



27 mai 2025

Anne-Lise Pelletier - T  odore AUTULY

Groupe B1

Compte rendu de travaux pratiques : SAR Audio

SAR Traitement Audio Matlab



IMT Atlantique

Bretagne-Pays de la Loire
  cole Mines-T  l  com

Table des matières

1	Introduction	4
2	Synthèse additive	4
2.1	Analyse d'un son harmonique	4
2.1.1	Violoncelle	5
2.1.2	Piano 1	5
2.1.3	Piano 2	6
2.1.4	Saxophone Alto	7
2.1.5	Saxophone Soprano	8
2.1.6	Trompette	9
2.2	Analyse de l'inharmonicité des deux sons de piano	11
2.3	Synthèse	12
3	Synthèse soustractive	16
3.1	Définition des signaux périodiques et tracé de leur spectre	16
3.2	Filtrage passe-bas d'ordre 1	18
4	Effets audio-numériques	21
4.1	Effets de réverbération	21
4.1.1	Mesure de réponse impulsionnelle	22
4.1.1.1	Estimation de la réponse impulsionnelle :	22
4.1.2	Convolution classique	24
4.1.3	Convolution rapide	25
4.2	Effet de retard	26
4.2.1	Réponse impulsionnelle théorique de l'effet de delay	26
4.2.2	Condition de stabilité du filtre de delay	27
4.2.3	Déterminer les vecteurs a et b	27
4.2.4	Analyse de l'effet de delay	28
4.2.5	Réponse en fréquence du filtre de retard	29
4.2.5.1	Module de la réponse :	29
4.2.5.2	Phase de la réponse :	29
4.2.6	Comparaison : théorique vs numérique	29

4.2.7	Test effet delay	30
4.2.8	Delay filtré	31
4.2.9	Réponse en fréquence de l'effet de retard filtré	31
4.3	Synthèse Karplus-Strong	32
5	Conclusion	34

1 Introduction

L'évolution des technologies numériques a profondément transformé les pratiques de traitement audio, en offrant des outils puissants pour analyser, manipuler et synthétiser des signaux audio. Ce projet s'inscrit dans le cadre du cours Electrical Engineering et vise à mettre en œuvre plusieurs techniques fondamentales à travers des expérimentations pratiques, réalisées à l'aide de l'environnement de développement Matlab.

Nous commencerons par explorer la synthèse additive, en analysant d'abord la structure harmonique d'un son à l'aide de la transformée de Fourier discrète. Cette analyse permettra de caractériser les composantes fréquentielles d'un instrument, d'étudier son inharmonicité et, par la suite, de reconstruire un signal en superposant plusieurs ondes sinusoïdales. On ajoutera une enveloppe ADSR, que l'on comparera à une synthèse effectuée via la transformée de Fourier inverse.

Nous aborderons ensuite la synthèse soustractive, une méthode complémentaire qui consiste à partir d'un signal riche en harmoniques, que l'on filtre afin de sculpter le timbre désiré. Cette approche fera appel à différents types de filtres numériques et permettra d'expérimenter leur impact sur le son.

La dernière partie du projet portera sur les effets audio-numériques, avec une attention particulière portée à la réverbération. Celle-ci sera modélisée par des systèmes linéaires invariants dans le temps et simulée via convolution. Nous étudierons également l'effet de retard, analysé sous l'angle de sa réponse impulsionnelle et fréquentielle.

À travers ces étapes, ce projet permettra de consolider les connaissances théoriques sur la représentation et la transformation des signaux audio, tout en offrant une approche concrète à la programmation de traitements audio.

2 Synthèse additive

2.1 Analyse d'un son harmonique

Afin d'analyser les caractéristiques fréquentielles de différents instruments, nous avons visualisé le spectre d'amplitude obtenu à l'aide de la transformée de Fourier discrète. Nous remarquons que le violoncelle et le saxophone ont des spectres similaires avec une sé-

rie d'harmoniques bien visibles. Le piano possède quant à lui un spectre plus resserré avec moins d'harmoniques présentes. La trompette possède un spectre similaire à celui du violoncelle et du saxophone mais avec des pics uniquement autour de la fréquence fondamentale du son.

Les fréquences fondamentales montrent que la plupart des instruments jouent approximativement la même note autour de 220 Hz, à l'exception du piano 2 et de la trompette qui jouent une octave plus haute autour de 440 Hz.

2.1.1 Violoncelle

Fréquence fondamentale du violoncelle : 220.8047 Hz.

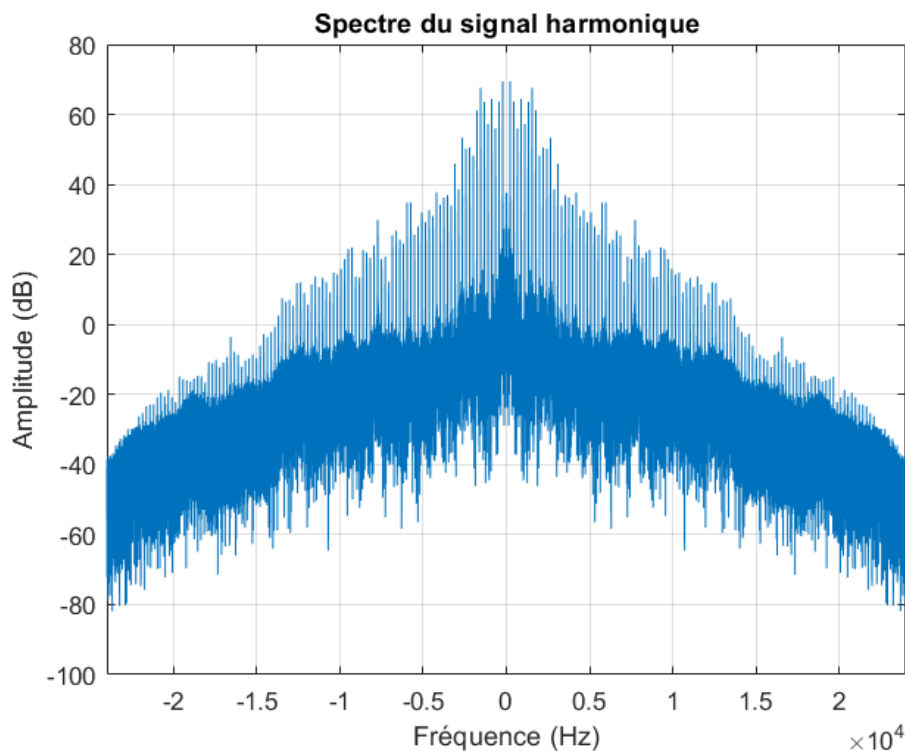


FIGURE 1 – Analyse spectrale d'une note de violoncelle

2.1.2 Piano 1

Fréquence fondamentale du piano 1 : 220.2939 Hz.

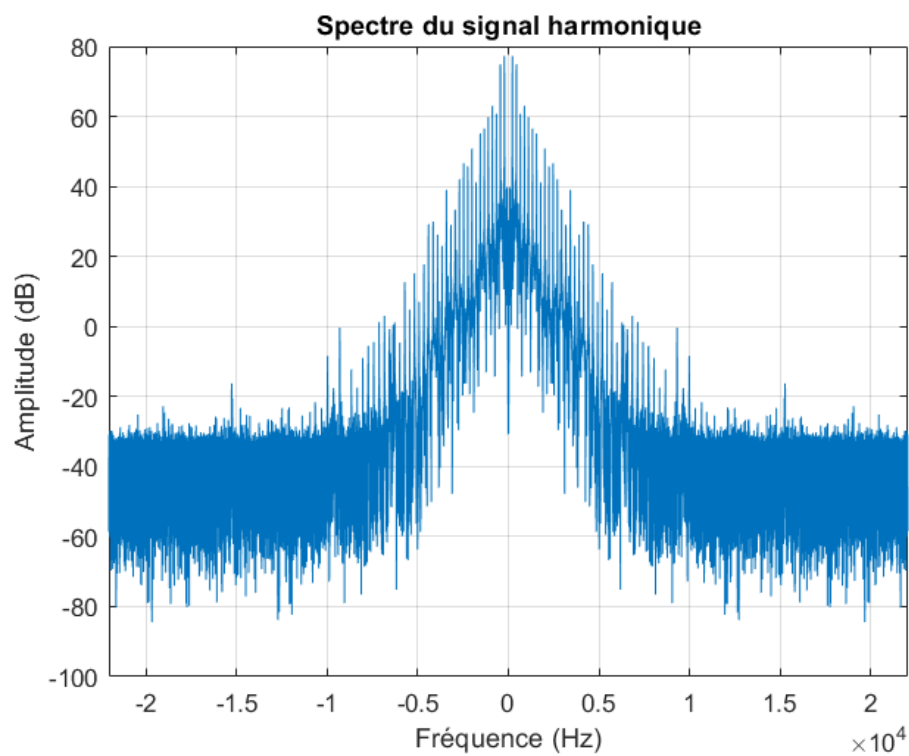


FIGURE 2 – Analyse spectrale d'une note de piano1

2.1.3 Piano 2

Fréquence fondamentale du piano 2 : 442.5712 Hz.

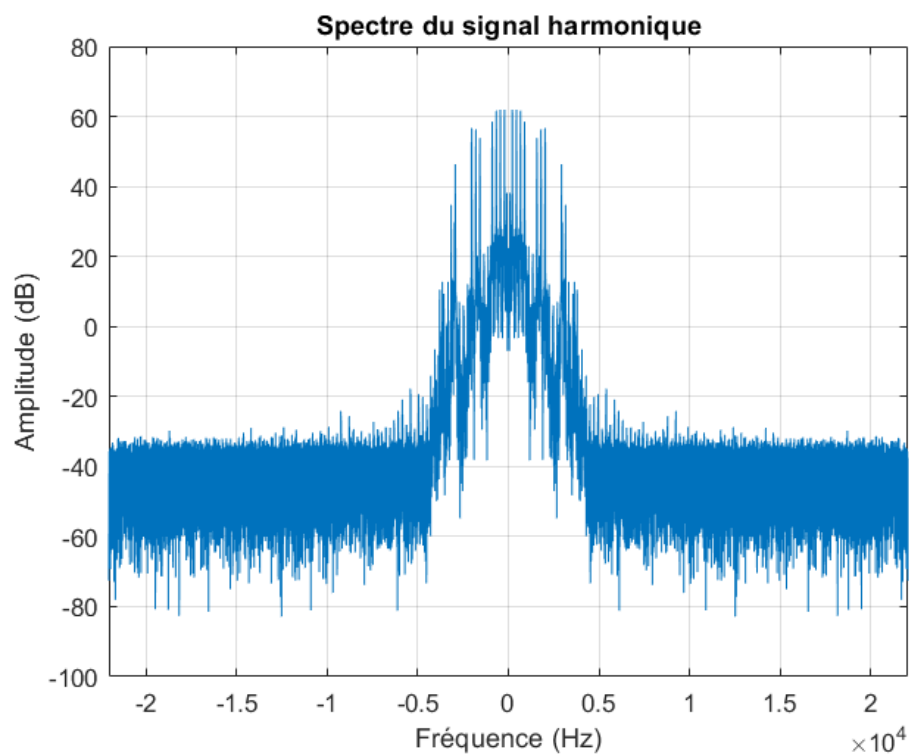


FIGURE 3 – Analyse spectrale d'une note de piano2

2.1.4 Saxophone Alto

Fréquence fondamentale du saphoxone alto : 220.6615 Hz.

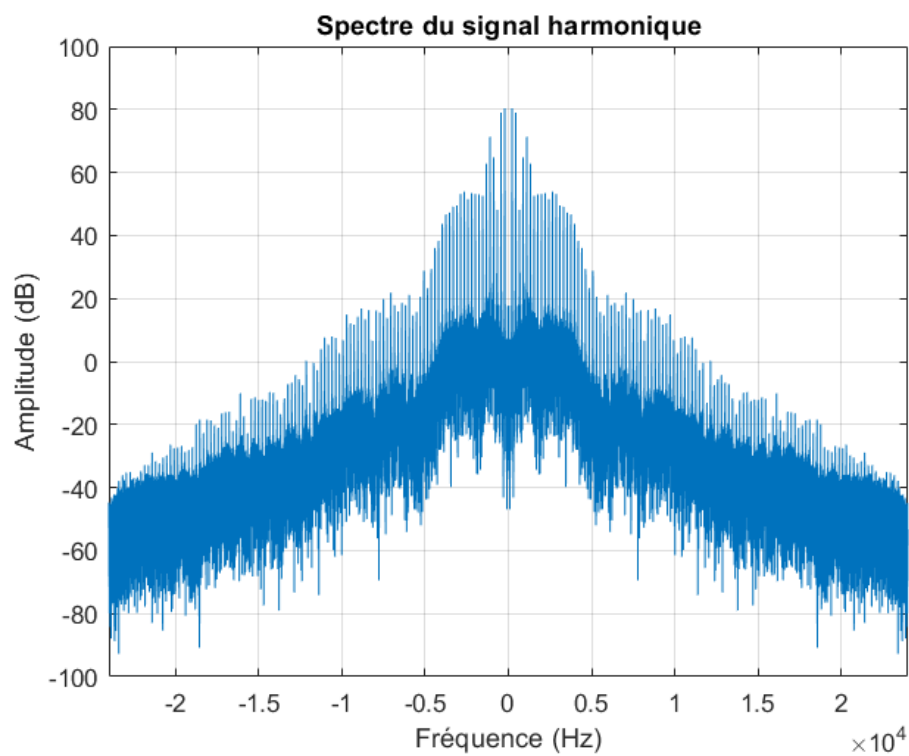


FIGURE 4 – Analyse spectrale d'une note de saxophone alto

2.1.5 Saxophone Soprano

Fréquence fondamentale du saxophone soprano : 221.2479 Hz.

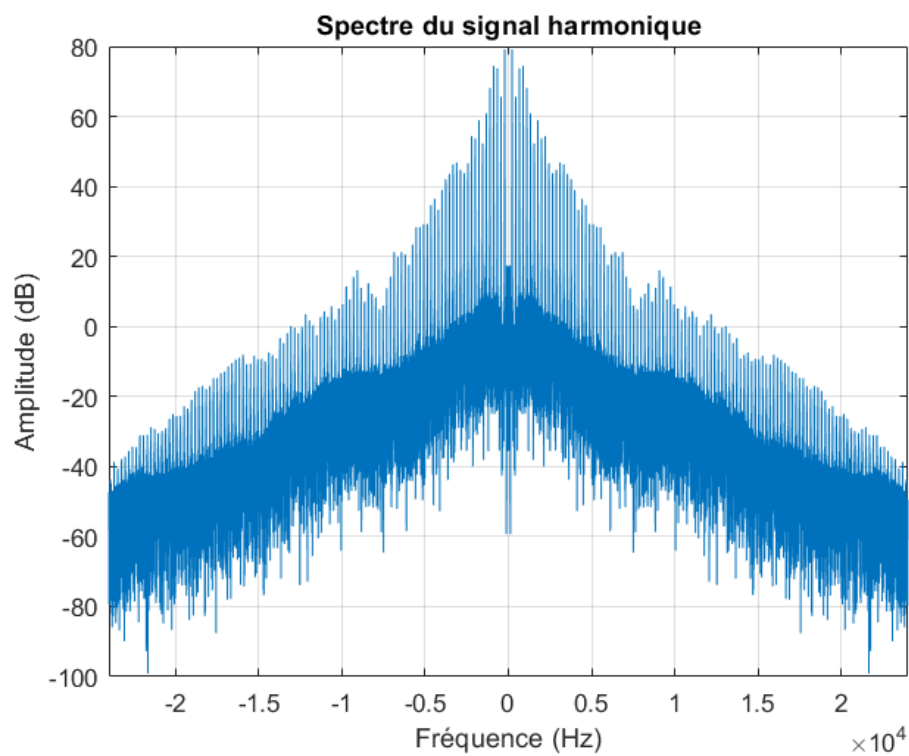


FIGURE 5 – Analyse spectrale d’une note de saxophone soprano

2.1.6 Trompette

Fréquence fondamentale de la trompette : 450.3337 Hz.

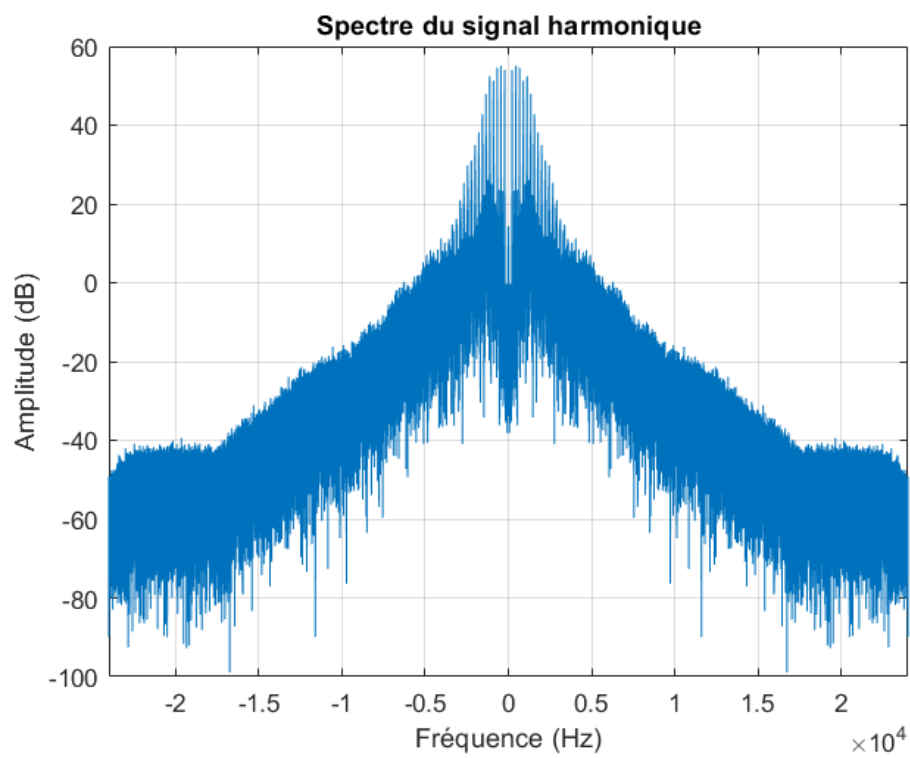


FIGURE 6 – Analyse spectrale d'une note de trompette

2.2 Analyse de l'inharmonicité des deux sons de piano

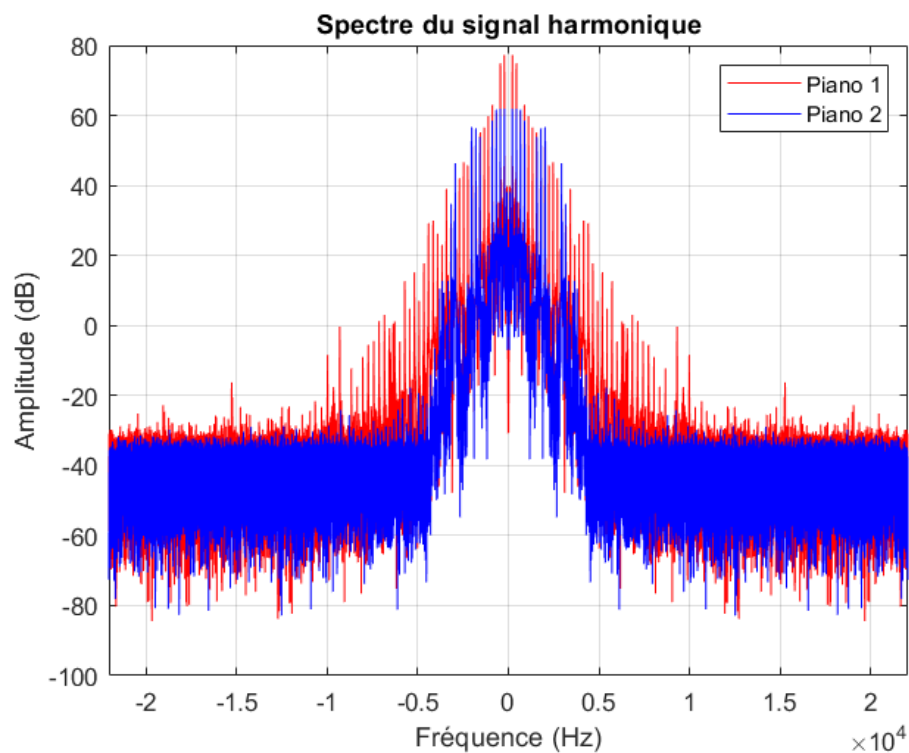


FIGURE 7 – Comparaison des spectres de deux pianos

Fréquence (Hz)	f_1	f_2	f_3	f_4	f_5	f_6
Théorique (Piano 1)	220	440	660	880	1100	1320
Mesurée (Piano 1)	220.29	440.58	660.88	881.17	1101.4	1321.7
Inharmonicité (Piano 1)	0	5.3	4.2	3.3	2.5	2.2
Théorique (Piano 2)	440	660	880	1100	1320	1540
Mesurée (Piano 2)	442.57	885.14	1327.7	1770.2	2212.8	2655.4
Inharmonicité (Piano 2)	0	8.8	5.6	3.1	1.0	0.42

TABLE 1 – Comparaison des fréquences harmoniques théoriques et mesurées pour deux pianos, et calcul de l'inharmonicité en cents.

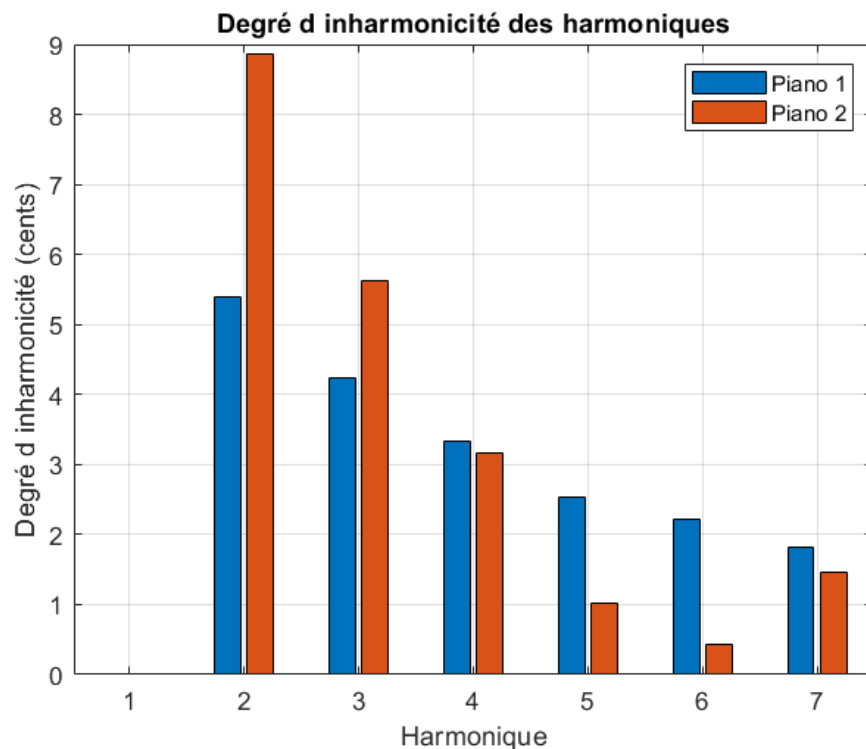


FIGURE 8 – Comparaison des inharmonicités des deux pianos

Le piano le plus harmonieux est le piano 1. En observant sa représentation spectrale, on remarque que les pics fréquentiels sont clairement définis et se répètent de manière régulière à des intervalles correspondant à des multiples entiers de la fréquence fondamentale, ce qui caractérise une structure harmonique. À l'inverse, le spectre du piano 2 présente des irrégularités par rapport au premier piano (très légères) et cela traduit une inharmonicité plus importante. Par conséquent, le piano 1 est perçu comme plus harmonieux, la partie synthèse se concentre sur ce dernier.

2.3 Synthèse

Ajout d'une enveloppe ADSR :

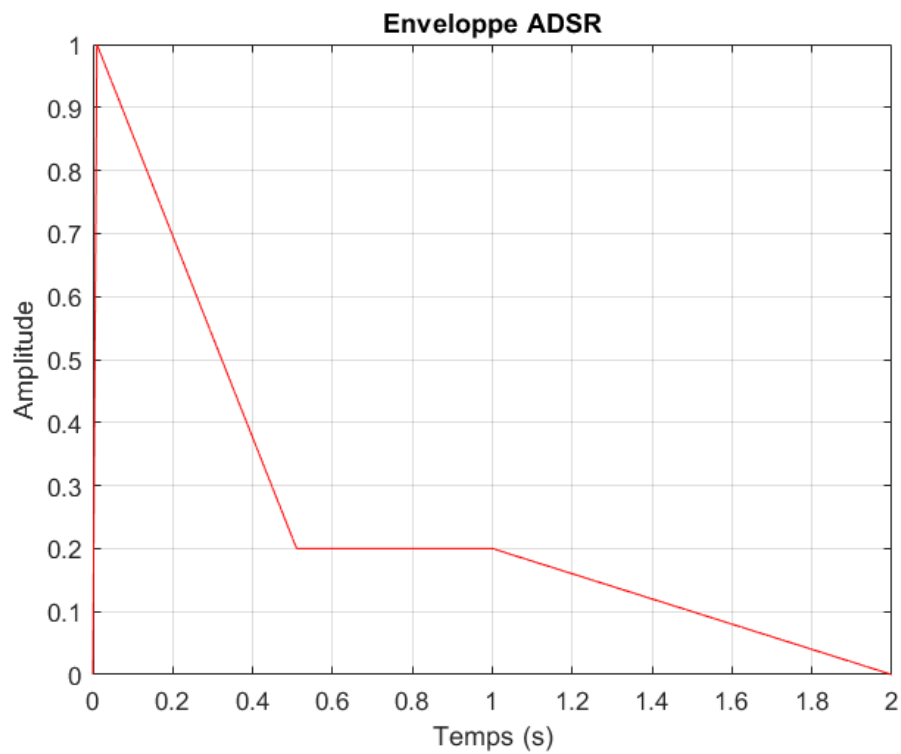


FIGURE 9 – Enveloppe ADSR

L'enveloppe ADSR (Attack, Decay, Sustain, Release) permet de moduler l'intensité d'un son au cours du temps, rendant ainsi sa synthèse plus réaliste et naturelle en reproduisant fidèlement l'évolution dynamique d'un instrument réel.

Voici ce que l'on obtient en superposant les sinusoïdes avec l'enveloppe ADSR :

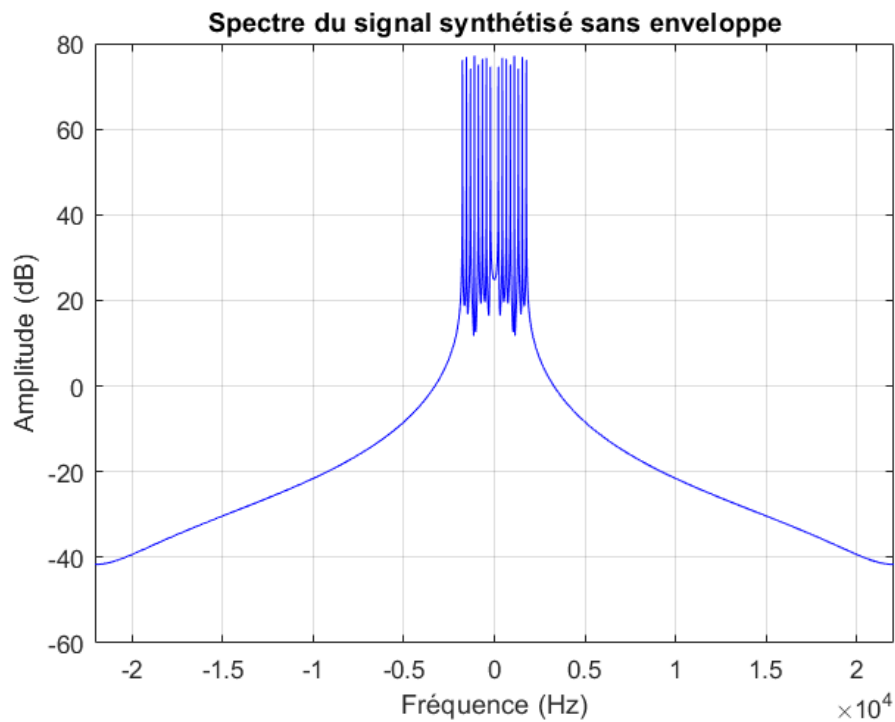


FIGURE 10 – Synthèse sonore sans enveloppe ADSR

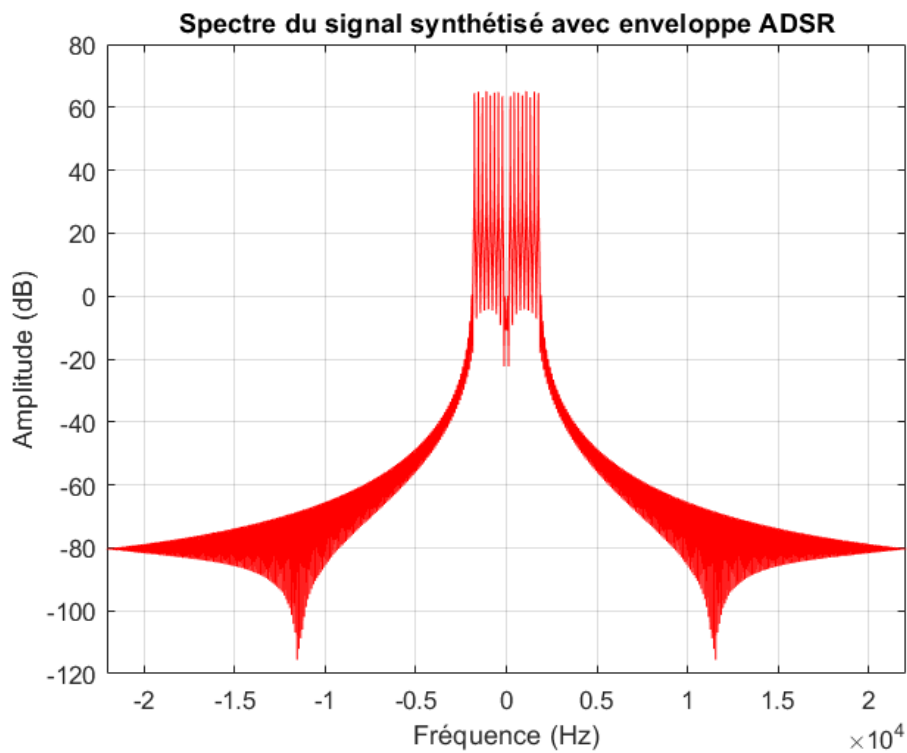


FIGURE 11 – Synthèse sonore avec enveloppe ADSR

Comparaison avec une synthèse par la transformée de Fourier discrète inverse :

2. Synthèse additive

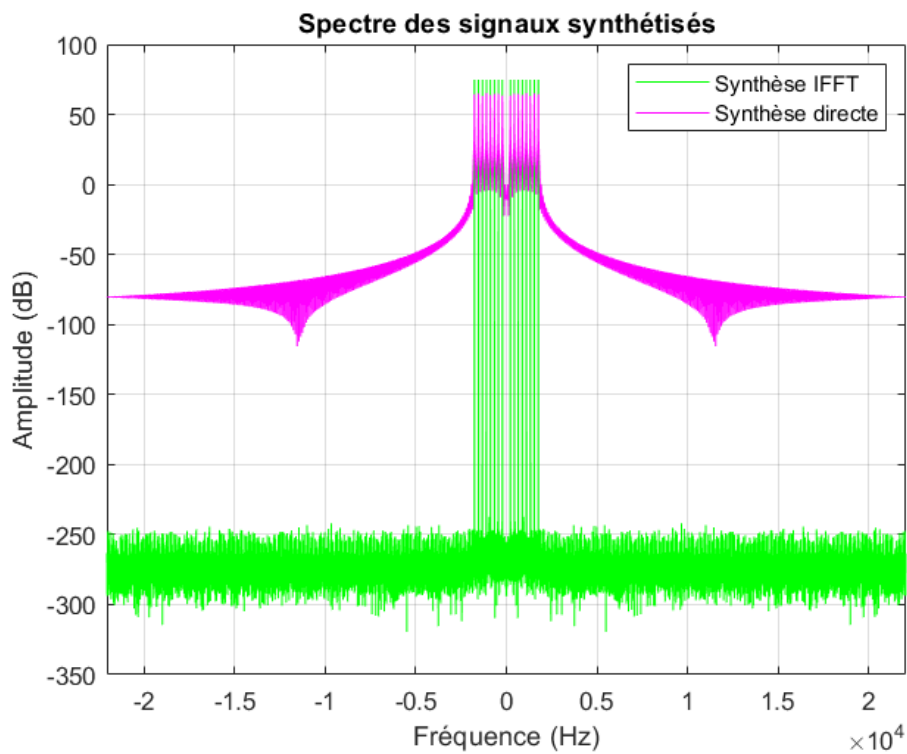


FIGURE 12 – Synthèse par IFFT et synthèse par somme directe

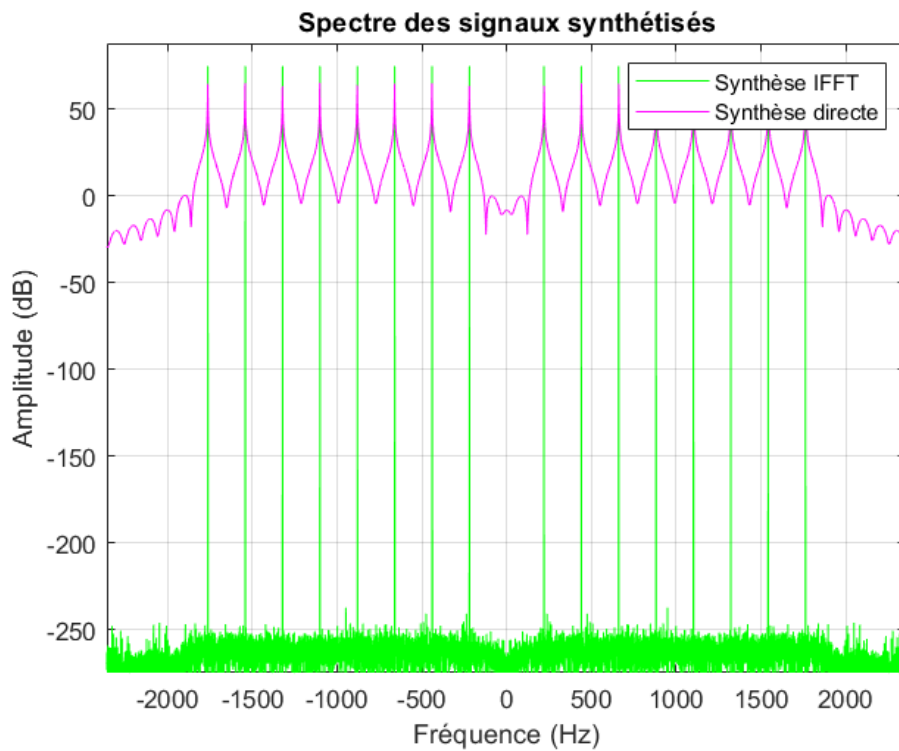


FIGURE 13 – Synthèse par IFFT et synthèse par somme directe avec zoom

En partant des mêmes amplitudes et fréquences harmoniques, nous avons réalisé une

synthèse sonore en utilisant cette fois la transformée de Fourier discrète inverse (IFFT). Le signal obtenu est identique à celui généré par la somme directe des sinusoïdes (en considérant les 2 parties roses éloignées de la fondamentale comme ayant un impact négligeable sur le spectre étant donné leur amplitude de -50dB). Cela s'explique par le fait que les deux méthodes reposent sur les mêmes composantes fréquentielles : l'une les additionne explicitement dans le domaine temporel, tandis que l'autre les reconstruit via une transformation mathématique inverse depuis le domaine fréquentiel.

3 Synthèse soustractive

3.1 Définition des signaux périodiques et tracé de leur spectre

Après avoir exploré la synthèse additive, nous abordons ici la synthèse soustractive. Cette technique consiste à partir d'un signal périodique (comme un signal carré ou dent de scie), de filtrer ce signal et obtenir le timbre désiré.

Pour calculer le spectre d'un signal carré de période T , centré et d'amplitude ± 1 , on peut utiliser son développement en série de Fourier. Ce signal ne contient que des composantes impaires, et les coefficients de Fourier décroissent proportionnellement à $\frac{1}{n}$. Le spectre en amplitude est donc :

$$X_n = \frac{4}{\pi n}, \quad \text{pour } n \text{ impair}$$

et

$$X_n = 0, \quad \text{pour } n \text{ pair.}$$

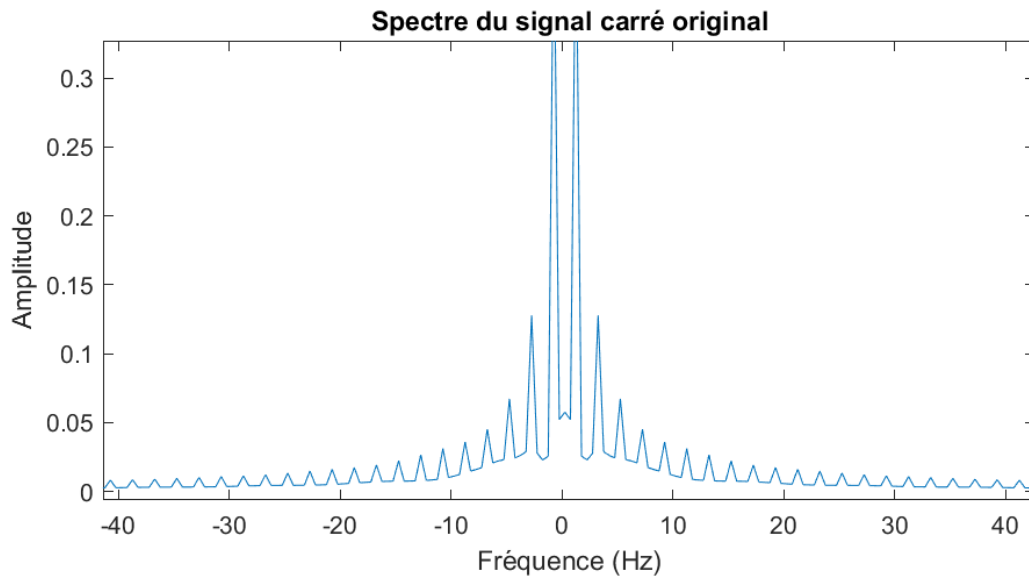


FIGURE 14 – Spectre d'un signal carré

Pour un signal dent de scie également centré et de même amplitude et période, le développement en série de Fourier donne cette fois toutes les composantes (paires et impaires), avec des coefficients également décroissants en $\frac{1}{n}$:

$$X_n = \frac{-1}{\pi n}, \quad \text{pour } n \geq 1$$

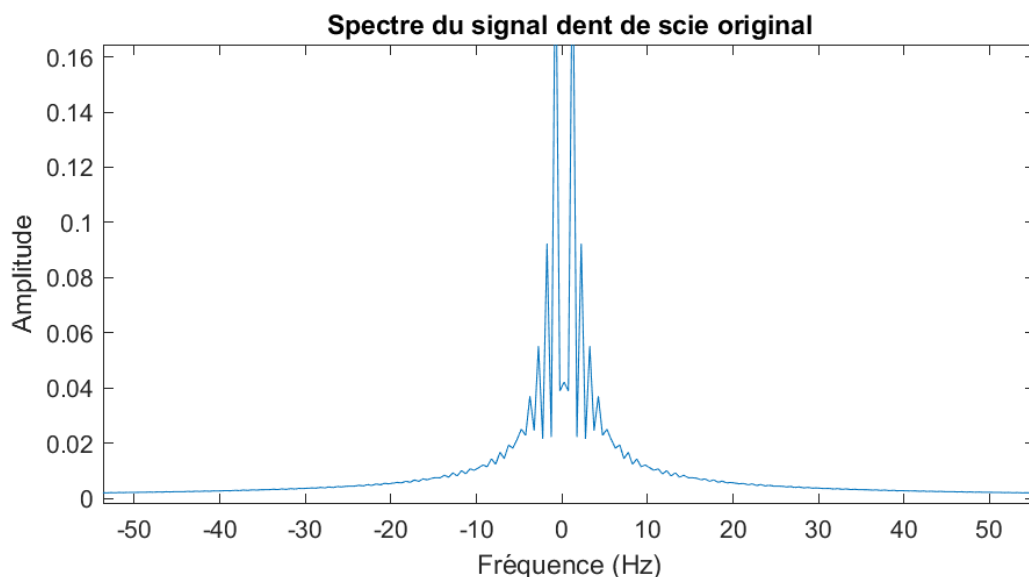


FIGURE 15 – Spectre d'un signal dent de scie

Si les coefficients du développement en série de Fourier deviennent nuls au-delà d'un certain rang, le spectre est alors à bande limitée. Cependant, dans la réalité, les signaux

possèdent souvent des spectres à bande illimitée, ce qui les rend non stockables numériquement. C'est l'une des principales limites du modèle numérique. De plus, la fréquence d'échantillonnage est elle aussi limitée, alors que certains signaux comportent un nombre infini d'harmoniques, avec des fréquences de plus en plus élevées. Néanmoins, avec la puissance des ordinateurs actuels, il est possible de traiter ces spectres de manière suffisamment précise. Même si le traitement reste imparfait, les différences sont généralement inaudibles pour l'oreille humaine.

3.2 Filtrage passe-bas d'ordre 1

Le filtrage passe-bas d'ordre 1 est défini par la relation suivante :

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

Il s'agit d'un filtre FIR (à réponse impulsionnelle finie), non récursif, d'ordre 1. Ce filtre réalise une moyenne glissante sur deux échantillons, ce qui a pour effet d'atténuer les hautes fréquences.

La fonction de transfert du filtre est :

$$H(z) = \frac{1}{2}(1 + z^{-1})$$

En remplaçant z^{-1} par $e^{-j\omega}$, on obtient la réponse fréquentielle :

$$H(e^{j\omega}) = \frac{1}{2}(1 + e^{-j\omega})$$

Le module de cette réponse fréquentielle est :

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

En traçant le spectre de la réponse impulsionnelle du filtre, on confirme nos résultats :

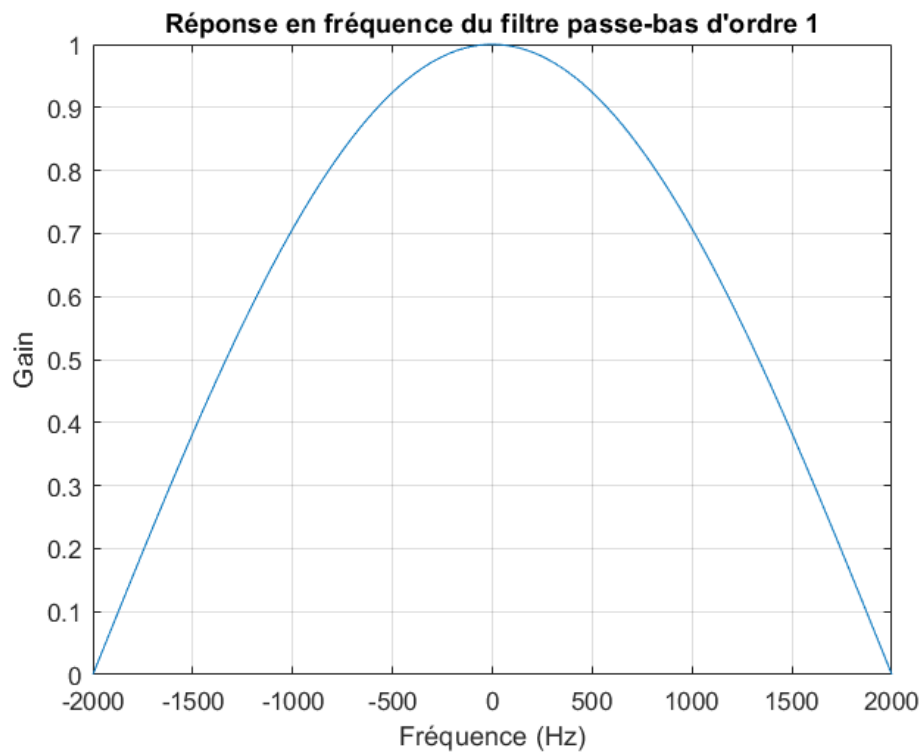


FIGURE 16 – Spectre de la réponse impulsionnelle du filtre passe-bas d'ordre 1

Le filtre laisse passer les basses fréquences ($\omega \approx 0$) avec un gain maximal de 1.

Nous appliquons le filtre passe-bas d'ordre 1 sur les signaux périodiques précédemment définis (carré et dent de scie) :

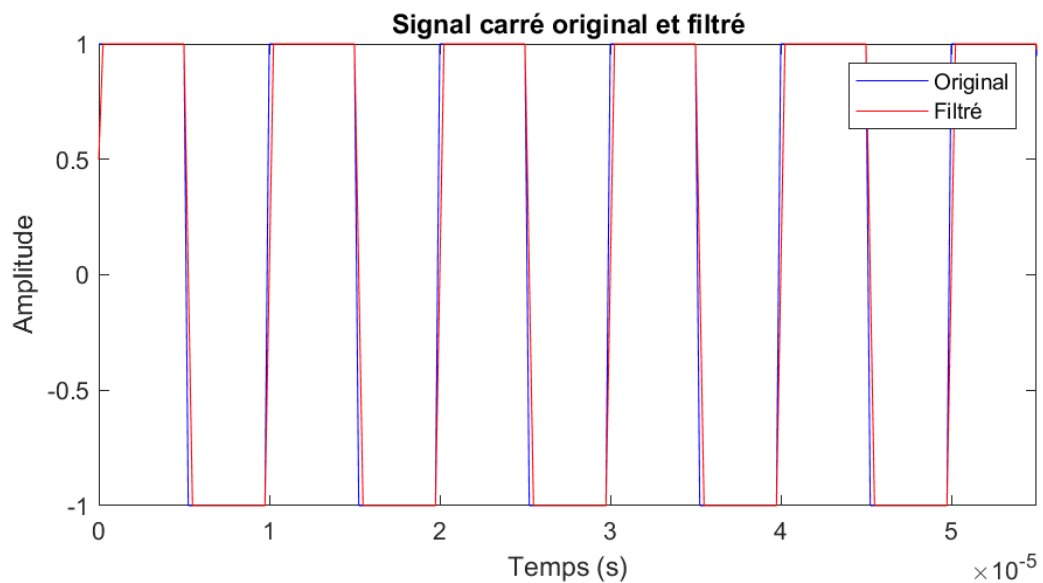


FIGURE 17 – Signal carré filtré par un filtre passe-bas d'ordre 1

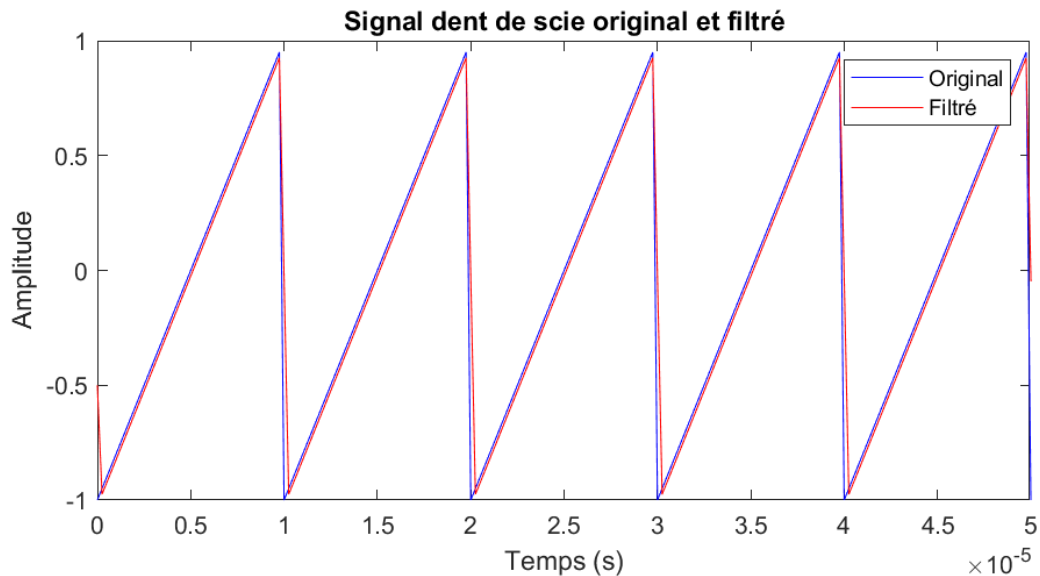


FIGURE 18 – Signal dent de scie filtré par un filtre passe-bas d'ordre 1

Nous avons volontairement choisi des fréquences très élevées (autour de 100 kHz) pour constater l'effet du filtre. Le filtre passe-bas d'ordre 1 atténue les hautes fréquences, cela entraîne un lissage lors des variations rapides du signal. Le signal carré et dent de scie deviennent plus arrondis.

Nous avons défini un filtre passe-bas d'ordre 2 par la suite :

$$d = \text{designfilt}('lowpassfir', 'FilterOrder', 2, 'HalfPowerFrequency', 0.20)$$

Nous obtenons alors les signaux filtrés suivants :

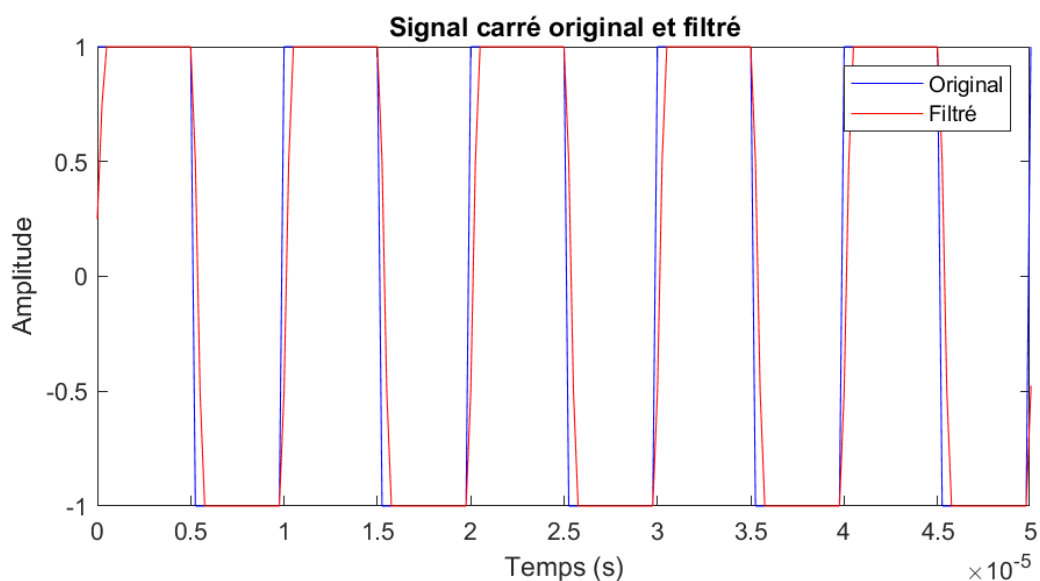


FIGURE 19 – Signal carré filtré par un filtre passe-bas d'ordre 2

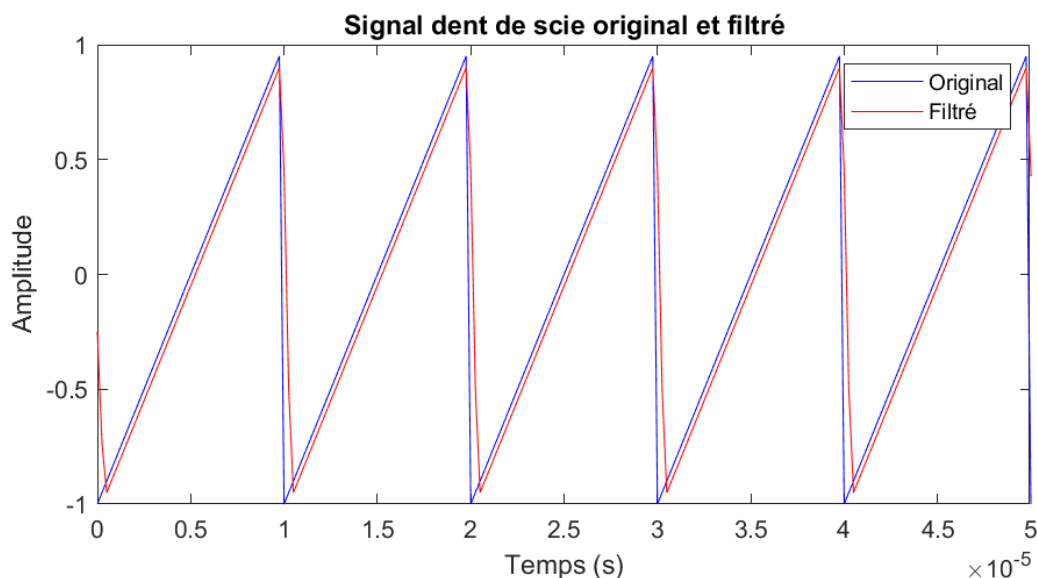


FIGURE 20 – Signal dent de scie filtré par un filtre passe-bas d'ordre 2

Le filtre passe-bas d'ordre 2 a un effet plus prononcé que le filtre d'ordre 1, car il atténue davantage les hautes fréquences. Le signal carré devient encore plus arrondi, et la dent de scie est lissée de manière significative. Nous n'avons pas constaté de différence notable entre les filtres RII et RIF, y compris après écoute des sons produits. Néanmoins, d'après le professeur présent les filtres RIF sont plus adaptés pour conserver la phase du signal mais demande une capacité de calcul plus importante, tandis que les filtres RII sont plus simples à implémenter mais peuvent introduire des déphasages.

4 Effets audio-numériques

Les effets audio-numériques sont présents dans la production sonore moderne, tant pour la création artistique que pour l'amélioration de la qualité d'un signal. Dans cette section, nous explorons l'implémentation de ces effets en modélisant des systèmes linéaires invariants dans le temps, particulièrement adaptés au traitement numérique du signal audio. Parmi les effets étudiés, on retrouvera la réverbération, la convolution et le delay.

4.1 Effets de réverbération

Il s'agit d'effets audio qui simulent la manière dont un son se propage et se réfléchit dans un espace fermé (comme une salle de TP). Numériquement, cet effet est souvent modélisé

comme un système linéaire invariant dans le temps à partir de sa réponse impulsionnelle, et peut être implémenté notamment par convolution.

4.1.1 Mesure de réponse impulsionnelle

Expression de la relation entre intercorrélation, autocorrélation et réponse impulsionnelle

L'expression de R_{yx} en fonction de R_{xx} et h est donnée par :

L'intercorrélation entre y et x , notée $R_{yx}(u)$, peut être exprimée comme la convolution entre h et l'autocorrélation R_{xx} du signal d'excitation x . Cette relation s'écrit :

$$R_{yx}(u) = (h * R_{xx})(u) = \sum_{m=-\infty}^{\infty} h(m)R_{xx}(u - m)$$

4.1.1.1. Estimation de la réponse impulsionnelle :

En supposant que l'autocorrélation du signal d'excitation est proche d'une impulsion de Dirac, c'est-à-dire :

$$R_{xx}(u) \approx \delta(u)$$

D'après la relation précédente :

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

on obtient :

$$R_{yx}(u) \approx h * \delta(u) = h(u)$$

Ainsi, l'intercorrélation $R_{yx}(u)$ constitue une estimation directe de la réponse impulsionnelle $h(u)$.

Nous avons calculé l'autocorrélation des signaux d'excitation $xe1$ et $xe2$:

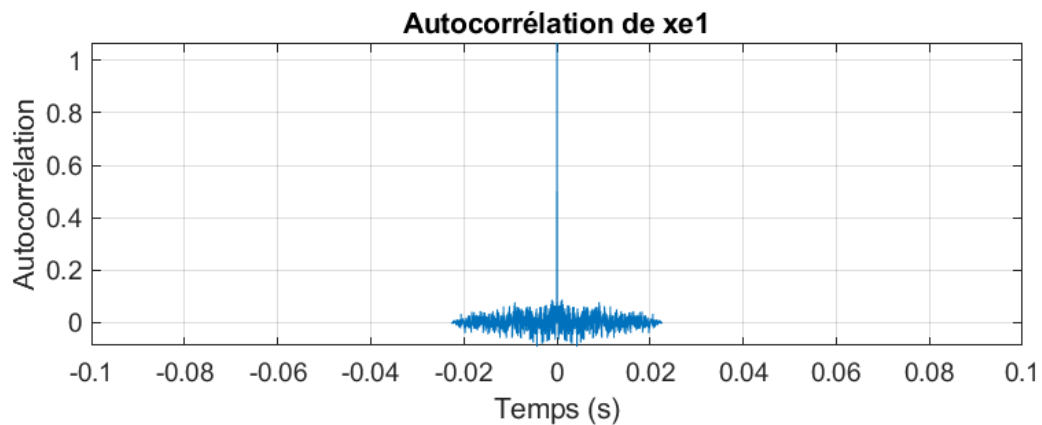


FIGURE 21 – Autocorrélation des signaux d'excitation

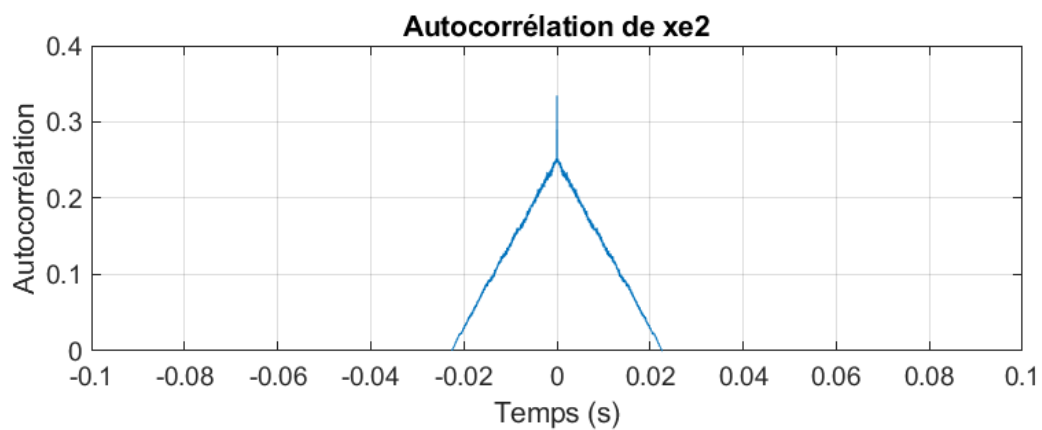


FIGURE 22 – Autocorrélation des signaux d'excitation

Nous pouvons en déduire que le signal $xe1$ correspond plutôt à un bruit blanc, tandis que le signal $xe2$ est plus proche d'une impulsion de Dirac. Le signal $xe2$ est donc plus adapté pour estimer la réponse impulsionnelle du système.

Nous obtenons sans surprise les réponses impulsionnelles estimées suivantes en fonction du temps. Un bruit blanc et un son d'amplitude décroissante pour le signal $xe2$ (comme après avoir tapé des mains dans une église par exemple) :

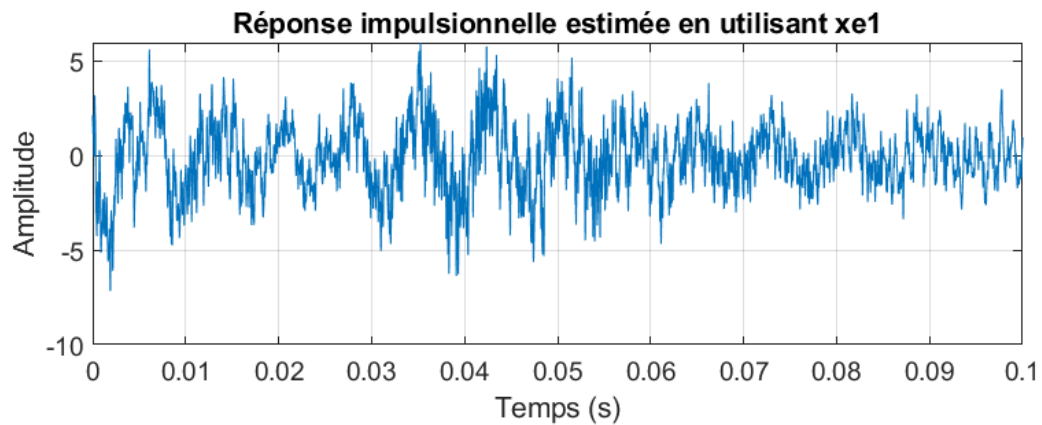


FIGURE 23 – Réponse impulsionnelle estimée pour le signal xe1

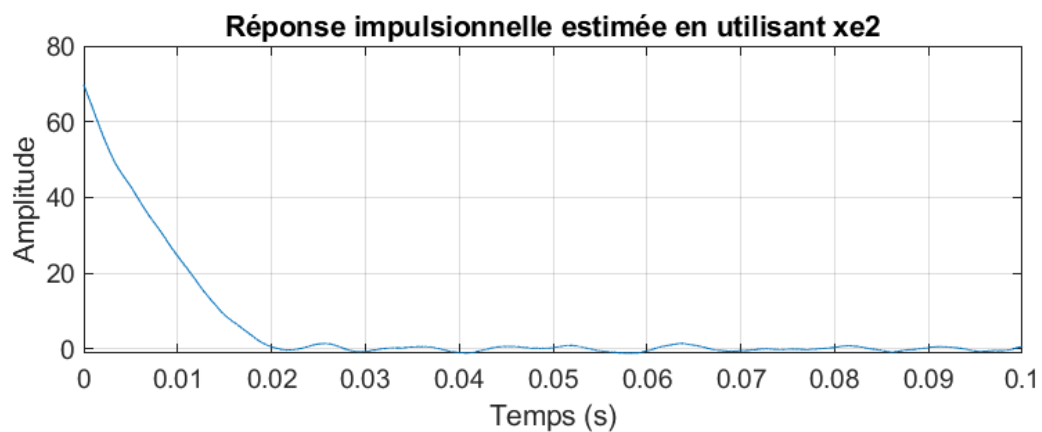


FIGURE 24 – Réponse impulsionnelle estimée pour le signal xe2

4.1.2 Convolution classique

Nous avons implémenté la convolution classique en utilisant la fonction `filter` de Matlab comme demandé dans le sujet. La réponse impulsionnelle h est convoluée avec le signal d'excitation $xe2$ pour obtenir le signal de sortie y et notre programme `test_effet_reverb.m` est capable de simuler la réverbération d'un signal audio de guitare en utilisant la réponse impulsionnelle estimée :

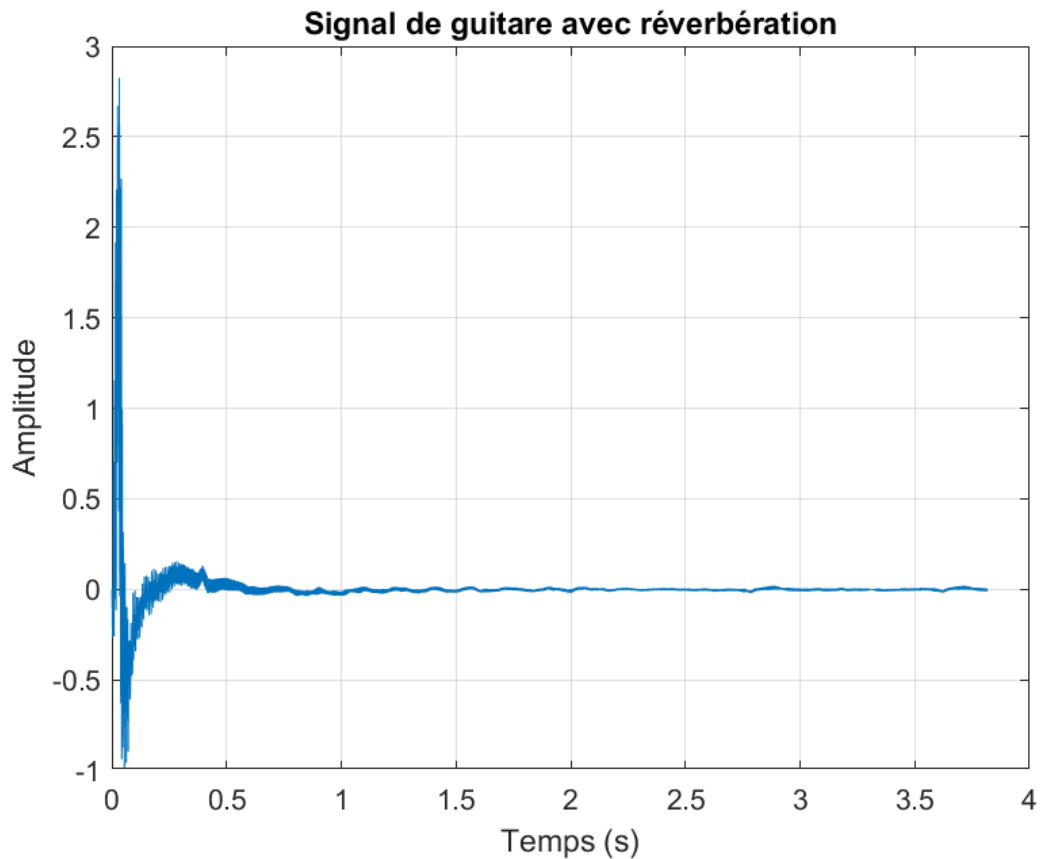


FIGURE 25 – Signal de sortie après convolution avec la réponse impulsionnelle

Le temps de calcul nécessaire pour la complétion de la fonction *effet_reverb.m* est 0.047532 s après calcul avec la fonction `tic` et `toc` de Matlab.

4.1.3 Convolution rapide

Nous réalisons la convolution rapide en convoluant la fonction dans le domaine fréquentiel ce qui utilise l'algorithme FFT pour accélérer le processus de convolution. Le signal de guitare avec réverbération obtenu est similaire à celui obtenu précédemment.

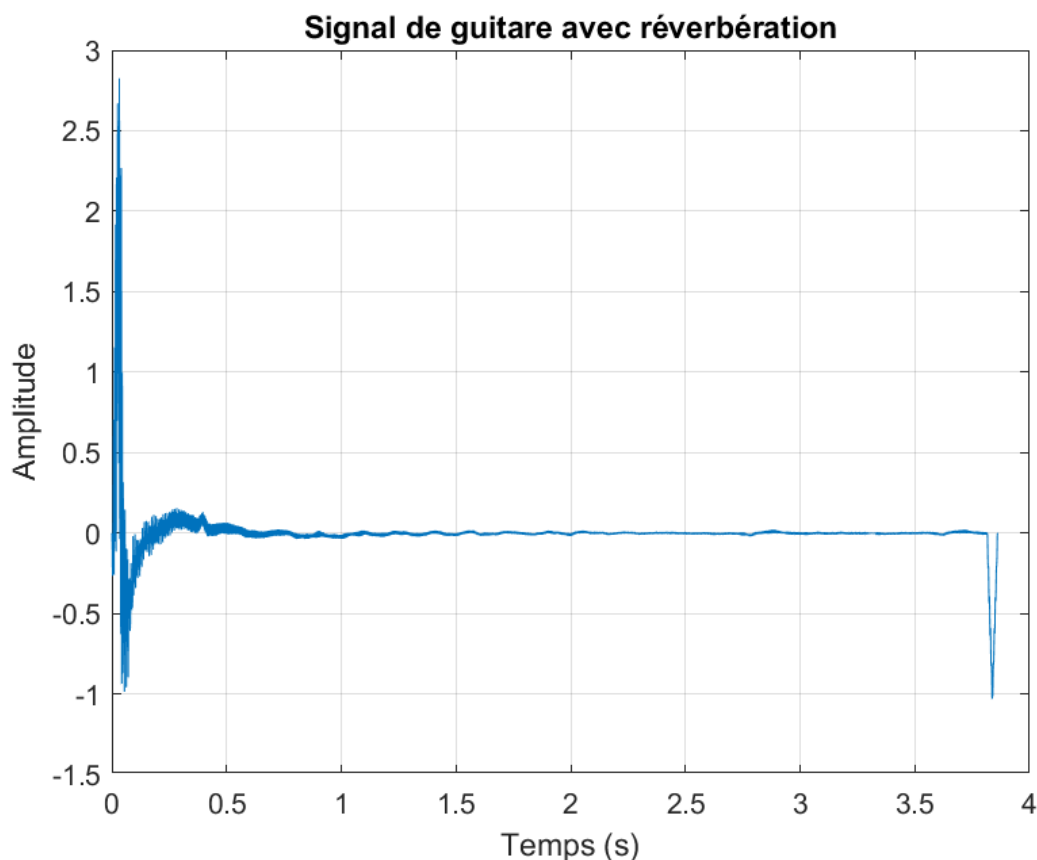


FIGURE 26 – Signal de sortie après convolution rapide avec la réponse impulsionnelle

Cette fois-ci, le temps de calcul nécessaire est de 0.015877 s, ce qui est plus rapide que la convolution classique. Par curiosité, nous avons aussi utilisé la fonction `conv` de Matlab pour réaliser la convolution rapide et le temps de calcul est alors du même ordre que celui de la convolution rapide. La méthode utilisant la FFT est mathématiquement équivalente à la convolution classique, néanmoins la documentation Matlab indique qu'il faut être vigilant au nombre de points sur lesquels est calculée la FFT. S'il est trop faible il y a un sous-échantillonnage du signal et donc un repliement de spectre. Il est donc important de s'assurer que le nombre de points de la FFT est suffisamment grand pour éviter ce problème.

4.2 Effet de retard

4.2.1 Réponse impulsionnelle théorique de l'effet de delay

Le système de delay est modélisé par l'équation de récurrence :

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

où g est un coefficient d'atténuation et τ est le nombre d'échantillons de retard.

Pour obtenir la réponse impulsionnelle $h(k)$, on applique une impulsion de Dirac à l'entrée, soit $x(k) = \delta(k)$. Le système devient :

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

$$\text{Pour } k = 0 : \quad y(0) = 1$$

$$\text{Pour } k = \tau : \quad y(\tau) = -g \cdot y(0) = -g$$

$$\text{Pour } k = 2\tau : \quad y(2\tau) = -g \cdot y(\tau) = -g \cdot (-g) = g^2$$

$$\text{Pour } k = 3\tau : \quad y(3\tau) = -g \cdot y(2\tau) = -g \cdot g^2 = -g^3$$

\vdots

En résolvant cette équation par récurrence, on trouve que $h(k)$ est nul partout sauf pour les instants multiples de τ , où il vaut $(-g)^n$.

4.2.2 Condition de stabilité du filtre de delay

La réponse impulsionnelle du filtre est :

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

Pour que ce filtre soit stable, la réponse impulsionnelle doit être absolument sommable :

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

Il s'agit d'une série géométrique convergente si et seulement si $|g| < 1$.

4.2.3 Déterminer les vecteurs \mathbf{a} et \mathbf{b}

L'équation du filtre de delay est donnée par :

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

En la réécrivant sous la forme standard :

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

on peut identifier les coefficients utilisés par la fonction `filter(b, a, x)` de Matlab.

Le vecteur a (dénominateur) contient les coefficients des sorties $y(k)$:

$$a = [1, 0, \dots, 0, g] \quad (\text{avec } \tau - 1 \text{ zéros})$$

Le vecteur b (numérateur) contient les coefficients des entrées $x(k)$:

$$b = [1]$$

4.2.4 Analyse de l'effet de delay

Nous avons implémenté l'effet de delay en utilisant la fonction `filter` de Matlab avec les coefficients a et b définis précédemment. Nous avons utilisé les paramètres suivants pour notre simulation :

- $f_e = 44100$ Hz - Fréquence d'échantillonnage
- $\tau = 500$ - Nombre d'échantillons de retard
- $g = 0.7$ - Coefficient d'atténuation

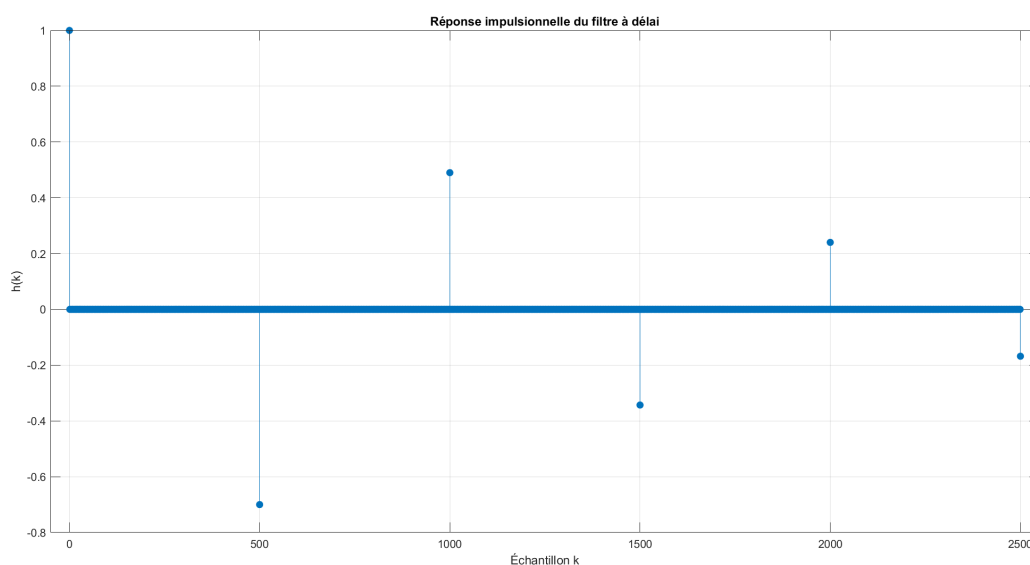


FIGURE 27 – Analyse de l'effet de delay

4.2.5 Réponse en fréquence du filtre de retard

Le filtre de delay est modélisé par l'équation de récurrence suivante :

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

Pour comprendre le comportement fréquentiel, on applique la transformée en z :

$$Y(z) = X(z) - gz^{-\tau}Y(z) \Rightarrow Y(z)(1 + gz^{-\tau}) = X(z) \Rightarrow H(z) = \frac{Y(z)}{X(z)} = \frac{1}{1 + gz^{-\tau}}$$

En remplaçant $z = e^{2j\pi\nu}$, on obtient la réponse :

$$\hat{h}(\nu) = \frac{1}{1 + ge^{-2j\pi\nu\tau}}$$

4.2.5.1 Module de la réponse :

$$1 + ge^{-2j\pi\nu\tau} = 1 + g \cos(2\pi\nu\tau) - jg \sin(2\pi\nu\tau)$$

Le module est alors :

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

4.2.5.2 Phase de la réponse :

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

Ce filtre se comporte comme un filtre en peigne, avec une réponse qui se répète régulièrement.

4.2.6 Comparaison : théorique vs numérique

Nous avons comparé la réponse en fréquence théorique du filtre de delay avec la réponse en fréquence obtenue numériquement à partir de la fonction `fft` de Matlab :

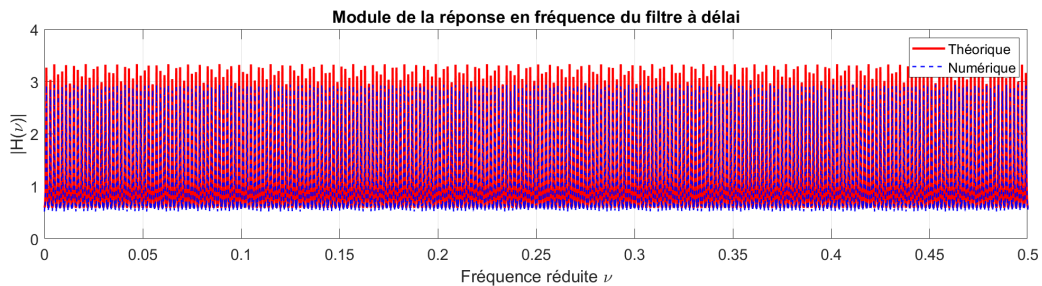


FIGURE 28 – Comparaison du module de la réponse en fréquence théorique et numérique du filtre de delay

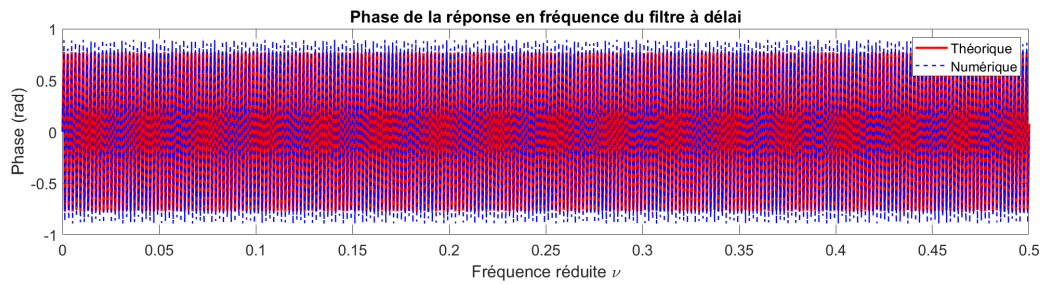


FIGURE 29 – Comparaison de la phase de la réponse en fréquence théorique et numérique du filtre de delay

Nous constatons que les deux réponses en fréquence sont très similaires, ce qui était attendu. En augmentant le nombre d'échantillons de la FFT, les différences deviennent négligeables.

4.2.7 Test effet delay

Nous obtenons la figure suivante en appliquant l'effet de delay sur un signal de piano :

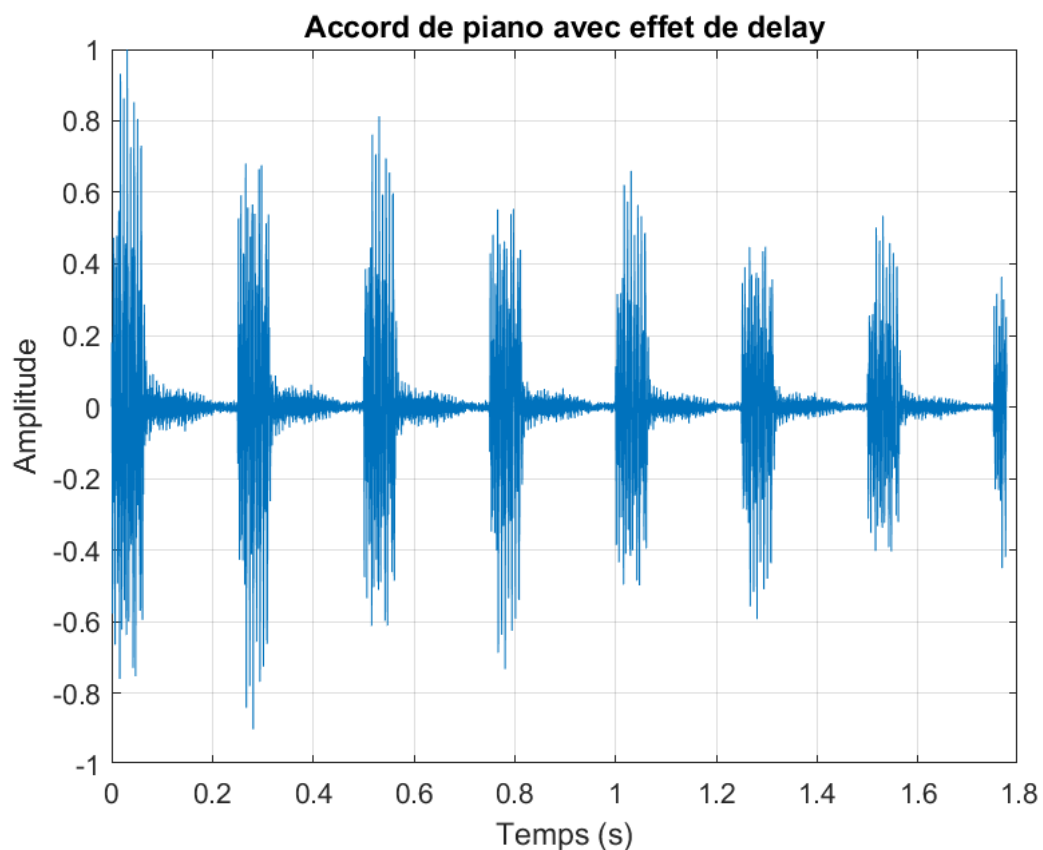


FIGURE 30 – Signal de piano avec effet de delay

4.2.8 Delay filtré

Nous avons implémenté un effet delay filtré avec un moyenne glissante sur K échantillons. Nous obtenons la figure suivante après application sur un son de piano :

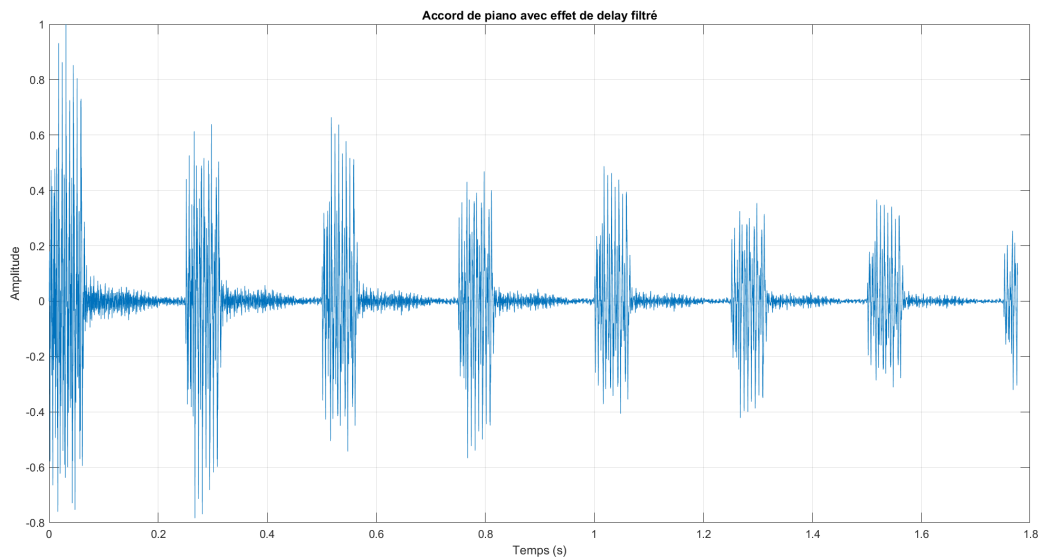


FIGURE 31 – Signal de piano avec effet de delay filtré

4.2.9 Réponse en fréquence de l'effet de retard filtré

Nous avons représenté la réponse fréquentielle et il y a en effet une "coupure" progressive des hautes fréquences :

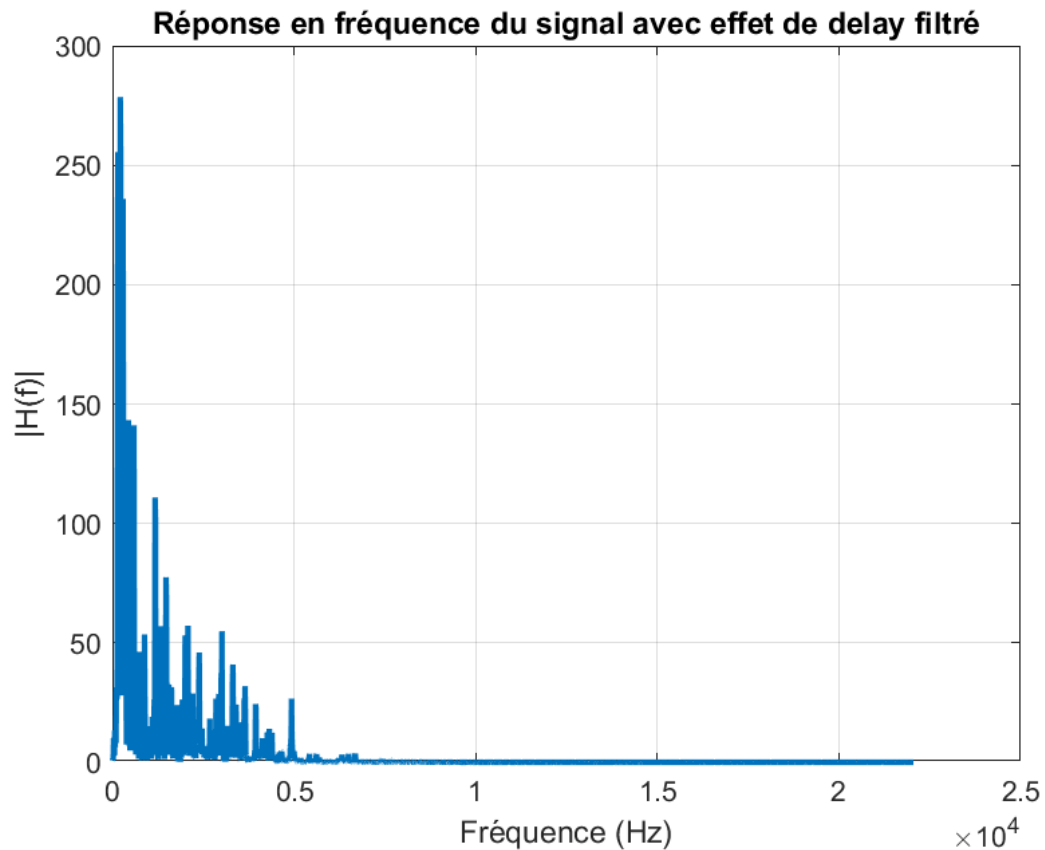


FIGURE 32 – Signal de piano avec effet de delay filtré

4.3 Synthèse Karplus-Strong

Nous avons implémenté l'algorithme de Karplus-Strong pour synthétiser un son de guitare. Les paramètres que nous avons utilisés sont les suivants :

- $f_e = 44100$ Hz - Fréquence d'échantillonnage
- $f_0 = 110$ Hz - Fréquence fondamentale souhaitée pour la génération du bruit blanc
- $N = \text{round}(f_e / f_0)$ - Retard
- $\text{duree} = 2$ s - Durée du son
- $g = 0.99$ - Coefficient d'amortissement
- $\alpha = 0.5$ - Coefficient du filtre de boucle

Les tests ont été réalisés en générant un bruit blanc de 2 secondes mais aussi sur la guitare utilisée précédemment. Nous obtenons les 2 spectres temporels suivants et nous avons écouté les sons qui sont très harmonieux à l'oreille :

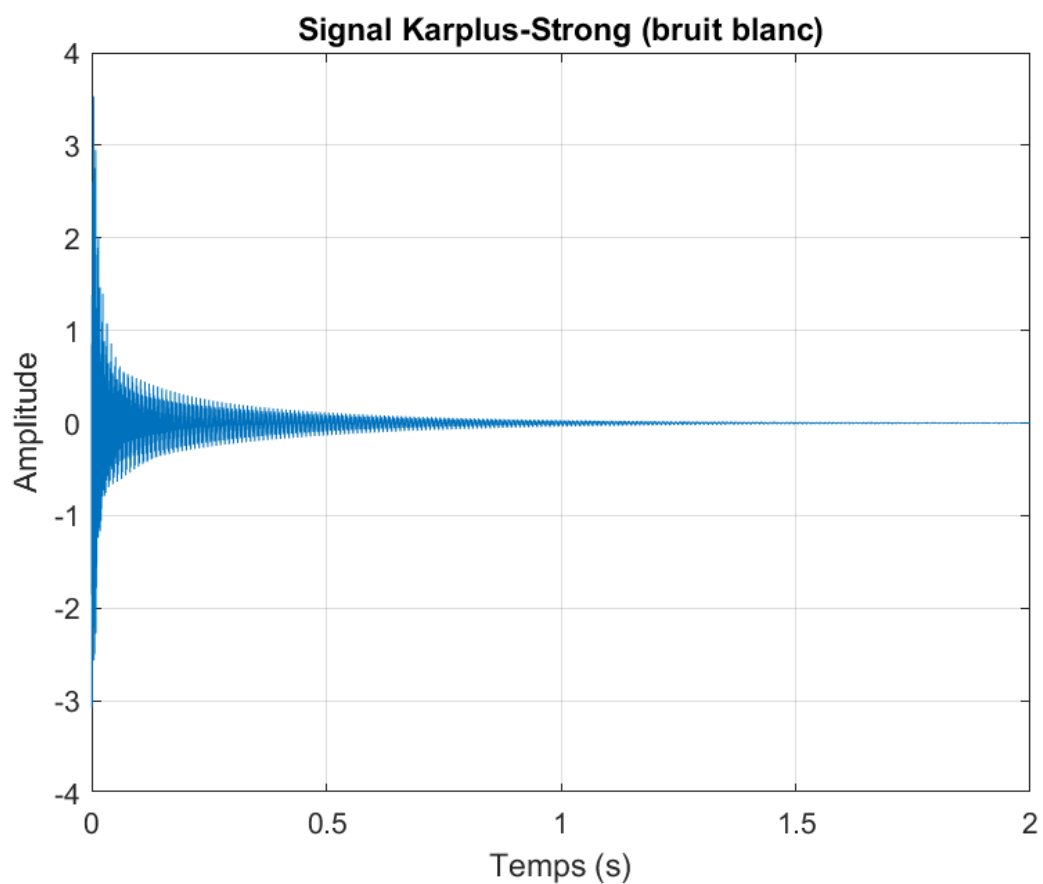


FIGURE 33 – Spectre temporel du son synthétisé à partir d'un bruit blanc

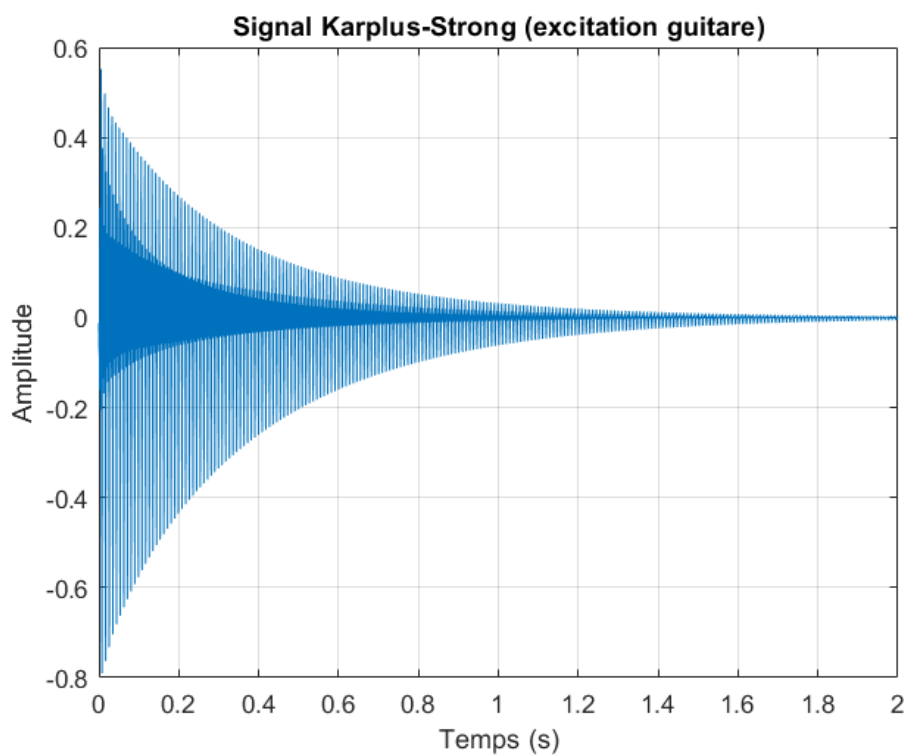


FIGURE 34 – Spectre temporel du son synthétisé à partir de la guitare

5 Conclusion

Ce projet nous a permis d'explorer en profondeur les techniques de traitement audio numérique, en mettant l'accent sur la synthèse additive et soustractive, ainsi que sur les effets audio-numériques. Nous avons appris à analyser les caractéristiques fréquentielles des signaux audio, à implémenter des synthèses sonores réalistes et à modéliser des effets tels que la réverbération et le delay. L'objectif de ce projet était de nous faire travailler sur les étapes préliminaires à la création d'un réel synthétiseur audio et de filtres réels avec Matlab en vue d'une utilisation par la suite sur carte électronique après simulation avec Vivado. Cela a d'ailleurs été réalisé dans la suite du projet (voir Gitlab-df).