



~

Camille Lerays - Samuel Vaton

Compte rendu SAR

UE electrical engineering



IMT Atlantique

Bretagne-Pays de la Loire
École Mines-Télécom

Table des matières

Introduction	3
1 Synthèse additive	3
1.1 Visualisation du spectre de notes d'instruments	3
1.2 Comparaison spectre piano1.wav et piano2.wav	5
1.3 Création d'un son sinusoïdale	6
1.4 Enveloppe ADSR	6
1.5 Reconstruction du signal à partir de <code>ifft()</code>	8
2 Synthèse soustractive	8
2.1 Calcul du spectre d'un signal carré et d'un signal en dents de scie	8
2.1.1 Calcul du spectre	8
2.2 Filtre passe bas d'ordre 1	11
2.2.1 Simulation	11
2.3 Comparaison signaux periodiques	13
2.4 Amélioration du signal produit par synthèse soustractive	14
3 Effets de réverbération	14
3.1 Calcul de R_{yx}	14
3.2 Approximation et méthode de calcul de h	15
3.3 Choix d'un signal adapté pour mesurer réponse impulsionnelle de la pièce	15
3.3.1 Méthodologie du calcul de la réponse impulsionnelle	16
3.4 Comparaison convolution classique et rapide	17
3.5 Effet de retard	19
3.6 Stabilité du filtre Delay	19
3.7 Création du filtre sur Matlab	20
3.8 Visualisation du filtre	20
3.9 Réponse du filtre en fréquence	20
3.10 Filtrage delay	22
3.11 Filtrage delay avec moyenne sur K échantillons	22
3.12 Réponse en fréquence de l'effet de retard filtré	23
4 Conclusion	23

Introduction

1 Synthèse additive

1.1 Visualisation du spectre de notes d'instruments

Nous obtenons les spectres suivants en dB :

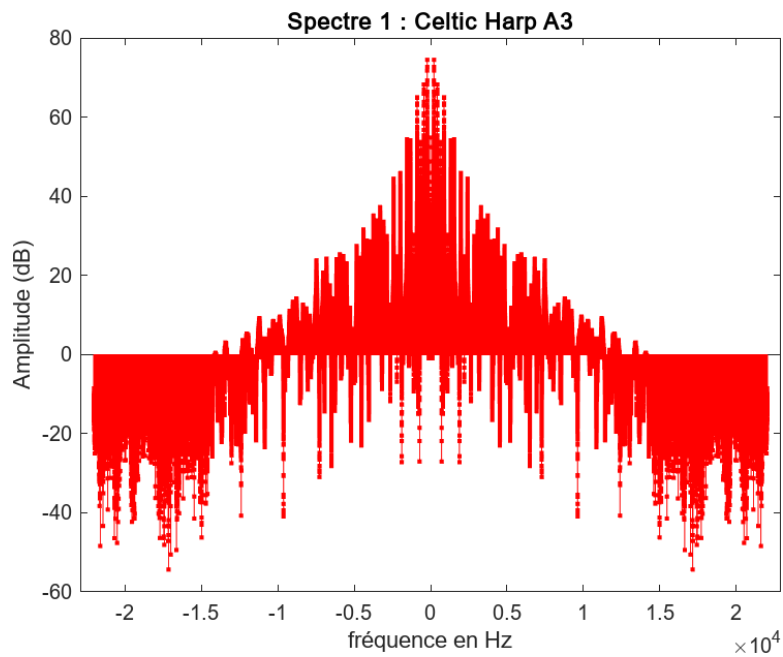


FIGURE 1 – Spectre en decibel d'une note de harpe

En faisant un zoom sur les figures avec matlab et en prenant la première valeur de fréquence la plus haute après 0 on obtient pour la fréquence fondamentale : $f_1 = 220.01 Hz$.

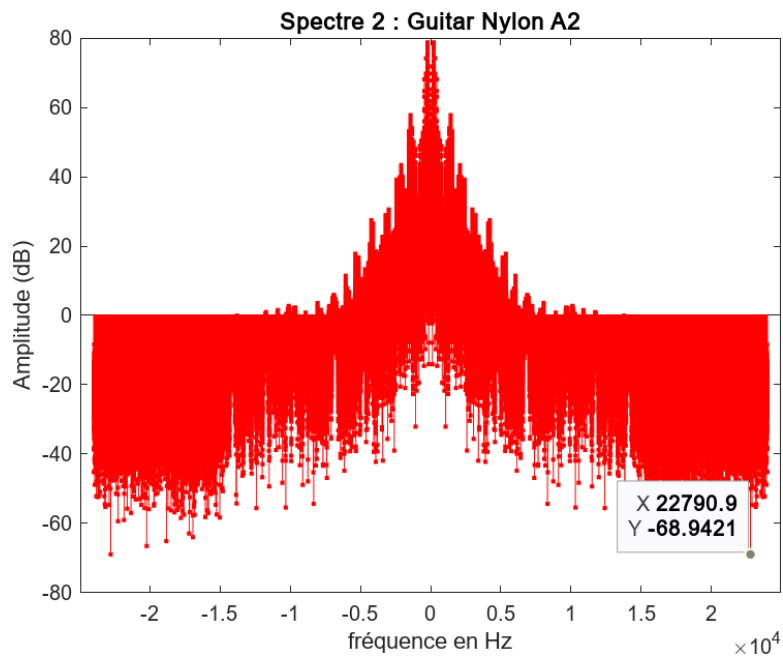


FIGURE 2 – Spectre en decibel d’une note de guitar en nylon

On obtient la fréquence fondamentale : $f_2 = 220.10Hz$.

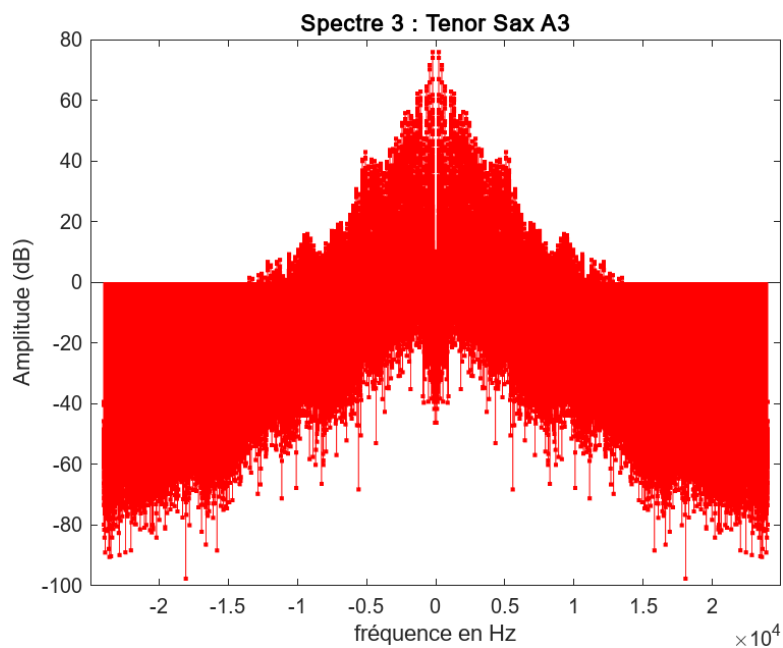


FIGURE 3 – Spectre en decibel d’une note de saxophone

On obtient la fréquence fondamentale : $f_3 = 220.96Hz$.

Nous avons déduit le spectre de la note de harpe du fichier : single_tone_celtic-harp-a3.wav, celui de la note de guitare du fichier : single_tone_guitar_nylon_string_a2.wav et celui de la note de saxophone du fichier : single_tone_sax-tenor-a3.wav

1.2 Comparaison spectre piano1.wav et piano2.wav

On affiche les spectres en décibels des fichiers piano1.wav en rouge et piano2.wav en bleu :

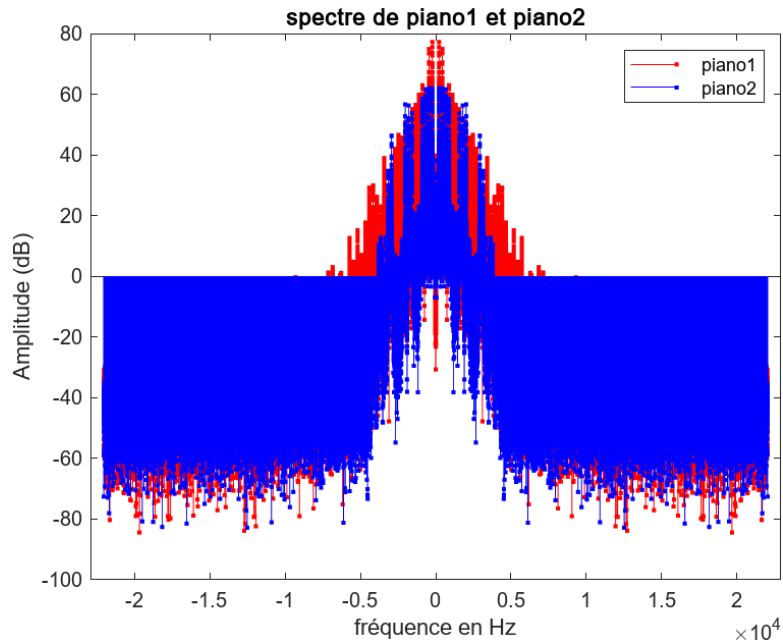


FIGURE 4 – Spectre en decibel d'une note de saxophone

Pour trouver les fréquences fondamentale, on effectue un zoom sur la figure ce qui nous donne :

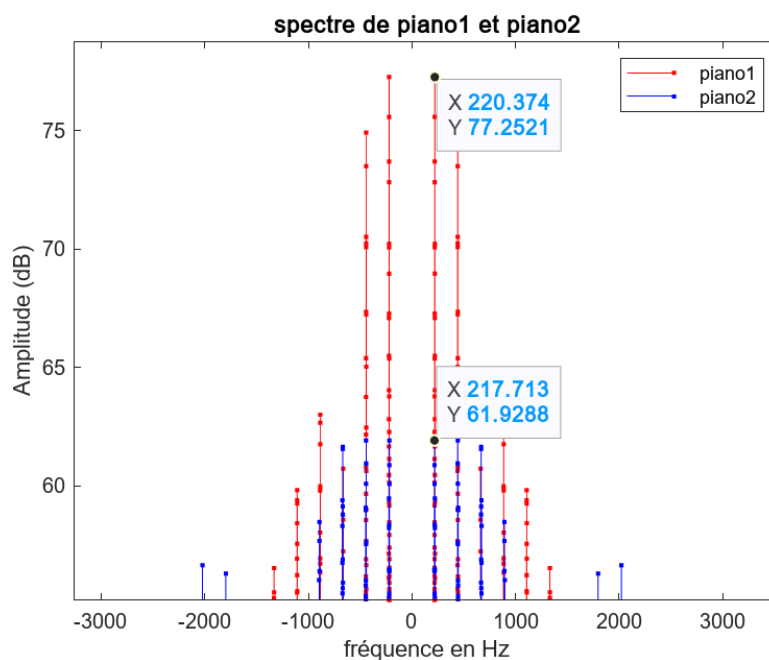


FIGURE 5 – fréquences fondamentales de piano1 et piano2

On a donc $f_{1,piano1} = 220.37Hz$ et $f_{1,piano2} = 217.71Hz$

On effectue une étude de l'inharmonicité jusqu'à la quatrième harmonique.

On trouve les harmoniques suivantes pour piano1 : 442.20, 663.22, 885.54 et 442.78, 667.42, 892.49 pour le piano2.

On utilise la formule $\xi_n = 1200 \times \log_2\left(\frac{f_n}{nf_1}\right)$ qu'on applique sur nos harmoniques et on obtient en moyenne : $\xi_{piano1} = 6.19$ cents et $\xi_{piano2} = 36.33$ cents.

Ainsi, on en déduit que le piano2 est moins harmonique que le piano1.

1.3 Création d'un son sinusoïdale

En utilisant la méthode précédente pour trouver les 8 premières harmoniques et les amplitudes associées dans piano1.wav, on reconstruit le signal suivant : $s(t) = \sum_{i=1}^8 A(f_i) \sin(2\pi f_i t)$

Ce qui donne le signal suivant dont l'amplitude est normalisée :

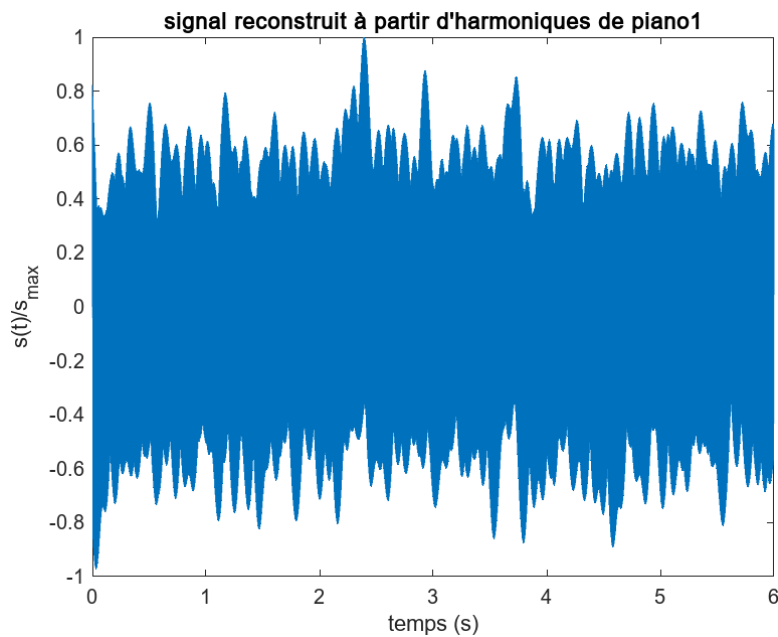


FIGURE 6 – Signal reconstruit à partir des 8 premières harmoniques de piano1.wav

1.4 Enveloppe ADSR

L'enveloppe ADSR est un filtre qui permet de transformer notre son en fonction de certains paramètres. Notre objectif est que le signal de la 9 ressemble au signal du piano1 qu'on affiche ci-dessous :

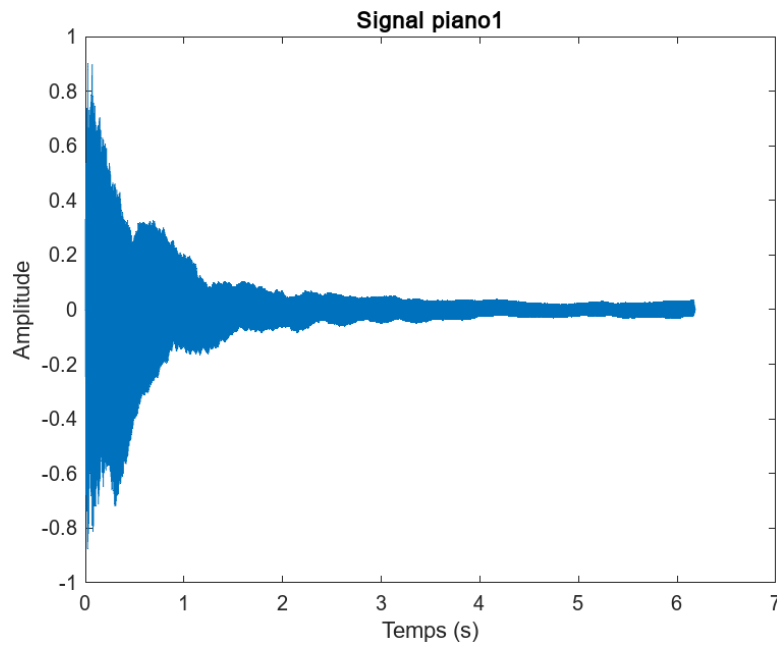


FIGURE 7 – Signal de piano1

L'ADSR pour attack, decay, sustain et release est un type de filtre qui utilise une enveloppe. Dans notre cas on prend une enveloppe qui correspond à notre signal piano1 afin que le signal crée à partir des harmoniques lui ressemble.

En superposant les signaux sur une même figure, on obtient :

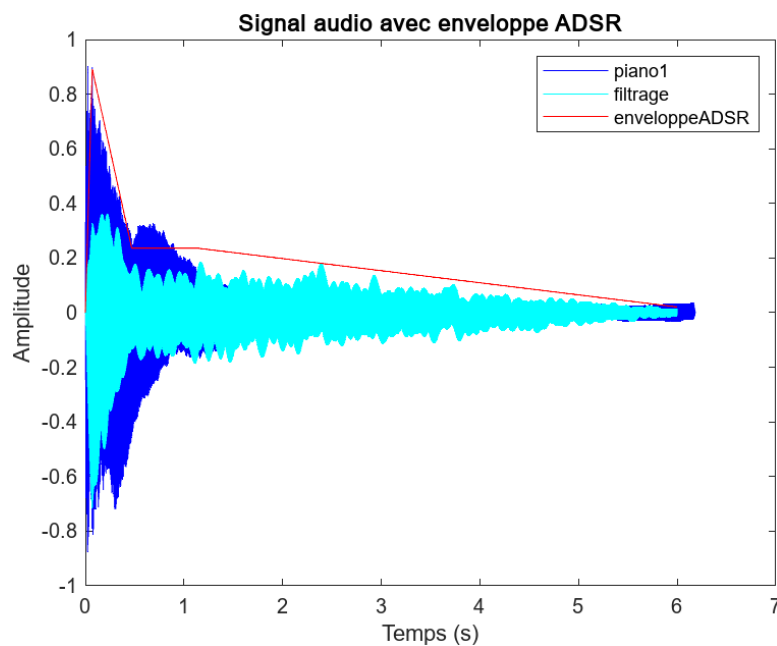


FIGURE 8 – signal reconstruit traité avec l'enveloppe ADSR en comparaison avec le signal piano1

A l'écoute le signal filtré ne ressemble pas beaucoup au piano1 mais il est filtré assez fortement et sa reconstruction se base uniquement sur les premières harmoniques de piano1, ce qui ne permet pas de recréer le son

très précisément. Cependant, la comparaison reste intéressante.

1.5 Reconstruction du signal à partir de `ifft()`

Pour cette partie nous obtenons le signal normalisé suivant :

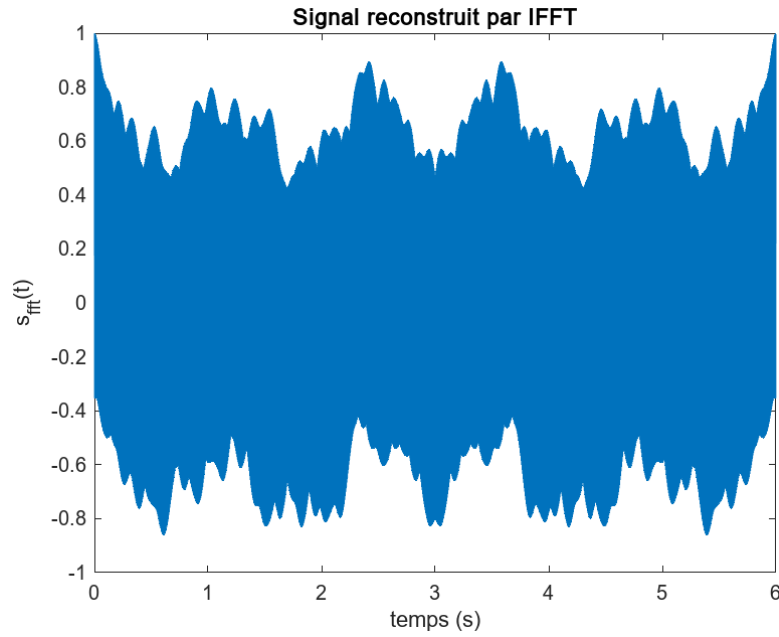


FIGURE 9 – Signal reconstruit à partir des harmoniques de piano1 et de la transformée de fourrier inverse

Nous remarquons que le signal crée ici est assez différent du signal crée directement en temporel. En effet, il y avait de nombreux pics sur l'autre signal alors que dans celui-là on peut voir une oscillation avec des temps de maintien légèrement plus importants.

Enfin, à l'oreille il semble beaucoup plus atténué que le premier.

2 Synthèse soustractive

2.1 Calcul du spectre d'un signal carré et d'un signal en dents de scie

2.1.1 Calcul du spectre

On considère des signaux **périodiques**, **centés**, de période T . Comme ils sont périodiques, on peut les écrire avec la série de Fourier complexe :

$$x(t) = \sum_{n=-\infty}^{+\infty} c_n e^{i2\pi n f_0 t}, \quad \text{où } f_0 = \frac{1}{T}$$

avec :

$$c_n = \frac{1}{T} \int_{-T/2}^{T/2} x(t) e^{-i2\pi n f_0 t} dt$$

■ Le signal carré $x(t)$ est défini par :

$$x(t) = \begin{cases} +1, & -\frac{T}{2} < t < 0 \\ -1, & 0 < t < \frac{T}{2} \end{cases} \quad \text{et } x(t+T) = x(t)$$

Calculons les coefficients c_n .

On a :

$$\begin{aligned} c_n &= \frac{1}{T} \left(\int_{-T/2}^0 1 \cdot e^{-i2\pi n f_0 t} dt + \int_0^{T/2} (-1) \cdot e^{-i2\pi n f_0 t} dt \right) = a_n + ib_n \\ &= \frac{1}{T} \left[\frac{e^{-i2\pi n f_0 t}}{-i2\pi n f_0} \right]_{-T/2}^0 - \left[\frac{e^{-i2\pi n f_0 t}}{-i2\pi n f_0} \right]_0^{T/2} \end{aligned}$$

En utilisant $f_0 = \frac{1}{T}$ et $e^{i\pi n} + e^{-i\pi n} = 2(-1)^n$:

$$c_n = \frac{1}{i\pi n} (1 - (-1)^n) \Rightarrow c_n = \begin{cases} \frac{2}{i\pi n}, & n \text{ impair} \\ 0, & n \text{ pair} \end{cases}$$

Nous obtenons donc le spectre suivant :

$$|c_n| = \begin{cases} \frac{2}{\pi|n|}, & n \text{ impair} \\ 0, & n \text{ pair} \end{cases} \quad \text{et} \quad \arg(c_n) = -\frac{\pi}{2} \cdot \text{sign}(n)$$

■ Le signal en dent de scie est donné sur $[-T/2, T/2]$ par :

$$x(t) = \frac{2}{T}t, \quad x(t+T) = x(t)$$

C'est aussi un signal impair.

Le calcul des coefficients nous donne :

$$c_n = \frac{2}{T^2} \int_{-T/2}^{T/2} t e^{-i2\pi n f_0 t} dt$$

Intégration par parties :

$$u = t, \quad dv = e^{-i2\pi n f_0 t} dt \Rightarrow v = \frac{e^{-i2\pi n f_0 t}}{-i2\pi n f_0}$$

$$c_n = \frac{2}{T^2} \left[\frac{te^{-i2\pi n f_0 t}}{-i2\pi n f_0} \Big|_{-T/2}^{T/2} - \int_{-T/2}^{T/2} \frac{e^{-i2\pi n f_0 t}}{-i2\pi n f_0} dt \right]$$

En simplifiant, on obtient :

$$c_n = \frac{i}{\pi n}, \quad n \neq 0, \quad c_0 = 0$$

Nous obtenons le spectre suivant :

$$|c_n| = \frac{1}{\pi|n|}, \quad \arg(c_n) = \frac{\pi}{2} \cdot \text{sign}(n)$$

Méthode numérique de vérification

Pour la méthode numérique, nous allons nous contenter de calculer le spectre en amplitude avec la `fft()`. Pour les 2 signaux, nous avons utilisé une fréquence d'échantillonnage $f_e = 10^4 \text{ Hz}$ pour éviter le repliement et une période $T = 0.1 \text{ s}$. Ce qui nous donne les courbes suivantes :

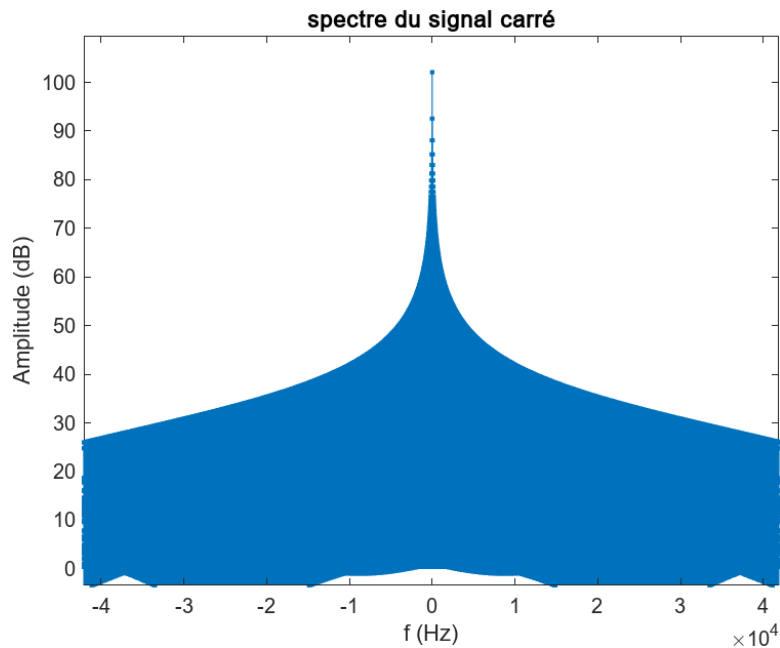


FIGURE 10 – Spectre du signal carré

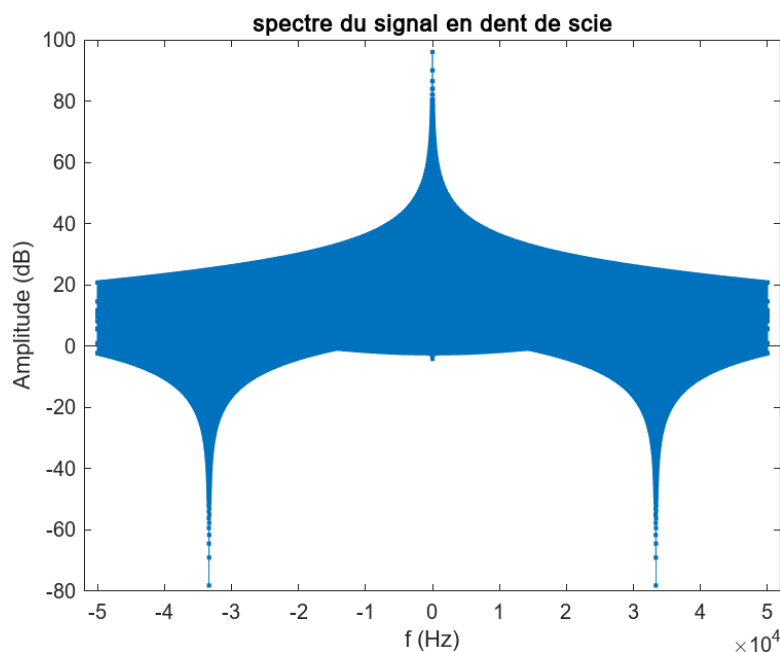


FIGURE 11 – Spectre du signal en dent de scie

Maintenant si on veut obtenir c_n , il nous faut trouver l'amplitude des harmoniques du signal à partir de nos

figures.

On a : $|c_n| = 10^{-\frac{A_{n,dB}}{20}}$ avec $A_{n,dB}$ l'amplitude en dB de la n-ième harmonique.

La limite de cette méthode est qu'elle ne permet que de récupérer le module de c_n et pas sa phase.

2.2 Filtre passe bas d'ordre 1

On considère un filtre défini par la relation de récurrence :

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

On cherche à calculer la transformée de Fourier de sa réponse impulsionnelle $h(k)$. Appliquons une impulsion $\delta(k)$ à l'entrée :

$$h(k) = \frac{1}{2}\delta(k) + \frac{1}{2}\delta(k-1)$$

En appliquant la transformée de Fourier, on obtient :

$$H(f) = \frac{1}{2}(1 + e^{-2\pi if}) = e^{-\pi if} \cos(\pi f)$$

Ainsi, le module et l'argument du spectre sont :

$$|H(f)| = |\cos(\pi f)|, \quad \arg(H(f)) = -\pi f$$

On remarque bien un filtre passe-bas, atténuant les hautes fréquences.

2.2.1 Simulation

Nous créons un signal carré de fréquence d'échantillonnage $f_e = 10^3 \text{ Hz}$ et de période $T = 0.1 \text{ s}$ pour lui appliquer le filtre.

Le spectre en décibel est donné ci-dessous :

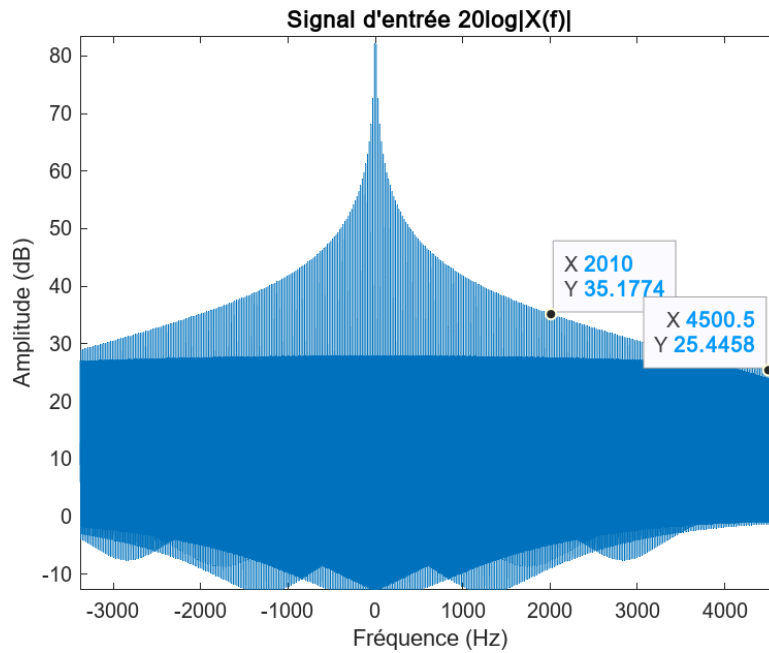


FIGURE 12 – Spectre du signal en dent de scie

Pour le filtrage, nous avons tracé 2 courbes, une où l'on applique le **filter(b,a,x)** à notre signal x et l'autre où nous réalisons le filtre théorique à l'aide des formules trouvées précédemment.

Nous observons alors la figure suivante :

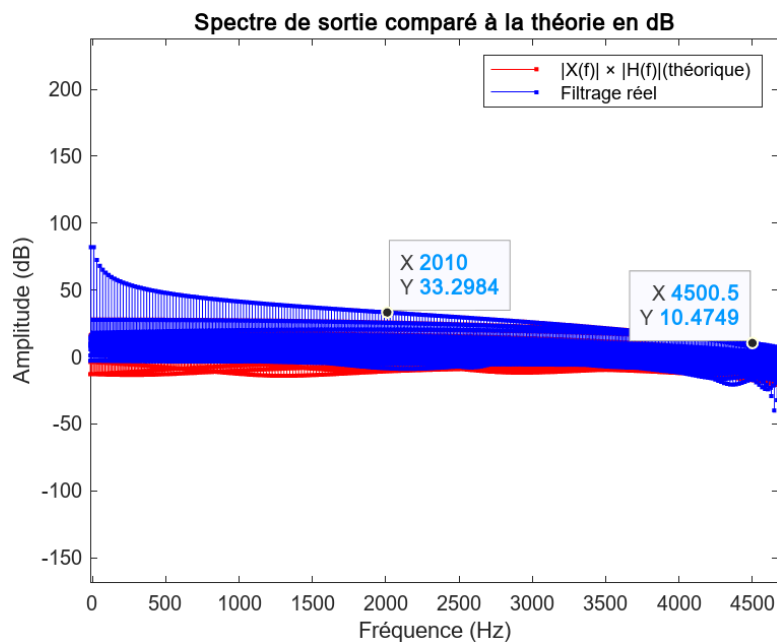


FIGURE 13 – Spectre du signal en dent de scie

Tout d'abord les spectres sont presque identiques entre la fonction **filter** et le théorique. Nous pouvons aussi remarquer la diminution en amplitude au fur et à mesure que les fréquences augmentent. Cela représente bien un filtre passe-bas.

2.3 Comparaison signaux periodiques

Pour la comparaison de nos signaux, nous allons procéder comme suit. Nous allons créer un signal carré et un en dent de scie. Après, nous filtrons les deux signaux avec le filtre de Matlab : `filter(b,a,x)` . Puis, on applique l'enveloppe ADSR développée précédemment afin de pouvoir comparer les sons obtenus à la synthèse additive précédente.

Nous commençons avec des signaux carrés et en dents de scie de fréquence d'échantillonnage $f_e = 44100Hz$ pour rester cohérent avec la synthèse additive et un période $T = 0.1s$.

Nous obtenons les courbes suivantes :

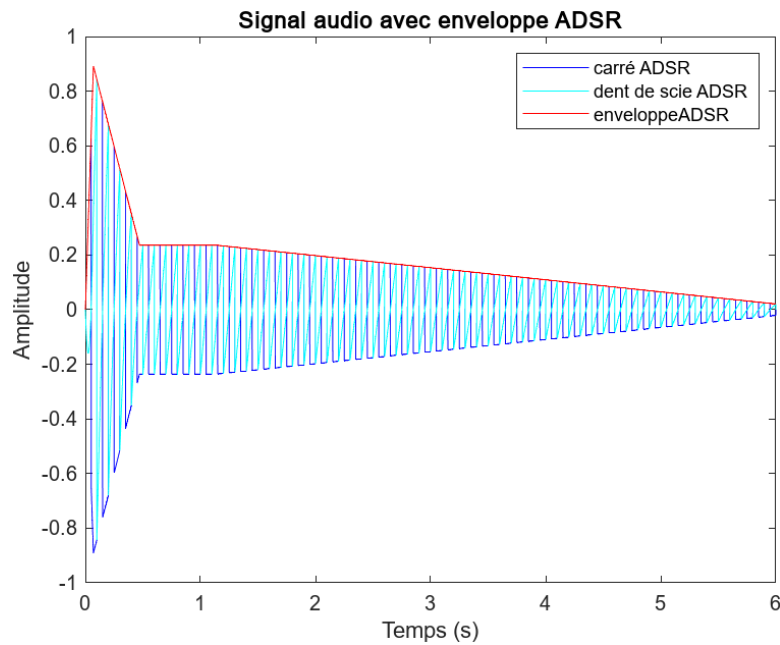


FIGURE 14 – Signal carré et en dent de scie avec l'enveloppe ADSR

A l'écoute nous entendons seulement des tapotements, nous supposons alors qu'il faut changer la fréquence de nos signaux car elle est trop faible.

Nous passons alors à une fréquence $f = 220Hz$ pour suivre la fondamentale de notre signal de piano1 ainsi que sa reconstruction à l'aide de sinusoides. Cela nous fait une période de $T = \frac{1}{220}s$.

Nous obtenons alors les signaux suivants :

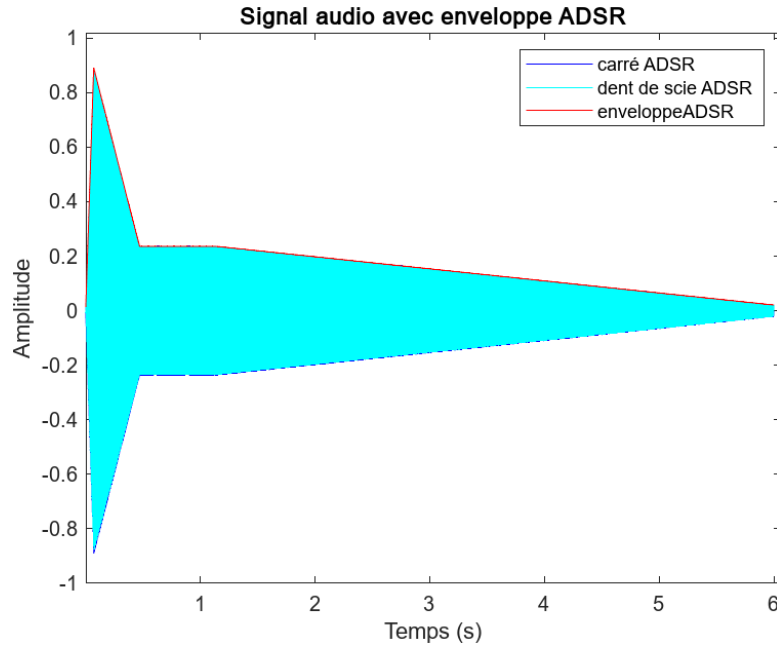


FIGURE 15 – Spectre du signal en dent de scie

Visuellement, les signaux sont confondus dans l'enveloppe ADSR, mais en écoutant les sons produits, nous entendons un son qui ressemble à un instrument synthétique, donc plus exploitable.

Le son semble beaucoup plus électronique qu'avec la synthèse additive.

2.4 Amélioration du signal produit par synthèse soustractive

Pour améliorer le filtrage, nous avons essayé plusieurs paramètres du filtre `designfilter()`. Nous en avons conclu que plus l'ordre était important, plus le signal de départ était filtré. Dans notre cas, il était important de beaucoup filtrer le signal pour le rendre moins perturbant.

Nous avons fini avec le filtre suivant : `designfilt('lowpassfir','FilterOrder', 40,'CutoffFrequency', 2500, 'SampleRate', 44100)` qui réduit l'aspect électronique du son et le rapproche du son produit avec la synthèse additive et donc du son de piano1 initial.

La synthèse additive reste à notre sens plus ressemblante au son original même si la recherche de paramètres nous laisse penser qu'un rapprochement par synthèse soustractive est envisageable.

3 Effets de réverbération

3.1 Calcul de R_{yx}

Nous avons :

$$R_{yx}(k) = y(k) * x^*(-k) = h(k) * x(k) * x^*(-k) = h(k) * R_{xx}(k)$$

3.2 Approximation et méthode de calcul de h

Si $R_{xx}(k) \approx d(k)$ alors : $R_{yx}(k) \approx h(k) * d(k)$.

Finalement :

$$R_{yx}(k) \approx h(k)$$

Pour la méthode d'estimation, il nous faut trouver un signal d'entrée $x(k)$ dont $R_{xx}(k) \approx d(k)$. On calcule alors R_{yx} et on en déduit $h(k)$.

3.3 Choix d'un signal adapté pour mesurer réponse impulsionnelle de la pièce

Pour choisir notre signal, il nous faut calculer l'autocorrélation des 2 signaux disponibles **xe1** et **xe2**. Nous récupérons alors l'autocorrélation et le décalage avec la fonction **xcorr**.

On trace alors l'autocorrélation des 2 signaux :

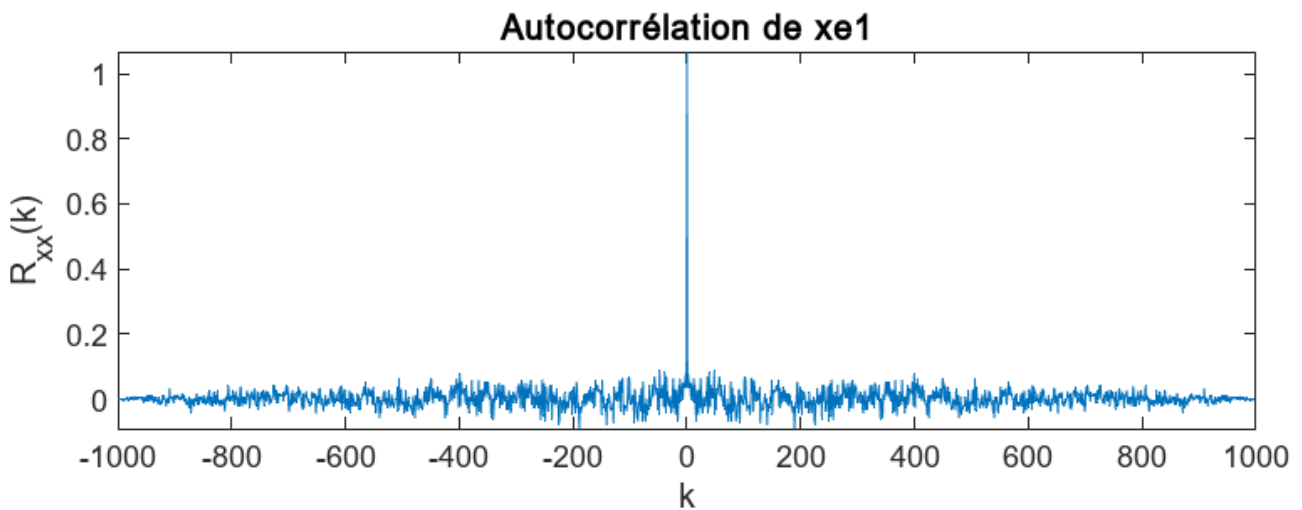


FIGURE 16 – Autocorrélation du signal xe1

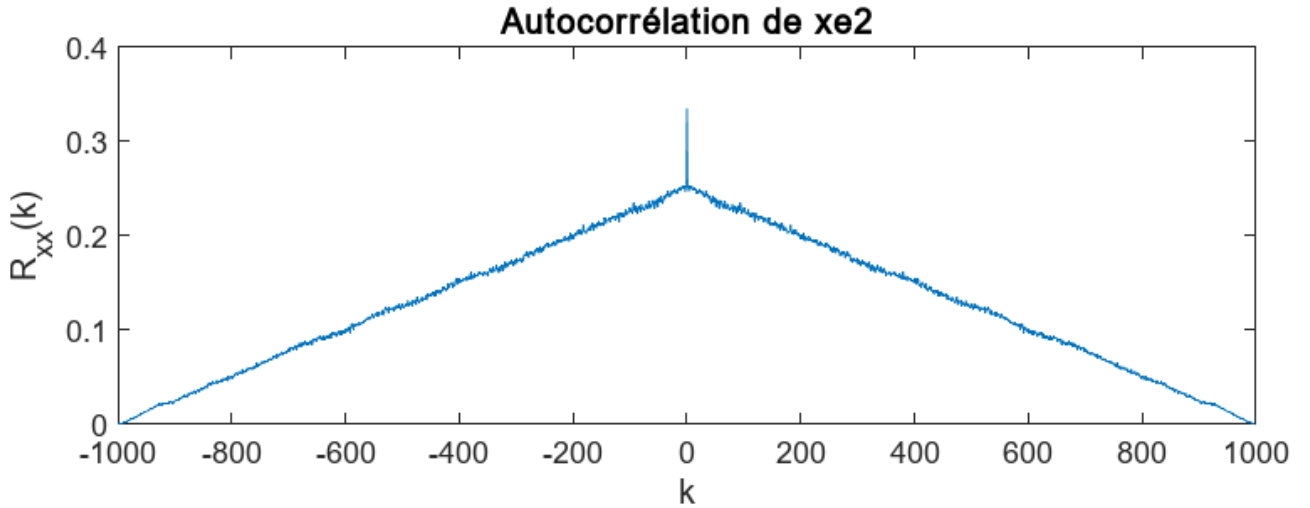


FIGURE 17 – Autocorrélation du signal xe2

Nous remarquons que l'autocorrélation de xe1 correspond à un Dirac, ce qui permet de valider les hypothèses précédentes. Ainsi, on peut choisir xe1 comme signal pour mesurer la réponse impulsionnelle de la pièce.

3.3.1 Méthodologie du calcul de la réponse impulsionnelle

Nous appliquons l'effet de la pièce à notre signal xe1 avec la fonction `simule_piece` pour obtenir notre signal y.

Ensuite, on utilise la fonction de corrélation `xcorr` pour calculer l'intercorrélation $R_{yx}(k)$ qui nous donne la réponse impulsionnelle $h(k)$.

Il nous reste à convertir notre discrétisation en temps en divisant k par f_e . Nous obtenons alors la réponse impulsionnelle suivante :

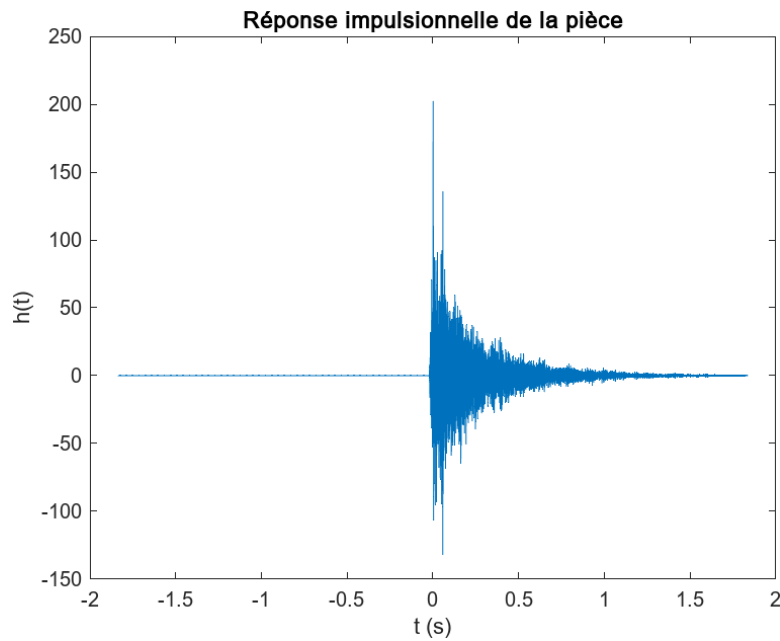


FIGURE 18 – Réponse impulsionnelle de la pièce en fonction du temps

3.4 Comparaison convolution classique et rapide

Pour la convolution classique, nous utilisons la fonction `effet_reverb` dont le temps d'exécution calculé en moyenne sur 5 essais donne : $0.311s$.

De son côté, la convolution rapide utilisant `effet_reverb_FFT` s'exécute en moyenne en : $0.289s$.

D'un point de vue auditif, les sons produits à partir des 2 reverbs semblent les mêmes. Nous traçons les signaux obtenus à partir des reverbs appliquées au son de guitare `nylon_guitar.wav` :

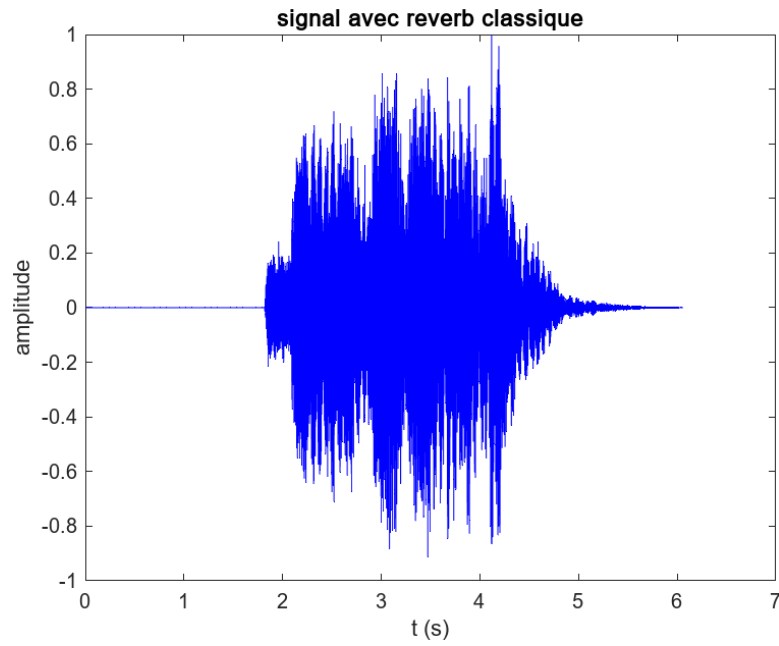


FIGURE 19 – Signal d’une guitare obtenu avec une convolution classique

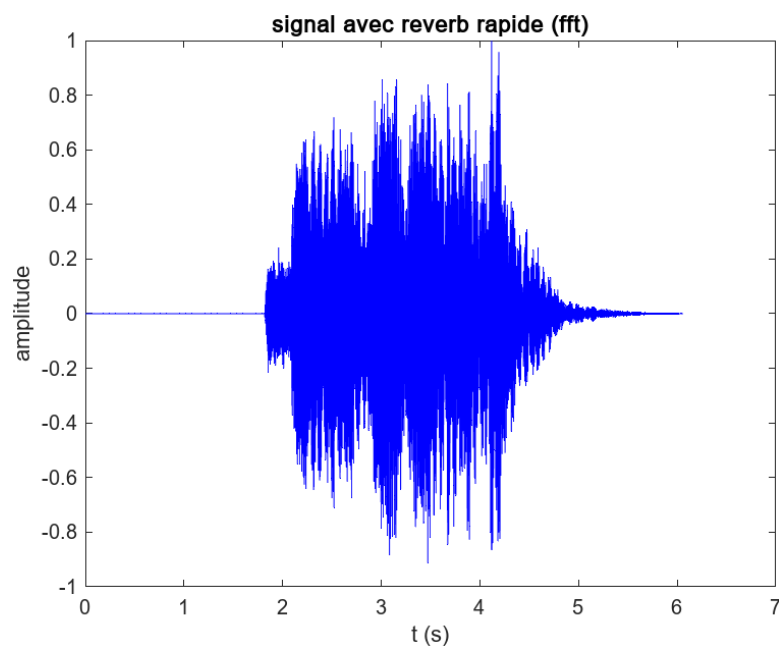


FIGURE 20 – Signal d’une guitare obtenu avec une convolution rapide

Nous remarquons que les tracés des signaux sont très similaires. Il est vrai que la convolution rapide avec la FFT semble être légèrement plus rapide en moyenne, mais les valeurs de temps restent très proches. D’après ces observations, nous pouvons conclure que ces méthodes produisent le même résultat.

3.5 Effet de retard

On considère le filtre défini par la relation de récurrence suivante (Delay) :

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

où $g \in \mathbb{R}$ est un coefficient d'atténuation et $\tau \in \mathbb{N}^*$ le nombre d'échantillons de retard du signal. On souhaite déterminer la réponse impulsionnelle $h(k)$. Pour cela on utilise : $x(k) = \delta(k)$.

En remplaçant dans l'équation du filtre, on obtient :

$$h(k) = \delta(k) - g \cdot h(k - \tau)$$

Nous construisons la solution par récurrence en supposant que le filtre est causal. C'est-à-dire : $\forall k \in \mathbb{N}, h(-k) = 0$.

On a alors :

$$h(0) = \delta(0) - g \cdot h(-\tau) = 1 \quad (\text{car } h(-\tau) = 0)$$

Si $k \in \{1, 2, \dots, \tau - 1\}$, alors :

$$h(k) = h(k - \tau) = 0 \quad (\text{car } k - \tau < 0)$$

Donc par récurrence :

$$h(k) = 0 \quad \text{si } k \notin \{n\tau \mid n \in \mathbb{N}\}$$

De plus :

$$h(\tau) = \delta(\tau) - g \cdot h(0) = -g$$

$$h(2\tau) = \delta(2\tau) - g \cdot h(\tau) = g^2$$

$$h(3\tau) = -g \cdot h(2\tau) = -g^3$$

D'où par récurrence :

$$h(n\tau) = (-g)^n \quad \text{pour } n \in \mathbb{N}$$

Enfin :

$$h(k) = \begin{cases} (-g)^n & \text{si } k = n\tau \\ 0 & \text{sinon} \end{cases}$$

3.6 Stabilité du filtre Delay

Pour obtenir la stabilité, il faut la convergence de $\sum_{k=0}^{+\infty} |h(k)| = \sum_{n=0}^{+\infty} |g^{n\tau}| = \sum_{n=0}^{+\infty} |g^\tau|^n$. Il nous faut donc $|g^\tau| < 1$ et finalement $|g| < 1$.

3.7 Création du filtre sur Matlab

Pour créer le filtre sur Matlab, il nous faut l'application **filter(b,a,x)** avec a et b des vecteurs en paramètres qui permettent de définir l'équation du filtre et x le signal à filtrer.

Le paramètre b correspond au vecteur des coefficients de la partie non réursive, ici $b = [1]$. Le paramètre a correspond au vecteur des coefficients de la partie réursive, ici $a = [1, 0, 0, \dots, g]$, avec le g en position $\tau + 1$.

En effet on peut modéliser la situation de la façon suivante : $\sum_{i=0}^N a_i \cdot y[n - i] = \sum_{j=0}^M b_j \cdot x[n - j]$ Il faut la mettre en lien avec l'équation (Delay) ce qui nous permet de trouver les paramètres a et b.

3.8 Visualisation du filtre

Pour représenter la réponse impulsionnelle, nous avons appliqué la fonction **filter(b,a,x)** avec les paramètres définis précédemment à un dirac. Avec notre programme, nous obtenons la représentation suivante de la réponse impulsionnelle :

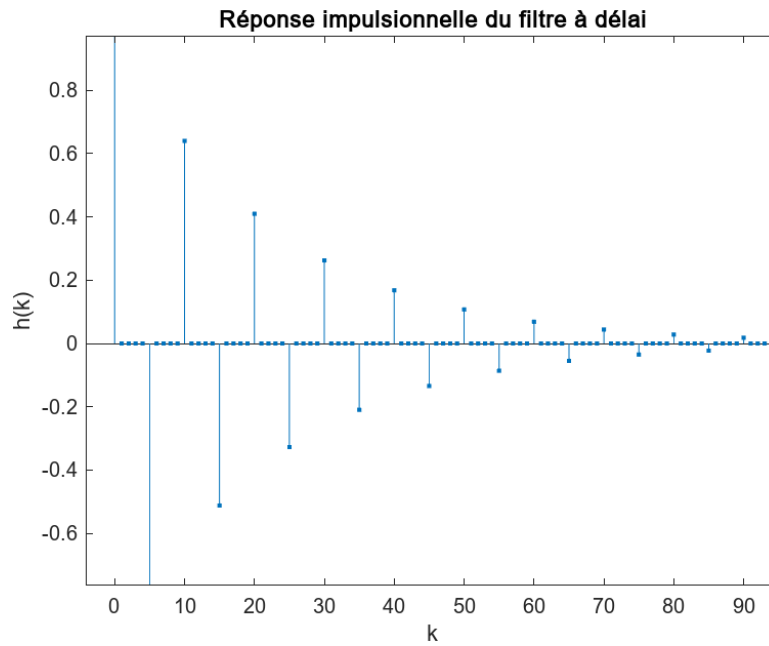


FIGURE 21 – Réponse impulsionnelle $h(k)$ du filtre à délai

Nous avons choisi un délai à 5 échantillons que nous pouvons observer. On remarque bien la diminution progressive du dirac en entrée.

3.9 Réponse du filtre en fréquence

On a l'expression de $h(k)$ suivante :

$$h(k) = \begin{cases} (-g)^n & \text{si } k = n\tau \\ 0 & \text{sinon} \end{cases}$$

On peut aussi l'écrire sous la forme :

$$h(k) = \sum_{n=0}^{\infty} (-g)^n \cdot \delta(k - n\tau)$$

La transformée de Fourier discrète (DTFT) de $h(k)$ est définie par :

$$H(\nu) = \sum_{k=-\infty}^{\infty} h(k) e^{-j2\pi\nu k}$$

En remplaçant $h(k)$, on obtient :

$$H(\nu) = \sum_{n=0}^{\infty} (-g)^n e^{-j2\pi\nu n\tau} = \sum_{n=0}^{\infty} (-ge^{-j2\pi\nu\tau})^n$$

Cette série géométrique converge si $|g| < 1$, et donne :

$$H(\nu) = \frac{1}{1 + ge^{-j2\pi\nu\tau}}$$

Le module de la réponse fréquentielle est :

$$|H(\nu)| = \left| \frac{1}{1 + ge^{-j2\pi\nu\tau}} \right| = \frac{1}{\sqrt{1 + 2g \cos(2\pi\nu\tau) + g^2}}$$

Et la phase est donnée par :

$$\arg(H(\nu)) = -\arg(1 + ge^{-j2\pi\nu\tau}) = -\arctan\left(\frac{g \sin(2\pi\nu\tau)}{1 + g \cos(2\pi\nu\tau)}\right)$$

Lorsque l'on trace la courbe de la réponse impulsionnelle obtenue théoriquement et celle obtenue numériquement, on obtient le graph suivant :

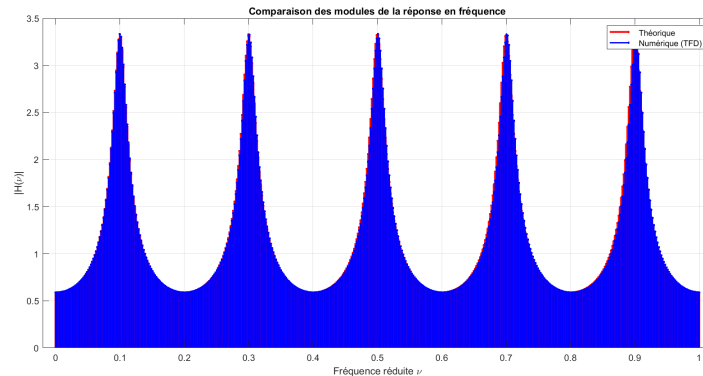


FIGURE 22 – Module de la réponse en fréquence théorique et numérique

On observe un léger retard de la part du signal numérique comme explicité sur l'image suivante :

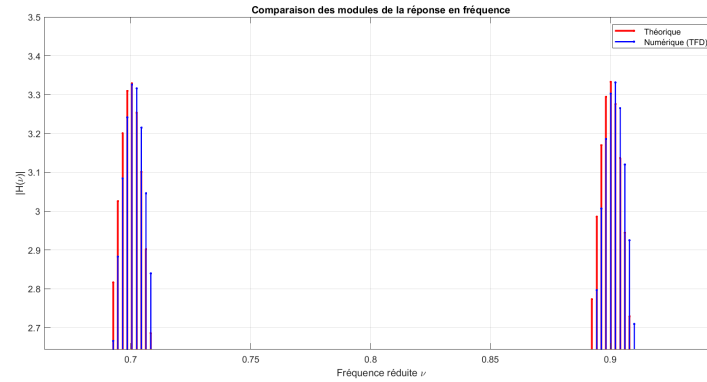


FIGURE 23 – Retard de la réponse théorique

Cela peut s'expliquer par la phase qui fait tourner notre spectre complexe lors de la TFD.

3.10 Filtrage delay

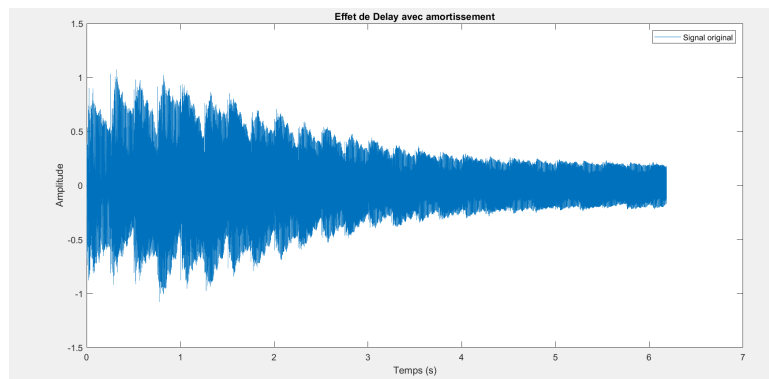


FIGURE 24 – Signal piano1 filtré avec le delay réalisé par moyenne sur K échantillons

3.11 Filtrage delay avec moyenne sur K échantillons

Nous appliquons notre programme avec les paramètres demandés et nous obtenons la figure suivante :

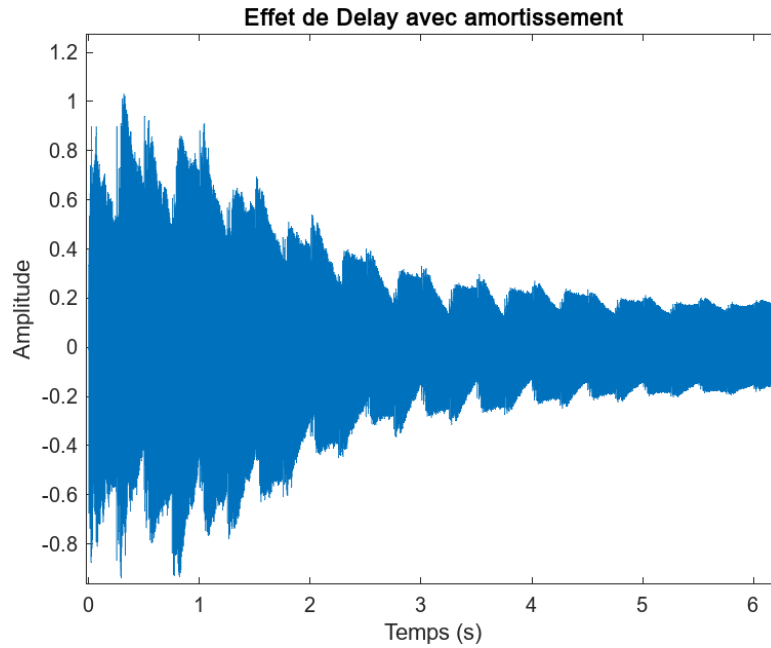


FIGURE 25 – Signal piano1 filtré avec le delay réalisé par moyenne sur K échantillons

3.12 Réponse en fréquence de l'effet de retard filtré

Pour terminer, nous affichons la réponse en fréquence du delay à l'aide de la fonction `freqz(b,a, nb_points, fe)` qui permet d'obtenir la réponse en fréquence du filtre directement à partir de ses paramètres a et b. Nous avons pris 2048 points et une fréquence d'échantillonnage $f_e = 44100 Hz$.

Nous obtenons le tracé suivant :

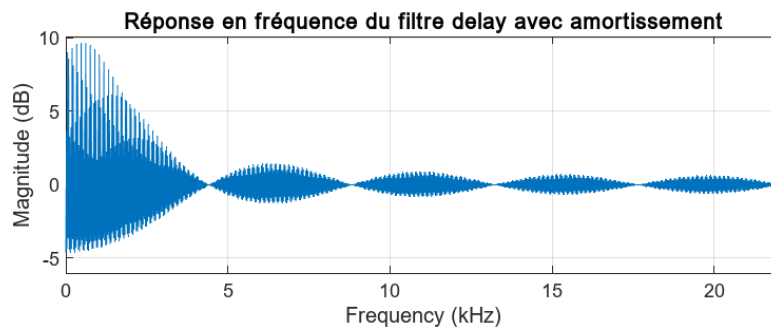


FIGURE 26 – Module de la réponse fréquentielle de l'effet de retard

Nous remarquons directement la baisse du module à partir de $4 kHz$. On a donc bien une coupure des hautes fréquences.

4 Conclusion

Pour conclure, nous avons effectué des constructions de signaux par méthode additive et soustractive et une étude d'effets applicables à des signaux sonores. Nous avons pu découvrir la construction de filtres tel que la

reverb et le delay. Ce sujet permet une première approche et une compréhension du fonctionnement de tels filtres très utilisés dans le domaine de la musique. À titre personnel, nous pratiquons de la musique notamment par ordinateur et utilisons donc ces filtres très régulièrement. C'est pourquoi nous avons trouvé leur étude et leur construction très enrichissante.