Réalisation d'effets audionumériques adaptatifs en temps réel et hors temps réel

V. Verfaille
LMA, CNRS
31, chemin Joseph Aiguier
13402 Marseille Cedex 20
verfaille@lma.cnrs-mrs.fr

Résumé

Les paramètres de contrôle des effets audionumériques peuvent évoluer dans le temps, soit manuellement par l'utilisateur, soit automatiquement par un programme. Nous proposons une automatisation de l'effet à partir de paramètres extraits du son. Trois effets auto-adaptatifs sont ainsi proposés : un ralenti-accéléré sélectif, une robotisation et un écho granulaire adaptatifs.

Mots clefs

ADAFx, effets audionumériques, adaptatif, extraction de paramètres, mapping, automatisation, temps réel, Max/MSP, Matlab.

Introduction

Les effets audionumériques (DAFx[1], pour *Digital Audio Effects*) sont la plupart du temps non adaptatifs : ils sont appliqués avec les mêmes valeurs de contrôle tout au long du son. Des logiciels de montage et mixage permettent à l'utilisateur de manipuler ces paramètres de contrôle en temps réel, ce qui peut aussi se faire sur du matériel analogique (pédale Wha-wha, paramètres d'un filtre, gain d'un écho, etc). Les effets audionumériques adaptatifs (ADAFx) sont contrôlés par des paramètres extraits du son. Cela signifie qu'une extraction de paramètres du son ainsi qu'une mise en correspondance de ces paramètres avec les paramètres de contrôle est nécessaire. Les effets adaptatifs correspondent à une généralisation des effets audionumériques.

Effets audionumériques adaptatifs (ADAFx)

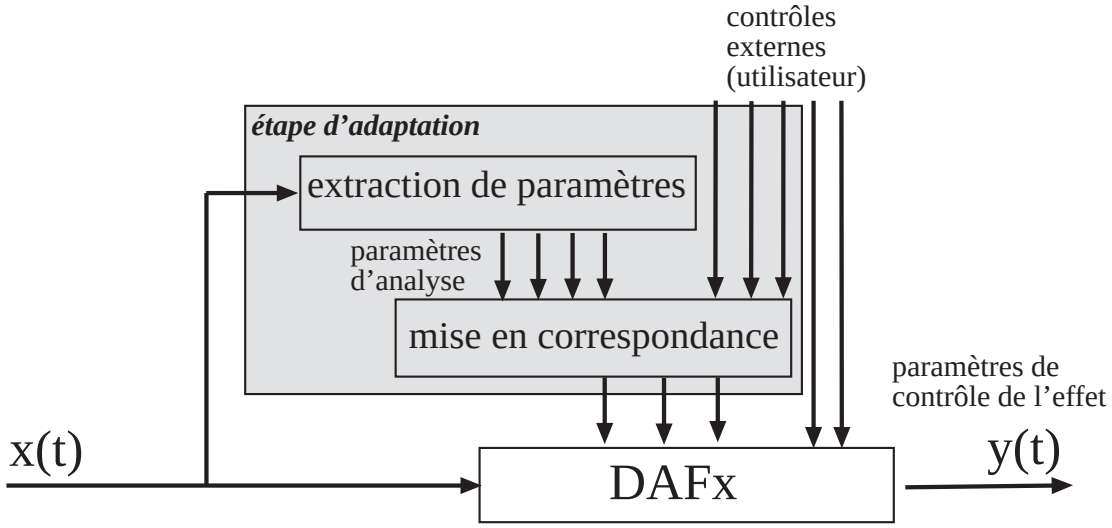
Les effets audionumériques adaptatifs sont des effets dont les paramètres de contrôle sont des courbes[2] extraites du son lui-même[3]. Le principe de ces effets est d'utiliser des paramètres propres au son pour automatiser les contrôles d'un effet. Les premiers exemples connus d'effets adaptatifs, connus sans cette appellation, sont des effets de contrôle de gain : connaissant la dynamique d'un signal, on la modifie en fonction de lois de correspondance. On obtient alors un limiteur ou un compresseur, selon la loi utilisée. Notre démarche consiste à généraliser cette idée à tous les effets possibles et imaginables afin de voir les apports musicaux possibles. Ces effets font partie de ce que l'on appelle « transformations et effets basés sur le contenu[4] ».
Afin de créer un effet adaptatif (cf. Exemple 1) il faut procéder en trois étapes :

1. L'extraction de paramètres ;
2. la mise en correspondance des paramètres extraits avec les paramètres de contrôle (*mapping*) ;
3. l'étape de transformation du signal (effet).

Nous présentons dans cet article trois effets : un écho granulaire adaptatif, un ralenti/accéléré temporel sélectif et une robotisation adaptative, les trois étant implémentés hors temps réel selon la technique du vocodeur de phase. Pour l'écho et la robotisation, une seconde version temps réel sous Max-MSP a été réalisée.
Ces effets vont être appliqués en particulier aux sons glissés (tout son comportant un changement rapide et continu de hauteur). Ces sons peuvent provenir de l'utilisation d'une technique instrumentale (vibrato, portamento, glissando) aussi bien qu'être directement une phrase musicale entière, ou un son électroacoustique. L'intérêt de ces sons est qu'ils comportent d'ores et déjà l'évolution d'un grand nombre de paramètres. De plus, en tant que sons, ils sont perceptivement intéressants pour la composition (par exemple : vibrato[5] ; transitions[6]).

---

[1] *DAFX - Digital Audio Effects*, Edited by Udo Zölzer, John Wiley & Sons, 2002.
[2] Arfib D., "Des courbes et des sons", *Recherches et applications en informatique musicale*, Paris, Hermès, 1998, pp. 277-286.
[3] Verfaille V., Arfib D., "A-DAFx: Adaptive Digital Audio Effects". *Proceedings of the DAFx01 Conference*, Limerick, 2001.
[4] Amatriain X., Bonada J., Loscos A., Arcos J. L. and Verfaille V., "Adressing the content level in audio and music transformations", submitted for a special issue of the *Journal of New Music Research.*
[5] Honing H., "The vibrato problem, comparing two solutions". *Computer Music Journal*, 19 (3), 1995.



Exemple 1 : Structure d'un effet adaptatif (ADAFx), avec $x(t)$ le signal d'entrée et $y(t)$ le signal de sortie. Les paramètres sont extraits du son, mis en correspondance avec les contrôles de l'effet.

Extraction de paramètres

On peut vouloir utiliser toutes sortes de paramètres afin de contrôler des effets. Les premiers qui viennent à l'esprit sont les critères psychoacoustiques (hauteur, sonie, brillance). Ces paramètres demandent un certain niveau d'abstraction (à la fois des connaissances sur le signal à court terme et à moyen terme, et des modèles de calcul performants), c'est pourquoi nous utilisons aussi des paramètres de plus bas niveau, plus aisément calculables en temps réel. Ainsi, des paramètres de signal tels que l'énergie, le centre de gravité spectrale, l'indice de voisement, la balance grave/aigu nous intéressent. De plus, des paramètres issus d'une analyse additive[7] sont utilisés : fréquence et module de la fondamentale ou d'une autre harmonique, voire d'un partiel, harmonicité, synchronisme des harmoniques[8], balance des harmoniques paires/impaires, énergie de la partie déterministe, de la partie transitoire[9], du résidu, etc. Nous utilisons aussi des paramètres relatifs à l'analyse de signaux en vue de leur segmentation[10]. En Exemple 2, nous pouvons voir l'évolution de six de ces paramètres extraits d'une voix.

Mise en correspondance des paramètres extraits du son et des contrôles d'un effet

L'évolution dans le temps des paramètres extraits du son est décrite à l'utilisateur au moyen de courbes. Les fonctions de mise en correspondance, ou *mapping*, que l'on utilise pour transformer ces paramètres du son en paramètres de contrôle de l'effet sont simples, explicites, statiques. Elles sont toutes constituées selon le schéma suivant : application d'une non-linéarité, combinaison linéaire, application d'une seconde non linéarité, calibrage.

---

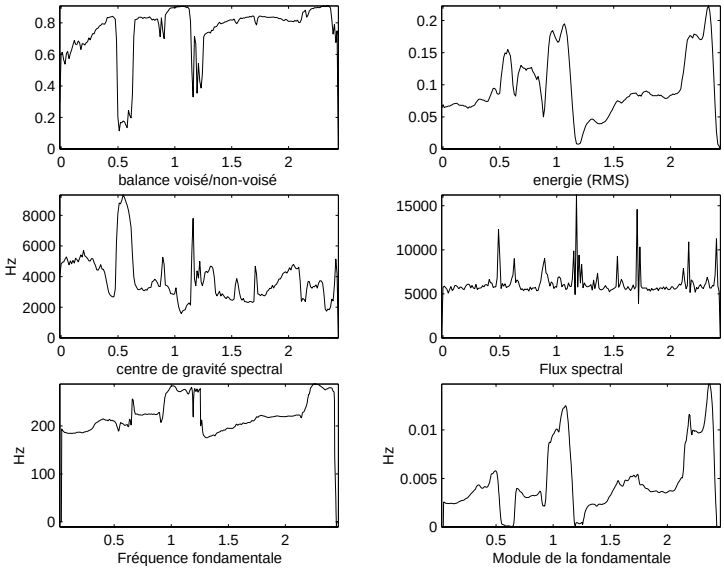
[6] Strawn J., "Analysis and Synthesis of Musical Transitions Using the Discrete Short-Time Fourier Transform", *Journal of the Audio Engineering Society*, volume 35, number 1/2, pp. 3-13, 1987.

[7] Serra X., "Musical Sound Modeling With Sinusoids Plus Noise", published in C. Roads, S. Pope, A. Picialli, G. De Poli, editors; ``Musical Signal Processing", Swets & Zeitlinger Publishers, 1997.

[8] Dubnov S. and Tishby N., ``Testing For Gaussianity and Non Linearity In The Sustained Portion Of Musical Sounds.", *Proceedings of the Journees Informatique Musicale*, 1996.

[9] Verma T., Levine S., and Meng T., ``Transient modeling synthesis: a flexible analysis/synthesis tool for transient signals", *Proceedings of the ICMC*, Greece, 1997.

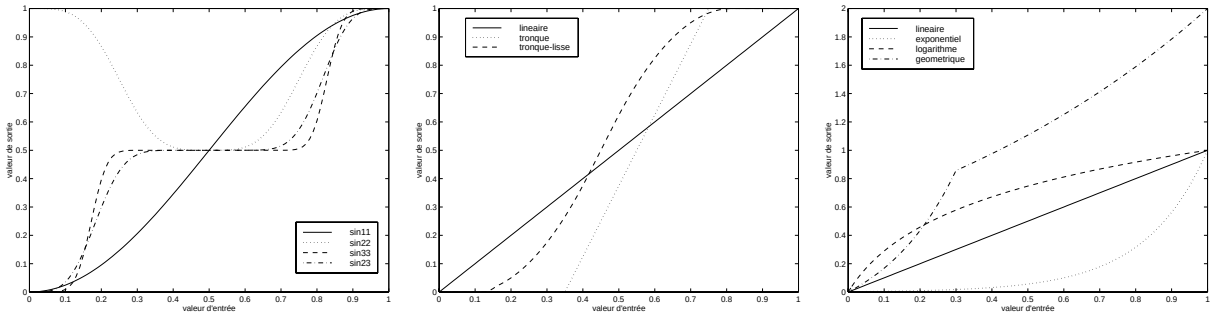
[10] Rossignol S., Rodet X., Soumagne J., Collette J.-L. and Depalle P., "Feature Extraction and Temporal Segmentation of Acoustic Signals", *Proceedings of the ICMC*, 1998.



Exemple 2 : quelques courbes de paramètres extraits d'une voix. De haut en bas et de gauche à droite : indice de voisement/non-voisement, énergie, centre de gravité spectrale, flux spectral, fréquence et module de la fondamentale.

Considérons K paramètres notés $G_k(t)$, $t=1,...,T$ $k=1,...,K$. La première étape consiste à appliquer une fonction linéaire ou non M_i à chacun de ces paramètres : on obtient $F_k(t)=M_{i_k}(G_k(t))$. Les différentes fonctions de *mapping* M_i que nous utilisons sont (cf. Exemple 3) :

- $M_1(F(t))=F(t)$ pour le *mapping* linéaire ;
- $M_2(F(t),\mu,\varepsilon)=\dfrac{1+\sin^{\mu}(...(\sin^{\mu}(\pi F(t)-0.5))...)}{2}$ pour le *mapping* sinusoïdal, avec μ la puissance et ε l'ordre (le nombre d'application de la fonction à elle-même) ;
- $M_3(F(t))=\dfrac{t_m H_{F(t)<t_m}+F(t)H_{t_m\le F(t)\le t_M}+t_M H_{F(t)>t_M}}{t_M-t_m}$ pour la troncature, avec H_a la fonction de Heaviside (dont la valeur est **1** si le résultat du test **a** est "vrai" et **0** sinon), et $[t_m;t_M]\in[0;1]$ l'intervalle de troncature ;
- $M_4(F(t))=(s_m)^{1-\frac{F(t)}{\alpha}}.H_{F(t)\le\alpha}+(s_M)^{\frac{(F(t)-\alpha)}{(1-\alpha)}}.H_{F(t)>\alpha}$ pour le *mapping* géométrique en deux parties, utilisé par le ralenti-accéléré sélectif, avec $s_m\le 1$ le facteur de contraction (accélération) et $s_M\ge 1$ le facteur de dilatation ou de ralenti (le facteur $\alpha\in[0;1]$ divise le segment $[0;1]$ en deux : la portion inférieure $[0;\alpha]$ est contractée, la portion supérieure $[\alpha;1]$ dilatée) ;
- $M_5(F(t))=10^{\delta(F(t)-\beta)}$ pour le *mapping* « exponentiel », avec δ est le facteur multiplicatif et β le facteur additif ;
- $M_6(F(t))=\log_{10}(\delta F(t)+\beta)$ pour le *mapping* logarithmique ;
- $M_7(F(t))=\begin{cases}\dfrac{1}{2\lambda-1}\displaystyle\sum_{k=-\lambda}^{\lambda}F(t+k), & t=\lambda+1,...,T-\lambda\\ \dfrac{1}{2t-1}\displaystyle\sum_{k=1}^{2t-1}F(k), & t=1,...,\lambda\\ \dfrac{1}{2(T-t+1)-1}\displaystyle\sum_{k=T-2t+1}^{T}F(k), & t=T-\lambda+1,...,T\end{cases}$ pour le lissage, avec λ l'ordre du lissage.



Exemple 3 *: Fonctions de mise en correspondance (mapping) :*
i) à gauche : sinusoïdes, $\sin_{1,1}$, $\sin_{2,2}$, $\sin_{3,3}$ *et* $\sin_{2,3}$
ii) au milieu : linéaire, tronqué de $t_m = 0.35$ *à* $t_M = 0.75$ *; tronqué de* $t_m = 0.25$ *à* $t_M = 0.65$ *puis lissé, d'ordre 14 ;*
iii) à droite : exponentiel avec $\delta = 2{,}5$ *et* $\beta = 1$, *logarithmique avec* $\delta = 10$ *et* $\beta = 1$ *; géométrique, contracté de* $s_m = 1/4$ *à 1 pour une valeur de contrôle entre 0 et* $\alpha = 0.35$, *et dilatée de 1 à* $s_M = 2$ *pour une valeur de contrôle entre* $\alpha = 0.35$ *et 1.*

La deuxième étape consiste à effectuer une combinaison linéaire des K paramètres après normalisation entre 0 et 1 (ou entre -1 et 1). En notant $F_k^M = \max_{t\in[1;T]}|F_k(t)|$ et $F_k^m = \min_{t\in[1;T]}|F_k(t)|$, nous obtenons la courbe :

$$F_g(t) = \frac{1}{\sum_{k=1}^{K}\gamma_k}\sum_{k=1}^{K}\gamma_k \frac{F_k(t) - F_k^m}{F_k^M - F_k^m}$$ avec γ_k le facteur de pondération du k^{ème} paramètre.

La troisième étape consiste à appliquer une fonction M_i linéaire ou non à la courbe pondérée, afin d'obtenir une nouvelle courbe, dont certaines caractéristiques sont ainsi mises en valeur.
La quatrième et dernière étape consiste à ajuster la courbe aux bornes du paramètre de contrôle Δ_m et Δ_M. La valeur du paramètre de contrôle de l'effet est alors donnée par :

$$\Delta(t) = \Delta_m + (\Delta_M - \Delta_m).M_i(F_g(t)), \quad t = 1,...,T$$

Ce *mapping* permet de prendre en compte différentes caractéristiques du son en même temps, afin de piloter un paramètre de contrôle. L'utilisation de la fonction non linéaire permet de focaliser le contrôle sur certaines particularités de chacun des paramètres extraits. Par exemple, la fonction sinus accroît la proximité de 0 et de 1, et rend la courbe plus saillante : l'appliquer plusieurs fois permet d'augmenter la saillance ; le *mapping* tronqué permet de se concentrer sur une portion du paramètre dans laquelle il varie d'une manière qui semble plus intéressante à l'utilisateur. Le *mapping* géométrique en deux parties concerne l'effet de ralenti-accéléré et permet de choisir quelle partie du son est contractée, et quelle partie et dilatée.

Effets implémentés et résultats

Nous avons implémenté plusieurs effets adaptatifs, dont certains ont déjà été présentés[11]. Nous nous focalisons dans cet article sur trois d'entre eux : le ralenti/accéléré sélectif, la robotisation et l'écho granulaire adaptatifs. Ils ont été implémentés hors temps réel à l'aide de la technique du vocodeur de phase et en temps réel sous Max/MSP (pour les deux derniers).
Pour chaque effet hors temps réel, le son est lu grain après grain avec un pas donné. Le grain est ensuite fenêtré. On calcule la transformée de Fourier à court terme (TFCT) à l'aide d'une FFT, ainsi que les paramètres que l'on désire extraire pour piloter l'effet. Le grain de synthèse est ensuite calculé en fonction de l'algorithme de l'effet, on lui applique une fenêtre, on superpose avec un pas donné les fenêtres successives, tout en normalisant le tout si le pas ou la taille du grain est variable. Cette normalisation se fait hors temps réel en calculant une enveloppe constituée des puissances des fenêtres de synthèse (avec la même méthode *overlap add*). A la fin du traitement, le son résultant est corrigé échantillon par échantillon en fonction de cette enveloppe.

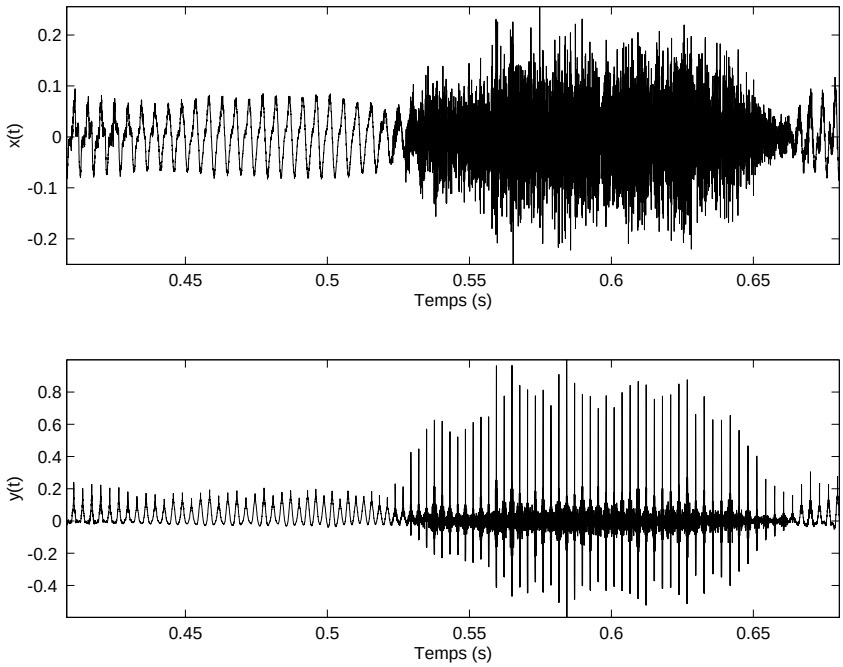
[11] Op. cit. Verfaille, Arfib 2001 ; ainsi que : Arfib D., Couturier J.M., Kessous L., Verfaille V., "Strategies of mapping between gesture parameters and synthesis model parameters using perceptual spaces.", submitted for a special issue on Mapping Strategies in *Organised Sound*.

Ralenti-accéléré sélectif

Pour le ralenti-accéléré sélectif, le paramètre de contrôle utilisé est le facteur d'étirement. Le pas de synthèse est fixe, et l'on modifie le pas d'analyse en fonction du paramètre de contrôle. Après fenêtrage du grain d'entrée, et en fonction du facteur d'étirement, on calcule le pas d'analyse, on déroule la phase de façon à la faire correspondre à la valeur ad hoc du grain de synthèse, puis on applique une fenêtre au grain résultant, et on l'ajoute en superposition (*overlap-add*) au son de synthèse.
Le facteur d'étirement peut varier entre une valeur inférieure à 1 et une valeur supérieure à 1, ce qui permet autant un ralenti qu'un accéléré temporel. Les artefacts du vocodeur de phase pour cet effet ne sont pas évités ; cependant, notre propos n'est pas de fournir le meilleur algorithme de ralenti-accéléré, mais bien de montrer l'intérêt de l'automation d'effets par des paramètres extraits du son.
Un ralenti-accéléré ne peut se faire en temps réel que dans le cadre de la synchronisation en temps réel d'une bande enregistrée sur un geste (métronome, baguette d'un chef d'orchestre). Nous ne l'avons pas encore réalisé. Hors temps réel, en utilisant par exemple un indice de voisement, on peut ralentir les parties voisées d'une phrase prononcée ou chantée par un humain, tout en conservant les consonnes, et ainsi proposer un ralenti qui conserve l'intelligibilité du texte. De manière plus générale, cet effet permet de réinterpréter une phrase musicale et d'en donner plusieurs versions.

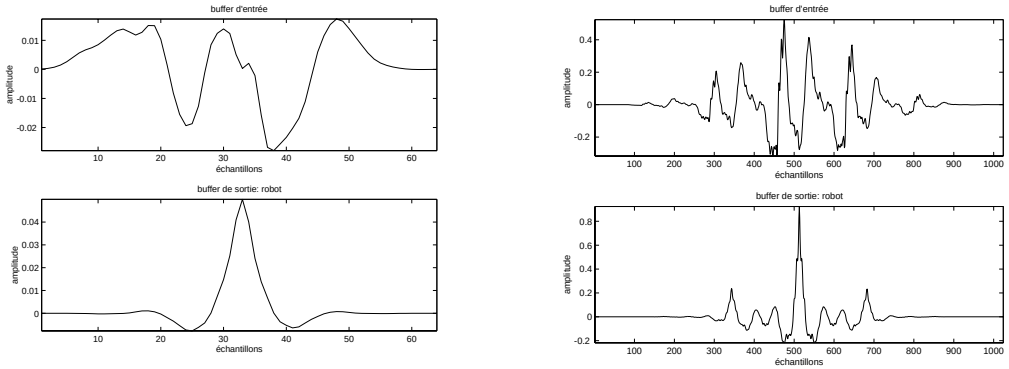
Robotisation adaptative

La robotisation "classique" consiste à synchroniser les phases de la TFCT d'un grain au centre du grain. Ceci s'opère en donnant à la phase de chaque *bin* (panier) fréquentiel de la TFCT la valeur 0, en ayant préalablement modifié la fenêtre[12]. Le grain reconstruit présente alors un ensemble de pics (cf. Exemple 4). On peut utiliser de petites fenêtres, de façon à n'en garder qu'un, ou de grandes fenêtres (cf. Exemple 5). Pour de grandes fenêtres, on remarque une combinaison d'effets : les transitoires sont amollis et une sorte de filtrage en peigne (dû à la coexistence de l'ancienne hauteur avec la hauteur imposée par l'effet) apparaît. L'information formantique présente dans le signal est globalement conservée, même si elle est localement lissée (cf. Exemple 6). Par contre, l'information de hauteur est gommée pour les petites fenêtres, ce qui permet d'en imposer une autre, en fonction du pas de synthèse (qui correspond à la période de la fondamentale). En changeant la valeur de ce pas, on obtient une voix de robot dont la hauteur change. L'implémentation en temps réel pose l'unique problème suivant : la gestion de *buffer* (mémoire tampon) de petite taille dans un autre *buffer* de plus grande taille. Les solutions à ce problème ne sont pas abordées dans cet article.

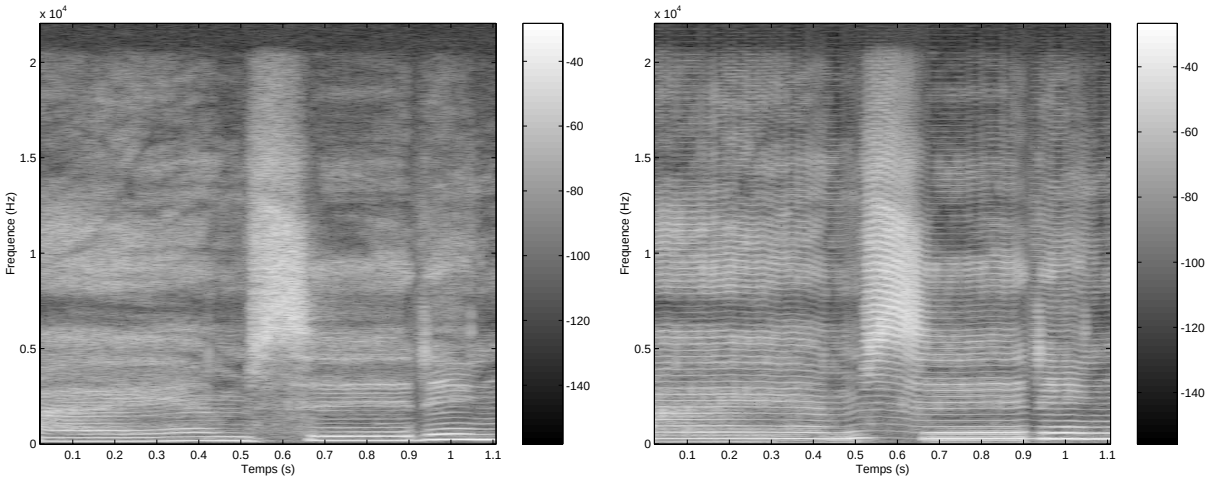


Exemple 4 : Exemple de robotisation, avec une fenêtre de 256 échantillons : signal d'entrée (en haut) et signal obtenu par robotisation adaptative (en bas).

[12] Op. cit. Serra, 1997.



Exemple 5 : Robotisation : pour un grain de 64 échantillons, un seul pic est créé (à gauche : grain avant transformaton en haut et après, en bas) ; pour 1024 échantillons (à droite), plusieurs pics sont créés, dont un prépondérant, qui correspond à celui qui apparaît lors de l'utilisation de la fenêtre de 64 échantillons.

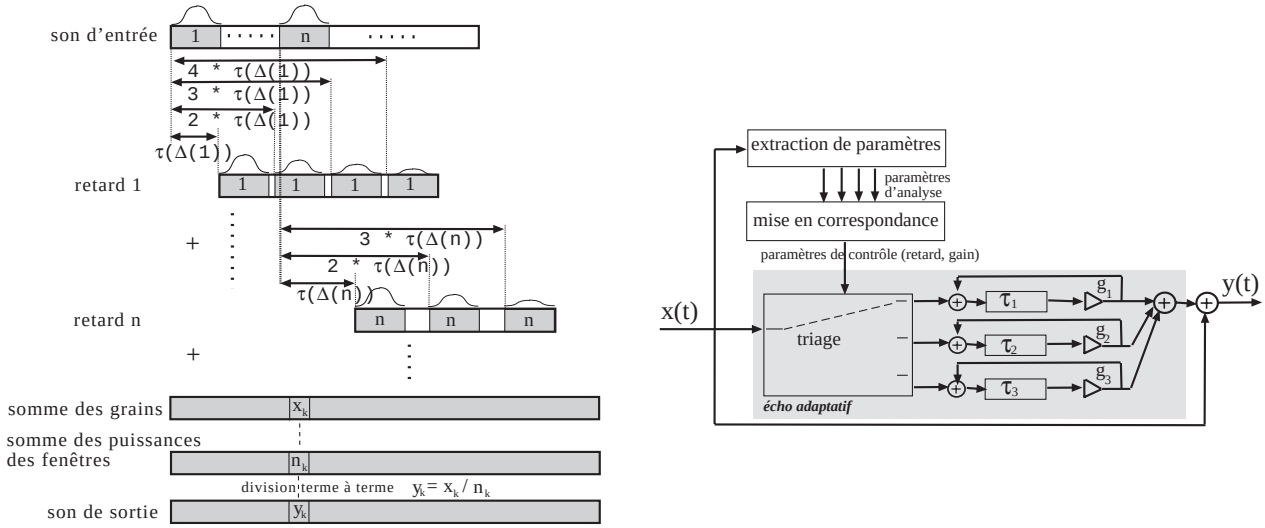


Exemple 6 : Sonagramme d'une robotisation adaptative (grain de 256 échantillons) : signal d'entrée (en haut) et de sortie (en bas) : l'information formantique est conservée, la hauteur est imposée par l'algorithme.

Cet effet permet, en utilisant en son d'entrée une voix normale, de la rendre robotique tout en conservant sa hauteur, ou en imposant une hauteur (donnée par un autre paramètre du son). La voix est en quelque sorte réinterprétée par un robot !

Echo granulaire adaptatif

L'écho granulaire adaptatif est un écho dont le temps de retard τ et le gain g peuvent varier : le grain de sortie est identique au grain d'entrée, mais recopié **k** fois avec un temps de retard τ et avec un facteur de gain $G(k) = g^k, \quad g \in [0;1[$. Le diagramme de cet effet hors temps réel est donné (cf. Exemple 7, gauche).



Exemple 7 : écho granulaire adaptatif : chaque grain est copié et déplacé, avec un facteur de gain et un temps de retard donné par des courbes de contrôle $\Delta(t)$ (diagramme de gauche). L'implémentation en temps réel de cet effet nécessite un nombre limité de lignes à retard (à droite).

L'implémentation hors temps réel correspond à la situation où le gain ainsi que le retard peuvent prendre toutes les valeurs données par leurs courbes respectives. Cependant, pour une implémentation en temps réel, on ne peut implémenter un nombre infini de lignes à retard. Il faut alors discrétiser les courbes des paramètres de contrôle, puis implémenter un ensemble de lignes à retard de longueurs et de gains différents, dans lesquelles on fera entrer les grains. Le « triage » se fait en fonction des courbes de contrôle (cf. Exemple 7 droite). Nous avons utilisé une discrétisation uniforme ainsi qu'une discrétisation non uniforme (prenant en compte les plateaux et les pics et creux extrêmes de la courbe à discrétiser).
A l'aide d'un tel écho adaptatif, dans chacune des deux versions, on peut rendre plus ou moins présentes certaines composantes du son. Par exemple, on peut répéter les attaques seules d'un son, et ne pas répéter la partie harmonique. On peut aussi utiliser un transducteur gestuel avec la version temps-réel afin de passer d'un réglage d'écho adaptatif à un autre de manière continue, et offrir ainsi un niveau de contrôle plus élevé.

Conclusions, perspectives

Les effets adaptatifs, implémentés sous Matlab hors temps réel et sous Max/MSP en temps réel, sont une généralisation des effets existants. Leur grand intérêt réside dans la possibilité de réinterpréter une phrase musicale, en changeant le timbre, la durée des notes ou la présence d'événements musicaux. Leur rendu perceptif est très explicite et permet des changements fins dans les propriétés d'un son. Leur application reste cohérente avec le son, puisque le contrôle de l'effet se réalise en fonction de paramètres du son lui-même.

Nous implémentons actuellement d'autres effets adaptatifs, des transformations sur le timbre ainsi que des effets adaptatifs croisés (l'effet est appliqué sur un son au moyen d'un paramètre de contrôle extrait d'un second son ou d'un geste physique), et leur manipulation en temps réel au moyen de transducteurs gestuels.