

Enseignement d'approfondissement

BELARDI Stéphane, JODER Cyril

4 décembre 2005

Synthèse de signaux percussifs

Table des matières

I	Introduction	3
II	Modélisation mathématique d'un signal percussif	4
1	Modélisation du phénomène	4
1.1	Les filtres AR(2)	4
1.2	Calcul de la réponse impulsionnelle	5
2	Déroulement théorique du travail	7
III	Réalisation en pratique	8
1	Détermination des différentes fréquences	8
2	Détermination de l'évolution de l'amplitude de la fréquence	10
3	Détermination du facteur ρ_i	10
IV	Les résultats obtenus	12
1	Synthèse sur différents instruments	12
1.1	Triangle	12
1.2	Cloche	12
1.3	Piano	13
1.4	Cymbale	13
1.5	Gong	13
2	Des idées d'amélioration	14
2.1	Les fréquences aigües	14
2.2	Amplitude de la fréquence	14
V	Conclusion	16

Deuxième partie

Modélisation mathématique d'un signal percussif

Une percussion peut être raisonnablement approximée par une excitation infiniment forte pendant une durée nulle. Cette excitation "contient" toutes les fréquences du spectre. On considère alors que le son produit par une percussion est dû à la résonance sur un temps plus ou moins long d'un petit nombre de fréquences. Les paramètres de ces résonances dépendent des propriétés physiques des matériaux qui s'entrechoquent. Cela correspond donc à un filtrage.

On modélisera alors l'excitation par un dirac en temps $t = 0$. On peut ensuite supposer qu'une fréquence ¹ résonne en s'atténuant. Le modèle choisi pour rendre compte de ce phénomène est celui d'un filtre auto-régressif d'ordre 2 ou *filtre AR(2)*. Dans ce modèle, l'atténuation est en exponentielle (les détails sont explicités plus loin).

Un son percussif sera donc modélisé ici comme une excitation de dirac filtrée par un banc de filtres AR(2), chaque filtre correspondant à une fréquence de résonance.

1 Modélisation du phénomène

1.1 Les filtres AR(2)

Les filtres AR(2) ont un comportement très simple. Si on nomme $s[n]$ le signal résultant du filtrage de l'excitation $g[n]$ par un filtre AR(2), on a :

$$s[n] = a_1 s[n-1] + a_2 s[n-2] + Ag[n]$$

où A , a_1 et a_2 sont les paramètres du filtre.

On a donc :

$$s[n] - a_1 s[n-1] - a_2 s[n-2] = Ag[n]$$

et dans le domaine fréquentiel :

$$(1 - a_1 e^{-i\omega} - a_2 e^{-2i\omega}) \hat{s}(e^{i\omega}) = A \hat{g}(e^{i\omega})$$

soit :

$$\hat{s}(e^{i\omega}) = \frac{A}{1 - a_1 e^{-i\omega} - a_2 e^{-2i\omega}} \hat{g}(e^{i\omega})$$

d'où on déduit la fonction de transfert du filtre :

$$\hat{h}(e^{i\omega}) = \frac{A}{1 - a_1 e^{-i\omega} - a_2 e^{-2i\omega}}$$

On veut évidemment n'utiliser que des filtres causaux et stables. La condition nécessaire et suffisante est que les pôles de la transformée en z soient de modules strictement inférieurs à 1. Donc, il faut et il suffit que les racines de l'équation :

$$z^2 - a_1 z - a_2 = 0 \tag{1}$$

soient de module strictement inférieur à 1.

¹Dans un premier temps, une seule fréquence. Bien sûr, par la suite, pour améliorer la qualité du son synthétisé, on utilisera plusieurs fréquences différentes

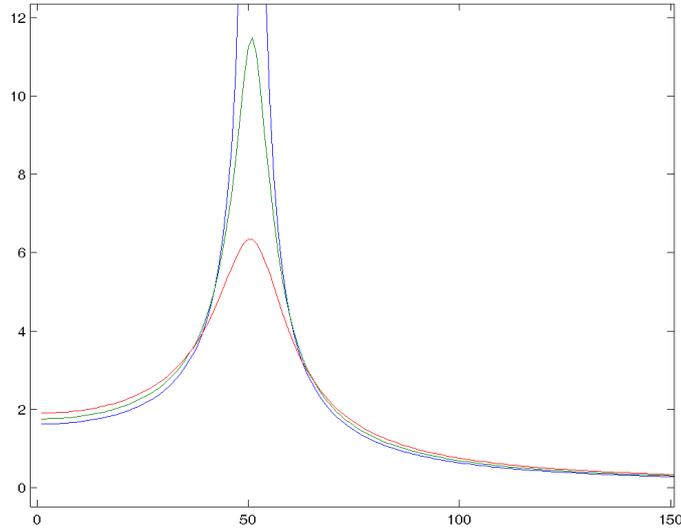


FIG. 1 – Fonctions de transfert à A et ω_0 fixés, avec $1 > |\rho_{rouge}| > |\rho_{vert}| > |\rho_{bleu}|$

1.2 Calcul de la réponse impulsionnelle

Pour calculer la réponse impulsionnelle, on utilise le fait que celle-ci est le signal de sortie du filtre excité par un dirac.

On a donc, pour tout n :

$$h[n] = a_1 h[n-1] + a_2 h[n-2] + A\delta[n]$$

Comme le filtre est causal, on a :

$$\begin{cases} h[n] = 0 & \text{si } x < 0 \\ h[0] = A \\ h[n] - a_1 h[n-1] - a_2 h[n-2] = 0 & \text{si } x < 0 \end{cases}$$

On peut donc voir $(h[n])_{n \geq 0}$ comme une suite récurrente d'ordre 2.

On tombe alors sur l'équation (1) de précédemment. On sait donc que les racines seront de modules strictement inférieurs à 1.

– Si les racines sont réelles $\{x_1, x_2\}$, les suites solutions de l'équation (1) sont de la forme :

$$h[n] = \begin{cases} k_1 x_1^n + k_2 x_2^n & \text{si } x_1 \neq x_2 \\ k_1 x_1^n + k_2 n x_1^n & \text{si } x_1 = x_2 \end{cases}$$

Ces deux formes de solutions forment des signaux qui ne sont pas intéressants dans notre étude car soit ils sont inaudibles, soit ils forment des "clics", qui n'ont pas de fréquence de résonance, et ne rentrent donc pas dans le modèle étudié.

- Le cas qui nous intéresse est donc celui où les deux racines sont complexes conjuguées $\{\rho e^{i\omega_0}, \rho e^{-i\omega_0}\}$. La suite récurrente est donc de la forme :

$$h[n] = k_1 \rho^n e^{ni\omega_0} + k_2 \rho^n e^{-ni\omega_0}$$

Les conditions initiales sont :

$$\begin{cases} h[-1] = 0 = \rho^{-1}(k_1 e^{-i\omega_0} + k_2 e^{i\omega_0}) \\ h[0] = A = k_1 + k_2 \end{cases}$$

Un calcul rapide donne :

$$\begin{cases} k_1 = \frac{A}{1 - e^{-2i\omega_0}} = \frac{A e^{i\omega_0}}{2i \sin(\omega_0)} \\ k_2 = \frac{A}{1 - e^{2i\omega_0}} = \frac{-A e^{-i\omega_0}}{2i \sin(\omega_0)} \end{cases}$$

On en déduit alors la réponse impulsionnelle du filtre :

$$h[n] = \frac{A}{\sin \omega_0} \rho^n \sin(\omega_0(n + 1))$$

Cette réponse est donc une sinusoïde (qui oscille à la fréquence de résonance)

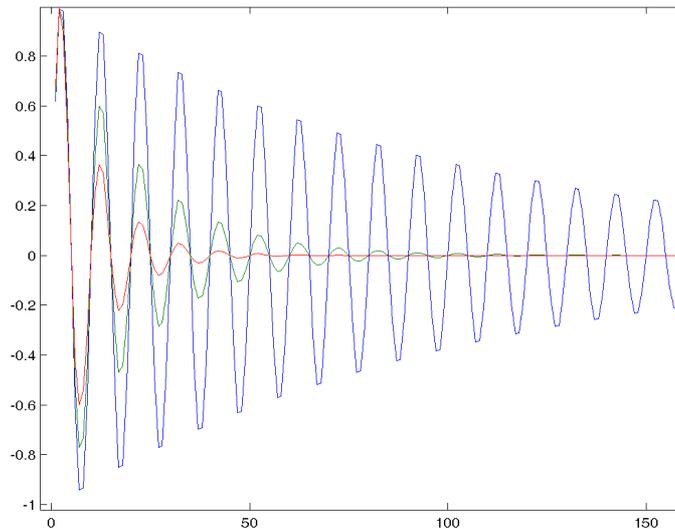


FIG. 2 – Réponses impulsionnelles des courbes précédentes

modulée en amplitude par une exponentielle décroissante, conformément aux hypothèses formulées plus haut. $\frac{A}{\sin(\omega_0)}$ représente alors l'amplitude initiale du son, ρ est le facteur d'amortissement ($\rho < 1$). Enfin, ω_0 est la pulsation du son. Notons que ρ est aussi lié à la "largeur" du pic de la fonction de transfert en ω_0 . Plus ρ est proche de 1, moins le son s'amortit et plus le pic est "étroit", et inversement.

2 Déroulement théorique du travail

Pour chaque son percussif, il suffira donc en théorie d'effectuer les opérations suivantes :

- A partir d'un son réel, et pour chaque fréquence de résonance ω_i
 1. mesurer exactement cette fréquence ω_i
 2. mesurer l'amplitude de cette fréquence à des temps successifs ²
 3. en déduire le facteur d'amortissement ρ_i et l'amplitude $A'_i = \frac{A_i}{\sin(\omega_i)}$
- Sauvegarder ces coefficients $(\omega_i, \rho_i, A'_i)_i$
- Synthétiser une percussion à l'aide de la formule suivante :

$$R[n] = \sum_{i=1}^n (A'_i \rho_i^n \sin(n+1)\omega_i)$$

²On considère dans cette partie que les fréquences de résonance sont assez éloignées pour que l'amplitude mesurée corresponde uniquement au filtre considéré, et que l'influence des filtres voisins soit négligeable. Cette amplitude doit alors être une fonction exponentielle décroissante du temps.

Troisième partie

Réalisation en pratique

Essayons maintenant de réaliser en pratique le déroulement établi dans le paragraphe précédent. On se heurte alors à plusieurs problèmes :

- Un son réel comporte toujours du bruit : comment distinguer les fréquences utiles des fréquences dues au bruit ?
- La mesure exacte d'une fréquence est assez compliquée : il faudra écrire un algorithme de détection de pics, qui induira inévitablement des erreurs.
- Une fois la fréquence bien déterminée, la mesure de son amplitude en plusieurs points n'est pas difficile. Mais il faut savoir jouer avec la précision de chaque point et le nombre de points (qui doit être suffisant pour effectuer l'étape suivante).
- La déduction du paramètre ρ_i n'est pas évidente : il faut élaborer un algorithme de régression exponentielle.
- L'étape de synthèse est elle sans problème particulier.

1 Détermination des différentes fréquences

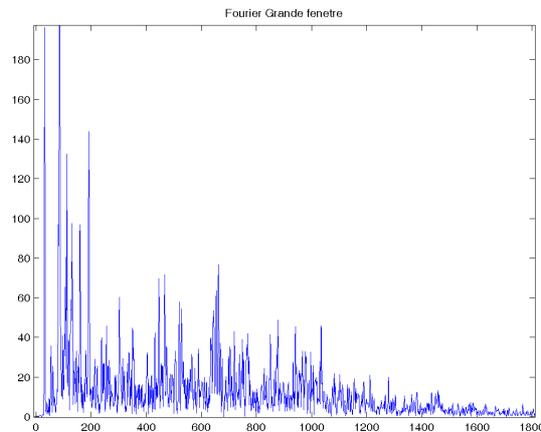


FIG. 3 – Exemple de transformée de Fourier d'un son "réel" : le spectre est bruité

On peut supposer que le bruit a une amplitude suffisamment faible par rapport au son enregistré. Ainsi, les fréquences du bruit ont une basse amplitude, et on ne travaille que sur les fréquences dont l'amplitude est supérieure à un certain seuil. Notre méthode a été la suivante :

- On ne travaille bien sûr avec uniquement la première moitié de la FFT, la deuxième partie étant redondante.
- On détermine la variance moyenne locale.
- Dès qu'un point vérifie les conditions suivantes :
 - Maximum local

- Amplitude au-dessus d'un certain seuil (que l'on fait varier en fonction de la fréquence)
- Variance en ce point supérieure à `tolérance` multiplié par la variance moyenne locale. (On prend `tolérance` de l'ordre de 1.5).
- Alors on sauvegarde son abscisse (fréquence) et son ordonnée (son amplitude)

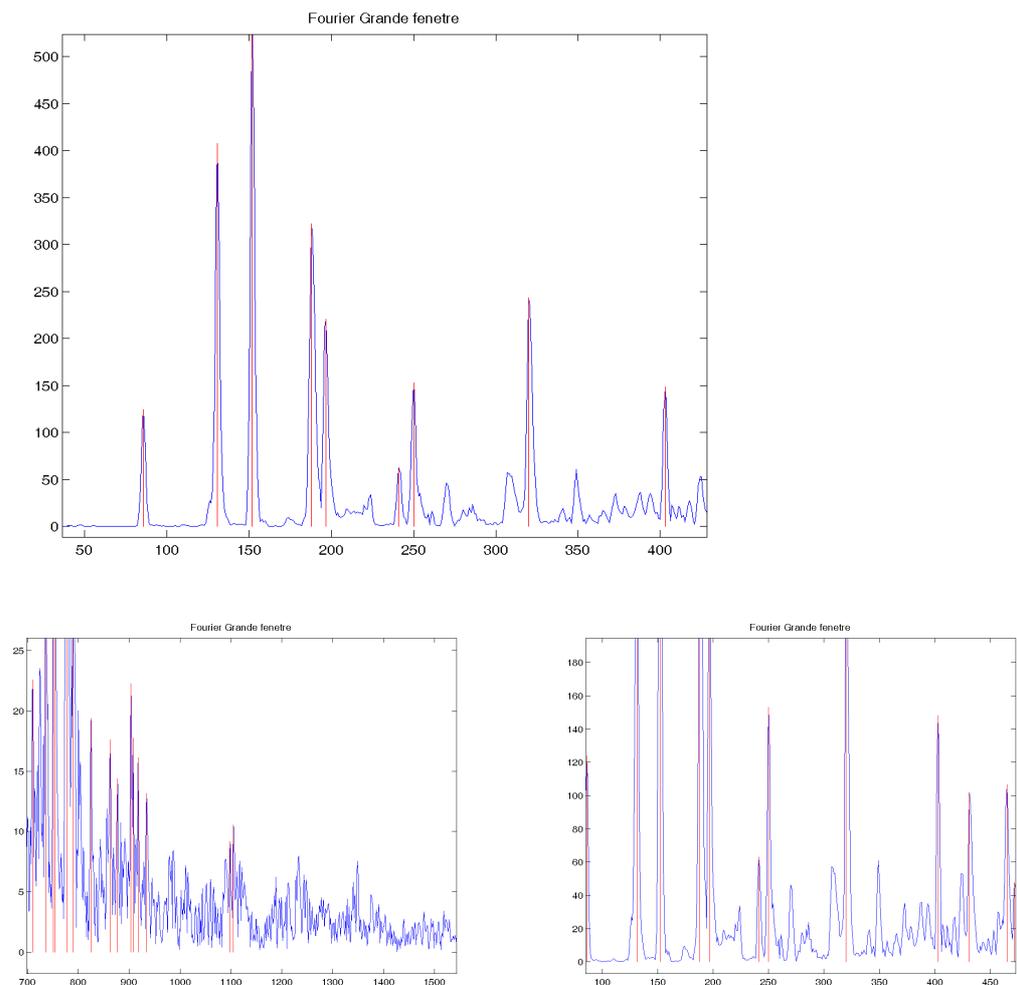


FIG. 4 – Exemples de détection de fréquences. Sur ces figures, les traits verticaux rouges représentent les fréquences détectées. On voit que certaines fréquences aigües (à gauche) ou que des fréquences entourées par des fréquences de trop grande importance (à droite) ne sont pas détectées.

Nous prenons donc le son réel, fenêtré avec une grande fenêtre (de l'ordre de 8000 ou 16000 points), puis nous appliquons notre algorithme `pics2` pour en retirer les fréquences sur lesquelles nous travaillerons ensuite.

2 Détermination de l'évolution de l'amplitude de la fréquence

Une fois que l'on connaît précisément la fréquence ω_i du filtre que l'on étudie, on cherche à déterminer les autres paramètres ρ_i et A_i . Pour cela, il faut connaître l'évolution temporelle de l'amplitude de la fréquence considérée.

Nous avons choisi la représentation la plus classique, c'est-à-dire une transformée de Fourier discrète à fenêtre glissante, avec une fenêtre plus petite que celle utilisée plus haut (de l'ordre de 1024 points). En fait, on utilise des fenêtres de taille variable, pour que cette taille soit un nombre entier de fois la période du son étudié quand cela est possible :

$$largeur_{fenetre} = \min(1024, kP), \text{ avec } P \text{ periode et } k \in \mathbf{N}$$

De cette façon, l'amplitude de Fourier ne dépend pas de la phase de l'oscillation. Pour une fréquence de résonance ω_i , le signal est de la forme :

$$h[n] = A'_i \rho_i^n \sin(\omega_i(n+1))$$

3 Détermination du facteur ρ_i

Il a déjà fallu déterminer un algorithme de régression exponentielle. Voilà le

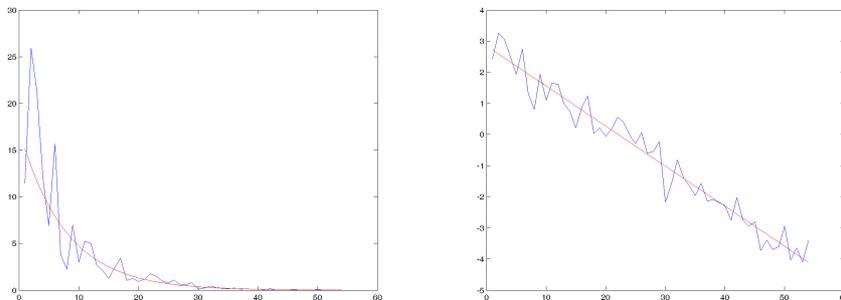


FIG. 5 – Résultat de l'algorithme "naïf" de régression exponentielle (à gauche) et logarithme de ces courbes (à droite)

résultat que l'on obtient avec un algorithme très simple de régression exponentielle, qui consiste à faire une régression linéaire sur le logarithme du signal. On remarque que cette régression ne donne pas de résultats suffisants pour notre étude. En effet, le `log` atténue les grandes différences, et donc a fait disparaître le pic initial.

Nous avons alors créé un algorithme qui minimise la norme $\sum_x f(x)^2 w(x)^2$ où f est le signal et w est une fonction poids. En prenant une fonction de poids de la forme

$$w(x) = \frac{1}{x^3} \text{ ou } e^{-x}$$

on trouve une régression qui "colle" beaucoup plus aux premiers points. Cette régression donne des résultats plus intéressants que l'algorithme naïf.

Mais on trouve encore un pic de départ est encore mal représenté. En effet, la phase du d'attaque (d'amplitude plus faible que le pic) nuit à la représentation de la hauteur de ce pic dans la régression. Nous avons donc coupé le signal de $t = 0$ à $t = t_0$ tel que $f(t_0) = \max(f)$. Ce procédé enlève la phase d'attaque et après régression, le pic est bien représenté (bonne amplitude et bonne décroissance). Cet algorithme n'est pas bon dans les basses amplitudes, mais cela n'a pas

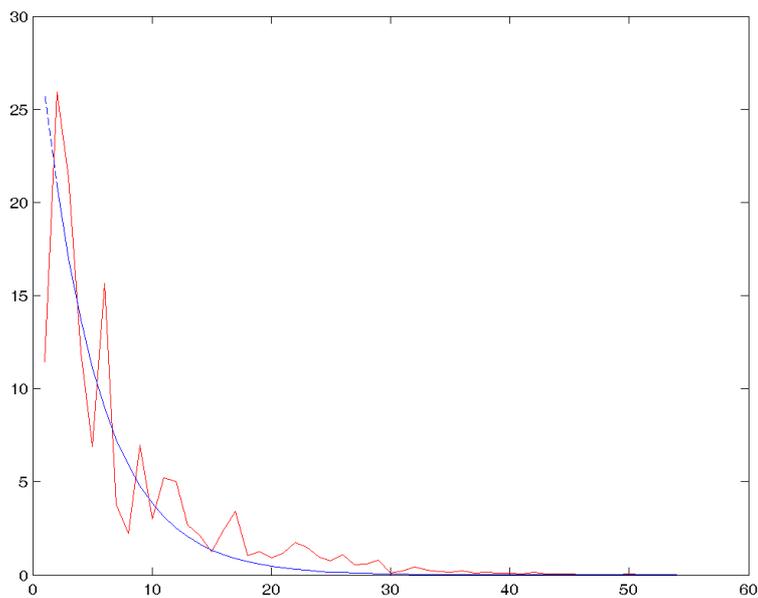


FIG. 6 – Le résultat de notre algorithme

grande importance : l'oreille distingue beaucoup moins bien les nuances à ces amplitudes. Cependant, on peut déjà s'attendre à ce que la décroissance du son va être amplifiée : le son synthétisé va donc paraître "étouffé".

Quatrième partie

Les résultats obtenus

Il est bien sûr difficile de retranscrire des résultats sonores sur un rapport. Cependant, certaines courbes peuvent permettre de se faire une idée des résultats que nous avons obtenus.

1 Synthèse sur différents instruments

1.1 Triangle

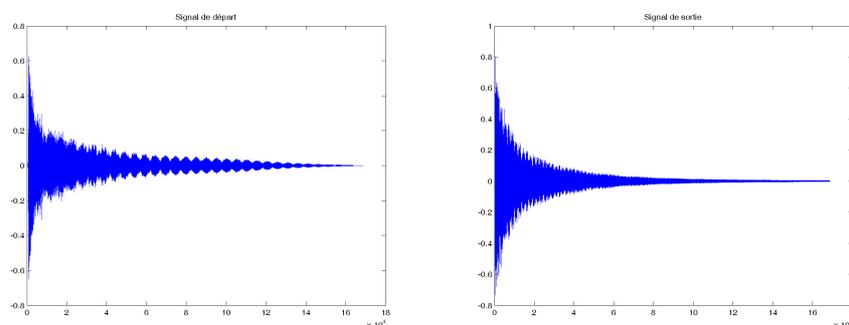


FIG. 7 – Le son réel d’un triangle et le son synthétisé

Notre algorithme donne de très bons résultats sur cet instrument. Le seul défaut est le rendu des battements un peu plus faible que dans la réalité : on le voit très bien sur la figure 7.

1.2 Cloche

Ce son a lui l’effet inverse : l’effet de battement est amplifié par rapport au son de départ (figure 8). Le son est également un peu vite ”étouffé”.

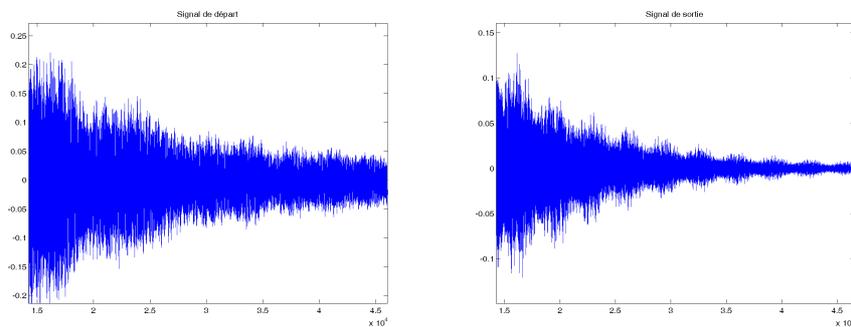


FIG. 8 – Le son réel d’une cloche et le son synthétisé

1.3 Piano

Cet instrument a été étonnamment bien synthétisé, d'autant plus qu'il s'agissait d'un accord (Figure 9). Le son en revanche est rapidement étouffé. Plus

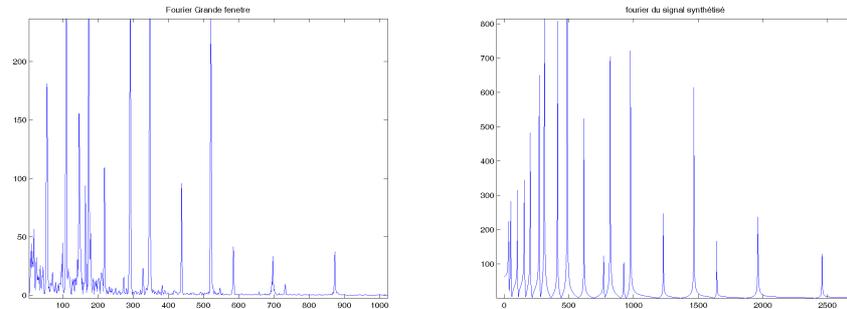


FIG. 9 – La représentation de Fourier du son réel d'un piano et du son synthétisé généralement, notre algorithme marche d'autant mieux que les fréquences sont isolées, comme c'est le cas pour le piano et le triangle.

1.4 Cymbale

La synthèse de cet instrument donne un son métallique, et la phase d'attaque est absente (Figure 10).

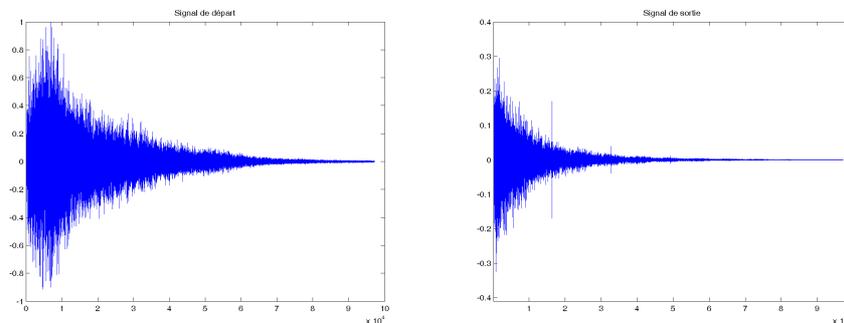


FIG. 10 – La représentation amplitude-temps du son réel d'une cymbale et du son synthétisé

1.5 Gong

La synthèse de ce son est particulièrement médiocre (son métallique, présence de 'clics', résonance très courte). Cela s'explique par le nombre important de fréquences mises en jeu (Figure 11).

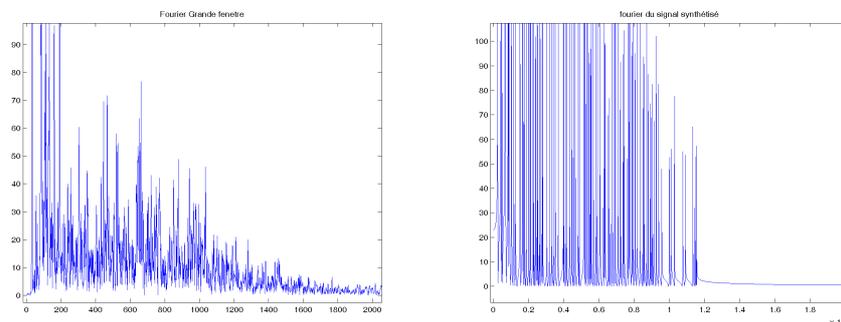


FIG. 11 – La représentation de Fourier du son réel d’un gong et du son synthétisé. Le son du gong a du mal à être synthétisé. (Les deux graphiques sont à la même échelle)

2 Des idées d’amélioration

2.1 Les fréquences aigües

On voit clairement sur la figure 11 que les fréquences ne se retrouvent pas dans la synthèse. Cela provient de l’algorithme de détection de pics qui ne donne pas de très bons résultats dans les hautes fréquences. En effet, certains sons percussifs contiennent des fréquences de résonance dans l’aigu avec des amplitudes faibles et leur détection est donc délicate. Certains sons semblent aussi contenir des ”bruits colorés” dans les fréquences aigües. Cela se traduit par de nombreux pics et creux d’amplitudes faibles dans la FFT, qui ne correspondent pas à des fréquences de résonance, mais à la réalisation d’un processus aléatoire.

Ces ”bruits” sont souvent importants pour la reconnaissance du son à l’oreille car leur énergie est assez importante, malgré les valeurs faibles de leur transformée de Fourier. Mais la faiblesse de ces valeurs ainsi que la structure de ”bruit” rend difficile la séparation des pics dûs au son étudié et de ceux dûs au bruit de fond.

Nous avons tenté de considérer tous les maxima locaux de la FFT comme des fréquences de résonance. Le son obtenu est effectivement plus réaliste, mais il ne se confond toujours pas avec le son de départ. De plus le bruit de fond présent dans l’enregistrement est reproduit dans la synthèse et les temps de calculs sont excessivement longs, ce qui rend cette méthode inexploitable. Et du point de vue de la compression, elle est très mauvaise puisque le nombre de filtres modélisés est grand (il peut varier jusqu’à la moitié du nombre de points de la FFT). La taille des données utilisées pour synthétiser le son est alors de l’ordre de la taille de l’enregistrement initial, ce qui est très grand.

2.2 Amplitude de la fréquence

On rappelle la formule établie en 2

$$h[n] = A_i \rho_i^n \sin(\omega_i(n+1))$$

Malheureusement, l’évolution temporelle réelle ne correspond pas exactement à ce résultat théorique. Cela est dû à plusieurs facteurs.

Tout d'abord, l'hypothèse sur "l'éloignement" des fréquences de résonance peut ne pas être satisfaite. On a alors des influences des fréquences voisines, comme par exemple des effets de battement. Une autre explication concerne le bruit de fond : pour certaines fréquences d'amplitudes faibles, la part due au bruit de fond peut ne pas être négligeable devant celle due au son analysé.

Enfin, certains sons ou composantes de sons ne sont tout simplement pas bien approximés par le modèle que nous utilisons, et l'évolution temporelle de certaines fréquences n'est alors pas celle prévue. Le son produit par une cymbale, par exemple, contient une phase d'augmentation d'intensité après le choc (figure 10), ce dont le modèle de filtres AR(2) ne rend pas compte. Le gong a lui une phase très rapide de descente, puis une décroissance très lente (figure 12), ce qui ne peut être représenté par notre modèle exponentiel.

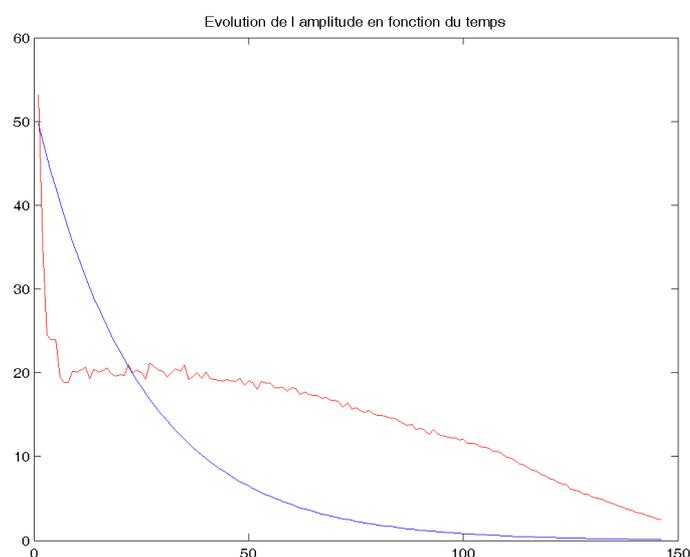


FIG. 12 – L'évolution de l'amplitude d'une fréquence d'un gong et sa régression

Nous avons donc tenté de modéliser cette phase d'attaque, dans le domaine fréquentiel, par une fonction affine, puis d'utiliser le modèle exponentiel dans la phase de descente. Mais, si non seulement un filtre représenté par une fonction affine n'est pas un filtre AR(2), on a également le problème de recollement entre les 2 phases. Cette solution s'éloignant trop du sujet initial et se trouvant être trop complexe, a été abandonnée.

Cinquième partie

Conclusion

Ce projet avait pour but de synthétiser un signal percussif, à l'aide d'un modèle simple. Si sur certains instruments (triangle, piano, cloche, ...) ce modèle exponentiel marche très bien, pour d'autres instruments, ce modèle se révèle être insuffisant. D'autres modèles plus avancés sont donc nécessaires pour permettre la synthèse de tels sons.

En supposant une telle synthèse réalisée, on aurait alors la possibilité de mélanger 2 percussions *a priori* totalement différentes, de créer une musique à l'aide d'une simple partition de percussions, de modifier la sonorité de percussions déjà existantes, voire peut-être de compresser certaines parties de fichiers musicaux par un algorithme de ce genre (plus efficace qu'un algorithme classique), et bien plus encore.

