



28 mai 2025

Groupe B2

Auteurs : Martin Leducq - Oren Torjman

Situation d'apprentissage riche

Traitement audio numérique



IMT Atlantique

Bretagne-Pays de la Loire
École Mines-Télécom

Table des matières

1 Synthèse additive	3
1.1 Analyse d'un son harmonique	3
1.2 Synthèse	6
2 Synthèse soustractive	7
3 Effets audio numériques	9
3.1 Effet de réverbération	9
3.2 Effet de retard	13
3.2.1 Analyse temporelle	13
3.2.2 Analyse fréquentielle	15
3.2.3 Amélioration : le filtre moyennneur	16
4 Conclusion	18

Avant propos

Cette situation d'apprentissage vise à mettre en pratique les notions théoriques étudiées au sein de UE Electrical Engineering.

Nous allons donc nous demander de quelle manière des l'électronique peut contribuer à la création est au traitement de signaux audio.

Nous verrons dans un premier temps comment un outil comme `Matlab` permet de réaliser la synthèse additive et soustractive de son échantillonné. Puis nous étudierons la manière dont l'agencement d'un circuit mélangeant des modules séquentiels et combinatoires peut créer un synthétiseur digital à l'aide du logiciel `vivendo` et d'une carte `Xilinx Artix 7` FPGA de la marque AMD. Finalement nous verrons que ce même logiciel et cette même carte peuvent créer un filtre numérique audio à l'aide de registres.

Remarque

Ce document ne traitera que le partie de synthèse réaliser sur `Matlab`. Le reste du travail est présent sur dans les fichiers `tp-filtre-etudiant-o24torjm` et `tp-synthe-etudiant-o24torjm`

Synthèse numérique audio

1 Synthèse additive

1.1 Analyse d'un son harmonique

Pour la plupart des instruments à cordes, le contenu fréquentiel d'une note est (quasi) harmonique, c'est-à-dire que les composantes fréquentielles sont placées sur des multiples entiers de la fréquence fondamentale, et données par

$$f_n = n f_1 \quad (1)$$

où correspond f_1 à la fréquence fondamentale et $n \in \mathbb{N}$. Il est remarquable de constater que les signaux subjectivement (ou culturellement) agréables à l'oreille comportent une structure mathématique aussi singulière.

Nous le vérifions donc grâce à un code rédigé sur MATHLAB (cf. Q11.m) :

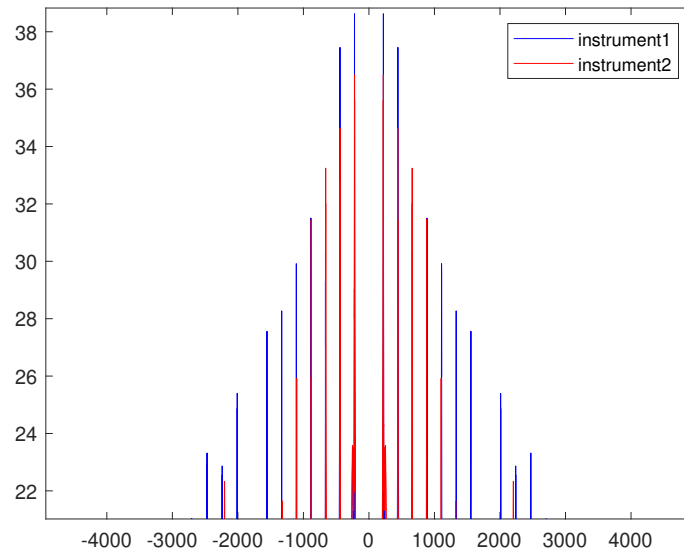


FIGURE 1 – Amplitude de la transformée de Fourier du son de deux instruments à cordes

Ici, les deux instruments écoutés sont `piano1` et `guitar_nylon_string_a3.wav`. On voit bien que les pics d'amplitude correspondant aux fréquences du signal sonore sont présent aux même fréquences étant des multiples de f_1 ici égal à 220 Hz.

En réalité, les instruments à cordes présentent rarement une structure harmonique aussi parfaite. En effet, l'équation qui régit le mouvement d'une corde réelle est en fait une combinaison de l'équation d'onde de la corde idéale et de celle d'une tige fixée aux deux extrémités pour laquelle les fréquences propres suivent une progression en n^2 . Le degré d'inharmonicité ξ_n de la n -ième harmonique se mesure en centième de demi-ton ou *cent* (i.e., rapport de fréquence de $2^{1/1200}$). Ce degré d'inharmonicité s'obtient pour la n -ième harmonique comme suit :

$$\xi_n = 1200 \times (\log_2(\hat{f}_n) - \log_2(nf_1)) \quad (2)$$

où \hat{f}_n est la n -ième fréquence propre mesurée sur le spectre.

Nous proposons ici de constater la présence d'inharmonicité sur les 2 sons de piano (`single_tone_piano1.wav` et `single_tone_piano2.wav`). Dans notre cas, $f_1 = 220 \text{ Hz}$.

1. Synthèse additive

Voici le graphe du quel nous avons extrait les données reportées dans les tableaux suivant (cf Q12.m).

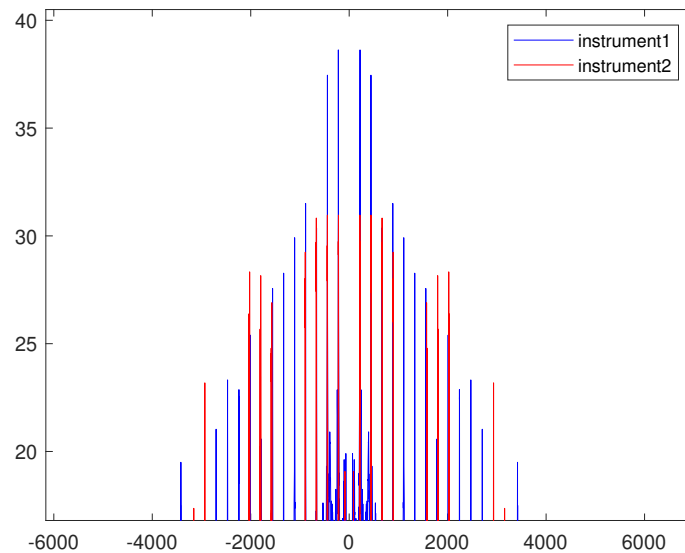


FIGURE 2 – Amplitude de la transformée de Fourier du son des deux pianos

Fréquence (Hz)	f_2	f_3	f_4	f_5	f_6	f_7
Théorique (Hz)	440	660	880	1100	1320	1540
Mesurée (Hz)	443	668	885	1108	1332	1556
Inharmonicité	11,8	20,9	9,8	12,5	15,7	17,9

TABLE 1 – Inharmonicité du single_tone_piano1.wav

Fréquence (Hz)	f_2	f_3	f_4	f_5	f_6	f_7
Théorique (Hz)	440	660	880	1100	1320	1540
Mesurée (Hz)	443	668	885	1587	1797	2022
Inharmonicité	11,7	20,9	9,8	635	534	471

TABLE 2 – Inharmonicité du single_tone_piano2.wav

On remarque ainsi que le piano1 est le plus harmonieux car la valeur totale de l'inharmonicité est la plus faible et le spectre des valeurs de fréquence associé à une amplitude significative ($A \geq 20dB$) est plus fourni.

1.2 Synthèse

Pour le son le plus harmonique (i.e. `single_tone_piano1.wav`), Nous avons récupéré les amplitudes de chacune des harmoniques (fréquence fondamentale et les 7 premières harmoniques). Dont voici les valeurs :

$i^{\text{ème}}$ Harmonique	1	2	3	4	5	6	7	8
f_i Fréquence (Hz)	220	443	668	885	1108	1332	1587	1784
A_i Amplitude (dB)	39	37	30	31,5	30	28	27,5	21

TABLE 3 – Valeurs relevées sur le gain de la transformée de Fourier de `single_tone_piano1.wav`

Nous avons ensuite généré un son en superposant les ondes sinusoidales aux fréquences et amplitudes correspondantes sous la forme :

$$s(t) = \sum_{i=1}^8 A_i \sin(2\pi f_i t) \quad (3)$$

Par la suite nous appliquons un filtre ADSR (Attack Decay Sustain Release) ce qui nous permet de rendre le son moins "numérique".

Nous avons utilisé également un autre méthode consistant à définir un transformée de Fourier par superposition de Dirac constituant chacun des sinus tel que : (en notant j la variable complexe)

$$\hat{s}(\nu) = \sum_{i=1}^8 \frac{A_i}{2j} (\delta_{f_i} - \delta_{-f_i}) \quad (4)$$

Finalement, les deux méthodes ne donne pas le même résultat. En effet, deux notes différentes semble être jouées. Le son généré par la somme de sinus est plus proche du son joué par la note de piano. La méthode de transformée de Fourier inverse semble être obsolète ou du moins de la façon dont nous l'avons implémentée

*
* * *

2 Synthèse soustractive

Dans la section précédente, la génération d'une note a été faite en partant de la fréquence fondamentale enrichie par des harmoniques. Nous allons maintenant aborder une autre technique partant d'un signal possédant naturellement des harmoniques (signal périodique) que nous allons ensuite filtrer. C'est le principe utilisé dans le célèbre synthétiseur TB-303 qui joua un rôle important dans le développement de la musique électronique dans les années 1980. Ce principe est illustré sur la figure suivante :

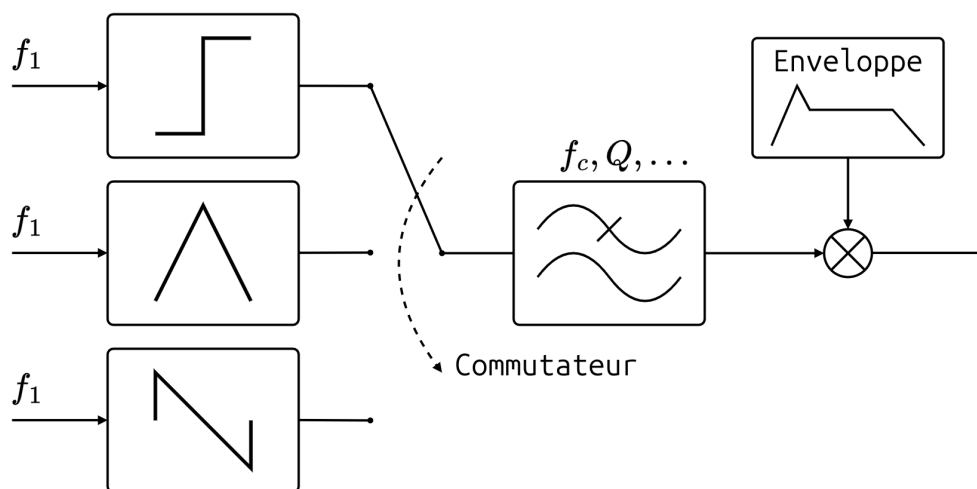


FIGURE 3 – Schéma représentant un synthèse soustractive de 3 type de signaux (carré, dent de scie, triangulaire)

Calculons le spectre d'un signal carré (centré, d'amplitude ± 1 , période T) et d'un signal dent de scie (centré, d'amplitude ± 1 , période T)

Pour le signal carré, nous allons prendre :

$$x(t) \rightarrow \begin{cases} 1 & \text{si } t \in [0; \frac{T}{2}] \\ -1 & \text{si } t \in [\frac{T}{2}; T] \end{cases}$$

Ainsi utilisons l'expression générale des coefficients de Fourier afin de trouver le spectre du signal, on a donc :

$$\forall n \in \mathbb{N}, a_n = \frac{2}{T} \int_0^T x(t) \cos\left(\frac{2\pi n t}{T}\right) dt, \quad b_n = \frac{2}{T} \int_0^T x(t) \sin\left(\frac{2\pi n t}{T}\right) dt \quad (5)$$

En remplaçant l'expression de $x(t)$ pour a_n et b_n on a finalement :

$$\boxed{\forall n \in \mathbb{N}^*, a_n = 0, \quad b_n = \frac{4}{n\pi} (1 - \cos(n\pi))}$$

Ainsi $S_{p_{carr}} = \{(2k+1)f_0 \mid k \in \mathbb{N}^*\}$

Pour le signal en dent de scie, nous appliquons la même démarche et nous obtenons :

$$\boxed{\forall n \in \mathbb{N}^*, b_n \neq 0}$$

Ainsi $S_{p_{saw}} = \{kf_0 \mid k \in \mathbb{N}^*\}$

Numériquement, on pourrait définir une fonction carrée ou en dent de scie sur un support fini (première limite) et calculer la `fft` et la `fftshift` ce qui nous donnera la transformée de Fourier du signal pour laquelle nous pouvons tracer son amplitude et regarder la position des pics selon la fréquence pour retrouver le spectre qui ne sera pas complet de part la limitation du tracé du spectre (deuxième limite). Nous serons de plus dépendant de la fréquence d'échantillonnage pour la précision de notre valeur (troisième limite).

Pour la synthèse soustractive, on s'intéresse au filtre passe bas d'ordre 1 suivant :

$$y(k) = \frac{1}{2} (x(k) + x(k-1))$$

Ce filtre est un filtre FIR non récursif d'ordre 1. Sa réponse fréquentielle est :

$$H(e^{j\omega}) = \frac{1}{2}(1 + e^{-j\omega}) = \cos\left(\frac{\omega}{2}\right) e^{-j\omega/2}.$$

Son module est :

$$|H(e^{j\omega})| = \left| \cos\left(\frac{\omega}{2}\right) \right|.$$

Le filtre conserve donc les basses fréquences et réduit progressivement l'amplitude des hautes fréquences.

Le spectre de sortie est donc le spectre d'entrée multiplié par le gain du filtre.

Nous avons vérifié expérimentalement l'effet de ce filtre par écoute du son filtré et il s'agit bien d'un filtre passe bas car les sons caractéristiques de hautes fréquences (kicks par exemple) sont filtrés.

Après écoute des sons produits par synthèse soustractive et par synthèse additive, on remarque que pour ceux de la synthèse additive, l'enveloppe a un effet instantané (on additionne cette enveloppe avec le signal) et le signal arrive donc progressivement (en fonction de l'enveloppe). A contrario pour la synthèse soustractive, on entend l'impact de l'enveloppe uniquement après son établissement

(on soustrait l'enveloppe au signal) ce qui fait qu'on entend d'abord le signal normalement puis il est atténué par l'enveloppe.

3 Effets audio numériques

3.1 Effet de réverbération

Une fois ces synthèses implémentées, nous allons nous intéresser à l'implémentation de systèmes linéaires invariants en temps en prenant comme cadre d'application les effets audio-numériques. Ces effets modifient un signal audio numérisé souvent dans un but artistique mais aussi parfois pour en corriger ses défauts.

La réverbération permet d'ajouter de l'espace au son en simulant sa propagation dans un environnement le plus souvent fermé (salle, amphithéâtre, arène, église, etc). Physiquement, la réverbération est un mécanisme complexe faisant intervenir des phénomènes de réflexion, réfractions, etc. Ces phénomènes sont traités par une discipline nommée l'"acoustique des salles". Lorsque l'environnement n'évolue pas en fonction du temps, la réverbération peut être modélisée par un système linéaire invariant en temps. La réverbération est alors complètement décrite par sa réponse impulsionnelle.

L'approche par convolution s'implémente facilement et produit des résultats très convaincants. Elle est composée des étapes suivantes :

- Mesure de la réponse impulsionnelle $h(k)$ (l'empreinte acoustique) d'un lieu.
- Convolution d'un son $x(k)$ par la réponse impulsionnelle.

La mesure de réponse impulsionnelle d'une salle est très utile pour contrôler l'acoustique de celle-ci ou pour recréer cette acoustique virtuellement en appliquant un effet de réverbération à un signal d'intérêt. Cette mesure n'est pas aussi aisée qu'on pourrait le penser car générer une impulsion acoustique est assez complexe et pose quelques problèmes en terme de répétabilité de la mesure et de saturation des éléments de la chaîne de captation (microphones ou haut-parleurs, convertisseur analogique-numérique, etc). Parmi les différentes alternatives de mesure possibles, nous allons nous intéresser à la méthode dite pseudo-impulsionnelle qui repose sur la génération de séquences ayant une fonction d'autocorrélation proche de l'impulsion unité.

Soit $x(k), k \in \mathbb{Z}$, le signal d'excitation et $y(k)$ le signal capté par un microphone positionné dans une pièce. On note $h(k)$ la réponse impulsionnelle de la pièce (incluant également la réponse de la

chaîne de génération-captation). On note $R_{yx}(u)$ la fonction d'intercorrélation entre y et x et $R_{xx}(u)$ la fonction d'autocorrélation de x .

On a alors :

$$R_{yx}(u) = \sum_n y(n+u) x(n)$$

d'où :

$$R_{yx}(u) = \sum_n \left(\sum_k h(k) x(n+u-k) \right) x(n)$$

Donc en intervertissant les sommes :

$$R_{yx}(u) = \sum_k h(k) \sum_n x(n+u-k) x(n)$$

En prenant $m = n + u - k$, on identifie la fonction d'autocorrélation R_{xx} :

$$\sum_n x(n+u-k) x(n) = R_{xx}(u-k)$$

D'où :

$$R_{yx}(u) = \sum_k h(k) R_{xx}(u-k)$$

Et donc :

$$R_{yx}(u) = h(u) * R_{xx}(u)$$

Si on suppose alors que $R_{xx}(u) \approx d(u)$, on a $R_{yx}(u) \approx h(u)$.

La réponse impulsionnelle est donc dans ce cas approximativement égale à l'intercorrélation entre x et y . Une méthode de mesure de la réponse impulsionnelle est donc de mesurer cette intercorrélation.

Pour ce faire on cherche à déterminer parmi les signaux dont on dispose, lequel est le plus approprié à la mesure de h .

Pour cela on peut visualiser et analyser l'autocorrélation de ces signaux.

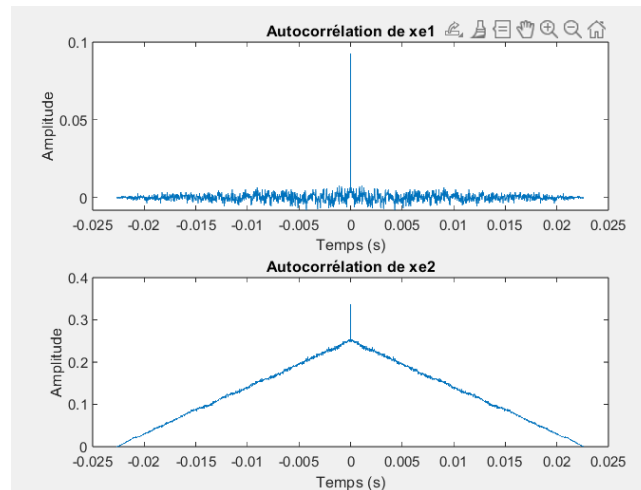


FIGURE 4 – Autocorrélation des signaux xe1 et xe2

L'autocorrélation du signal x_{e1} est un pic centré en 0 c'est à dire $R_{xx}(u) \approx d(u)$ comme les conditions que l'on souhaite.

En revanche l'autocorrélation du signal x_{e2} présente des variations peut importe la fréquence ce qui rend compliqué la mesure de l'intercorrélation.

On choisit donc de garder le signal x_{e1} pour mesurer la réponse impulsionnelle.

On applique alors l'effet de la pièce dont on veut récupérer la réponse impulsionnelle sur le signal d'excitation de la pièce grâce à la fonction `simule_piece`. On calcule ensuite la fonction d'intercorrélation (qui correspond à h) du signal obtenu avec le signal d'excitation x_{e1} .

On obtient donc la réponse impulsionnelle suivante :

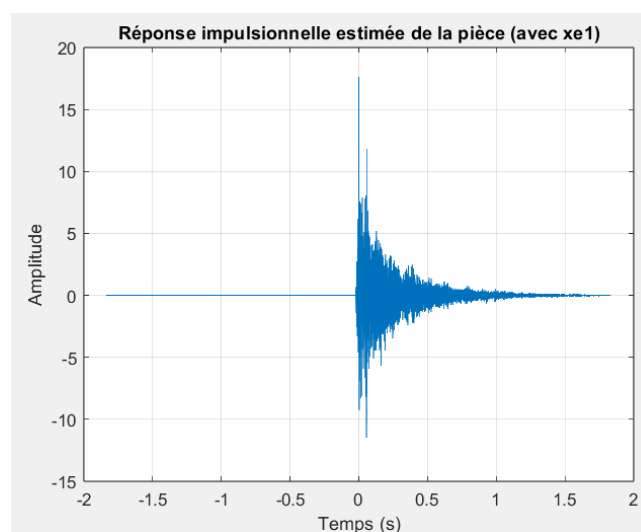


FIGURE 5 – Réponse impulsionnelle de la pièce en fonction du temps

Maintenant que nous avons la réponse impulsionnelle du filtre étudié, nous pouvons mettre en forme l'effet de réverbération qui nous intéressait de prime abord. Cet effet s'obtient par convolution de la réponse impulsionnelle avec le signal source, c'est à dire $y(k) = h(k) * x(k)$.

Nous avons donc d'abord implémenter la fonction `effet_reverb` décrite sur la figure 6.

```
function y = effet_reverb(x,h)
% S'assurer que les signaux sont des vecteurs colonnes
x = x(:);
h = h(:);

% Application de la réverbération via la convolution
y = conv(x,h);
end
```

FIGURE 6 – Fonction `effet_reverb` en Matlab

On teste ensuite la fonction sur un son de guitare avec le code suivant :

3. Effets audio numériques

```
1 % Normalisation du signal d'excitation
2 xe = xel / max(abs(xel));
3
4 % Simulation du signal de sortie dans la pièce
5 ys = simule_piece(xe,fe); % ys = signal mesuré à la sortie de la pièce
6
7 % Estimation de la réponse impulsionnelle via corrélation croisée
8 [h_est, lags] = xcorr(ys, xe);
9
10 % Centrage de la réponse impulsionnelle estimée
11 [~, idx_max] = max(abs(h_est)); % Trouve l'indice du maximum (origine temporelle)
12 lags = lags - lags(idx_max); % Décalage pour que t = 0 corresponde au pic
13
14 % Axe temporel en secondes
15 t = lags / fe;
16
17 [x, Fe] = audioread('single_tone_guitar_nylon_string_a2.wav');
18
19 % Application de la réverbération
20 tic;
21 y_reverb = effet_reverb(xel, h_est);
22 temps_calcul = toc;
23 fprintf('Temps de calcul : %.4f secondes\n', temps_calcul);
24 y_reverb = y_reverb / max(abs(y_reverb));
25 audiowrite('Q35.wav', y_reverb, Fe);
```

FIGURE 7 – Script test_effet_reverb permettant de tester l'effet de réverbération

On obtient un temps de calcul nécessaire à l'exécution de la fonction effet_reverb de 0.0111 seconde.

Pour diminuer le temps de calcul, on envisage d'implémenter l'effet de réverbération dans le domaine fréquentiel en utilisant la FFT. La sortie du filtre s'écrit alors :

$$y_2(k) = IFFT(FFT(h(k)).FFT(x(k)))$$

On modifie alors la fonction effet_reverb en effet_reverb_FFT comme sur la figure 8.

```
1 function [y, temps] = effet_reverb_FFT(x, h)
2 % Entrées :
3 % x : signal d'entrée
4 % h : réponse impulsionnelle (réverbération)
5
6 % Étape 1 : Déterminer la taille pour le calcul FFT (longueur max)
7 NFFT = max([length(x) length(h)]);
8
9 % Étape 2 : Appliquer la FFT sur les deux signaux (zéro-padding)
10 tic;
11 X = fft(x, NFFT);
12 H = fft(h, NFFT);
13
14 % Étape 3 : Multiplication dans le domaine fréquentiel
15 Y = X .* H;
16
17 % Étape 4 : Retour dans le domaine temporel
18 y = real(ifft(Y));
19
20 % Étape 5 : Enregistrer le temps de calcul
21 temps = toc;
22 end
```

FIGURE 8 – Fonction effet_reverb_FFT

Après test de la fonction effet_reverb_FFT, on obtient un temps de calcul nécessaire à l'exécution de fonction de 0.0964 seconde. On obtient un temps de calcul similaire en ordre de grandeur, néanmoins celui ci dépend de la taille du signal d'entrée, il faudrait tester le système avec un signal plus long.

La méthode utilisant la FFT est donc équivalente à la convolution classique, si elle est bien implémentée (zero-padding, taille adéquate, pas de repliement de spectre). On obtient à l'oreille deux sons presque identiques bien que cette deuxième méthode perde un peu en précision par rapport à la première.

3.2 Effet de retard

Les effets de retard (delay) sont très utilisés en musique. Ces effets permettent d'ajouter de l'espace en simulant les réflexions des ondes acoustiques dans un espace clos. Dans cette pré-détermination, vous allez analyser la réponse impulsionnelle et la réponse fréquentielle de l'effet de delay. Ici, le symbole $z^{-\tau}$ désigne un retard de τ échantillons.

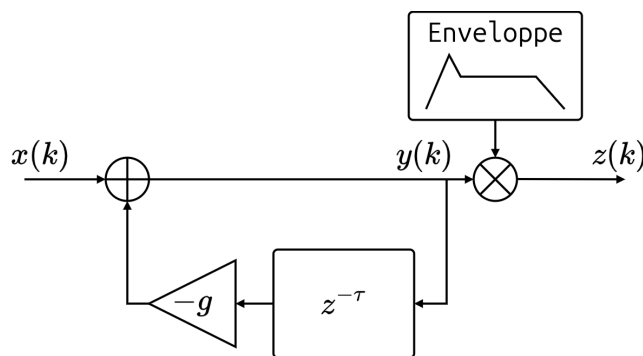


FIGURE 9 – Schéma numérique symbolisant le retard

Pour simuler un effet de retard, nous allons considérer le système décrit par le schéma de la figure ci-dessus. Mathématiquement, cet effet peut être modélisé par la relation de récurrence suivante ($\tau \in \mathbb{N}$) :

$$y(k) = x(k) - gy(k - \tau) \quad (6)$$

Le système décrit par (Delay) désigne un filtre IIR puisque le système possède une partie récursive non nulle. La partie récursive permet de simuler les réflexions du son. Ainsi le signal de sortie est renvoyé avec τ échantillons de retard et un coefficient d'atténuation g . Dans la suite, le signal $x(k)$ est supposé échantillonné à $F_e = 44100$ Hz.

3.2.1 Analyse temporelle

Montrons ainsi par récurrence que, pour la relation (6) décrivant le retard, on a $h(k) = \begin{cases} (-g)^n & \text{si } k = n\tau, \\ 0 & \text{sinon.} \end{cases}$

Par définition $h(k)$ est la réponse du filtre pour $x(k) = \delta(k)$. Ainsi (6) devient :

$$h(k) = \delta(k) - gh(k - \tau)$$

Pour $n = 0$:

$$h(0) = 1$$

Ensuite, $\forall k \in]0; \tau[$, $h(k) = 0$. On obtient ceci car, $\forall k \neq 0$, $\delta(k) = 0$ (il en est de même pour $k \in]-\tau; 0[$

3. Effets audio numériques

Soit $n \in \mathbb{N}$ tel que $h(n\tau) = (-g)^n$ et $\forall k \in](n-1)\tau; n\tau[$, $h(k) = 0$ montrons ainsi que la proposition est vraie au rang $n+1$:

$$h((n+1)\tau) = \delta((n+1)\tau) - gh(n\tau) = 0 - g \times (-g)^n = (-g)^{n+1}$$

Et $\forall k \in]n\tau; (n+1)\tau[$, $h(k) = 0$. On obtient ceci car, $\forall k \in]n\tau; (n+1)\tau[$, $\delta(k) = 0$ et $h(k) = 0$

Finalement l'hypothèse est vraie au rang 0 et héréditaire pour $n \in \mathbb{N}$ alors la proposition est vraie sur \mathbb{N}

Pour que ce filtre soit stable il faut que la somme de l'amplitude de sa réponse impulsionnelle soit bornée i.e.

$$\sum_{k=0}^{\infty} |h(k)| < \infty \Leftrightarrow \sum_{n=0}^{\infty} |(-g)|^n < \infty$$

Cette condition revient à avoir $|g| < 1$

Pour pouvoir exploiter ce filtre sur nos machines nous utilisons la fonction `filter(a, b, x)` de Matlab avec $a = [1 \underbrace{0 \dots 0}_{\tau-1 \text{ fois}} g]$ et $b = [1]$

Ce qui nous donne la figure suivante pour la réponse impulsionnelle temporelle :

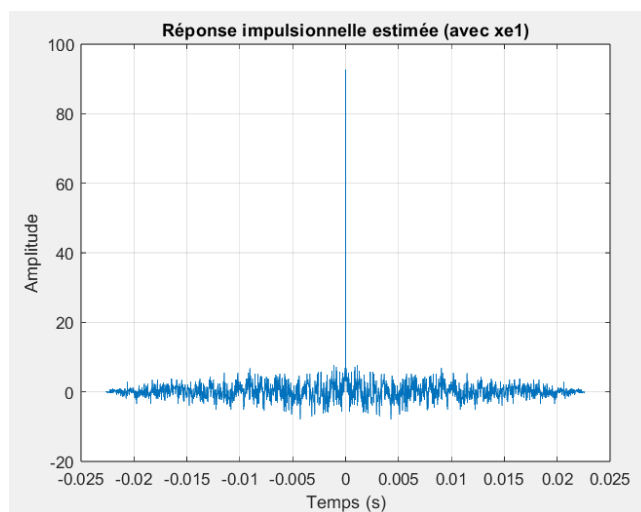


FIGURE 10 – Réponse impulsionnelle de l'effet de retard

On remarque bien ici les oscillations de l'amplitude entre des valeurs positives et négatives du filtre symbole du terme de $-g$ se trouvant dans l'équation. Cependant, nous ne devrions voir une figure tout à fait différente (définie seulement pour des temps positifs et étant nulle pour les points différents des $n\tau$). Nous avons pas réussi à en déterminer la cause.

3.2.2 Analyse fréquentielle

Déterminons maintenant la réponse fréquentielle du filtre retard. Pour ceci nous repartons de l'expression récursive du filtre :

$$y(k) = x(k) - g y(k - \tau)$$

En prenant la transformée de Fourier, on obtient la relation entre les spectres :

$$Y(e^{2i\pi\nu}) = X(e^{2i\pi\nu}) - g e^{-2i\pi\nu\tau} Y(e^{2i\pi\nu})$$

On regroupe les termes en Y :

$$Y(e^{2i\pi\nu}) (1 + g e^{-2i\pi\nu\tau}) = X(e^{2i\pi\nu})$$

Donc la fonction de transfert (réponse fréquentielle) est :

$$\hat{h}(\nu) = \frac{Y(e^{2i\pi\nu})}{X(e^{2i\pi\nu})} = \frac{1}{1 + g e^{-2i\pi\nu\tau}}$$

On calcule maintenant le module et la phase de la fonction de la transformée de la fonction de transfert du filtre :

Module :

$$|\hat{h}(\nu)| = \left| \frac{1}{1 + g e^{-2i\pi\nu\tau}} \right| = \frac{1}{|1 + g e^{-2i\pi\nu\tau}|}$$

On développe le module en utilisant la forme exponentielle complexe :

$$e^{-2i\pi\nu\tau} = \cos(2\pi\nu\tau) - i \sin(2\pi\nu\tau)$$

Donc :

$$|1 + g e^{-2i\pi\nu\tau}|^2 = (1 + g \cos(2\pi\nu\tau))^2 + (g \sin(2\pi\nu\tau))^2$$

$$= 1 + 2g \cos(2\pi\nu\tau) + g^2 (\cos^2(2\pi\nu\tau) + \sin^2(2\pi\nu\tau))$$

$$= 1 + 2g \cos(2\pi\nu\tau) + g^2$$

Donc le module est :

$$|\hat{h}(\nu)| = \frac{1}{\sqrt{1 + 2g \cos(2\pi\nu\tau) + g^2}}$$

Phase :

$$\arg(\hat{h}(\nu)) = -\arg(1 + g e^{-2i\pi\nu\tau})$$

Avec :

$$\arg(1 + g e^{-2i\pi\nu\tau}) = \arctan\left(\frac{g \sin(2\pi\nu\tau)}{1 + g \cos(2\pi\nu\tau)}\right)$$

Donc :

$$\arg(\hat{h}(\nu)) = -\arctan\left(\frac{g \sin(2\pi\nu\tau)}{1 + g \cos(2\pi\nu\tau)}\right)$$

Nous pouvons alors constater que le module du filtre est maximal pour les fréquences réduites $\nu_k = \frac{2k+1}{2\tau}$ ($k \in \mathbb{N}$) et que $|\hat{h}(\nu_k)| = \frac{1}{1-g}$. Notre effet est donc un filtre en peigne.

Voici une illustration graphique des résultats établis :

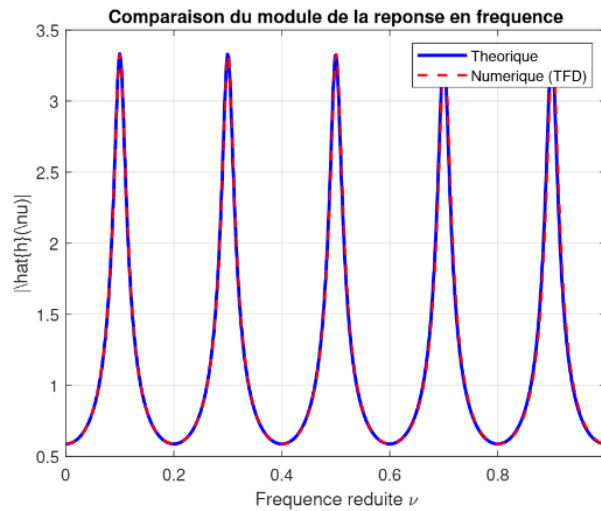


FIGURE 11 – Comparaison des modules des transformées de Fourier théorique et numérique du filtre

Nous retrouvons bien l'aspect "peigne" précédemment établi avec des fréquences d'apparitions régulièrement espacées ($\Delta\nu = 0.2$).

Nous avons implémenté par la suite une fonction `effet_delay` qui permet à paramètres d'entrée fixés (ν , τ , F_e , g) d'appliquer le filtre à n'importe quel signal d'entrée. Nous allons tester cette fonction sur un accord de piano.

3.2.3 Amélioration : le filtre moyennneur

A l'écoute, nous observons que cet effet produit des sonorités "synthétiques". Pour rendre l'effet plus naturel, nous allons modifier le son après réflexion en filtrant le signal dans la boucle de retour. Cette modification est illustrée par la figure suivante :

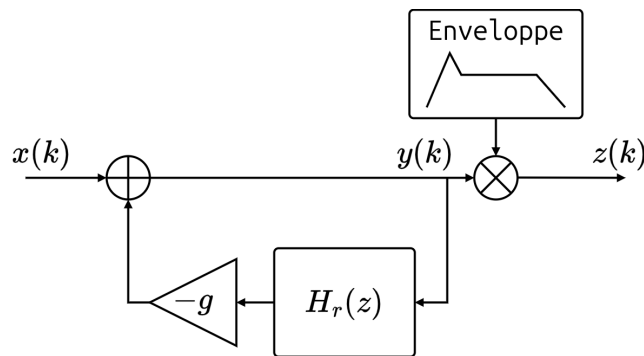


FIGURE 12 – Schéma numérique symbolisant le filtre appliqué à l'accord de piano

Dans la boucle de retour, nous choisissons un filtre dont la réponse impulsionnelle est donnée par :

$$h_r(k) = \begin{cases} \frac{1}{K} & \text{si } k = 0, 1, \dots, K-1 \\ 0 & \text{sinon} \end{cases}$$

Ce filtre réalise une moyenne glissante sur K échantillons. L'effet de delay s'exprime alors par la relation de récurrence suivante :

$$y(k) = x(k) - \frac{g}{K} \sum_{n=0}^{K-1} y(k - \tau - n)$$

Nous avons donc implémenté une fonction `effet_delay_filtre` qui permet à paramètres d'entrée fixés (ν , τ , F_e , g) d'appliquer le filtre moyennneur à n'importe quel signal d'entrée.

L'écoute du résultat montre que le filtre $h_r(k)$ semble "couper" progressivement les hautes-fréquences du signal. Vérifions cela graphiquement :

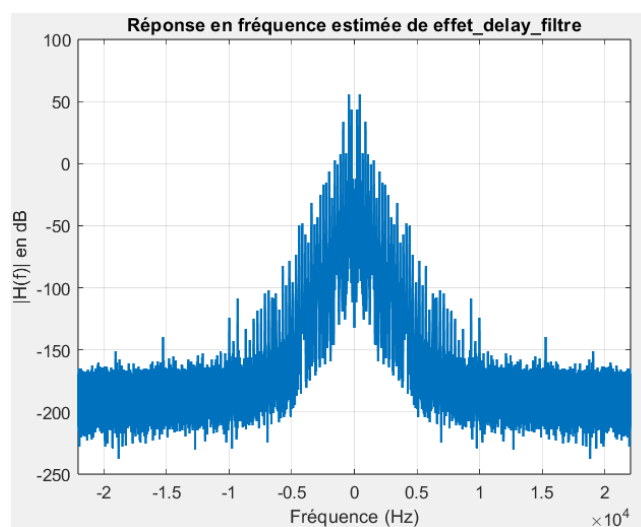


FIGURE 13 – Module de la réponse en fréquence du filtre au piano

On remarque ici que le module de la réponse du filtre va très vite s'effondrer symbole du fait que le filtre coupe les hautes fréquences.

4 Conclusion

Nous avons implémenter en Matlab différentes approches du traitement du signal audio, d'abord avec les synthèse additives et soustractives qui permettent l'application d'une enveloppe au signal acoustique traité, puis avec l'application de l'effet de réverbération à ce signal. Cet effet a été implémenté de différentes façons, que ce soit par convolution classique ou par convolution rapide de sorte à améliorer le temps de calcul. Nous avons également implémenté un effet retard simulant les réflexions des ondes acoustiques dans un espace clos.

Plus globalement l'étude des effets audio, notamment la réverbération et le delay, a permis d'appliquer des concepts de systèmes linéaires invariants en temps, d'explorer la convolution, l'analyse fréquentielle et de mettre en œuvre des filtres IIR et FIR. Nous avons aussi comparé différentes méthodes d'implémentation (domaine temporel vs fréquentiel), ce qui nous a donné une vision pratique des compromis entre précision et efficacité.