination de la fonction d'aire du conduit vocal.*  
CNET Lannion, 2 Février 1972.
- [20] DODGE C., JERSE T. A.  
*Computer Music.*  
Schirmer Books, New York, Collier Macmillan, London, 1985. 8 Composition with Computers, pp 265-279.
- [21] DODGE CHARLES, JERSE THOMAS A.  
*Computer Music Synthesis, Composition, and Performance*  
Schirmer books, ISBN 0-02-873100-X
- [22] DUTILLEUX PIERRE.  
*Simple to Operate Digital Time-Varying Filters.*  
1989, Preprint 2757, 86th AES Convention, Hamburg, March.

- [23] FLANAGAN J. L.  
*Speech Analysis Synthesis and Perception.*  
1972, Springer-Verlag.



- [24] HAMMING R. W.  
*Digital Filters.*  
1989, Prentice-Hall. ISBN : 0-13-212895-0.
- [25] HECQUET ANTHONY.  
*L'environnement MIDI et ses applications musicales.*  
1989, SYBEX.
- [26] LE BRUN MARC.  
*Digital Waveshaping Synthesis.*  
1979, J. Audio Eng. Soc., Vol. 27, No 4, April, pp. 250-265.
- [27] MAILLIARD B.  
*Les Distortions de MUSIC-V.*  
Cahiers Recherche/Musique, No 3, Synthétiseur ordinateur, INA-GRM, 1976.
- [28] MATHEWS M. V., MILLER J. E., MOORE F.R., PIERCE J. R., RISSET J. C.  
*The Technology of Computer Music.*  
The MIT Press, 1969. pp 68-73.
- [29] MATHEWS MAX V., PIERCE JOHN R.  
*Current Directions in Computer Music Research.*  
1989, The MIT Press. ISBN: 0-262-13241-9.
- [30] MATHEWS MAX V.  
*Analysis and Synthesis of Timbres.*  
Music Room and Acoustics, Publications of the Royal Swedish Academy of Music 17, Stockholm 1977.
- [31] MATHEWS MAX. V, KOHUT J.  
*Electronic Simulation of Violin Resonances.*  
J. Acoust. Soc. Am., Vol. 53, 1973, pp 1620-1626.
- [32] MIDI 1.0 SPECIFICATION.  
*International MIDI User's Group (IMUG).*  
1983, P.O. Box 593, Los Altos, CA, 94022.
- [33] MITO MICHIHARU, TAKAHASHI KIYOSHI, KUROKAWA SHUJI,  
NAKAMURA SHOGO, KUBOTA TADAHIRO.  
*Real Time Pitch Detection with Digital Signal Processor.*  
J-7, The Second Joint Meeting of ASA and ASJ, Nov. 1988.
- [34] MOORER J. A.  
*The Use of Linear Prediction of Speech in Computer Music Applications.*  
1979, J. Audio Eng. Soc., Vol. 27, No. 3, pp 134-140, March.
- [35] MOORER JAMES A.  
*On the Transcription of Musical Sound by Computer.*  
1977, Computer Music Journal, November.
- [36] MURRAY HILL  
*A General Program to Perform Sampling Rate Conversion of data by Rational*

*Ratios.*

IEEE Acoustics, speech, and Signal Processing Society. Bell Laboratories, , NJ 07974

- [37] NOMURA NORIO, YOSHIDA YUICHI.  
*A Pitch Extraction Method Using Higher Order Joint Moment.*  
J-5, The Second Joint Meeting of ASA and ASJ, Nov. 1988.
- [38] PIERCE JOHN R.,  
*traduit de l'anglais par BERQUIER Françoise, avec la collaboration de Risset Jean Claude.*  
*Le son musical, musique, acoustique et informatique.*  
1984, Pour la Science, BELIN, Paris. ISBN 2-90218-35-X.
- [39] PIERCE JOHN R.  
*The science of musical sound*  
Scientific american library, ISBN 0-7167-1508-2
- [40] PRESS W. H., FLANNERY B. P., TEUKOLSKY S. A., VETTERLING W. T.  
*Numerical Recipes, The Art of Scientific Computing.*  
Cambridge University Press, 1987.
- [41] RABINER L. R. , GOLD B.  
*Theory and Application of Digital Signal Processing.*  
Prentice-Hall, Englewood Cliffs, N. J., 1975.
- [42] RABINER L. R., SHAFER R. W.  
*Digital Processing of Speech Signals.*  
Prentice Hall, Inc., Englewood Cliffs, N. J., 1978.
- [43] REGALIA P. A., MITRA S. K., VAIDYANATHAN P. P.  
*The Digital All-Pass Filter: A Versatile Signal Processing Building Block.*  
Proceedings of the IEEE, Vol 76, No. 1, Jan. 1988.
- [44] RIGDEN JOHN S.  
*Physics and the sound of music*  
John Wiley & sons, ISBN 0-471-02433-3
- [45] RISSET J. C.  
*An Introductory Catalogue of Computer Synthesized Sounds.*  
Bell Telephone Laboratories, Murray Hill, N. J., 1969.
- [46] RISSET JEAN-CLAUDE.  
*Musical Acoustics.*  
1978, Handbook of Perception, Vol. IV, Academic Press, CARTERETTE et FRIEDMAN Eds., pp. 521-564.
- [47] SAPIR SYLVIANE.  
*Processeurs Audio-Numériques Temps-Réel et Informatique Musicale: Mise en œuvre d'un Système d'Exploitation et Utilisation pour la Recherche, la Production et la Pédagogie Musicales.*  
1987, Faculté des sciences de Luminy, Aix-Marseille II, Thèse Docteur d'Etat-Sciences, Septembre.

- [48] SCHAEFFER P.  
*Traité des objets musicaux.*  
Seuil, Paris, 1966.
- [49] SMITH J. O.  
*Introduction to Digital Filter Theory.*  
CCRMA Report No. STAN-M-20. April 1985.
- [50] SMITH J. O.  
*techniques for digital design and system identification with application to the violin*  
CCRMA, department of Music, univ. Stanford Raport No. STAN-M-14