2.5.6 Effet Rotary sur les enceintes (Leslie)

Définition

La radiation d'un son émis par un haut-parleur rotatif crée un effet de spatialisation saisissant. L'effet Rotary consiste à mettre en rotation une enceinte. Cet effet rend vivants même les sons sourds et stables en les animant de qualités variant dans le temps.

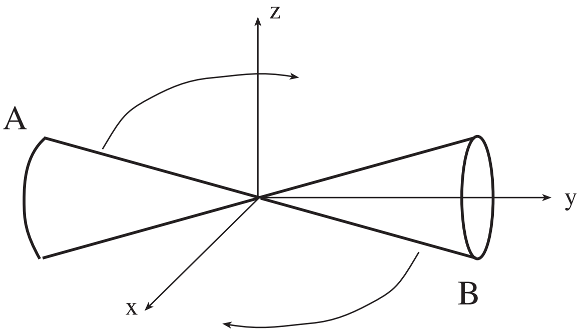

Fig. 2.22: *Système Rotary (à enceintes rotatives) permettant l'effet Leslie.*

Fonctionnement

Le mécanisme originel d'enceinte rotative fut la Cabine Leslie, qui envoyait un signal entrant dans deux mécanismes rotatifs séparés : un pavillon rotatif pour les hautes fréquences et un écran rotatif (bloquant et débloquant un boomer stationnaire) pour les basses fréquences. Une télécommande pour la vitesse du moteur permettait au musicien d'ajuster la vitesse de rotation. Le pavillon résonant de la Cabine Leslie la rend immédiatement identifiable. Ce système fut à l'origine construit pour enrichir le son d'orgues électriques, tels le Hammond B3, avec lequel il était souvent couplé.

Les effets induits par la rotation sont nombreux, mettant en jeu le vibrato de l'effet Doppler, le filtrage variant dans le temps, les déphasages, les distorsions causées par la turbulence de l'air, les réflexions d'échos des surfaces adjacentes, ainsi que les caractéristiques de transfert des amplificateurs et des enceintes utilisés.

Le pavillon est double, le haut-parleur A ayant ses modifications de hauteur et d'intensité dans la direction opposée à celle du haut-parleur B.

Tous ces effets simultanés sont difficiles à simuler de façon convaincante en utilisant le traitement numérique du signal. Une combinaison de modulateurs (pour l'intensité) et de lignes à retard (pour moduler la hauteur) permet de simuler assez bien cet effet. La simulation de l'effet Doppler des deux pavillons opposés est réalisée par deux lignes à retard modulées avec des configurations de vibrato en déphasages de π radians. La caractétistique de directionnalité des haut-parleurs tournant peut être rendue par une modulation en amplitude des signaux en sortie des lignes à retard. Cette modulation est synchrone à la modulation de la ligne à retard, de manière à ce que le mouvement en arrière du pavillon produise une baisse de la hauteur et de la puissance du son. Au point de demi-tour, l'intensité est minimal et la hauteur non modifiée par rapport au son original. Le mouvement inverse, de l'arrière vers l'auditeur, produit les effets inverses : hausse de la hauteur et de l'intensité. Un effet stéréophonique de haut-parleurs tournant est obtenu par un mixage inégal des deux lignes à retard dans les sorties correspondant aux canaux gauche et droit.

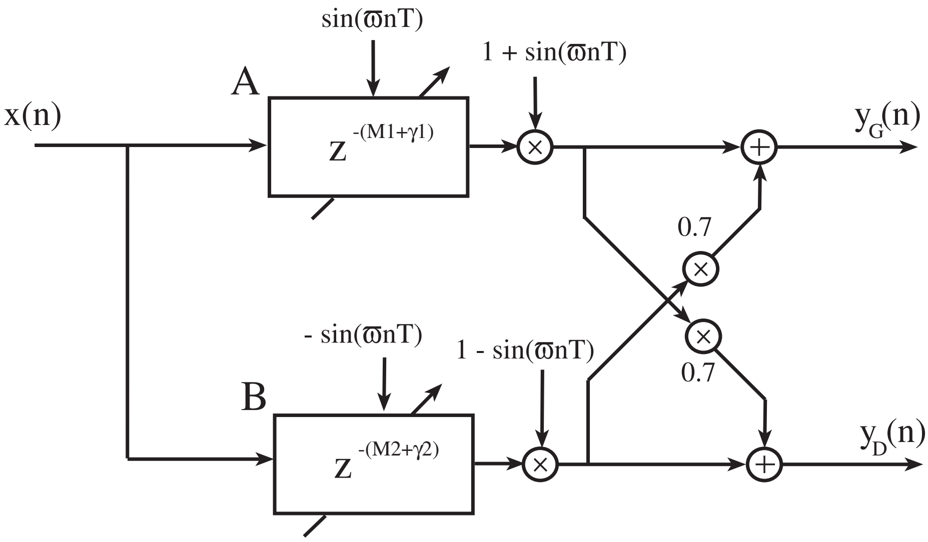

Fig. 2.23: *Diagramme de l'effet leslie, ou enceintes rotatives.*

2.5.7 Rendu de distance et d'espace : 3D avec casque

Introduction

Le rendu de distance et d'espace en trois dimensions avec casque (ou en écoute binaurale) consiste à modéliser les filtrages dûs aux réflexions des ondes sur la tête, le cou et le torse, avant d'arriver dans les conduits auditifs. Ainsi, lors de l'écoute au casque, le signal envoyé directement dans les conduits auditifs correspondra à celui d'un signal ayant parcouru l'espace dans les mêmes conditions. Pour réaliser cet effet, il faut d'abord parler de localisation et de sa perception, puis de la manière d'externaliser un son qui, à l'écoute au casque, paraît être situé dans la tête. Enfin, nous aborderons les fonctions de transfert relatives à la tête (*HRTF, Head-Related Transfert functions*). Pour plus de détails sur ces techniques, nous conseillons vivement la lecture de [16].

Perception de la localisation

La localisation d'un son est perçu grâce à trois données du signal sonore stéréophonique :

- la différence d'intensité interaurale (*IID, Interaural Intensity Difference*), utilisée par le système auditif pour les fréquences supérieures à 1500 Hz ;
- la différence de temps d'arrivée interaurale (*ITD, Interaural Time Difference*), pour les fréquences inférieures à 1500 Hz ;
- Les fonctions de transfert relatives à la tête (*HRTF*), qui permettent de level l'ambiguïté du cône de confusion entre l'avant et l'arrière, lorsque les ITD et IID sont identiques.

La prise en compte des mouvements de la tête a son importance, puisque l'on n'est jamais parfaitement immobile, et que les petits mouvements impliquent de petites variations sur lesquelles le cerveau (système d'analyse différentielle par excellence) se base pour son analyse de localisation. Nous rappellons que la position d'une source dans l'espace est donnée en coordonnées sphériques par la distance (rayon), l'azimuth et l'élévation (cf. fig.2.16).

Codage de l'azimuth par les différence interaurales

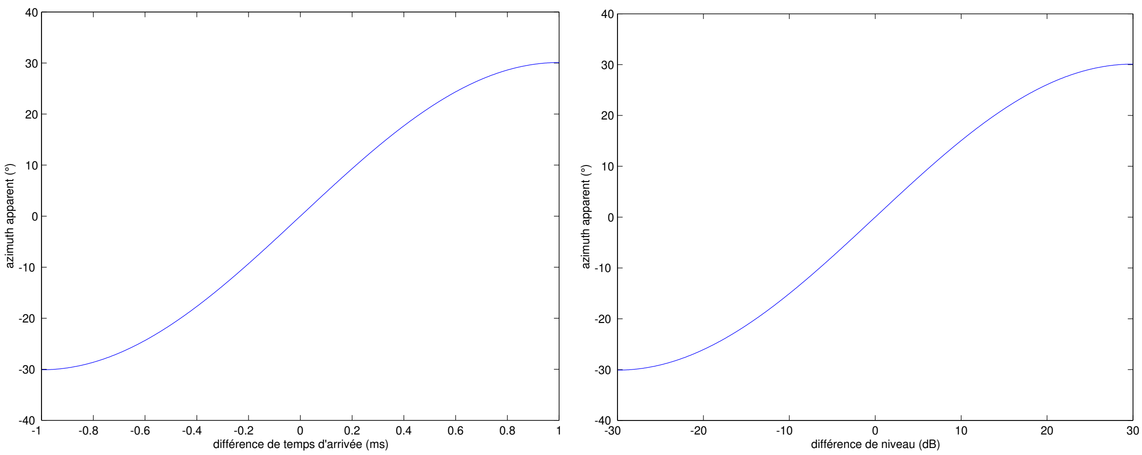

Fig. 2.24: *Azimuth perçue en fonction de la différence de temps (gauche) et d'intensité (droite).*

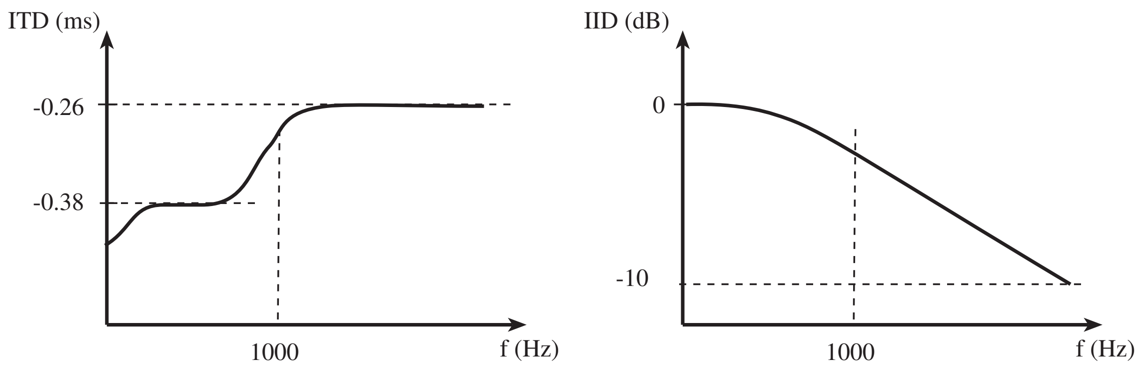

Fig. 2.25: *Différence de temps (gauche) et d'intensité (droite) interaurale dépendant du temps.*

L'amplitude et le temps de délai entre les deux canaux acoustiques permettent de coder l'azimuth d'un son, mais pas son élévation ni sa distance (le son est localisé dans la tête de l'auditeur). Lorsque l'on fait dépendre les différences interaurales de temps (ITD) et d'intensité (IID) de la fréquence, on obtient une bien meilleure localisation. La diffraction du son dans l'air implique que ITD soit plus élevé en basses fréquences qu'en hautes fréquences. La limite basse fréquence est donnée par :

$$ITD = \frac{1.5\delta}{c} \sin\theta \tag{2.37}$$

avec θ l'angle d'incidence (rad), δ la distance interaurale (m) et c la célérité du son dans l'air (ms^{-1}). La limite est pour environ 1000 Hz : en dessous, ITD vaut environ -0.38 ms, au-dessus environ -0.26 ms. Pour la différence d'intensité IID, elle décroît de 0 dB à 0 Hz à -2 dB à 1000 Hz, puis ensuite régulièrement jusqu'à -10 dB pour 20 kHz.

De plus, les différences interaurales changent avec la distance source-auditeur, du fait que les répartitions en fréquences sont modifiées par le filtrage dans l'air (c'est principalement audible pour les basses fréquences).

Externalisation

Lorsque les signaux sonores arrivant dans les deux oreilles sont parfaitement corrélés, le son est internalisé (entendu directement dans la boîte cranienne, à égale distance des deux oreilles). Par contre, lorsqu'il sont décorrélés, même partiellement (ce qui est le cas par exemple pour un son réverbéré : les signaux arrivant aux deux oreilles sont la somme de réflexions aux parois suite à des trajets différents), le son est externalisé. La corrélation entre les deux signaux se mesure ainsi :

$$r(\tau) = \lim_{T\to+\infty} \frac{1}{2T\nu} \int_{-T}^{+T} x_L(t)\ x_R(t+\tau)dt \tag{2.38}$$

avec τ la différence de temps entre les deux canaux et ν le facteur de normalisation défini comme suit :

$$\nu = \frac{1}{2T}\sqrt{\int_{-T}^{+T} x_L(t)^2 dt\ \int_{-T}^{+T} x_R(t)^2 dt} \tag{2.39}$$

Lorsque r vaut 1, les signaux sont parfaitement corrélés ; on les dit cohérents. Lorsque r vaut -1, les signaux sont en opposition de phase. Lorsque r vaut 0, les signaux sont non corrélés.

Il faut savoir que trois méthodes de décorrélation existent : la décorrélation par couple de filtres, la décorrélation par filtres passes-tout à pôles aléatoires et la décorrélation par réseau de délai à réinjection. Nous ne rentrerons cependant pas dans les détails de ces méthodes.

Fonctions de transfert relatives à la tête (HRTF)

Comme expliqué précédemment, ces fonctions de transfert correspondent au filtre composé du torse, du cou et de la tête, ainsi qu'aux échos dûs aux épaules et aux réflexions sur le cou. Ces fonctions de transfert sont personnelles, varient grandement d'une personne à l'autre. La manière de les calculer précisemment pour une personne donnée consiste à introduire des microphones dans les conduits auditifs, et calculer la fonction de transfert réelle entre le signal envoyé sur les haut-parleurs et celui enregistré aux microphones. Il existe aussi des méthodes pour les modéliser [16].

2.5.8 Rendu de distance et d'espace : 3D avec haut-parleurs

Introduction

Le rendu de distance et d'espace en 3D à l'aide de haut-parleurs peut s'obtenir de trois manières : par la reconstruction holophonique (il s'agit de reproduire un champ acoustique 2D ou 3D par 3D-panning ou par procédé Ambisonic), par la technique transaurale (basée sur le binaural, et modifié de façon à fonctionner sur 2 haut-parleurs) ou par une méthode basée sur l'effet de précédence (un modèle de l'effet de salle est utilisé pour ajouter explicitement un délai sur l'une des voies : c'est moins précis en localisation mais aussi moins sensible aux changements de position de l'auditeur). Nous présentons 3 de ces techniques.

Localisation avec plusieurs haut-parleurs

Chaque oreille de l'auditeur reçoit deux contributions : une contribution directe, du haut-parleur d'un côté vers l'oreille de ce même côté, et une contribution croisée (cf. fig. 2.26) :

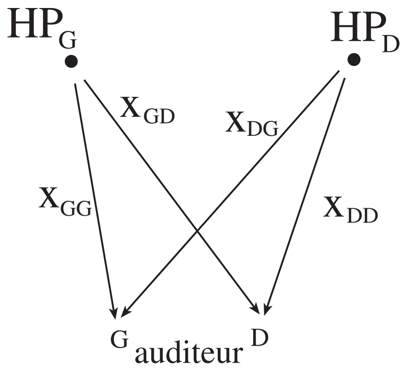

Fig. 2.26: *Signaux reçus par chaque oreille de l'auditeur sur un système à deux haut-parleurs.*

$$x_G(t) = x_{GG}(t) + x_{DG}(t) \tag{2.40}$$
$$x_D(t) = x_{DD}(t) + x_{GD}(t) \tag{2.41}$$

Si on émet une sinusoïde dans les deux canaux, avec un pur gain A_D à droite et un pur retard τ_D, on entendra deux sinusoïdes distinctes, une à chaque oreille :

$$X_G(\omega) = 1 + A_D A_H e^{-j\omega(\tau_D+\tau_H)} \tag{2.42}$$
$$X_D(\omega) = A_D e^{-j\omega\tau_D} + A_H e^{-j\omega\tau_H} \tag{2.43}$$

avec A_H et τ_H le gain et le délai donnés par la HRTF relative à l'oreille contralatérale à la direction du haut-parleur. Remarquons qu'en basse fréquence, sachant que $A_H \approx 1$, un pur délai au haut-parleur sera perçu comme une pure différence de niveau, et réciproquement. En hautes fréquences, le masquage de la tête n'est plus négligeable, et les valeurs données A_H et τ_H peuvent provoquer un effet en contradiction avec ce que donneraient A_{HP} et τ_{HP} au niveau des haut-parleurs.

3D-panning

Le 3D-panning consiste d'abord à utiliser un grand nombre de haut-parleurs situés en demi-sphère autour de l'auditeur. La localisation en azimuth est donnée par un panning entre les deux haut-parleurs les plus proches. La localisation en azimuth et élévation nécessite quant à elle 3 haut-parleurs. La généralisation se fait mathématiquement par les opérations matricielles [16].

Transaural audio

Le transaural audio consiste à utiliser un signal binaural, et à le faire passer au travers d'un ensemble de filtres pour corriger les contributions croisées des haut-parleurs, de façon à ce que chaque oreille n'entende que ce qu'elle aurait entendu en binaural. Les filtres utilisés font apparaître deux blocs fonctionnels : un filtre en treillis et un filtre en peigne [16]. Le grand défaut de cette technique est qu'il faut que la tête soit positionnée à un endroit très précis, équidistant des haut-parleurs, et surtout ne doit pas en bouger. De plus, il ne faut pas de réverbération naturelle dûe à la salle (ce qui, en pratique, est approché en éloignant le plus possible les haut-parleurs des murs) [16].

2.6 Traitements modifiant le timbre

Les traitements portant sur le timbre existent en très grand nombre. Nous allons en aborder quelques uns, parmi les plus “classiques”, à savoir : le chorus, le *flanger*, le *phasing*, la distorsion, les modifications de l'enveloppe spectrale (dont le changement de formants), l'équalisation, le filtrage en peigne, la synthèse croisée, le *morphing*, l'interpolation spectrale, la robotisation et enfin l'effet wha-wha.

2.6.1 *Flanger*

Définition

Le flanger (ou *flanging*) est un effet de modulation de la fréquence d'un signal sonore. A l'origine, il provenait de la lecture simultanée d'un même signal sur deux appareils analogiques différents : la très légère différence de vitesse de défilement est à l'origine de cet effet. Numériquement, il s'obtient à l'aide d'une ligne à retard, et est basé sur le même principe que le chorus (cf. 2.6.2). Ces deux effets sont très légèrement différents du **phasing** (cf. 2.6.3). L'effet perceptif est assez saisissant, et certains se plaisent à y entendre le son d'un avion à réactions.

Le nom de *flanging* vient de l'anglais “*flange*”, qui signifie “bord”, à savoir le bord de la bobine sur laquelle il faut appuyer légèrement pour créer l'effet.

Fonctionnement

L'effet de flanging s'obtient en dupliquant le signal d'entrée à l'aide d'une ligne à retard, dont le temps de retard $\tau_{flanging}$ varie avec le temps (c'est un oscillateur à basse fréquence, OBF, ou *LFO* pour *Low Frequency Oscillator*), cf. fig. 2.27.

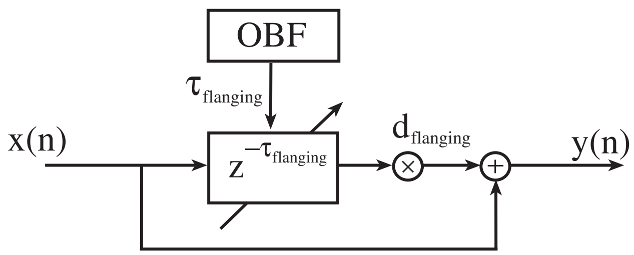

Fig. 2.27: *Diagramme du flanger.*

Le délai est trop court pour être entendu comme tel (inférieur à 50 millisecondes, alors que l'oreille humaine entend un délai comme tel à partir de 50 à 70 millisecondes). Par contre, le retard crée un effet de filtrage fréquentiel, en créant un série de nœuds dans le spectre en fréquences du signal. Les nœuds correspondent à des fréquences où le module du spectre est nul (cf. fig.2.28), et sont espacés régulièrement.

Ces nœuds résultent d'interférences destructives entre les deux signaux additionnés, lorsque deux composantes sinusoïdales s'additionnent avec quasiment le même module et en opposition de phase. La profondeur $d_{flanging}$ de l'effet correspond au taux de mixage de l'effet avec le son original. Pour une profondeur nulle, auncun signal n'est ajouté, donc aucun nœud n'apparaît. Pour une profondeur maximale (égale à 1), les nœuds descendent jusqu'à atteindre 0 dans la réponse en fréquence (soit $-\infty$ dB). Le temps de retard varie selon un oscillateur

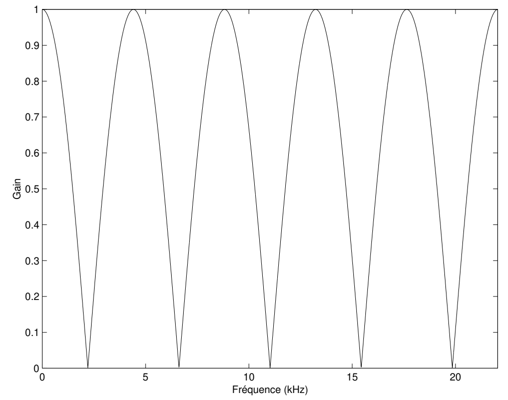

Fig. 2.28: *Réponse en fréquence du filtrage en peigne, retard de* 0.23 *ms.*

à basses fréquences. Ce peut être un oscillateur sinusoïdal, triangulaire, logarithmique (cf. fig.2.29).

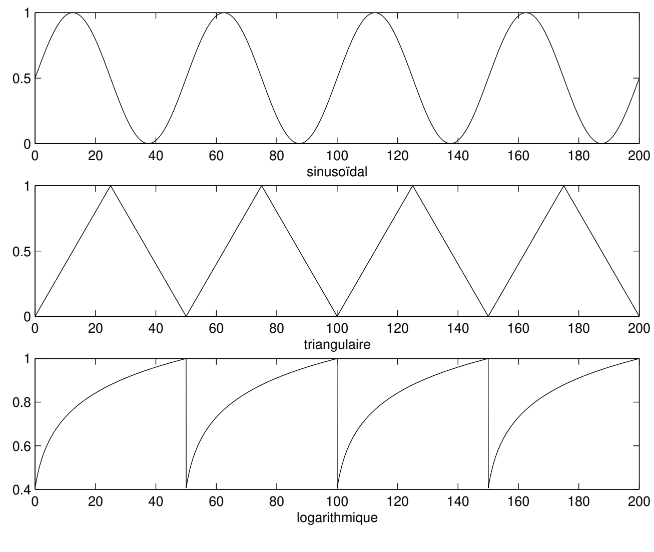

Fig. 2.29: *Types d'oscillations généralement utilisées pour l'effet de* flanging : *sinusoïdal, triangulaire, logarithmique. Le temps est indiqué en ms.*

Les paramètres de contrôle de cet effet concernent principalement la forme d'onde. Elle se définit par l'amplitude de la forme d'onde $b_{flanging}$ (ou profondeur de balayage), qui s'additionne au délai fixe $\tau_{flanging}$ pour donner le délai variable qui contrôle la ligne à retard ; par la profondeur ou le taux de mixage $d_{flanging}$ variant entre 0 et 1 et faisant varier l'acuité des nœuds ; par la forme de l'oscillateur basse fréquence ; par sa fréquence, ou plutôt son inverse, le taux $t_{flanging}$ et par le gain de réinjection $G_{flanging}$. Ce gain peut être positif ou négatif (entre -1 et 1), de façon à pouvoir supprimer où ajouter de l'effet.

Le paramètre $b_{flanging}$ détermine l'espacement entre les nœuds dans la réponse en fréquences. Une faible valeur impliquera une faible variance pour le délai, alors qu'une grande valeur impliquera une grande variance pour le délai, rendant l'effet de plus en plus perceptible.

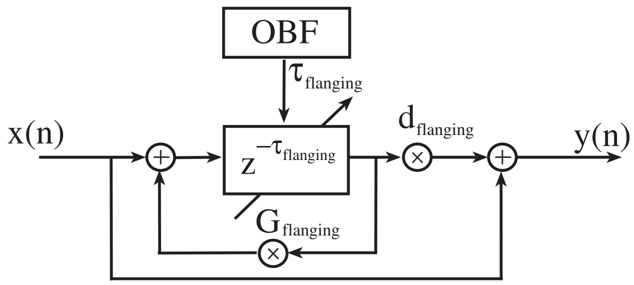

Fig. 2.30: *Diagramme du flanger avec réinjection*

Compléments d'information

Pour créer un flanger stéréophonique, on utilise deux flangers en quadrature de phase (déphasage de $\pi/2$ entre les deux formes de l'oscillateur basses fréquences).

Les différences principales entre le flanger et le chorus (cf. 2.6.2) sont l'absence de réinjection pour le chorus et la plage de valeurs possible pour le temps de délai (20 à 30 ms pour le *flanger*, contre 1 à 10 ms pour le chorus).

Rappel des paramètres de l'effet

Les paramètres permettant de caractériser cet effet sont :

- $d_{flanging}$ la profondeur (le taux de mixage), entre 0 et 1 ;
- $\tau_{flanging}$ le délai minimum de l'effet (20 à 30 ms);
- $b_{flanging}$ la profondeur du balayage (amplitude de la forme d'onde à basse fréquence, quelques ms) ;
- $t_{flanging}$ la fréquence de l'oscillateur basse fréquence (quelques Hz);
- $G_{flanging}$ le taux de réinjection.

2.6.2 Chorus

Définition

Le "chorus" est un choeur, un groupe de chanteur. L'effet chorus transforme donc le son d'un instrument de telle sorte qu'il semble à l'auditeur que plusieurs instruments jouent en même temps la même partition. Cela ajoute de l'épaisseur au son.

Fonctionnement

L'algorithme utilisé pour l'effet de chorus est très simple, et partagé (à une éventuelle complication près) par le flanger (cf. 2.6.1).

Lorsque deux personnes jouent en même temps la même ligne mélodique, ils ne sont pas tout à fait à l'unisson, ni parfaitement accordés. Le retard entre les deux instruments se modélise avec une ligne à retard. Lorsque le retard de cette ligne varie au cours du temps, on reproduit l'effet de léger désaccord entre le son original et le son traité par transposition du son original. Le diagramme du chorus est le suivant (cf. fig. 2.31) :

Il ne reste plus qu'à mixer le son original et le son traité pour obtenir le chorus.

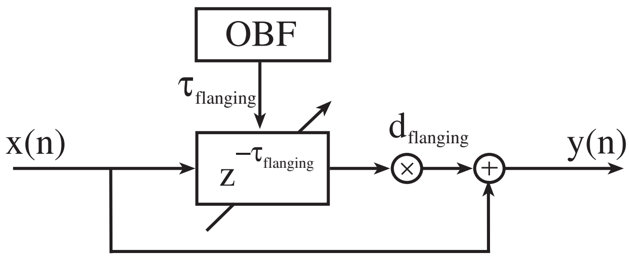

Fig. 2.31: *Diagramme du chorus.*

La structure proposée est quasiment la même que celle du flanger. Les différences principales entre le flanger et le chorus sont l'absence de réinjection pour le chorus et la plage de valeurs possible pour le temps de délai (20 à 30 ms pour le *flanger*, contre 1 à 10 ms pour le chorus).

La quantité de changement de hauteur dépend de la forme d'onde de l'oscillateur basses fréquences : les zones à fortes variations de la forme d'onde produisent beaucoup de changements de hauteur, alors que les zones de faibles variations impliquent peu de changements de hauteur. Aussi, en augmentant la fréquence de l'oscillateur, on augmente les changements de hauteur.

Intérêt de cet effet

Cet effet permet d'enrichir un son en donnant la sensation de plusieurs instrumentistes jouant simultanément la même partition. L'utilisation de plusieurs chorus simultanés rend l'effet encore plus intéressant. il remplace à sa manière les méthodes des luthiers par exemple, qui doublaient les cordes (guitare 12 cordes, mandoline, banjo) afin de créer cet effet chorus.

Compléments d'information

Pour créer un chorus multiple (par exemple stéréophonique, cf. fig.2.32), il suffit de multiplier les étages de chorus, et de définir pour chaque étage le taux de mixage, la forme de l'oscillateur, les déphasages.

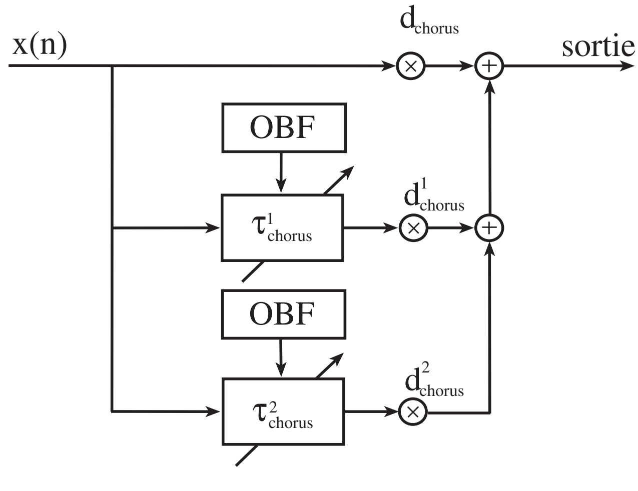

Fig. 2.32: *Diagramme du chorus stéréophonique.*

Rappel des paramètres de l'effet

Les paramètres permettant de caractériser cet effet sont :

- d_{chorus} la profondeur (le taux de mixage), entre 0 et 1 ;
- τ_{chorus} le délai minimum de l'effet (1 à 10 ms);
- b_{chorus} la profondeur du balayage (amplitude de la forme d'onde à basse fréquence, quelques ms) ;
- t_{chorus} la fréquence de l'oscillateur basse fréquence (quelques Hz);
- G_{chorus} le taux de réinjection.

2.6.3 *Phaser*, décalage de phase (phase shifting/phasing)

Définition

Le *phasing* consiste à effectuer un filtrage du spectre du son en créant des nœuds, c'est-à-dire donnant une valeur nulle au module du spectre pour des fréquence données. Le flanger (cf. 2.6.1) et le chorus (cf. 2.6.2) sont des cas particulier du *phaser*. Le *phasing* est plus général que le *flanging* et le chorus en ceci que les espacement entre les noeuds qui apparaissent dans le spectre de fréquence du signal est variable pour le *phasing*, et constant pour le *flanging* et le chorus.

Fonctionnement

Les nœuds que l'on désire créer dans le spectre d'amplitude du son sont le plus souvent implémentés à l'aide de filtres passe-tout (cf. fig.2.33).

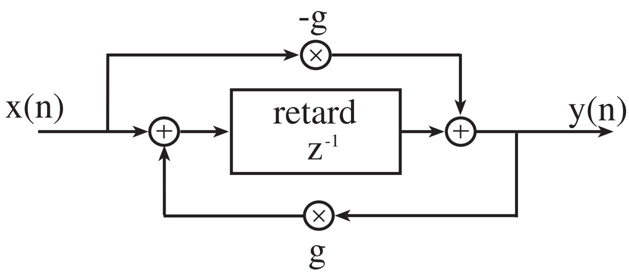

Fig. 2.33: *Diagramme du filtre passe-tout.*

Ces filtres ne modifient en rien le spectre d'amplitude ; par contre, ils introduisent des déphasages (modifications dans le spectre de phase). Le gain est unitaire pour toute les fréquences, mais la réponse en phase est non-linéaire. De ce fait, lorsque l'on va additionner le son original avec celui contenant des retards de phase, on va obtenir un son dont le spectre d'amplitude contiendra des trous aux noeuds désirés, par addition de composantes en opposition de phase.

Tout comme pour le chorus et le *flanger*, on définit la profondeur de l'effet $d_{phasing}$ comme étant le taux de mixage de l'effet avec le son original (entre 0 et 100 %). La plage de balayage $b_{phasing}$ correspond à la largeur en fréquence (Hz) sur laquelle chaque nœud peut se déplacer autour d'une valeur fixe (parfois donnée par l'utilisateur). Le gain de réinjection $r_{phasing}$

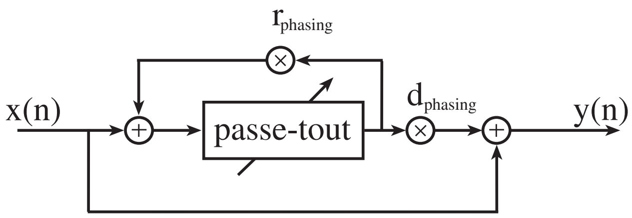

Fig. 2.34: *Diagramme du phasing.*

correspond encore à la même chose que pour les effets précédemment décrits. La fréquence de balayage $t_{phasing}$ correspond à la fréquence à laquelle les nœuds se déplacent (nombre de déplacement aller-retour par seconde).

Compléments d'information

Un *phaser* stéréophonique se réalise de même qu'un chorus ou un *flanger* stéréophonique, à l'aide de deux filtres passe-tout, et un mixage de gain croisé entre les deux lignes (cf. fig.2.35).

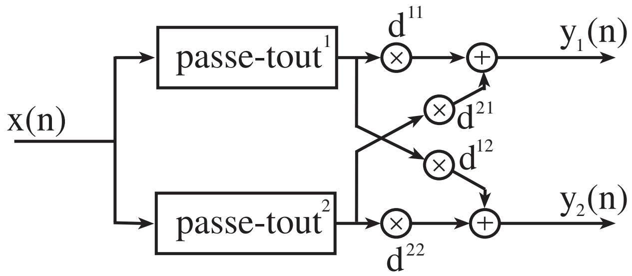

Fig. 2.35: *Diagramme du phasing stéréophonique*

Rappel des paramètres de l'effet

Les paramètres permettant de caractériser cet effet sont :

- $d_{phasing}$: profondeur de l'effet (entre 0 et 1) ;
- $b_{phasing}$: largeur de la plage de balayage (quelques Hz) ;
- $r_{phasing}$: gain de réinjection (entre 0 et 1) ;
- $t_{phasing}$: fréquence de balayage (quelques Hz).

2.6.4 Distorsion

Définition

La distorsion d'un signal audio est un effet indésirable dans une installation haute-fidélité, et très désirable dans des situations musicales, du fait qu'elle ajoute des harmoniques au signal.

Fonctionnement

La distorsion s'obtient en faisant circuler le signal à travers une fonction de transfert non-linéaire :

$$y(n) = f(x(n)) \qquad \text{avec } f \text{ non-linéaire} \tag{2.44}$$

La fonction de transfert f, non linéaire, fait apparaître dans le spectre des composantes qui n'y existaient pas.

Distorsion de sons harmoniques

Si le signal $x(n)$ est harmonique, il peut être décomposé en une somme de sinusoïdes $x(n) = \sum_{i=-p}^{p} e^{j\omega_i n}$. Si la fonction f se développe en série de Taylor-MacLaurin autour de 0 :

$$f(u) = f(0) + u\, f'(0) + \frac{u^2\, f''(0)}{2!} + \frac{u^3\, f'''(0)}{3!} + \ldots = \sum_{i=0}^{\infty} \frac{u^i}{i!}\frac{d^i f}{du^i} \tag{2.45}$$

alors on remarque que le signal $x(n)$ passé à travers la fonction f apparaît comme une somme pondérée de ses puissances entières, on obtient donc :

$$y(n) = \sum_{i=0}^{\infty} \alpha_i (x(n))^i \tag{2.46}$$

avec $(x(n))^i = \sum_{-p \leq j(k) \leq p} e^{j(\omega_{j(0)} + \omega_{j(1)} + \ldots + \omega_{j(i-1)})}$.

Pour un signal composé de sinusoïdes réelles, les termes ω_i apparaîssent par paires de signes opposés. Par suite, le signal de sortie $y(n)$ contient des composantes sinusoïdales dont les fréquences sont toutes les sommes et les différences des multiples des fréquences originales ω_i. Ainsi, un couple de sinusoïdes $x(n) = \cos(\omega_1 n + \Phi_1) + \cos(\omega_2 n + \Phi_2)$ donnera en sortie un signal qui en général contiendra les fréquences ω_1, ω_2, $(\omega_1 + \omega_2)$, $(\omega_1 - \omega_2)$, $2\,\omega_1$, $2\,\omega_2$, etc. Si $x(n)$ est périodique, alors $y(n)$ est également périodique, de même période, mais de contenu spectral différent.

Une manière d'implémenter cette distorsion non linéaire consiste à utiliser les **polynômes de Chebyshev**. Ces polynômes T_i possèdent la propriété suivante :

$$T_i(\cos\theta) = \cos(i\theta) \tag{2.47}$$

et se calculent par récurrence selon la formulation :

$$\begin{aligned} T_0(x) &= 1 && (2.48)\\ T_1(x) &= x && (2.49)\\ T_{n+1}(x) - 2x\, T_n(x) + T_{n-1}(x) &= 0 && (2.50) \end{aligned}$$

La propriété 2.47 est très intéressante, car elle permet de doser le degré de distorsion obtenu par le système. En choisissant pour $f(u)$ une somme pondérée de polynômes de Chebyshev, on fait apparaître dans le signal $f(\cos(\omega n))$ une somme pondérée de certaines fréquences multiples de ω. On peut ainsi contrôler le nombre et les amplitudes des harmoniques qui apparaîssent après transformation (cf. [1], [10]). L'un des intérêts de ces fonctions est que le signal de sortie est limité en fréquences : auncun repliement n'est possible.

Puisque la fonction $f(u)$ est non-linéaire, la distorsion générée par son application au signal $x(n)$ dépend fortement de l'amplitude maximale du signal : pour un signal de très faible niveau, $f(u)$ peut être considérée comme linéaire (si son terme constant est nul), et la distorsion devient faible. Dans le cas contraire, le niveau du son de sortie n'augmente pas linéairement avec le niveau d'entrée, mais le spectre s'enrichit. Cette sensibilité de la distorsion à l'amplitude peut modéliser les caractéristiques d'instruments acoustiques. En effet, on peut jouer d'un instrument acoustique de façon plus "dure", par exemple en pinçant une corde de guitare avec force, en soufflant de façon stridante dans un saxophone, ou en frappant sur un tambour avec intensité, ce qui enrichit le spectre en hautes fréquences. Avec la distorsion non-linéaire, nous pouvons simuler cet effet en faisant passer le signal dont l'amplitude varie à travers une fonction non-linéaire, obtenant un signal de sortie dont le spectre varie.

Des études (cf. [10]) ont été menées pour corriger l'intensité du son de sortie, de manière à la normaliser. Ainsi, on peut s'assurer qu'en plus de l'enrichissement spectral produit par la distorsion, l'intensité de sortie est identique à l'intensité d'entrée.

Compléments d'information

A partir du moment où l'on utilise des fonctions autres que les polynômes de Chebyshev, plus rien n'assure que le son produit soit harmonique lorsque le son d'entrée l'est. Ainsi, des combinaisons inharmoniques de partiels ainsi que de strucutres formantiques peuvent être obtenues (cf. [1]). La fonction de transfert peut être donnée graphiquement, évoluer dans le temps. Dans ces cas, il est nécessaire de s'assurer qu'aucun repliement n'a lieu, à l'aide d'un filtrage passe-bas à la demi-fréquence d'échantillonnage.

2.6.5 Modifications de l'enveloppe spectrale

Modifier l'enveloppe spectrale peut se faire de plusieurs manières : soit en l'étirant ou la contractant, soit en la décalant. Nous ne parlerons pas ici des autres manières de faire à partir de deux sons, telle l'interpolation spectrale ou la synthèse croisée abordées plus loin (cf. 2.6.9).

Décalage de l'enveloppe

Le décalage de l'enveloppe spectrale consiste à décaler de d_{spec} Hz l'enveloppe du spectre de magnitude vers les basses fréquences ou vers les hautes fréquences.

Décaler l'enveloppe du spectre de magnitude nécessite quelques précautions. Quelque soit le modèle utilisé pour l'estimation de l'enveloppe spectrale (LPC, cepstre) et pour la modification (modèle additive, vocodeur de phase, modèle soustractif), un décalage vers les basses fréquences revient à une troncature de l'enveloppe, un décalage vers les hautes fréquences nécessite un troncature à la demi-fréquence d'échantillonnage, pour éviter le repliement.

Pour le vocodeur de phase, si l'on désire décaler d'un nombre entier de paniers de fréquence, aucune interpolation n'est nécessaire. Par contre, des lorsque l'on veut décaler l'enveloppe d'un nombre réel de paniers de fréquences, on peut interpoler l'enveloppe (par interpolation linéaire, ou par splines). Cela dit, l'oreille n'est pas assez sensible pour entendre de si fines variations sur l'enveloppe, sauf si celle-ci est très spéciale (par exemple un filtre passe-bande très étroit).

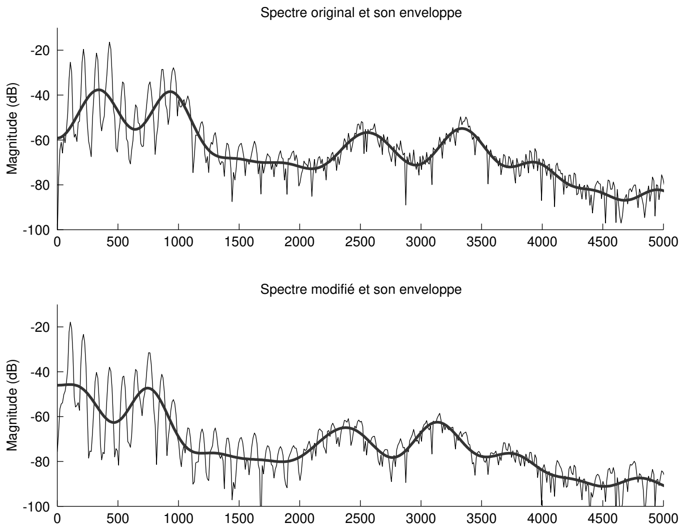

Fig. 2.36: *Décalage de l'enveloppe spectrale : spectre et son enveloppe avant (fig. haut) et après (fig. bas) décalage de* $-200\ Hz$.

Contraction/étirement de l'enveloppe

La contraction ou l'étirement de l'enveloppe spectrale consiste à modifier son échelle par un facteur correctif γ_{spec}, constant ou non. Les précautions énoncées précédemment concernant les hautes fréquences sont une fois de plus de mise.

La nouvelle enveloppe est calculée par interpolation de l'enveloppe du spectre original (interpolation linéaire ou splines).

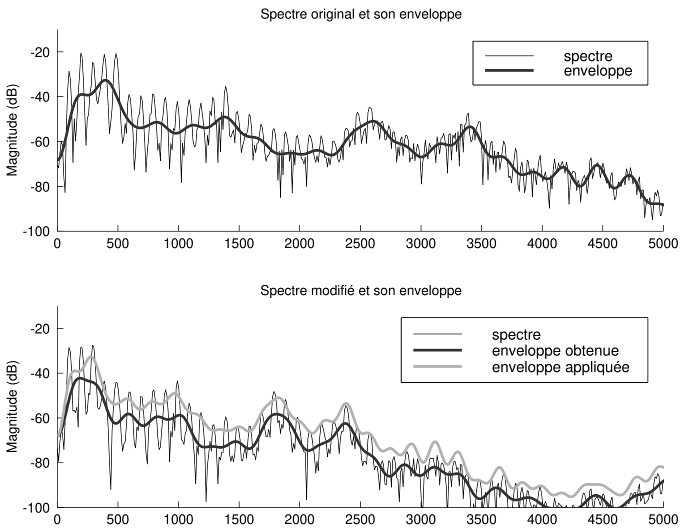

Fig. 2.37: *Contraction de l'enveloppe spectrale : spectre et son enveloppe avant (fig. haut) et après (fig. bas) étirement d'un facteur* $\gamma_{env} = 0.7$.

Intérêt de cet effet

Cette transformation permet d'obtenir entre autres un effet "Donald duck", du fait que les formants ne correspondent plus, après transformation, à la manière dont un être humain aurait produit le son.

Compléments d'information

L'étirement ou la contraction de l'enveloppe spectrale peuvent s'obtenir de manière involontaire (changement de hauteur ne conservant par l'enveloppe spectrale, cf. 2.4.2) ou volontairement, par les modèles additifs, vocodeur de phase ou par changement d'échelle temporelle PSOLA puis ré-échantillonnage. L'utilisateur indique le type de transformation qu'il veut appliquer : décalage en linéaire ou logarithme, multiplication (étirement/contraction fréquentielle).

2.6.6 Décalage du spectre, avec ou sans duplication

Définition

Le décalage du spectre consiste à prendre un spectre de magnitude $\rho_a(f)$, et à le décaler en fréquences $\rho_s(f) = \rho_a(f - \delta)$ de $\delta\ Hz$.

Fonctionnement

Lorque l'on dispose d'une représentation additive du signal, en décaler le spectre consiste à en décaler toutes les composantes : l'enveloppe spectrale du résidu, celle du transitoire, et chacune des harmoniques. Cela se réalise très aisément. Pour un son représentation avec le vocodeur de phase, c'est un peu plus délicat car le décalage seul des magnitudes ne suffit pas, il faut aussi recalculer les phases. Lorsqu'on reste dans le domaine temporel, le modulateur à bande latérale unique (BLU) permet de réaliser ce décalage aisément. Par contre, dès lorsque l'on utilise le modulateur en anneau, on réalise une copie du spectre, et chacune des deux copies est décalée en sens inverse, l'une vers les fréquences positives, l'auitre vers les fréquences négatives.

Dans tous les cas, on fera attention à la plage de validité des fréquences (de même que pour les décalages et étirements/contractions de l'enveloppe spectrale) : les fréquences négatives obtenues doivent être éliminées, de même que celles supérieures à la demi-fréquence d'échantillonnage.

Intérêt de cet effet

Le décalage simple du spectre permet, pour de très faibles décalages de modifier uniquement le spectre (et donc le timbre) : lorsque la fréquence fondamentale n'est pas trop différente de l'écart entre deux partiels successifs, pour peu qu'il soient bien équi-répartis, la hauteur entendue est celle de l'écart entre les harmoniques. Soient $f_{a,i} = if_0$ les fréquences des harmoniques analysées. Pour δ petit, le son composé des partiels de fréquences $f_{s,i} = if_0 + \delta$ aura pour hauteur f_0. Par contre, dès que δ atteint quelques Hertzs, ce n'est plus vrai, le son devient inharmonique, dissonnant, le timbre est modifiée et la hauteur n'est plus perçue.

C'est encore plus sensible pour les modulations en anneau, du fait que deux composantes proches à la place de chaque harmonique provoquent des battements.

Rappel des paramètres de l'effet

Les paramètres permettant de caractériser cet effet sont :

- le décalage δ en Hz ;
- le type de traitement utilisé (pour la modulation en anneau, le spectre est dupliqué).

2.6.7 Equaliseur

Définition

Un équaliseur est un outil de transformation du spectre de magnitude d'un signal. Etant donné un ensemble de filtres, chacun s'appliquant à une bande de fréquences précise, l'équalisation consiste à donner des gains à chaque filtre, afin d'amplifier ou diminuer chaque bande de fréquence.

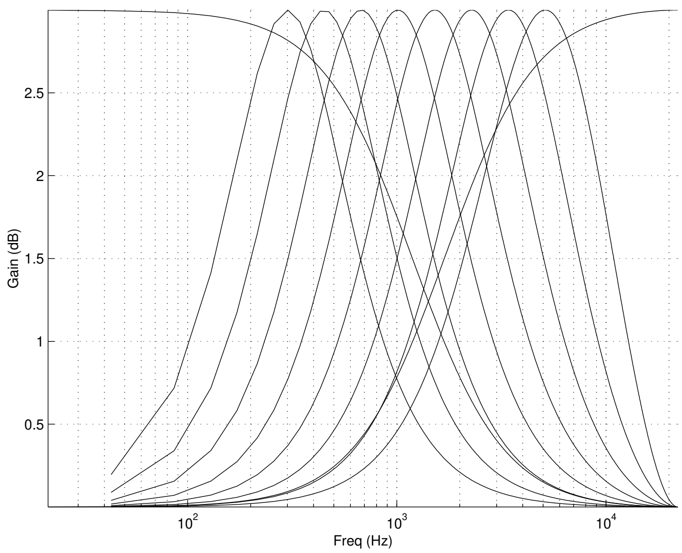

Fig. 2.38: *Equaliseur : filtres à étage aux extrémités et filtres en pic au milieu.*

Fonctionnement

Le spectre de magnitude est décomposé en bandes de fréquences à l'aide de filtres : des filtres à étages aux extrémités, et des filtres en pics entre les deux extrémités.

Chaque filtre est définit par son type (passe-bas ou passe-haut, en pic), son ordre (premier ou second), sa fréquence de coupure f_c^i, son gain G_i. Les filtres en pic (ou passe-bande) possèdent un paramètre supplémentaire, la largeur de bande f_b. Des filtres passe-bande couramment utilisés sont les filtres à Q constant : Q est le facteur qualité, obtenu par le rapport $Q = \frac{f_b}{f_c}$. Conserver Q constant revient à augmenter la largeur de bande lorsque la fréquence centrale du filtre augmente. En échelle logarithmique, cela donne des filtres à largeur de bande constante (en $log(Hz)$).

Tous ces paramètres sont manipulés à l'aide de potentiomètres (solution matérielle) ou d'une interface graphique faite de potentiomètres (solution logicielle).

Rappel des paramètres de l'effet

Les paramètres permettant de caractériser cet effet sont :

- le type de chaque filtre (passe-bas ou passe-haut, en pic) ;
- l'ordre de chaque filtre (premier ou second) ;
- la fréquence de coupure f_c^i du filtre ;
- la largeur de bande fréquentielle f_b^i du filtre en pic ;
- le gain G_i du filtre.

2.6.8 Filtrage en peigne

Définition

Le filtrage en peigne consiste à filtrer le son avec une sorte de peigne harmonique, soit pour augmenter, soit pour diminuer les modules du spectre aux multiples d'une pseudo-fondamentale.

Filtrage à réponse impulsionnelle finie

Soit $x(n)$ un signal entrant dans une ligne à retard de longueur L. Si l'on ajoute le signal en entrée de la ligne à retard au signal en sortie, avec g_x le gain, on obtient le signal :

$$y(n) = x(n) + g_x\ x(n-L) \tag{2.51}$$

correspondant au filtrage en peigne à réponse impulsionnelle finie, de fonction de transfert :

$$H(z) = 1 + g_x\ z^{-L} \tag{2.52}$$

Pour une fréquence d'échantillonnage F_e, le délai temporel (ou temps de retard) induit par le passage à travers la ligne à retard est alors $\tau = \frac{L}{F_e}$.

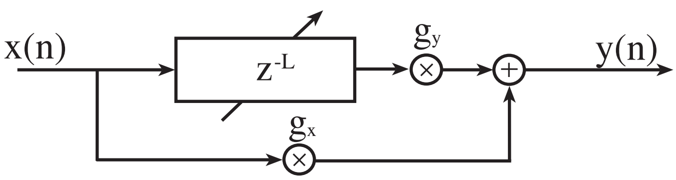

Fig. 2.39: *Diagramme du filtre RIF en peigne*

Dans le cas où le gain g_x est positif, le filtre amplifie toutes les fréquences multiples de $\frac{1}{\tau} = \frac{F_e}{L}$. et atténue toutes les fréquences entre ces multiples. Le gain varie en amplitude entre $1 - g_x$ et $1 + g_x$ (et en décibels entre $20\log_{10}(1 - g_x)$ et $20\log_{10}(1 + g_x)\ dB$). La réponse en fréquence de ce filtre ressemble à un peigne, les trous dans le spectre sont de plus en plus prononcés lorsque le gain $|g_x|$ augmente en valeur absolue (cf. fig. 2.40).

Selon le délai du filtre, le son produit sera perçu comme une coloration (filtrage, transformation du timbre, pour des τ petits) ou comme un écho (τ grand).

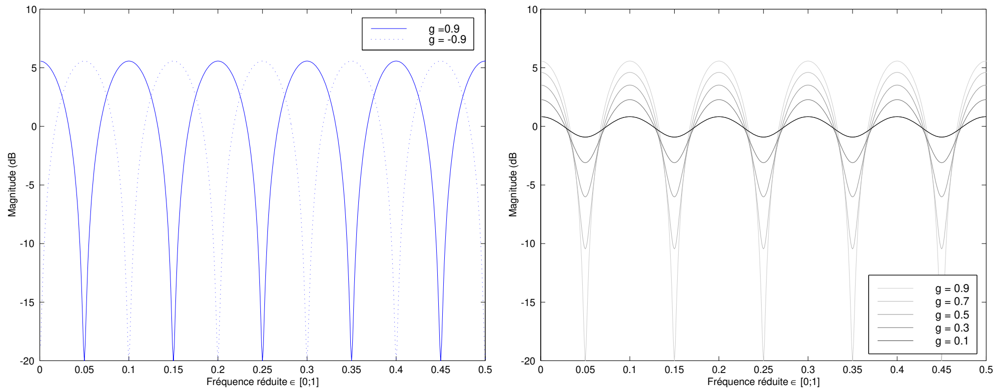

Fig. 2.40: *Réponse en fréquence d'un filtre en peigne FIR : à gauche pour $g_x = 0.9$ et $g_x = -0.9$, à droite pour $g_x = 0.1$, $g_x = 0.3$, $g_x = 0.5$, $g_x = 0.7$, $g_x = 0.9$.*

Filtrage à réponse impulsionnelle infinie (RII)

Soit $x(n)$ un signal entrant dans une ligne à retard de longueur L. Si l'on réinjecte le signal en sortie de la ligne à retard dans la ligne à retard, avec g_y le gain, et que l'on ajoute ce signal au son d'entrée de la ligne à retard, on obtient un signal :

$$y(n) = \beta\ x(n) + g_y\ y(n-L) \tag{2.53}$$

correspondant au filtrage en peigne à réponse impulsionnelle infinie (RII), de fonction de transfert :

$$H(z) = \frac{\beta}{1 - g_y\ z^{-L}} \tag{2.54}$$

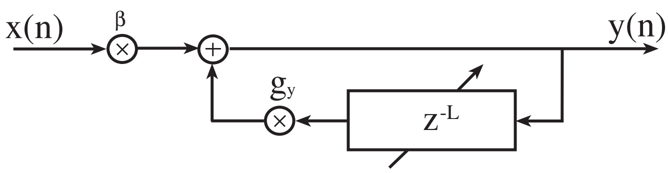

Fig. 2.41: *Diagramme du filtre RII en peigne*

La boucle de réinjection implique une réponse impulsionnelle infinie. Le filtre n'est stable que pour $|g_y| \leq 1$. Dans le cas où le gain g_y est positif, le filtre amplifie toutes les fréquences multiples de $\frac{1}{\tau} = \frac{F_e}{L}$, et atténue toutes les fréquences entre ces multiples. Le gain varie en amplitude entre $\frac{1}{1-g_y}$ et $\frac{1}{1+g_y}$ (soit en décibels entre $-20\log_{10}(1+g_y)$ et $-20\log_{10}(1-g_y)\ dB$). Les pics sont de plus en plus étroits et amplifiés lorsque le gain $|g_y|$ augmente en valeur absolue (cf. fig. 2.42).

Filtre en peigne universel

Il s'agit de la combinaison des filtres RIF et RII. Le signal en sortie d'un filtre en peigne universel est :

$$y(n) = \beta\ x(n) + g_x\ x(n-L) + g_y\ y(n-L) \tag{2.55}$$

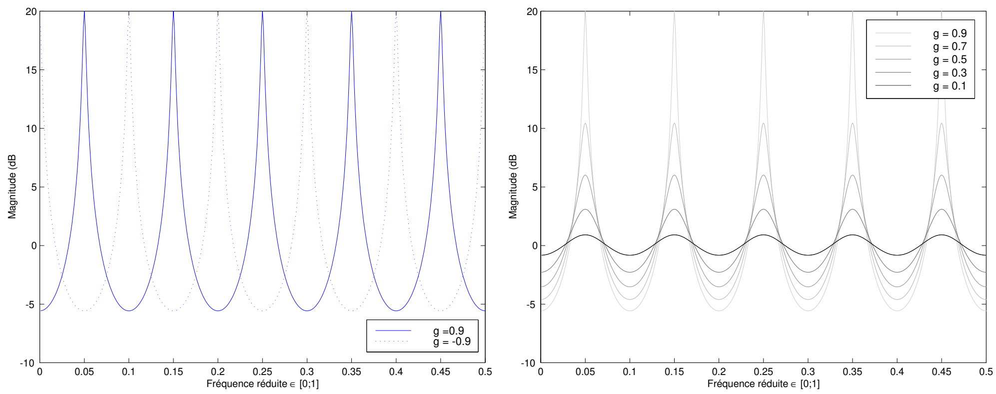

Fig. 2.42: *Réponse en fréquence d'un filtre en peigne FIR : à gauche pour $g_y = 0.9$ et $g_y = -0.9$, à droite pour $g_y = 0.1$, $g_y = 0.3$, $g_y = 0.5$, $g_y = 0.7$, $g_y = 0.9$.*

Sa fonction de transfert est :

$$H(z) = \frac{\beta + g_x\ z^{-L}}{1 - g_y\ z^{-L}} \tag{2.56}$$

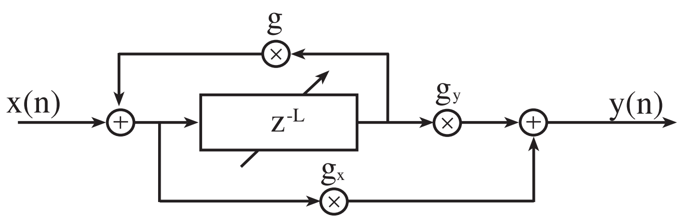

Fig. 2.43: *Diagramme du filtre en peigne universel*

Selon les valeurs des coefficients g, g_x et g_y, on obtient différentes configurations :

	g	g_x	g_y
RIF	X	0	Y
RII	1	X	0
passe-tout	a	$-a$	1
délai	0	0	1

Intérêt de cet effet

Cet effet de filtrage en peigne permet de donner une hauteur à un son (bruit) n'en ayant pas. En effet, l'ajout de pics équi-répartis revient à ajouter des harmoniques ; ce qui est le plus surprenant, c'est que si on filtre un bruit blanc par un filtre en peigne vers le bas, c'est-à-dire si l'on réalise des trous équi-répartis dans le spectre, une fois encore, on impose une hauteur clairement audible (plus qu'une simple coloration) au son.

Le filtrage en peigne permet aussi de supprimer des harmoniques, par exemple dans l'analyse additive, lorsque l'on a supprimé les harmoniques du résidu imparfaitement : un filtrage en peigne permet de "nettoyer" le résidu.

2.6.9 Synthèse croisée, *morphing*, interpolation spectrale

Définitions

Les effets de synthèse croisée, de *morphing* ou de mutation, et d'interpolation spectrale semblent correspondre à la même chose. En fait, même si certains résultats sonores peuvent se ressembler, la démarche n'est pas la même et les sons résultants (de manière générale) non plus.

Mutation La mutation correspond à la création d'un son à partir des spectres d'amplitude ρ_1, ρ_2 et de phase Φ_1, Φ_2 de deux sons : le spectre d'amplitude de synthèse ρ_s et celui de phase Φ_s sont donnés arbitrairement, et le son est reconstruit par FFT inverse. Le modèle sous-jacent est le modèle temps-fréquence (vocodeur de phase). La transformée de Fourier résultant est donnée simplement par :

$$Y(f) = \rho_s(f).e^{j\Phi_s(f)} \tag{2.57}$$

La représentation de Fourier obtenue n'est pas forcément valide. De ce fait, pour imposer une représentation valide, on calcule la transformée de Fourier inverse, puis on lui applique la transformée de Fourier : les spectres d'amplitude et de phase sont alors valides.

$$Y(f) = \mathcal{F}\left(\mathcal{F}^{-1}\left(\rho_s(f).e^{j\Phi_s(f)}\right)\right) \tag{2.58}$$

On peut par exemple prendre la magnitude du premier son et la phase du second (cf. fig. 2.44).

$$Y(f) = \mathcal{F}\left(\mathcal{F}^{-1}\left(\rho_1(f).e^{j\Phi_2(f)}\right)\right) \tag{2.59}$$

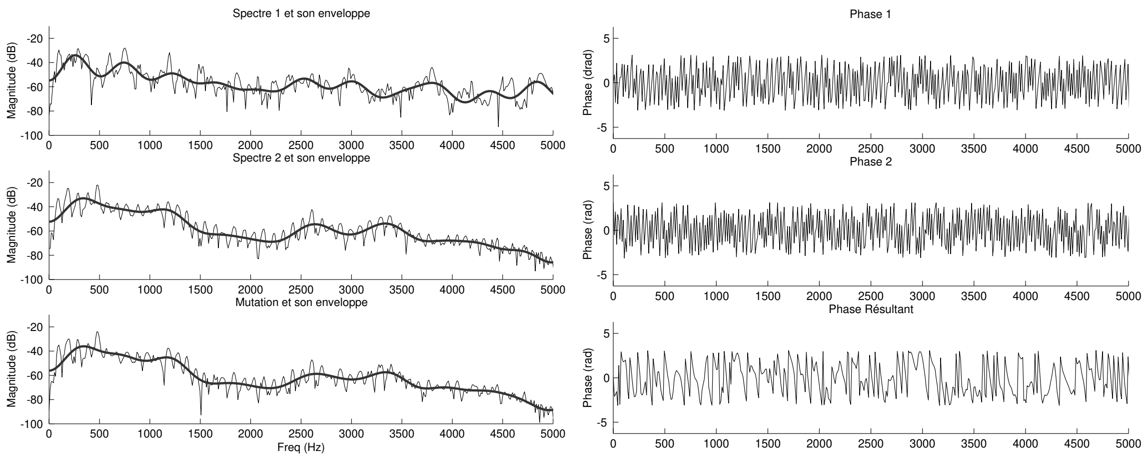

Fig. 2.44: *Mutation : le son résultant est obtenu par resynthèse à partir du spectre d'amplitude du son 1 (en haut à gauche) et du spectre de phase du son 2 (au milieu à droite). Le spectre de phase résultant (en bas à droite) après transformation de Fourier de la transformée de Fourier inverse est différent de celui du son 2, du fait que la représentation de Fourier imposée n'était pas valide.*

Synthèse croisée La synthèse croisée ou **effet vocodeur** (*vocoding*) consiste à prendre la source d'un son $e_1(n)$ et à lui appliquer le filtre du second $f_2(n)$. Les deux sons analysés sont représentés fréquentiellement comme le produit d'une excitation E avec un filtre H : $X_i(f) = E_i(f).H_i(f)$. On utilise ici le modèle source-filtre. Notons que E est la TFCT de l'excitation et $H(f) = \exp(C(f))$ est l'enveloppe spectrale obtenue à partir du cepstre ou de la LPC.

La transformée de Fourier du son résultant est donnée par :

$$Y(f) = E_1(f).H_2(f) \tag{2.60}$$

Cet effet et permet par exemple de faire parler un instrument de musique (principe des “vocodeurs” en musique électronique, de la *talk-box* : le son de la guitare est blanchi et passe par la cavité buccale où il est filtré). La hauteur provient alors du signal d'excitation.

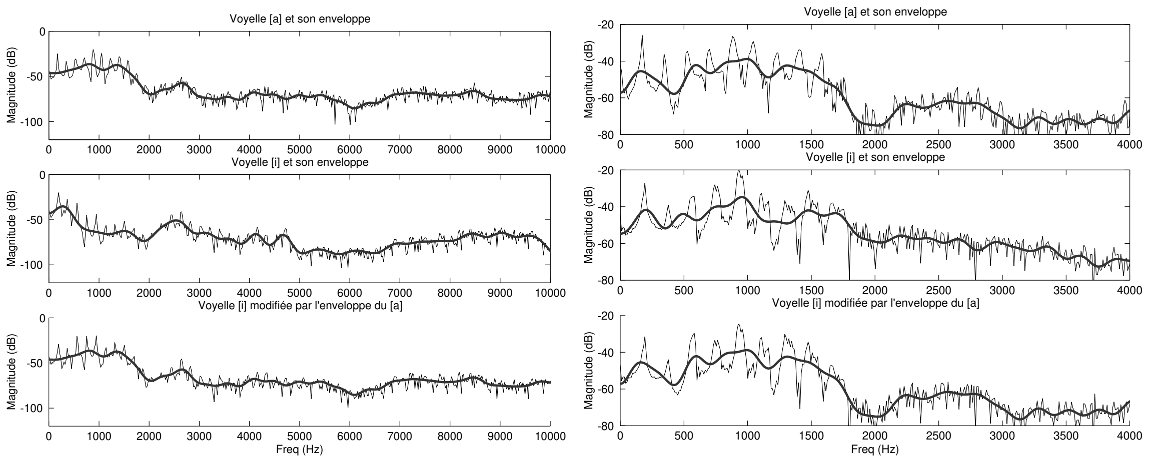

Fig. 2.45: *Synthèse croisée : application de l'enveloppe spectrale du [a] (fig. haut) à la voyelle [i] (fig. milieu) : l'enveloppe du spectre obtenu est identique à celle du [a] (fig. bas). Les figures de droite sont un zoom sur la partie basse fréquence, et montrent que la hauteur d'origine est conservée : seule l'enveloppe spectrale est modifiée.*

Morphing Le *morphing* quant à lui est définit différemment selon les auteurs. En effet, l'analogie avec le *morphing* visuel n'est pas aisé. Doit-on considérer un son qui progressivement, dans son évolution temporelle, glisse d'un timbre vers un autre de manière continue (une seule note), discrète (plusieurs notes, chacune différente de ses voisines) ou encore un timbre hybride, entre les deux ? Considérons que cet effet consiste à créer un son hybride à partir de plusieurs sons (2 ou plus)[2]. Par exemple, on peut créer un timbre situé entre le piano et la flûte, à partir d'une analyse additive des deux sons. Le timbre hybride est alors obtenu en interpolant les données spectrales des deux analyses : pour deux sons de même hauteur, les amplitudes et fréquences des pics ne sont pas identiques (le piano voit ses partiels dévier du multiple de la fondamentale au fur et à mesure que l'on monte dans les harmoniques, par exemple, ce qui n'est pas le cas de la flûte). Le son de synthèse, cf. 2.61, est obtenu par ajout de la partie harmonqiue, obtenue par moyenne ou interpolation entre ces valeurs, avec la

[2]auquel cas, on peut répondre aux autres définitions dès que l'on sait créer ces hybrides, par des évolutions temporelles des coefficients d'hybridation.

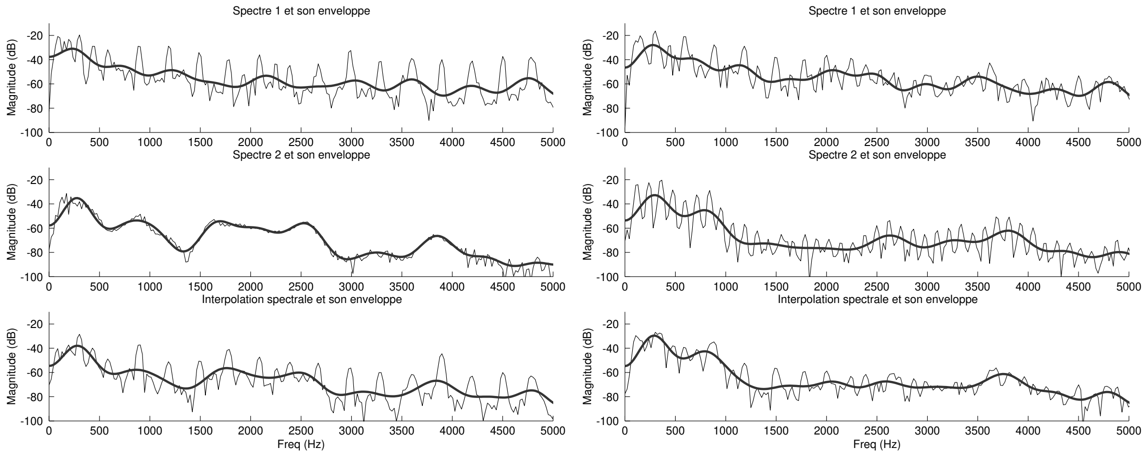

Fig. 2.46: *Interpolations spectrales (à deux instants différents) : spectre d'amplitude et enveloppe du son 1 (fig. haut), du son 2 (fig. milieu) et du son interpolé, avec* $e_1 = 0.6$*,* $e_2 = 0.4$*,* $s_1 = 0.1$ *et* $s_2 = 0.9$ *d'après (2.62).*

partie stochastique, obtenue par filtrage d'un bruit blanc à l'aide d'un filtre dont l'enveloppe spectrale est la moyenne des deux enveloppes des parties stochastiques des sons analysés.

$$x_s(n) = \tilde{x}(n) + \tilde{r}(n) \quad \text{avec} \quad \tilde{r}(n) = \alpha.r_1(n) + \beta.r_2(n) \tag{2.61}$$

Interpolation spectrale Un résultat proche peut s'obtenir par interpolation spectrale : les deux sons analysés sont représentés fréquentiellement comme le produit d'une excitation E avec un filtre H (modèle source-filtre). Le son de synthèse est obtenu par l'interpolation à la fois sur les excitations et sur les enveloppes spectrales :

$$Y(f) = [e_1 E_1(f) + e_2 E_2(f)] \, . \, [c_1 H_1(f) + c_2 H_2(f)] \tag{2.62}$$

L'interpolation spectrale se différencie du simple mixage, qui s'écrit dans le domaine fréquentiel :

$$Y(f) = k_1 E_1(f) H_1(f) + k_2 E_2(f) H_2(f) \tag{2.63}$$

Notons enfin que l'interpolation spectrale peut se faire sur M sons :

$$Y(f) = \left[\sum_{i=1}^{M} e_i E_i(f)\right] . \left[\sum_{i=1}^{M} c_i H_i(f)\right] \tag{2.64}$$

D'autre part, remarquons que lorsque $e_1 = 0$ et $s_2 = 0$ (de même lorsque $e_2 = 0$ et $s_1 = 0$), l'interpolation spectrale n'est ni plus ni moins qu'une synthèse croisée ; la transformée de Fourier résultant est donnée par :

$$Y(f) = (e_1.c_2) E_1(f).H_2(f) \tag{2.65}$$

ce qui correspond, à un facteur multiplicatif près, à la transformée de Fourier d'une interpolation spectrale, comme indiqué équation (2.60). La synthèse croisée est donc un cas particulier d'interpolation spectrale.

2.6.10 Robotisation

Définition

La robotisation s'applique à des sons de voix parlé ou chanté. Elle tranforme une voix humaine en voix métallique, robotique. Le traitement correspondant consiste à mettre à remplacer la voix par un train d'onde. La fréquence du train d'onde détermine la hauteur de la voix robotique.

Fonctionnement

La robotisation consiste à transformer la forme d'onde de la source originale en un train d'onde. La manière d'obtenir le train d'onde est simple : on modifie les phases d'un grain en les remplaçant par à 0 pour chaque FFT. Ceci revient à caler chaque sinusoïde de façon à obtenir un pic au centre du grain (cf. 2.47). La somme des sinusoïdes donne un pic principal au centre du tampon. L'information sur le contenu fréquentiel (les formants) proprement dit est donc conservé, mais pas l'information de déroulement de phase.

Entre deux fenêtres, le signal original est remplacé par des 0 (*zero-padding*). La hauteur de la voix de robot est déterminée par le pas entre deux fenêtres. En changeant la taille de la fenêtre, on sélectionne une plus ou moins grande partie du signal de départ : la voix est donc plus ou moins robotique selon la taille du grain. Une bonne taille est entre 64 (cf. fig. 2.47) et 1024 échantillons (cf. fig. 2.48). On remarque que plus la taille de grain augmente, plus des pics secondaires apparaissent. Ceci provoque dans le son un effet de deuxième hauteur : la hauteur originale du son réapparaît dans la voix de robot. Il est donc conseillé de prendre de petites fenêtres lorsqu'on ne désire pas conserver la hauteur originale.

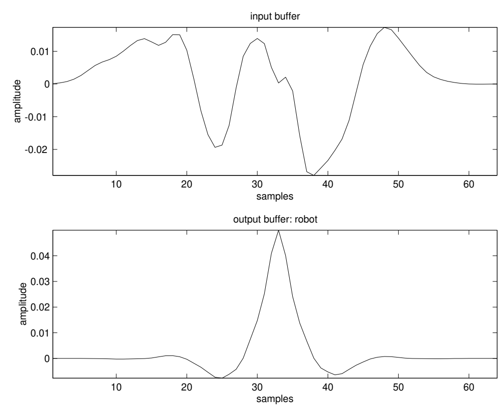

Fig. 2.47: *Robotisation appliquée sur un grain de 64 échantillons : grain d'origine (fig. supérieure) et après traitement (fig. inférieure). Un pic central (et un seul) est créé au centre du grain.*

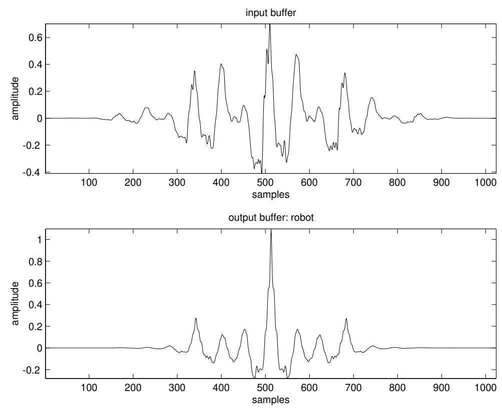

Fig. 2.48: *Robotisation appliquée avec un grain de 1024 échantillons : grain d'origine (fig. supérieure) et après traitement (fig. inférieure). Plusieurs pics sont créés, l'un au centre du grain (principal) et les autres de part et d'autre.*

Intérêt de cet effet

Tout d'abord, l'effet de robotisation est très utilisé dès lors que l'on veut donner une connotation "machine parlante" : les films de science-fiction y font largement appel lorsqu'ils donnent

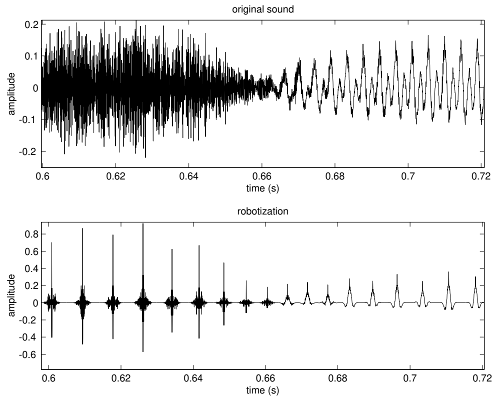

Fig. 2.49: *Robotisation appliquée avec un grain de 256 échantillons : forme d'onde d'origine (fig. inférieure) et forme d'onde du son après une robotisation ayant conservé la hauteur (fig. supérieure). La fréquence du train d'onde est identique à la fréquence perçue du signal d'origine.*

une voix à des machines s'adressant à des humains.

Si l'on fait varier le pas entre deux grains, c'est -à-dire la fréquence du train d'onde, la hauteur de la voix du robot change. On peut ainsi, en donnant en entrée à l'effet la hauteur perçue du son d'origine, obtenir une voix de robot de même hauteur ! D'autre part, on peut réinterpréter des phrases musicales en pilotant la hauteur du robot avec différents descripteurs extraits de la voix d'origine (robotisation adaptative).

Rappel des paramètres de l'effet

Les paramètres permettant de caractériser cet effet sont :

- la taille du grain (ou de la fenêtre) ;
- le pas entre deux grains (la période).

2.6.11 Wha-wha

Définition

L'effet wha-wha est utilisé principalement par les guitaristes. il consiste, par le biais d'une pédale contrôlant un rhéostat (ou potentiomètre), à passer d'un filtrage imitant le [u] à un filtrage imitant le [a].

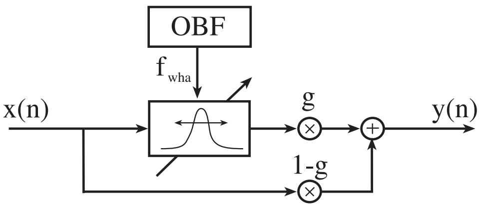

Fig. 2.50: *Diagramme de l'auto-wha.*

Fonctionnement

L'effet wha-wha consiste à modifier un filtre paramétrique (via sa fréquence centrale ou de résonance) passe-bande de petite bande passante. La fréquence de résonance est placée à la fréquence du premier formant de la voyelle [a] ou [u]. Le signal filtré est ensuite mixé avec le signal direct, comme indiqué fig. 2.50.

Compléments d'information

Lorsque le fréquence de résonance est donnée par un oscillateur basse-fréquence (comme indiqué sur le diagramme), il s'agit alors d'un auto-wha, ou effet wha-wha automatique. Un autre type d'effet wha-wha est celui qui se déclenche à l'attaque : lorsque le son s'enrichit en hautes fréquences (ce qui est le cas lors des attaques des notes de guitares, par exemple), on déclenche le mouvement rapide du filtre de la position donnant le son [u] vers le son [a]. Le retour à la position de repos (donnant le son [u]) se fait à une vitesse différente. De même que pour le compresseur ou l'expandeur, on utilise deux temps : le temps de montée ou d'attaque, et le temps de descente, ou de relaxation. On dit dans ce cas que l'effet wha-wha est sensitif.

Rappel des paramètres de l'effet

Les paramètres permettant de caractériser cet effet sont :

- la fréquence centrale du filtre résonant ;
- le type de contrôle sur cette fréquence (gestuel, automatique, sensitif à l'attaque) ;
- le temps de montée et celui de relaxation pour la wha-wha sensitive.

2.7 Traitements modifiant plusieurs paramètres à la fois

Une hypothèse de base à la représentation des sons par leurs indices perceptifs est que ces indices définissent un espace orthogonal. En réalité, ceci n'est jamais parfaitement vérifié. En effet, augmenter la sonie d'un son implique une modification du timbre et de la hauteur perçue. Modifier le spectre (la répartition en fréquence et l'amplitude des harmoniques) implique une modification du timbre mais aussi de la hauteur perçue. Un filtrage peut modifier à la fois la perception du timbre mais aussi de la localisation. Ces quelques exemples montrent les faiblesses de cette représentation. Toutefois, sans être trop exigeant, on peut considérer que l'espace perceptif est suffisant pour décrire les dimensions des sons.

Des traitements sonores, indépendammet des effets perceptifs décrits précédemment, peuvent modifier volontairement plusieurs paramètres perceptifs à la fois. Nous en présentons ici quelques uns, à savoir la transposition sans conservation de durée ni ne formants.

2.7.1 Transposition sans conservation de la durée ni des formants

Définition

La transposition sans conservation de la durée ni des formants consiste à modifier l'échantillonnage d'un signal, de façon à modifier la hauteur perçue. Ce traitement ne conserve ni les formants, ni la durée du signal.

Fonctionnement

Prenons l'exemple analogique des magnétophones à bande. De manière à pouvoir caler un morceau ou trouver une zone temporelle particulière, une fonction de lecture à vitesse variable existe. C'est la même chose pour un disque vinyl, lorsque le *disk jokey* cale ses morceaux : il peut lire à l'endroit ou à l'envers le son enregistré sur le support vinyl, et ce à différentes vitesses.

Le fait de lire le son à des vitesses différentes correspond, en numérique, à avoir un signal échantillonné à une fréquence $F_e^1\ Hz$ et à le lire à la fréquence d'échantillonnage $F_e^2\ Hz$. La durée du signal n'est pas conservée, puisqu'il faut soit F_e^1, soit F_e^2 échantillons par seconde pour produire le son.

D'autre part, une sinusoïde de fréquence $F_e^1/10$ a 10 périodes par seconde lorsqu'elle est échantillonnée à $F_e^1\ Hz$ alors qu'elle en a $F_e^2\ Hz$ lorsqu'elle est échantillonnée à $F_e^2\ Hz$. Le fait de changer la fréquence d'échantillonnage implique que cette sinusoïde aura $10\frac{F_e^2}{F_e^1}$ périodes par seconde, ce qui signifie tout simplement que la hauteur change d'un facteur $\frac{F_e^2}{F_e^1}$.

Cet effet est simple à mettre en œuvre, et utilisé notamment pour caler des bandes, disques vinyls.

Compléments d'information

L'effet s'obtient aussi bien par ré-échantillonnage que par lecture à vitesse variable (cf. mmoire circulaire 1.7.1) ou par modulation de la longueur d'une ligne à retard (cf. modulation de frquence et de phase 1.5.4).

Rappel des paramètres de l'effet

Le paramètre permettant de caractériser cet effet est le ratio de transposition γ. Toutefois, on peut considérer que deux paramètres sont à indiquer :

- la fréquence d'échantillonnage de départ $F_e^1\ Hz$;
- la fréquence d'échantillonnage cible $F_e^2\ Hz$.

2.7.2 Changement de genre

Définition

Le genre d'un humain correspond à son sexe. Une voix d'homme est différent d'une voix de femme par le spectre et par le fondamental utilisé pour parler. Changer le genre d'une voix consiste alors à changer la voix d'un homme en celle d'une femme, ou le contraire.

Fonctionnement

L'ambitus de fréquence de l'homme est différent de celui de la femme. Modifier cet ambitus par transposition avec préservation des formants est une première étape (par exemple, monter le son d'une octave pour passer d'une voix d'homme à une voix de femme, ou baisser le son d'un octave pour passer d'une voix de femme à une voix d'homme).

Ensuite, il a été montré que les formants (notamment le premie) d'une même voyelle varient lorsque la fondamentale évoluent. Une manière de modéliser cela consiste à effectuer un décalage de l'enveloppe spectrale à la suite de la transposition. Pour passer d'une voix d'homme à une voix de femme, celui-ci se fait uniquement pour les fréquences au dessus de 100 Hz. En dessous de 100 Hz, on ne décale pas l'enveloppe spectrale. Entre 100 et 500 Hz, on décale linéairement de 0 à 50 Hz l'enveloppe spectrale. Au-dessus de 500 Hz, on ne décale pas plus l'enveloppe

Pour passer d'une voix de femme à une voix d'homme, le décalage se fait inversement, dans la même plage $[100; 500]Hz$;

Intérêt de cet effet

Il permet de passer d'une voix d'homme à une voix de femme, et réciproquement. On imagine aisément que dès lors que l'on sait quantifier les différences entre une voix d'adulte et d'enfant par exemple, on peut aussi réaliser un changement de timbre tel que passer d'une voix d'homme à une voix d'enfant, d'une voix d'enfant à une voix de femme, etc.

2.7.3 Changement de transition

Définition

Différents types de transitions existent, dont le legato (ou lié), le staccato, le piqué, le glissando (ou glissé). Le changement de transition consiste à modifier le type de transitions entre notes d'instruments de musique.

Fonctionnement

La manière de procéder pour modifier une transition consiste d'abord à reconnaître la transition, puis à appliquer une modification de hauteur et de dynamique afin de satisfaire les règles décrivant la transition cible. L'ajout ou le retrait du transitoire d'attaque permet d'affiner les résultats audibles.

Intérêt de cet effet

Cet effet permet par exemple de prendre une mélodie Midi jouée à l'aide d'un synthétiseur et d'y ajouter de la vie lors des transitions. Il permet aussi de changer l'expressivité (et la reconnaissance du timbre) d'instruments : ainsi, si on prend une mélodie de piano, que l'on supprime les transitoires et remplace les transitions par des glissandi rapides, le piano devient difficilement reconnaissable (un nouveau timbre apparaît).

Cet effet est encore à l'état de recherche.

2.7.4 Changement de prosodie

Définition

La prosodie correspond aux micro-variations de hauteurs d'une phrase de voix parlée. La manière de la modifier consiste donc à segmenter le signal en phonèmes, et à altérer la hauteur pour les différents phonèmes.

Fonctionnement

La segmentation du signal se fait selon des méthodes usuelles (cf. le cours de françois Pachet). La fréquence fondamentale est extraite pour chaque segment, et analysée en terme de variations (moyenne, écart-type). Le son est resynthétisé avec un ratio de changement de hauteur calculé de manière à modifier la prosodie (amplifier ou diminuer les micro-variations).

Compléments d'information

Cet effet est un exemple d'effet adaptatif, où des paramètres extraits du son pilotent l'effet audionumérique. Il est encore à l'état de recherche.

2.7.5 Changement d'expressivité

Définition

Pour cet effet, on entend expressivité non pas en son sens cognitif, mais via les attributs psychoacoustiques qui s'y rapprochent, à savoir la hauteur et le temps. En effectuant des variations à la fois sur le temps et sur la hauteur, on peut en effet produire de fort changements d'expressivité d'une voix parlée. Il faut pour cela procéder à une segmentation du signal sonore.

Fonctionnement

La segmentation du signal se fait selon des méthodes usuelles. La fréquence fondamentale est modifiée comme indiqué dans l'effet de changement de prosodie. Les variations de durée se font quant à elle dans des ratios proches de 1, entre 0.5 et 2 par exemple, en utilisant un changement d'échelle adaptatif pour conserver la cohérence du son.

Compléments d'information

Cet effet est un autre exemple d'effet adaptatif, où des paramètres extraits du son pilotent l'effet audionumérique. Les exemples d'utilisation sont nombreux, notamment pour l'ajout d'expressivité aux voix synthétiques (répondeurs téléphoniques, messages sonores dans les gares, synthèse de parole sur ordinateur).

Chapitre 3

Implémentations et contrôle des effets audionumériques

Parlons maintenant des différents techniques d'implémentations, et des difficultés qui découlent de chaque.

3.1 Implémentation temps réel

3.1.1 Implémentation échantillon par échantillon

L'implémentation échantillon par échantillon consiste à prendre un échantillon en entrée, et à donner l'échantillon traité dans un temps moyen inférieur ou égale au temps d'acquisition de l'échantillon suivant. Cette implémentation est la plus proche de l'implémentation en machine (DSP), et nécessite une optimisation des calculs du traitement.

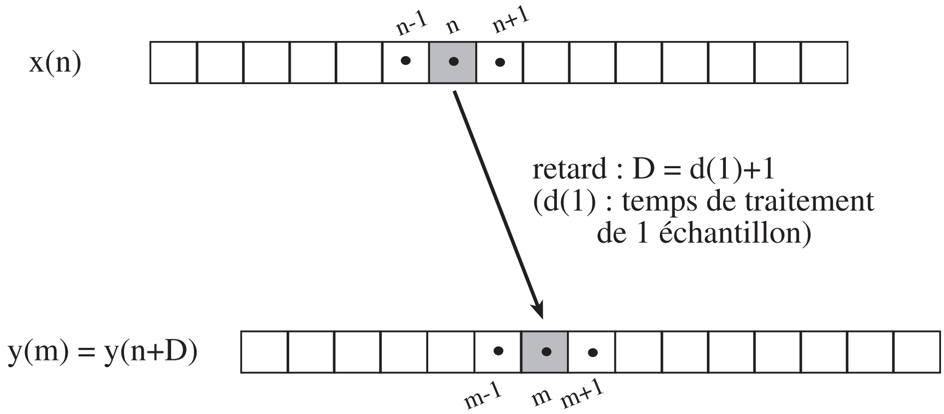

Fig. 3.1: *Implémentation échantillon par échantillon : le retard D est fonction du temps de calcul du traitement d'un échantillon* $d(1)$

L'avantage principal de cette implémentation est :

- le temps-réel (délai entre entrée et sortie d'un ou de quelques échantillons).

Les inconvénients inhérents à l'implémentation échantillon par échantillon sont :

- les traitements doivent être rapides, réalisables sur un cycle d'échantillonnage ;

- les traitements doivent être simples, réalisables échantillon par échantillon (domaine temporel) ;
- des codes de calcul bien optimisés sont peu lisibiles (portions de langage assembleur et astuces d'optimisation).

3.2 Implémentation en temps différé

3.2.1 Implémentation par blocs

L'implémentation par blocs consiste à découper le signal à traiter en blocs, et à traiter successivement chaque bloc. Ces blocs peuvent être des fenêtres d'analyse (vocodeur de phase, analyse aditive), ou un ensemble de N fenêtres d'analyse. Ces blocs peuvent se superposer (cf. la TFCT).

Le délai entre le bloc entrant et le bloc traité est au minimum de la durée du bloc (en nombre d'échantillon) : il faut attendre d'avoir reçu les T_{bloc} échantillons avant de les traiter, et le traitement peut nécessiter un temps supérieur à une période d'échantillonnage (1/44100 s).

Cette technique permet de mieux répartir la charge de calcul que la technique échantillon par échantillon (cf. 3.1.1) : le traitement de certains échantillons peut nécessiter plus de temps que pour d'autres échantillons ; il peut aussi y avoir une étape d'initialisation commune au traitement de tous les échantillons du bloc, pour des traitements spécifiques.

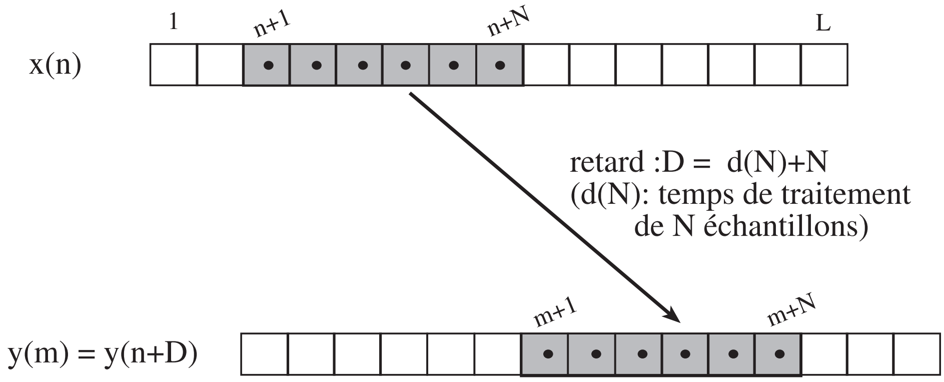

Fig. 3.2: *Implémentation bloc par bloc : le retard D est fonction du temps de calcul du traitement de N échantillons $d(N)$.*

Les avantages principaux de cette implémentation sont :

- l'implémentation est temps-réel ou quasiment (faibles retard, de 20 à 30 ms) pour de petits blocs ;
- les traitements complexes (notamment spectraux) sont rendus possibles ;
- en cas de lecture-écriture dans des fichiers, c'est plus rapide que l'implémentation échantillon par échantillon.

Les inconvénients inhérents à l'implémentation par bloc sont :

- le temps de délai de l'ordre de 20 ms pour des blocs (TFCT) de 2048 échantillons à 44 100Hz : ce n'est pas vraiment temps-réel, le délai peut poser problème pour certaines applications. Pour des blocs plus grands, c'est encore pire.

3.2.2 Implémentation vectorielle

Cette implémentation est liée au traitement en temps différé. Le seul cas où elle peut être utilisée en temps réel est celui d'applications spécifiques utilisant de grands délais entre un son et sa version traitée.

Dans cette approche, le son numérisé est considéré comme un seul vecteur. Le traitement consiste en une succession d'unités de traitement prenant un vecteur en entrée, et donnant un vecteur en sortie. Ce type de traitement est naturel dans des langages tel que **Matlab**.

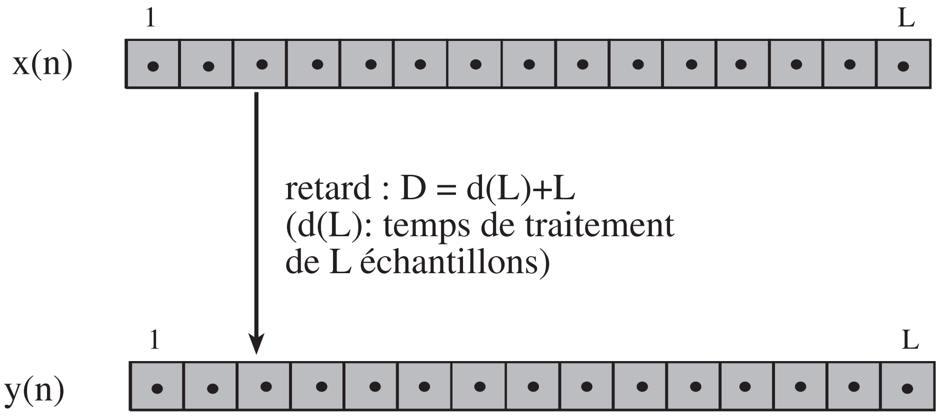

Fig. 3.3: *Implémentation par vecteur : le retard D est fonction du temps de calcul du traitement de tous les échantillons* $d(L)$.

Les avantages principaux de cette implémentation sont :

- chaque procédure de traitement est indépendante des autres, et peut être testée indépendamment ;
- le temps de traitement peut être très court dès lors que les procédures de traitement sont optimisées ;
- un effet est constitué d'une succession de procédures.

Les inconvénients inhérents à l'implémentation vectorielle sont :

- le coût en terme de mémoire : il faut suffisamment d'espace mémoire pour stocker tous les vecteurs, y compris les variables intermédiaires ;
- cette implémentation n'est pas utilisable en temps réel, puisque le son d'entrée est supposé entièrement connu avant le traitement (sauf dans le cas évoqué en introduction).

3.3 Contrôle et *mapping*

Le problème du contrôle est important en traitement sonore. Tout d'abord, le type de contrôle que l'on veut sur l'effet déterminera s'il faut utiliser du temps réel ou non. Ainsi, un contrôle donné par une courbe peut se faire en temps différé, alors qu'un contrôle donné par un transducteur gestuel est plus adapté pour des traitements en temps réel.

Le *mapping* est la mise en correspondance entre des paramètres en entrée (les coordonnées d'un transducteur gestuel, des valeurs, des courbes) et les contrôles de l'effet (les paramètres qui le définissent). Différentes stratégies de mise en correspondance existent, notamment le *mapping* explicite (sa formulation mathématique est donnée via des équations) ou implicite (le modèle de correspondance est non-linéaire, et seul son principe est décrit, par exemple les réseaux de neurones). Un *mapping* en trois couches constitue une voie de recherche prometteuse [2] : la première couche va du transducteur gestuel à une représentation perceptive des gestes, la troisième couche relie les paramètres perceptifs du son au paramètres de synthèse ou de traitement, et la couche intermédiaire relie les paramètres perceptifs du geste à ceux du son.

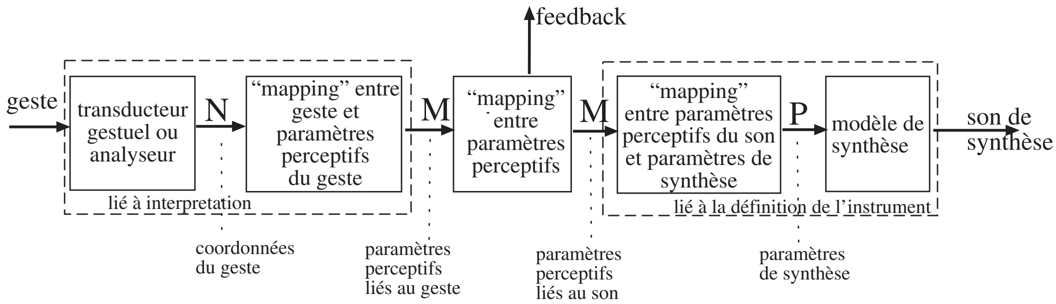

Fig. 3.4: Mapping *en trois étapes, avec utilisation d'une couche "perceptive" comme intermédiaire entre paramètres du geste et paramètres de synthèse ou de l'effet.*

Le *mapping* comprend aussi des étapes d'ajustement (*scaling* ou *fitting*), de façon à caler les plages de variation des paramètres et des contrôles.

3.4 Contrôle avec interfaces graphiques

Le contrôle par interfaces graphiques est le plus simple, très fréquemment utilisé du fait que tout ordinateur dispose au moins d'un écran, d'une unité centrale, d'un clavier et d'une souris. L'écran est alors utilisé pour présenter les paramètres de contrôle de l'effet (ex : *plug-in*, *patch* Max/MSP, jMax ou Pure Data), et la souris et ses boutons pour sélectionner (activer) un paramètre, et le faire varier. Des courbes et indicateurs (jauges, potentiomètres, valeur numérique) permettent un retour visuel à l'utilisateur.

Voici quelques points clefs à respecter lors de la réalisation d'interfaces graphiques utilisateurs (IGU, ou *GUI, graphical User Interfaces*) :

- la visibilité : elle permet à l'utilisateur de voir ce qu'il peut faire de tel ou tel outil ;
- la transparence : l'utilisateur ne voit pas ce qui lui a été volontairement caché (les calculs entrepris par l'ordinateur), mais il a accès à une visualisation de la tâche réalisée, selon l'image mentale qu'il s'est faite du processus ;
- prévisibilité : le système doit réaliser la tâche à laquelle l'utilisateur naïf s'attend, à partir de ses connaissances et à l'aide de métaphores (par exemple avec la représentation d'un potentiomètre, d'un clavier) ;
- consistance : l'application doit être prévisible dans tout contexte du programme et d'une application à l'autre ;

- intégrité : l'interface doit protéger les données précieuses, même lorsque l'utilisateur effectue une mauvaise manipulation ; il doit être possible d'annuler une action, et de revenir en arrière ;
- concision : l'interface doit être concise, tant au niveau du contrôle (raccourcis clavier, menu déroulants, valeurs par défaut, etc) qu'au niveau de l'affichage écran via la fenêtre de l'application (tout particulièrement l'information importante qui doit être claire et précise) ;
- apparence à l'écran : l'apparence doit être soignée, claire et ordonnée ; brillance, contraste, couleurs, textures, clignotements doivent être utilisés pricipalement pour leur sens, avant les raisons esthétiques ;
- adaptabilité : l'utilisateur doit pouvoir, sans avoir à programmer, réaliser l'interface correspondant à ses besoins et son niveau de connaissances ;
- guide : tout utilisateur peut avoir parfois besoin de réponses à certaines questions ; si l'interface ne se comprend pas d'elle-même, une manuel en ligne ou une aide contextuelle peut être utile.

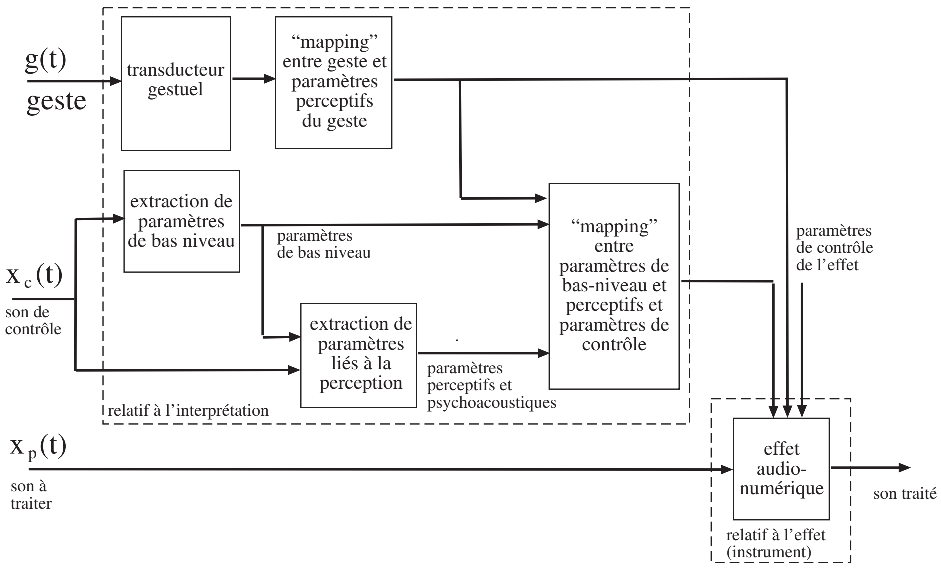

Fig. 3.5: *Diagramme de l'effet audionumérique adaptatif contrôlé gestuellement [2]*

3.5 Contrôle gestuel

L'utilisation du geste comme moyen de contrôle se fait à l'aide de transducteurs gestuels : la souris, le clavier, mais aussi les claviers midi, le joystick, les capteurs électromagnétiques de position, de vitesse, d'accélération, d'orientation (dont le *radio baton* et le *flock of birds*), les capteurs de pression, le volant de jeu et les pédales associées, le gant de réalité virtuelle, les *racks* de potentiomètres, les capteurs à ultrasons, les capteurs de percussion, etc.

Les données transitent entre le transducteur et l'ordinateur peut être au format Midi (c'est le cas pour Max/MSP, jMax, Pure Data), ou série, USB, selon le transducteur utilisé.

3.6 Contrôle automatisé

Le contrôle automatisé peut se faire de deux manières. La première consiste à utiliser une interface graphique pour faire apparaître la valeur des contrôles de l'effet, et les modifier en temps réel à l'aide de la souris. Les séquenceurs midi et audio, tel **ProTools**, **Cubase**, **Logic Audio**, **Digital Performer**, permettent généralement de réaliser cette tâche.

Une autre manière de faire est d'automatiser le contrôle par des paramètres extraits du son. C'est le principe déjà évoqué des effets audionumériques adaptatifs (ou *A-DAFx, Adaptive Digital Audio Effects*, [14]).

Ces effets permettent un contrôle cohérent avec le son, puisque les courbes de contrôle proviennent de descripteurs du son. Ils permettent notamment des effets ou synthèses croisées, et réintroduisent le contrôle gestuel au niveau du *mapping* entre descripteurs et contrôles de l'effet, permettant un contrôle de plus haut niveau. Les descripteurs les plus souvent utilisés sont les suiveurs d'enveloppe et de hauteur. D'autres descripteurs, tels les partiels et leurs modules, les enveloppes spectrales, le centroïde, l'indice de voisement (voisé/non voisé), le détecteur de sons (silence/son) sont aussi très utiles, d'autant plus qu'il sont déjà présents dans certains effets, à l'étape d'analyse.

Chapitre 4

Travaux Pratiques

Les Travaux Pratiques proposent de réaliser soi-même des effets et transformations sonores. La première partie des TP se fait sous **Matlab**, et verra la réalisation d'effets en temps différé (hors temps-réel). La seconde partie des TP se fait sous **Pure Data** (**Pd, Public Domain**), environnement graphique de traitements et de synthèse en temps-réel.

Le but de la première partie sous l'environnement de programmation semi-formel Matlab est de comprendre les bases de traitement de signal utilisées par quelques effets, et de connaître les fonctions permettant de les réaliser. Une fois ces bases acquises, nous nous pencherons sur des programmes existant, qui mettent en œuvre des techniques expliquées en cours.

La seconde partie quant à elle va permettre de se dégager de l'implémentation pour découvrir le contrôle (par la manipulation des paramètres de chaque effet) en temps-réel.

La durée totale des TP est de 6 à 8 heures, selon la vitesse à laquelle le cours avance. Les sessions Matlab seront les deux premiers TP, le (ou les deux) dernier(s) sera consacré au temps-réel.

4.1 TP 1 : Bases, filtrages et modulateurs (Matlab, temps différé, 2 heures)

4.1.1 Bases de Matlab

Matlab a été réalisé pour la manipulation de matrices. Depuis, les possibilités offertes par ce logiciel ont beaucoup évoluées. L'une des principales carctéristiques à retenir est la possibilité de manipuler des vecteurs directement, comme de simples variables.

Lancer Matlab

Double-cliquez sur l'icône Matlab. C'est parti ! L'environnement présente une fenêtre de commandes et un explorateur de disque dur. Lorsque vous sélectionnez un fichier, il s'ouvre dans l'éditeur Matlab. Pour l'éxecuter, utilisez le raccourci clavier `Ctrl E`, ou le menu `Fichier Sauver et exécuter`. Une autre manière de faire consiste à copier le contenu du fichier, et le déposer dans l'éditeur.

A l'aide !!!

Une aide en ligne est disponible, notamment dans l'interpréteur de commandes : tapez `help`. Si vous cherchez l'aide d'une fonction précise, tapez `help <nom_de_fonction>`.

Sans commentaire

Toute portion de code située à droite du signe `%` est interprétée comme un commentaire. Chaque ligne se termine par un `;`, sauf dans le cas où l'on désire afficher le résultat. Ainsi, l'interpréteur donnera les résultats suivants :

```
a = 2;
a = 2

a =

     2
```

Générer une sinusoïde

Soit le signal définit par $x(t) = \sin(2\pi F_0 t + \Phi)$. Pour le créer sous l'environnement Matlab, il faut définir la fréquence F_0, la fréquence d'échantillonnage F_e, la longueur du signal que l'on veut créer (le nombre d'échantillons) :

```
Fe = 44100;              % frequence d'echantillonnage : 44,1 kHz
L = 2*Fe;                % duree : deux secondes
F0 = 400;                % fondamentale : 400 Hertz
t = (1:L)/Fe;            % vecteur Temps : instant d'echantillonnage
x = sin(2*pi*F0*t);      % signal resultant
```

La fonction `soundsc` permet d'écouter un signal après l'avoir normalisé. Le premier argument est le signal, le second correspond à la fréquence d'échantillonnage.

```
soundsc(x,Fe)
```

Remarquons tout de suite que pour réaliser un ré-échantillonnage du son, il suffit de changer la valeur de `Fe`. Ainsi, on peut réaliser extrêmement facilement notre premier effet : le changement de vitesse de lecture (qui, rappellons-le, modifie la hauteur d'un facteur γ et la durée du signal d'un facteur $\frac{1}{\gamma}$). Pour lire le son deux fois moins vite (et l'entendre deux fois plus bas, soit à l'octave inférieure) :

```
soundsc(x,Fe/2)
```

Au contraire, pour lire le son deux fois plus vite (et l'entendre deux fois plus haut, soit à l'octave supérieure) :

```
soundsc(x,Fe*2)
```

Malheureusement, la fonction `soundsc` n'accepte que des fréquences d'échantillonnage inférieures à 48 kHz. Il faut donc générer une sinusoïde à la fréquence d'échantillonnage de 16 kHz, par exemple.

Visualiser

Pour visualiser une fonction mathématique discrétisée et stockée dans un tableau de valeur, utilisez la fonction `plot`. Dans le cas de notre sinusoïde par exemple, on aimerait que l'axe des abcisse corresponde au temps (en secondes). Si l'on exécute la commande `plot(x)`, les abcisses correspondent aux numéros d'échantillons. La commande `plot(t,x)` quant à elle affichera le signal avec des abcisses correspondant aux secondes.

Transformée de Fourier

La transformée de Fourier s'obtient à l'aide de la fonction `fft`. Son premier argument est le signal à transformer. Si sa taille est une puissance de 2, le calcul est direct. Sinon, l'algorithme utilisé est celui de le TFD classique. Si l'on désire faire du bourrage de zéros volontairement, on peut ajouter un second argument à la fonction : la taille du signal une fois allongé (de façon à ce que sa longueur soit la puissance de 2 directement supérieure). Le module est obtenu avec la fonction `abs`, la phase avec la fonction `angle`. Regardons le spectre d'amplitude (en décibel) de notre sinusoïde.

```
y = x(1:2048);           % on tronque la sinusoide
Y = fft(y);
plot(20*log10(abs(Y))); % affichage basique
plot((1:2048)/2048, 20*log10(abs(Y))); % affichage en frequences reduites
plot((1:2048)/2048*Fe, 20*log10(abs(Y))); % affichage en frequences (Hertz)

Y = fft(y, 4096);        % bourrage de zeros
hold on;                 % pour superposer plusieurs graphiques
plot((1:4096)/4096*Fe, 20*log10(abs(Y)),'r'); % affichage en freq (Hz)
```

Fenêtres

On sait que n'appliquer "aucune" fenêtre au signal avant d'en calculer la Transformée de Fourier, cela revient à lui appliquer la pire des fenêtres : la fenêtre rectangulaire. Aussi, on va appliquer au signal une fenêtre triangulaire (`triang`), une fenêtre de Hanning (`hanning`) ou une fenêtre de Blackman (`blackman`). La manière de faire est de multiplier point par point deux vecteurs : les opérations point par point utilisent les opérandes classiques ($*$, $/$, $+$, $-$), précédés du symbole . "point". Le résultat de la multiplication de deux vecteurs `x` et `w` est obtenu par `y = x .* w;`.

Regardons les Transformées de Fourier de chacune de ces fenêtres. Pour ce faire, nous utilisons la fonction `fftshift`, qui permet de centrer les points en plaçant les fréquences négatives à gauches des fréquences positives (même si cela semble leur place la plus logique, le stockage des transformée de Fourier dans Matlab est tel que les modules et phases correspondant aux fréquences négatives sont placées après les modules et phases correspondant aux fréquences positives, dans le même vecteur). Il faut donc bien faire attention, lors des affichages aux fréquences mises en abcisses.

```
NF = 2048;
Nfen = 50;
w_rect = ones(Nfen,1);                % fenetre rectangulaire
w_triang = triang(Nfen);              % fenetre triangulaire
w_hanning = hanning(Nfen);            % fenetre de Hanning
```

```
w_blackman = blackman(Nfen);            % fenetre de Blackman
w = w_triang;
figure(1);                              % choix de la figure
plot((-NF/2:NF/2-1)/NF*44100, 20*log10(abs(fftshift(fft(w,NF)))))
xlabel('Frequences (Hz)')
ylabel('magnitude (dB)')
title('TF de la fenetre triangulaire')
```

Profitons-en pour regarder le rapport en dB entre les maxima du lobe primaire et du lobe secondaire, ainsi que la largeur du lobe primaire de chaque fenêtre.

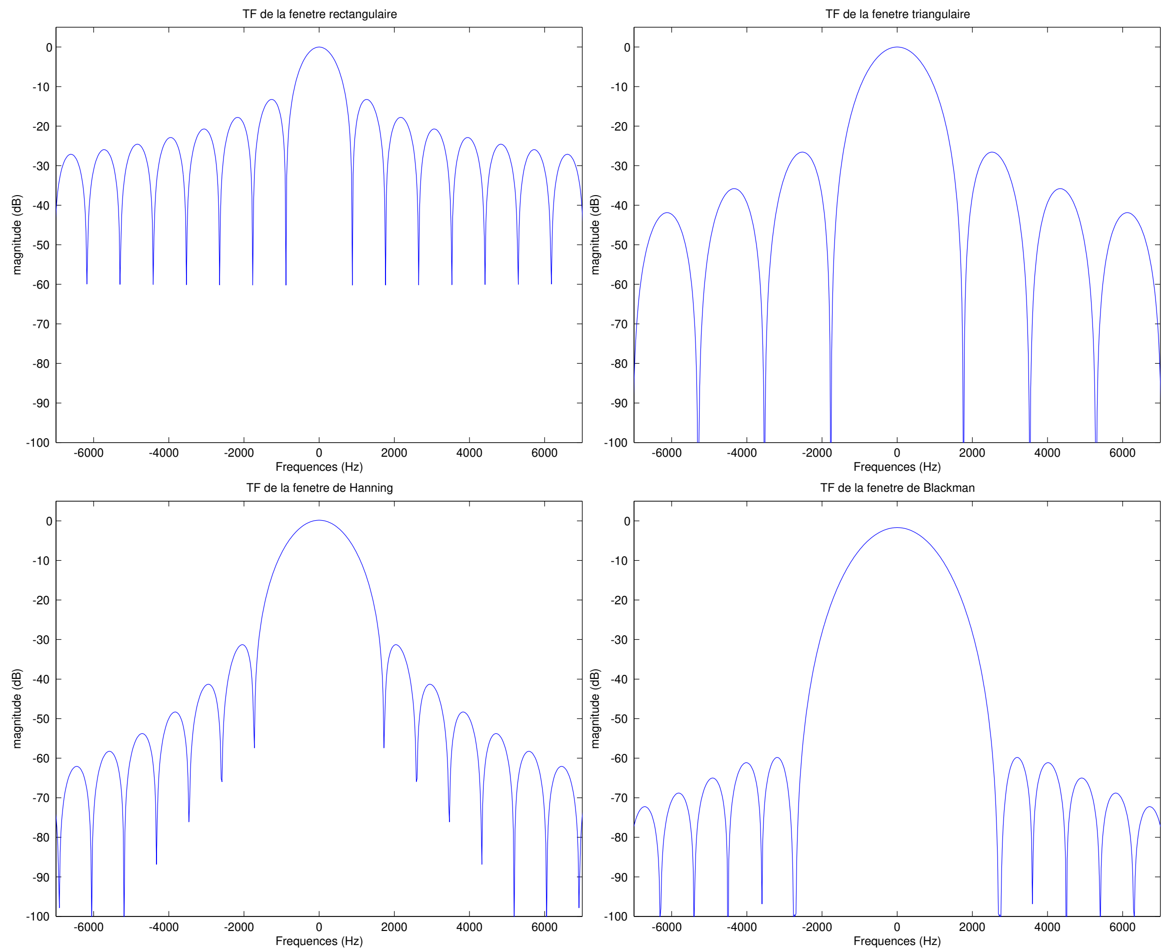

Fig. 4.1: *Transformées de Fourier des fenêtres rectangulaire, triangulaire, de Hanning et de Blackman*

Polynômes, filtres

Sous Matlab, un polynôme est donné par un vecteur, contenant les coefficients dudit polynôme, par ordre de puissance décroissante. Ainsi, le polynôme $x^3 + 2x^2 - x - 3$ est représenté par le vecteur `[1 2 -1 -3]` $\approx (1,\ 2,\ -1,\ -3)$. Remarquons que ce vecteur est un vecteur ligne, et peut s'écrire avec des virgules séparant chaque valeur, pour plus de lisibilité : `[1, 2, -1, -3]`. L'utilisation des points virgules dans les vecteurs correspond au passage à la ligne :

`[1; 2; -1; -3]` $\approx (1,\ 2,\ -1,\ -3)^T$. La transposition sous Matlab se fait à l'aide du symbole “apostrophe” `'`, ce qui donne `[1; 2; -1; -3]` $\approx$ `[1, 2, -1, -3]'`.

Soit le filtre en peigne non récursif. Sa formule de récurrence est $y(n) = x(n) + cx(n-p)$. Prenons par exemple $p = 4$, le filtrage se réalise avec la fonction `filter` en utilisant le polynôme `[c 0 0 0 1]` :

```
a = [0.95 0 0 0 1];
b = 1;
y = filter(b, a, x);
soundsc(x,Fe);
```

Une fois définit, un filtre peut être visualisé par sa réponse en fréquence à l'aide de la fonction **freqz** de Matlab.

```
freqz(b,a);      % affichage de la reponse en frequence
                 % et en phase du filtre
zplane(b,a);     % representation dans le cercle unite des zeros
                 % et des poles du filtre
```

4.1.2 Modulateurs

Modulateur en amplitude

1. Générer un signal sinusoïdal $x_m(t)$ de fréquence fondamentale $F_m = 100\ Hz$ et de durée 2 secondes.

2. Générer un signal sinusoïdal $x_p(t)$ de fréquence fondamentale $F_p = 440\ Hz$ et de durée 2 secondes.

3. Générer le signal $y(t) = x_m(t) \times (1 + x_p(t))$ résultant d'une modulation en amplitude de x_m et x_p. L'écouter et tracer dans une même figure sa transformée de Fourier ainsi que celles de la porteuse et de la modulante.

4. Quelles observations faîtes-vous ?

5. Recommencer avec $F_m = 300Hz$ et $F_p = 400\ Hz$. Quelles observations faîtes-vous ?

6. Recommencer avec $F_m = 5Hz$ et $F_p = 400\ Hz$. Quelles observations faîtes-vous ?

Modulateur en anneau

1. Générer un signal sinusoïdal $x_m(t)$ de fréquence fondamentale $F_m = 100\ Hz$ et de durée 2 secondes.

2. Générer un signal sinusoïdal $x_p(t)$ de fréquence fondamentale $F_p = 440\ Hz$ et de durée 2 secondes.

3. Générer le signal $y(t) = x_m(t) \times x_p(t)$ résultant d'une modulation en anneau de x_m et x_p. L'écouter et tracer dans une même figure sa transformée de Fourier ainsi que celles de la porteuse et de la modulante.

4. Quelles observations faîtes-vous ?

5. Recommencer avec $F_m = 300Hz$ et $F_p = 400\ Hz$. Quelles observations faîtes-vous ?

6. Recommencer avec $F_m = 5Hz$ et $F_p = 400\ Hz$. Quelles observations faîtes-vous ?

4.1.3 Filtres en peigne

Nous allons créer le filtre en peigne à RIF définit par : $y(n) = x(n) + 0.9\,x(n-10)$ à l'aide d'une ligne à retard puis à l'aide de la fonction de filtrage de Matlab, afin de montrer l'équivalence entre une ligne à retard et un filtre récursif.

1. Créer un vecteur `LigneRetard` de 10 valeurs, initialisé à 0.

2. Créer un signal bruit blanc $x(n), n = 1, \ldots, 1024$ à l'aide de la fonction de génération de nombres aléatoires `randn`.

3. A l'aide d'une boucle, calculer le signal $y(n)$ d'après la formule de récurrence.

4. Calculer analytiquement la réponse en fréquence du filtre, puis numériquement (les variables `i`et `pi` sont connues de l'environnement Matlab, aussi une formulation du type `H = 1+g*(exp(-2*i*pi*10*t)))` est possible et conseillée.

5. Tracer la fonction de transfert, ou module de la réponse en fréquence du filtre (en décibels).

6. Retour au filtrage classique : connaissant la formulation de récurrence du filtre, calculer les polynômes `a` et `b` et le filtre `h` associé.

7. Utiliser la fonction `freqz` pour visualiser le module de la réponse en fréquence du filtre. Comparer à celle obtenue précédemment par la méthode "ligne à retard".

8. Utiliser la fonction `zplane` pour visualiser les pôles et les zéros du filtre.

9. Calculer maintenant le signal `y2` résultant du filtrage de `x` par `h`.

10. Visualiser sur un même graphique les signaux `y` et `y2`.

11. Appliquer à nouveau chaque méthode (ligne à retard et filtre), en plaçant la commande `tic` avant la première ligne et la commande `toc` après la dernière ligne. Ces commandes déclenchent respectivement le démarrage et l'arrêt d'un compteur, en secondes. Comparer ainsi les temps de calcul des deux méthodes. Conclure.

Modifier le programme précédent de façon à créer le filtre en peigne à RII définit par : $y(n) = x(n) + 0.9\ y(n-10)$. Réaliser à nouveau les visualisation des modules des réponses à fréquence.

Modifier encore le programme précédent de façon à créer le filtre en peigne universel définit par : $y(n) = \beta\ x(n) + g_x\ x(n-L) + g_y\ y(n-L)$. Réaliser à nouveau les visualisation des modules des réponses à fréquence, pour différentes valeurs de β, g_x et g_y bien choisies.

4.2 TP 2: le Vocodeur de Phase en temps différé (Matlab, 2 heures)

Ce deuxième TP est consacré aux effets et transformations utilisant des représentations spectrales du son (vocodeur de phase, SMS). Nous n'utiliserons que le vocodeur de phase, ce qui est déjà suffisamment courageux de votre part !!!

Plutôt que de partir de vos connaissances et écrire entièrement des programmes, il vous est proposé de partir de programmes déjà existants, disponibles sur le site du livre DAFx (`http://www.dafx.de/`), que nous allons modifier petit à petit, afin d'apprendre à les utiliser.

4.2.1 Vocodeur de phase

Etude d'un programme réalisant une analyse-resynthèse à l'aide du vocodeur de phase.

1. Ouvrir le programme `TP2_1_nothing_pv.m` dans l'éditeur de Matlab. Que représente la variable `n1` ?

2. Modifier le nom du fichier son au format *.wav* que l'on désire traiter. Exécuter le prgramme. Que remarquez-vous à l'écoute du son traité ?

3. A quoi servent les lignes de code suivantes :
   `DAFx_in = [zeros(WindowLength, 1); DAFx_in; ...`
   `zeros(WindowLength-mod(L,n1),1)] / max(abs(DAFx_in));` ?

4. A quoi sert la fonction `fftshift` ?

5. Où doit-on placer les traitements temps-fréquences que l'on désire effectuer ?

6. A quoi peut servir la variable `n2` ?

4.2.2 Synthèse croisée

Utilisation du cepstre pour effectuer une synthèse croisée, en appliquant les formants d'un son à un second son.

1. Ouvrir le programme `TP2_2_cross.m` dans l'éditeur de Matlab. Choisir deux fichiers son (le signal d'excitation doit être riche, par exemple un extrait d'orchestre symphonique).

2. Quel est le rôle de la valeur 0.00001 utilisée dans la ligne de commande
   flog = log(0.00001+abs(f)); ?

3. Quelle fenêtre de troncature est utilisée pour le calcul de l'enveloppe spectrale à l'aide du cepstre ?

4. Comment est calculée la phase du grain ? Utiliser la ligne de commande :
   *grain = (real(ifft(f1.*f2))).*w2;*
   pour répondre.

4.2.3 Changement de hauteur sans conservation des formants

Réalisation d'une transposition sans conservation des formants, à l'aide du vocodeur de phase.

1. Ouvrir le programme `TP2_3_pshift_pv.m` dans l'éditeur de Matlab.
2. Identifier ce que représentent les variables `delta_phi`, `omega`, *phi* et *phi0*.
3. Expliquer la méthde d'interpolation utilisée.
4. Que représente `lx` ?
5. Peut-on dire que les grains de synthèse sont centrés (ie. ont le même indice temporel) ? Sinon, comment y remédier ?
6. Quelle est la relation entre le coefficient de transposition et la variable `tstretch_ratio` ?
7. Dans le cas où le coefficient de transposition varie dans le temps, quels problèmes pourraient se poser ?
8. Que manque-t-il pour conserver les formants ?

4.2.4 Changement de hauteur avec conservation des formants

Réalisation d'une transposition avec conservation des formants, à l'aide du vocodeur de phase, avec utilisation du cepstre pour conserver les formants.

1. Ouvrir le programme `TP2_4_pshift_formant.m` dans l'éditeur de Matlab.
2. A quoi correspond, selon-vous, la variable `grain_formant` ?
3. A quoi correspond, selon-vous, la variable `corr` ?
4. Dans quelle ligne est réalisée la correction sur les formants ?
5. Dans quelle ligne est réalisée la transposition ?

4.2.5 Changement de durée

Modifiation de l'échelle temporelle d'un son par vocodeur de phase.

1. Ouvrir le programme `TP2_5_tstretch_real_pv.m` dans l'éditeur de Matlab.
2. Les paramètres `n1` et `n2` ont-ils la même valeur ? Sinon, expliquer le rôle que peut avoir `n1`.
3. Quelle valeur donner à `n1` pour obtenir un son quatre fois plus long ? Et pour obtenir un son quatre fois plus court ?
4. Dans les deux cas, en appliquant la transformation à une voix, que remarque-t-on au niveau des consonnes ?

4.2.6 Modification du timbre

Réalisation d'un effet de robotisation.

1. Ouvrir le programme `TP2_6_robot.m` dans l'éditeur de Matlab.
2. Quel traitement est réalisé pour le calcul du grain de synthèse `grain` ?
3. Aucune information de phase n'est donnée pour le calcul de la FFT inverse. Qu'est-ce que cela signifie ?
4. Que vaut le vecteur résultat entre deux grains de synthèse ?
5. Comment est fixée la hauteur du son résultant ?
6. Que se passe-t-il si l'on veut faire varier cette hauteur dans le temps ?

4.3 TP 3: modulateurs et lignes à retard en temps-réel (Pure Data, 2 heures)

Ce TP porte sur l'utilisation de modulateurs et de lignes à retard, à l'aide de l'environnement graphique temps-réel **Pure Data**. Lancer Pure Data, à l'aide de la commande `pd` dans une fenêtre "terminal".

La fenêtre fig. 4.2 s'ouvre.

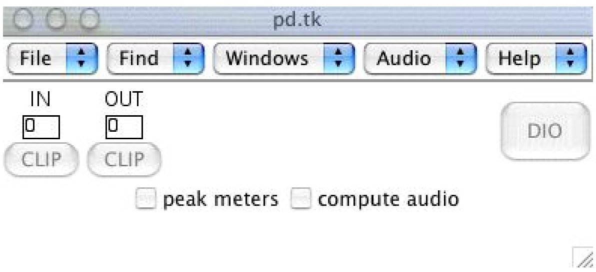

Fig. 4.2: *0.pd*

4.3.1 Modulateur en anneau

1. Ouvrir le *patch* `1.ring-mod`, présenté fig. 4.3.
2. L'utiliser en ouvrant un fichier son, en choisissant une fréquence de modulation puis en lançant le calcul du son (DSP) à l'aide du bouton "run the transformation". Que remarquez-vous pour une fréquence de 4 Hz ?
3. Et pour une fréquence de 40 Hz ?
4. Quelle différence y a-t-il entre la modulation à 40 Hz et la modulation à -40 Hz ? Est-elle audible ?

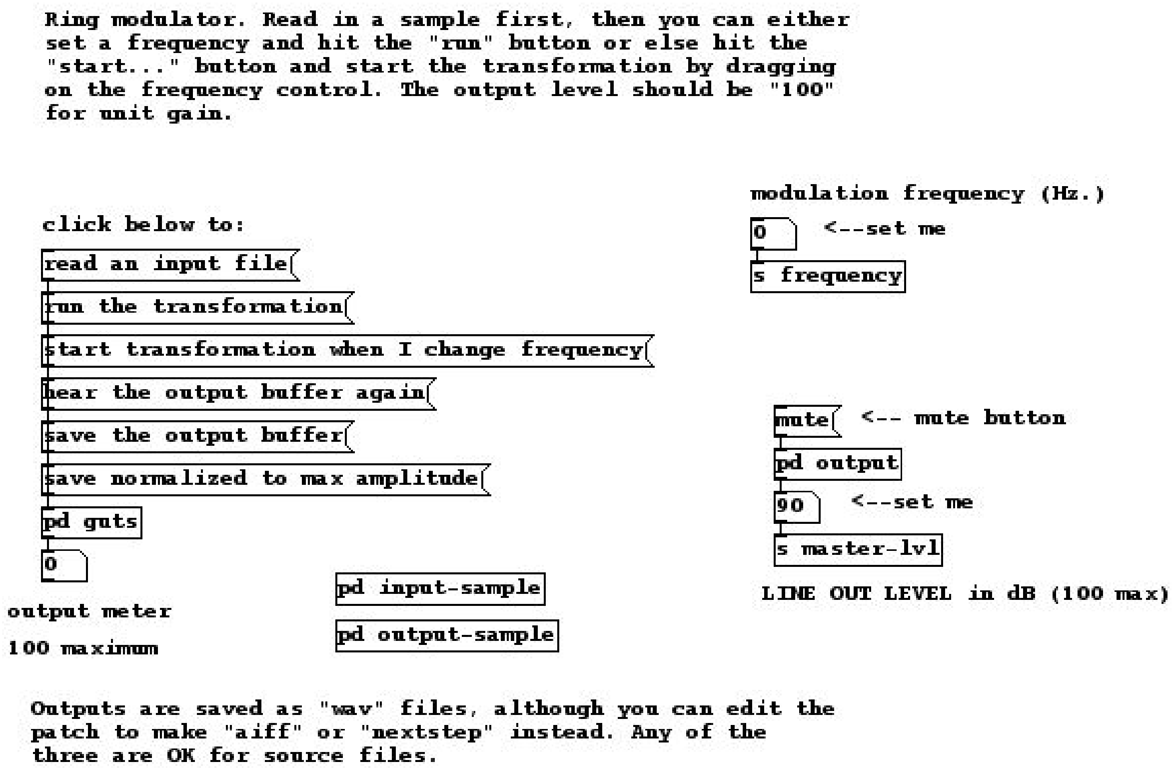

Fig. 4.3: *Patch de modulateur en anneau.*

4.3.2 Flanger

Ouvrir le *patch* `6.flanger`, présenté fig. 4.4. L'utiliser en ouvrant un fichier son, puis en lançant le calcul du son (DSP).

A quoi conduit la manipulation des curseurs et boîtes à nombres ?

4.3.3 Réverbération

Ouvrir le *patch* `5.reverb`, présenté fig. 4.5. Il implémente le calcul d'une réverbération.

A quoi conduit la manipulation des curseurs et boîtes à nombres ?

4.3.4 Transposition par modulation de ligne à retard

Ouvrir le *patch* `56.delay.pitchshift`, présenté fig. 4.6. Il implémente la transposition (changement de hauteur) en modulant une ligne à retard.

A quoi conduit la manipulation des curseurs et boîtes à nombres ?

4.3.5 Bonus : synthèse granulaire

Ouvrir le *patch* `4.looper`, présenté fig. 4.7. Il permet d'effectuer une synthèse granulaire, effet particulier où le son est généré par ajout superposition de grains lus à des endroits choisis par l'utilisateur, et avec une taille de grain fixée par l'utilisateur.

A quoi conduit la manipulation des curseurs et boîtes à nombres ?

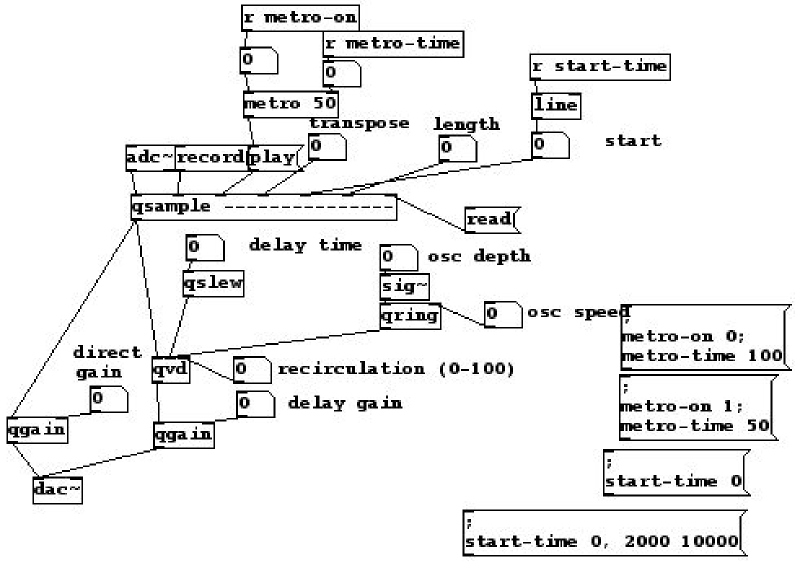

Fig. 4.4: *Patch de flanging.*

4.4 TP 4: le Vocodeur de Phase en temps-réel (Pure Data, 2 heures)

4.4.1 Vocodeur de phase : transposition et changement de durée

Ouvrir le *patch* `3.phase.vocoder`, présenté fig. 4.8. Il implémente le vocodeur de phase, et permet d'effectuer des transpositions avec conservation de la durée, et des changements de durée avec conservation de la hauteur.

A quoi conduit la manipulation des curseurs et boîtes à nombres ?

4.4.2 Vocodeur de phase : convolution

Ouvrir le *patch* `08.convobros`, présenté fig. 4.9. Il implémente le vocodeur de phase, et permet d'effectuer des synthèses croisées. L'enveloppe spectrale est extraite à l'aide du cepstre, et est appliquée au second son après blanchiment par le cepstre.

A quoi conduit la manipulation des curseurs et boîtes à nombres ?

4.4.3 Vocodeur de phase amélioré

Ouvrir le *patch* `10.phaselockedvoc`, présenté fig. 4.10. Il implémente le vocodeur de phase avec l'amélioration proposée par Laroche et Dolson : les phases sont bloquées autour des pics de la TFCT, ce qui permet de limiter les calculs (et donc diminuer la charge de calcul). Ce *patch* permet d'effectuer des transpositions avec conservation de la durée, et des changements de durée avec conservation de la hauteur.

A quoi conduit la manipulation des curseurs et boîtes à nombres ?

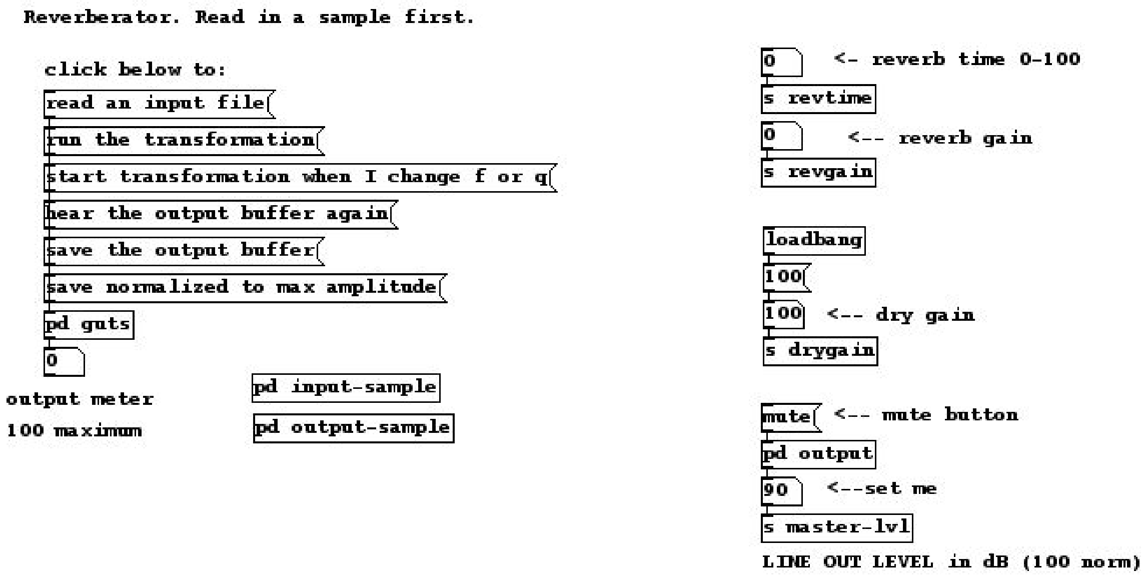

Fig. 4.5: *Patch de réverbération artificielle.*

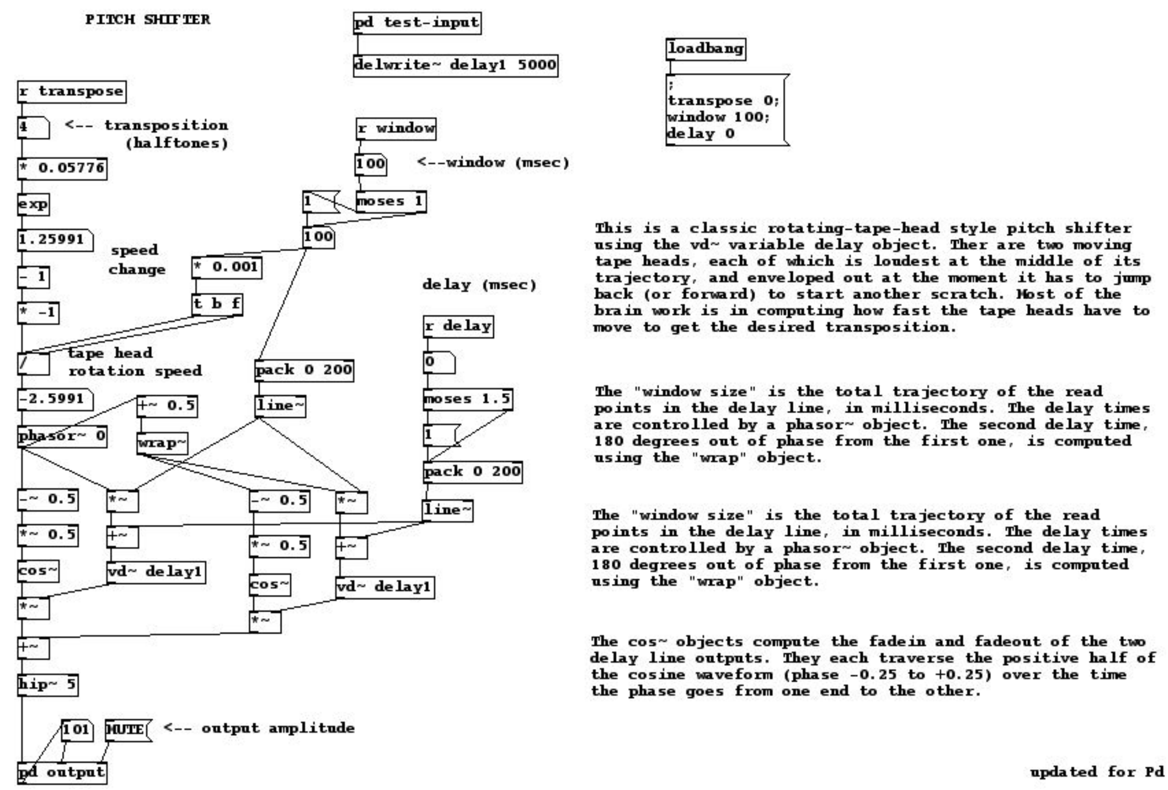

Fig. 4.6: *Patch de changement de hauteur par modulation de ligne à retard*

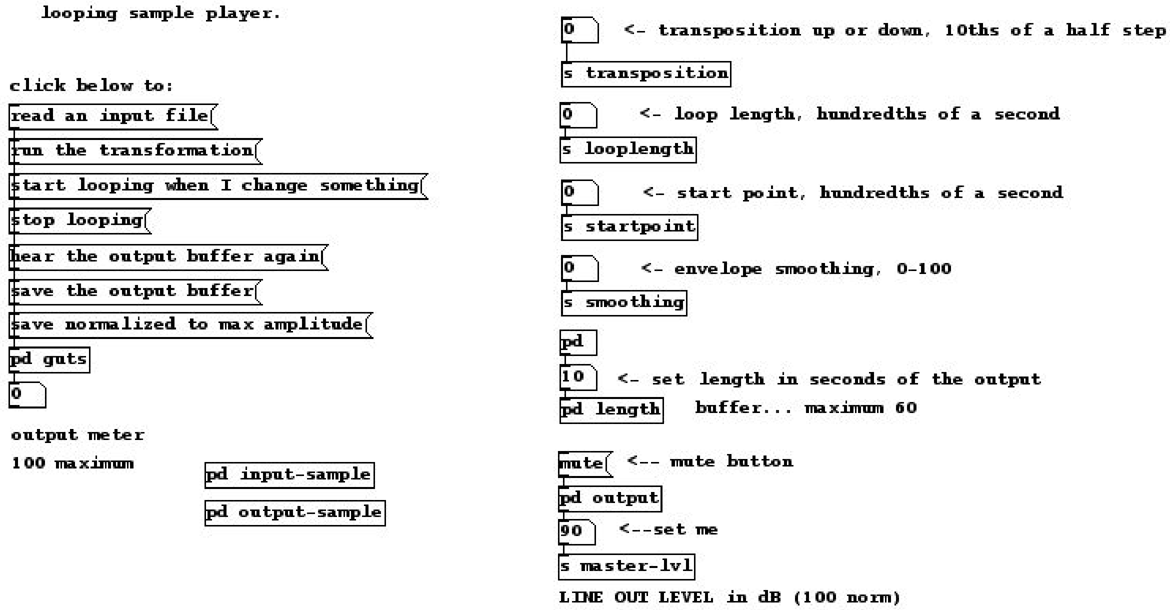

Fig. 4.7: *Patch d'échantillonnage (synthèse granulaire)*

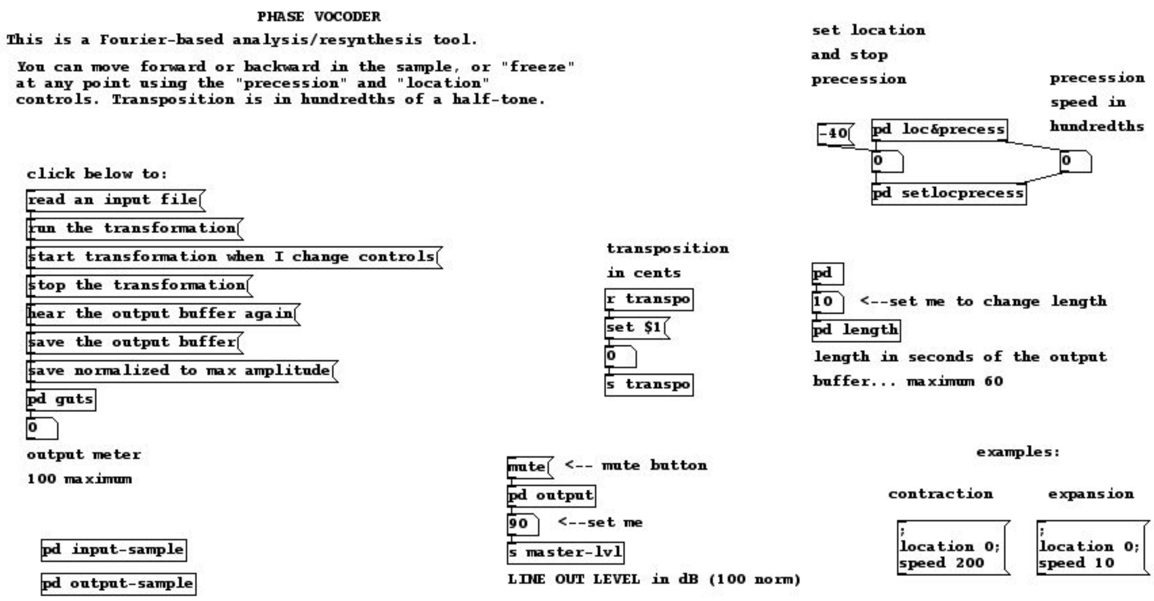

Fig. 4.8: *Patch de transposition et changement de durée par vocodeur de phase.*

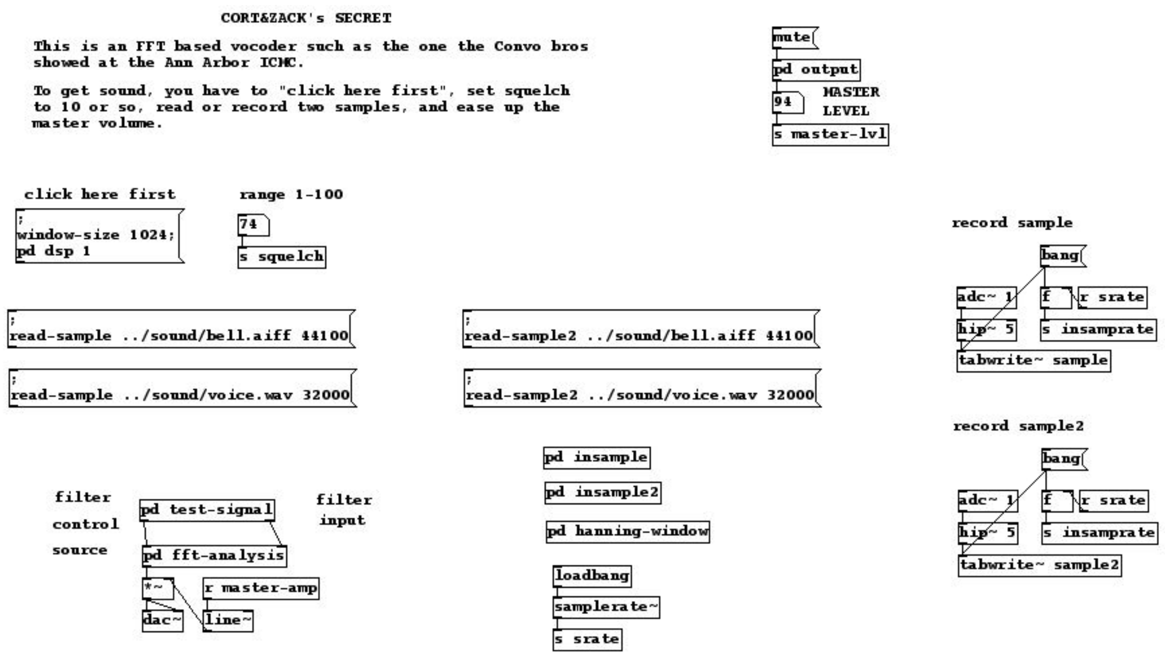

Fig. 4.9: *Patch de synthèse croisée par vocodeur de phase*

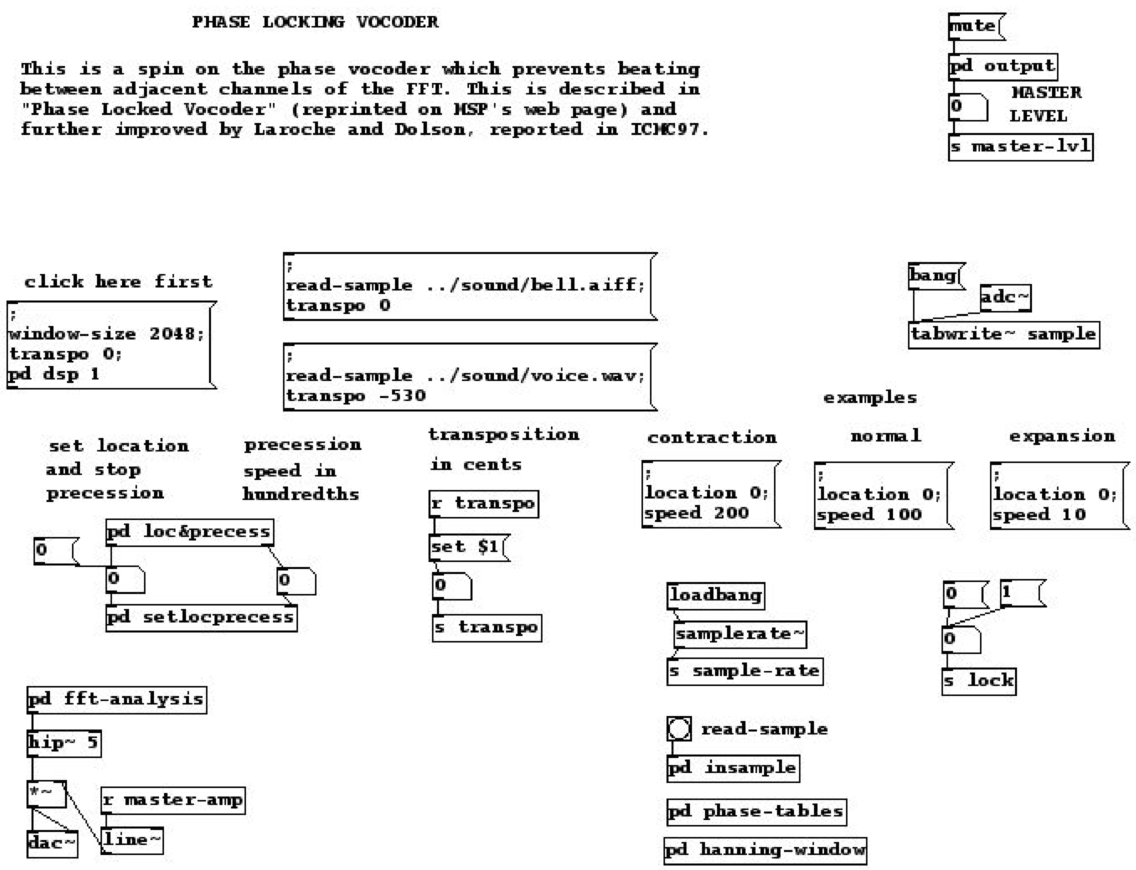

Fig. 4.10: *Patch de changement d'échelle temporelle par vocodeur de phase avec blocage de phase (Laroche, Dolson).*

Bibliographie

[1] Arfib D. Digital synthesis of complex spectra by means of multiplication of non linear distorded sine waves. *J. Audio Eng. Soc.*, 27:250–265, 1979.

[2] Arfib D., Couturier J. M., Kessous L., and Verfaille V. Strategies of mapping between gesture parameters and synthesis model parameters using perceptual spaces. *Organised Sound*, Mapping Strategies Issue in 2002.

[3] Music Technology Group. Sms, http://www.iua.upf.es/sms/, 2000.

[4] Music Technology Group. Clam, http://www.iua.upf.es/mtg/clam/, 2002.

[5] Maher J. and Beauchamp J. Fundamental frequency estimation of musical signals using a two-way mismatch procedure. *Journal of the Acoustic Society of America*, 95, 1993.

[6] McAulay R. J. and Quatieri T. F. Speech analysis/synthesis based on a sinusoidal representation. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 34(4), 1986.

[7] Henri Torgue Jean-François Augoyard, editor. *Répertoire des effets sonores*. Parenthèses, 1995.

[8] Desainte-Catherine M. and Marchand S. Structured additive synthesis: Towards a model of sound timbre and electroacoustic music forms. *Proc. of the International Computer Music Conference (ICMC'99)*, pages 260–3, 1999.

[9] Grey J. M. *An exploration of musical timbre*. PhD thesis, Stanford University, 1975.

[10] Le Brun M. Digital waveshaping synthesis. *J. Audio Eng. Soc.*, 27:757–768, 1979.

[11] Iverson P. Auditory stream segregation by musical timbre: Effects of static and dynamic acoustic attributes. *Journal of Experimental Psychology: Human Perception and Performance*, 21:751–763, 1995.

[12] McAdams S. and J.C. Cunibile. Perception of timbral analogies. In *Philosophical Transactions of the Royal Society*, volume series B 336, pages 383–389, London, 1992.

[13] Verma T., Levine S., and Meng T. Transient modeling synthesis: a flexible analysis/synthesis tool for transient signals, 1997.

[14] Verfaille V. and Arfib D. Adafx: Adaptive digital audio effects. In *Proceedings of the DAFx'01 Workshop in Limerick, Ireland*, December 2001.

[15] Serra X. and Smith J. O. A sound decomposition system based on a deterministic plus residual model. *Journal of the Acoustic Society of America, Supp. 1*, 89(1):425–434, 1990.

[16] Udo Zolzer, editor. *DAFX - Digital Audio Effects*. John Wiley & Sons, 2002.