



# SAR Traitements audio

**Étudiantes :** Niamatellah LAHKIM, Houda DAOURI

**Référent :** Frédéric GUILLOUD

**Date :** 28 mai 2025

# Table des matières

<b>1 Synthèse additive</b>	<b>4</b>
1.1 Analyse d'un son harmonique . . . . .	4
1.2 Synthèse . . . . .	7
<b>2 Synthèse soustractive</b>	<b>9</b>
<b>3 Effets audio-numériques</b>	<b>12</b>
3.1 Expression de la corrélation . . . . .	12
3.2 Méthode d'estimation de la réponse impulsionnelle . . . . .	13
3.3 Sélection du signal d'excitation . . . . .	13
3.4 Estimation expérimentale de la réponse impulsionnelle . . . . .	14
3.5 Application de la réverbération par convolution classique . . . . .	15
3.6 Application de la réverbération par convolution directe . . . . .	15
3.7 Application de la réverbération par convolution rapide . . . . .	17
3.8 Comparaison des méthodes de convolution . . . . .	18
3.9 Démonstration par récurrence . . . . .	19
3.10 Condition de stabilité du filtre . . . . .	20
3.11 Détermination des vecteurs <b>a</b> et <b>b</b> . . . . .	21
3.12 Obtention de la réponse impulsionnelle via <b>filter</b> . . . . .	21
3.13 Réponse fréquentielle théorique du filtre à délai . . . . .	22
3.14 Comparaison entre la réponse fréquentielle théorique et celle obtenue par FFT . . . . .	23
3.15 Implémentation de la fonction <b>effet_delay</b> . . . . .	24
3.16 Test de la fonction avec paramètres imposés . . . . .	25
3.17 Implémentation de la fonction <b>effet_delay_filtre</b> A REVOIR . . . . .	27
3.18 Test de la fonction <b>effet_delay_filtre</b> . . . . .	27
3.19 Test de la fonction <b>effet_delay_filtre</b> . . . . .	29

## Table des figures

1	Spectre centré - Trompette ( <code>single_tone_trumpet_a3</code> ) . . . . .	5
2	Spectre centré - Guitare ( <code>single_tone_guitar_nylon_string_a2</code> ) . . . . .	5
3	Spectre centré - Piano ( <code>single_tone_piano1</code> ) . . . . .	6
4	Amplitudes normalisées des 8 premières harmoniques mesurées dans le signal <code>piano1.wav</code> . . . . .	8
5	Enveloppe ADSR appliquée au signal synthétique . . . . .	9
6	Autocorrélations normalisées de <code>xe1</code> et <code>xe2</code> , affichées entre $-1000$ et $+1000$ échantillons . . . . .	14
7	Réponse impulsionnelle estimée de la pièce . . . . .	15
8	Comparaison des signaux original et réverbéré d'un son d'une guitare, après convolution avec la réponse impulsionnelle $h(k)$ . . . . .	16
9	Comparaison du signal original et du signal réverbéré par convolution rapide (TFD). . . . .	18
10	Réponse impulsionnelle $h(k)$ obtenue numériquement. . . . .	22
11	Comparaison entre le module de la réponse fréquentielle obtenu par FFT et celui calculé théoriquement. . . . .	24
12	Signal original : <code>single_tone_piano1.wav</code> . . . . .	26
13	Signal après application de l'effet delay . . . . .	26
14	Signal original : <code>single_tone_piano1.wav</code> . . . . .	28
15	Signal traité avec effet delay via filtre ( $g = 0,9$ , $\tau = 0,25 \cdot F_e$ ). . . . .	29

# Introduction

Le traitement numérique du signal audio est une discipline clé à l'intersection de l'ingénierie, de l'informatique et de la musique. Il regroupe un ensemble de techniques permettant d'analyser, de modifier et de générer des sons à l'aide d'outils numériques. Ces méthodes sont largement utilisées dans des domaines variés : amélioration de la qualité sonore, production musicale assistée par ordinateur, transmission vocale, ou encore conception d'effets sonores pour le cinéma.

Ce projet a pour objectif de mettre en pratique les concepts fondamentaux du traitement du signal audio, à travers des applications concrètes et des cas d'usage réels. Il se structure en quatre axes majeurs, chacun représentant un pilier essentiel de cette discipline :

- **L'échantillonnage** : première étape de la numérisation d'un signal acoustique, où un signal continu est converti en signal discret. La qualité de cette conversion détermine la précision des traitements ultérieurs.
- **L'analyse fréquentielle** : décomposition du signal dans le domaine spectral afin d'en étudier les composantes harmoniques et le timbre.
- **La synthèse sonore** : génération de sons via des modèles mathématiques, incluant ici deux approches complémentaires : la synthèse *additive* (superposition d'harmoniques) et la synthèse *soustractive* (filtrage d'un signal complexe).
- **Les effets audio-numériques** : transformation du signal à des fins esthétiques ou techniques, avec un focus particulier sur la réverbération et l'effet de délai (*delay*).

Ce rapport détaille les démarches méthodologiques adoptées pour chaque partie, les fondements théoriques mobilisés, les protocoles expérimentaux mis en œuvre avec **Matlab**, ainsi que l'analyse critique des résultats obtenus.

## 1 Synthèse additive

### 1.1 Analyse d'un son harmonique

#### Question 1.1 :

Nous avons visualisé le spectre d'amplitude des signaux audio à l'aide de la transformée de Fourier discrète (FFT) et de la fonction `fftshift` pour centrer le spectre. Les signaux proviennent de trois instruments différents : trompette, guitare et piano.

La fréquence fondamentale  $f_1$  a été déterminée en repérant la première fréquence positive correspondant à un maximum d'amplitude dans le spectre. Cette fréquence fondamentale est :

- **Trompette** :  $\approx 450$  Hz
- **Guitare** :  $\approx 220$  Hz
- **Piano** :  $\approx 220$  Hz

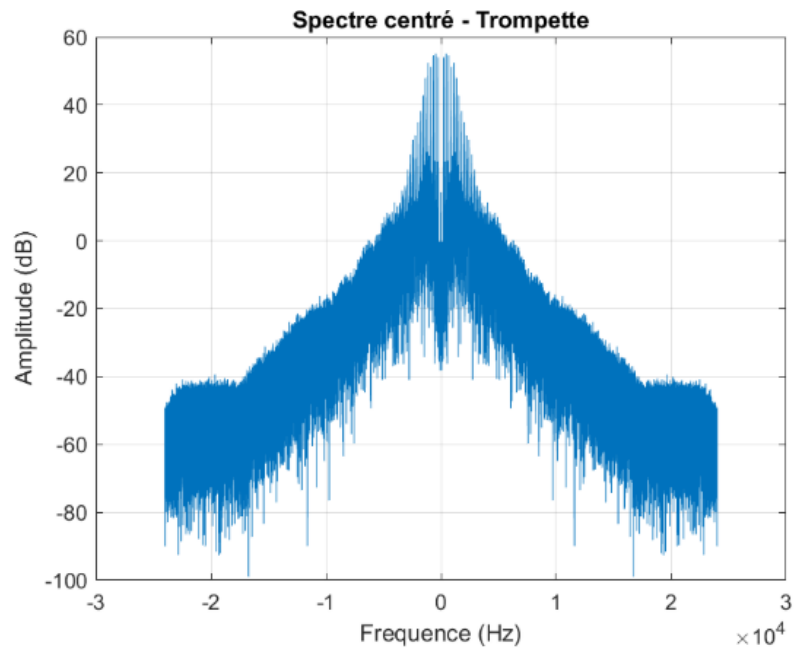


FIGURE 1 – Spectre centré - Trompette (single\_tone\_trumpet\_a3)

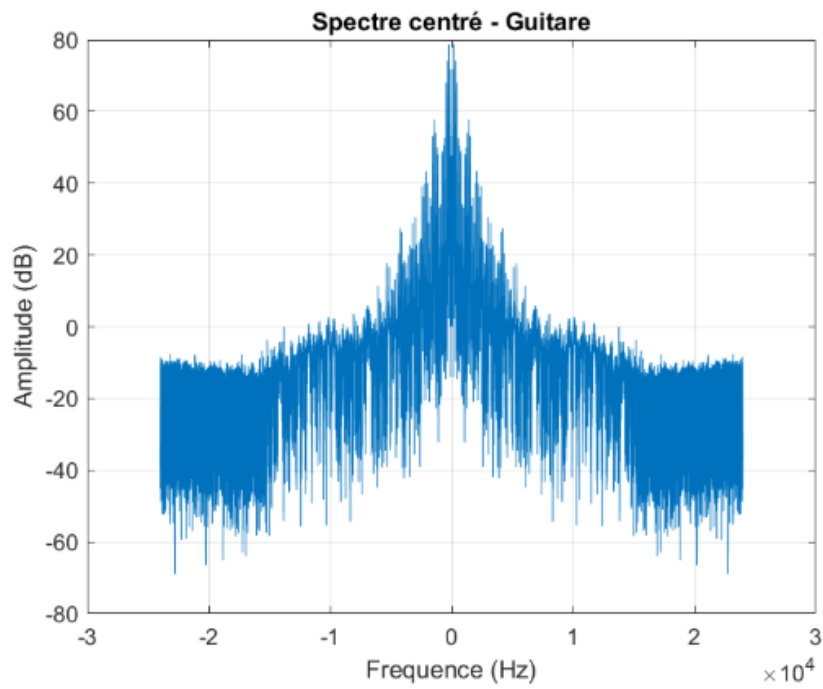


FIGURE 2 – Spectre centré - Guitare (single\_tone\_guitar\_nylon\_string\_a2)

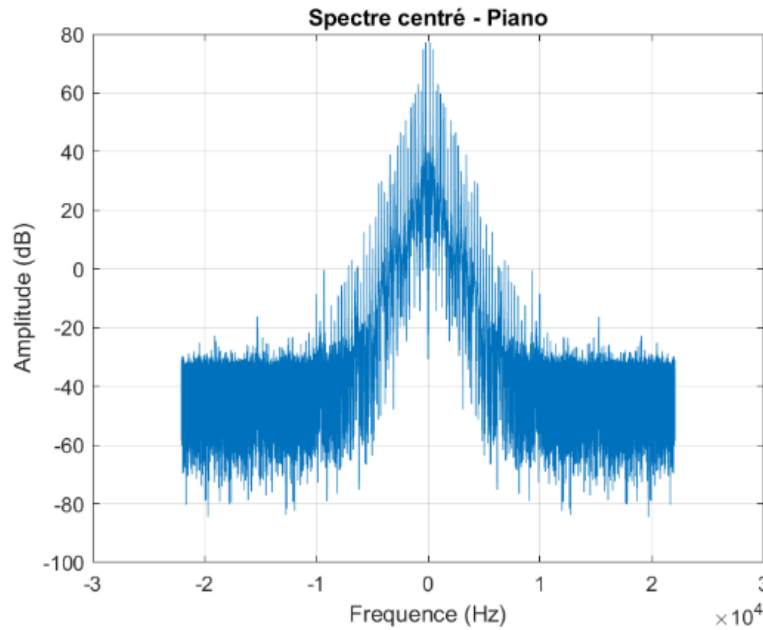


FIGURE 3 – Spectre centré - Piano (single\_tone\_piano1)

On observe que les spectres sont symétriques autour de zéro. Le contenu harmonique est visible à travers les pics régulièrement espacés. La trompette présente une fréquence fondamentale plus élevée, ce qui est cohérent avec la hauteur perçue plus aiguë du son.

### Question 1.2 :

Dans cette question, nous devons comparer deux sons de piano (`piano1.wav` et `piano2.wav`) afin de déterminer lequel est le plus harmonique. Un son est considéré comme harmonique lorsque ses composantes fréquentielles se trouvent à des multiples entiers de la fréquence fondamentale  $f_1$ . On évalue l'écart à cette structure idéale à l'aide de l'inharmonicité, mesurée en cents par la formule :

$$\xi_n = 1200 \cdot (\log_2(f_n) - \log_2(n \cdot f_1))$$

où :

- $f_n$  est la fréquence mesurée du  $n$ -ième pic harmonique,
- $n \cdot f_1$  est sa valeur théorique.

Pour le piano 1, la fréquence fondamentale est de  $f_1 = 220,37$  Hz).

Pour le piano 2, la fréquence fondamentale est de  $f_1 = 442,78$  Hz)

Les deux pianos présentent une structure globalement harmonique, mais le piano 1 se distingue par une progression régulière de l'inharmonicité (de 0 à 12 cents), ce qui est le signe d'une structure cohérente et stable. À l'inverse, le piano 2 présente une inharmonicité plus irrégulière, avec un pic de +29,01 cents pour l'harmonique 2, suivi de variations incohérentes, dont un passage négatif à l'harmonique 6. Cela reflète un comportement acoustique moins stable.

Nous en concluons que le piano 1 est plus harmonique que le piano 2, car ses fréquences mesurées sont plus proches des valeurs théoriques attendues. Son inharmonicité évolue de façon régulière, tandis que celle du piano 2 est plus irrégulière.

<b>Harm.</b>	$f_{th}$ (Hz)	$f_{mes}$ (Hz)	Inharm. (cents)
1	220.37	220.37	0.00
2	440.75	442.04	5.07
3	661.12	663.22	5.50
4	881.50	885.54	7.92
5	1101.87	1108.34	10.13
6	1322.24	1331.46	12.02
7	1542.62	1552.48	11.03

TABLE 1 – Tableau des fréquences pour le piano 1

<b>Harm.</b>	$f_{th}$ (Hz)	$f_{mes}$ (Hz)	Inharm. (cents)
1	442.78	442.78	0.00
2	885.57	892.49	29.01
3	1328.35	1335.71	26.19
4	1771.13	1781.09	19.68
5	2213.92	2221.71	12.35
6	2656.70	2654.97	-13.24
7	3099.48	3103.81	-5.42

TABLE 2 – Tableau des fréquences pour le piano 2

## 1.2 Synthèse

### Question 1.3 :

Dans cette question, nous nous sommes appuyés sur le signal `piano1.wav`, précédemment identifié comme le plus harmonique. L'objectif était de reconstruire un son synthétique à partir de ses composantes spectrales les plus significatives, à savoir les huit premières harmoniques.

Pour cela, nous avons d'abord réalisé une transformée de Fourier sur le signal afin d'en extraire les fréquences et amplitudes des huit premiers pics situés autour des multiples de la fréquence fondamentale. Ces valeurs ont été déterminées à l'aide d'une recherche locale autour de chaque fréquence théorique, ce qui permet de prendre en compte l'inharmonicité naturelle du piano.

Les amplitudes obtenues ont ensuite été normalisées par rapport à la plus grande, de manière à conserver leurs proportions relatives sans dépasser une amplitude unitaire. La synthèse du signal a été réalisée par superposition de huit sinusoïdes correspondant aux fréquences mesurées, chacune pondérée par son amplitude.

Nous avons choisi une durée du signal synthétique à 6 secondes, ce qui est suffisant pour percevoir le timbre du son tout en restant cohérent avec la durée approximative du fichier d'origine.

Enfin, nous avons produit une représentation graphique des amplitudes normalisées en fonction des fréquences mesurées. Cette courbe permet de visualiser la décroissance naturelle des harmoniques en intensité. Le son, restitué déjà une partie du timbre du piano, notamment la richesse harmonique de son spectre stable.

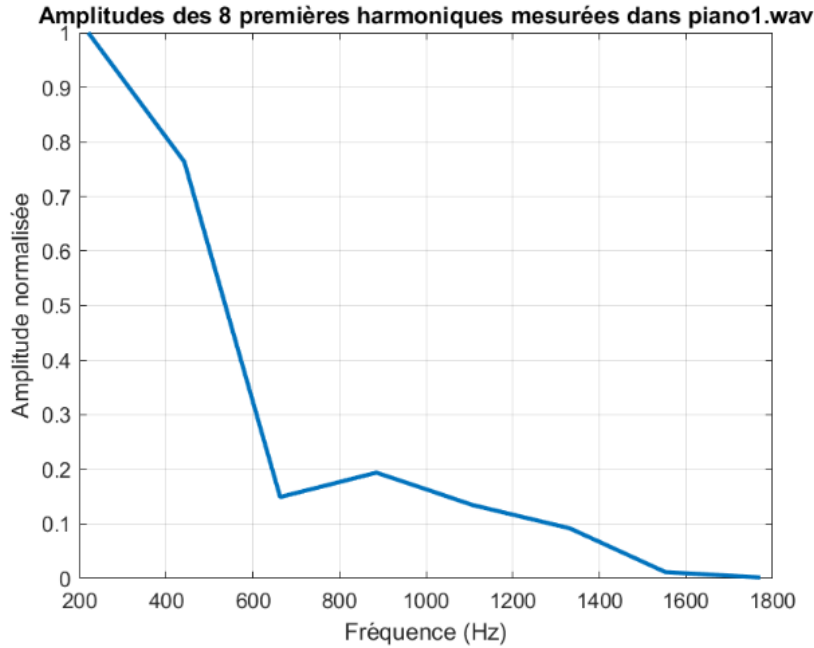


FIGURE 4 – Amplitudes normalisées des 8 premières harmoniques mesurées dans le signal piano1.wav.

#### Question 1.4 :

Afin de rendre le son synthétique obtenu à la question précédente plus réaliste et plus proche du timbre d'un piano réel, une enveloppe de type ADSR (Attack, Decay, Sustain, Release) a été appliquée au signal.

L'enveloppe utilisée est définie par les paramètres suivants, exprimés en pourcentage de la durée totale du signal, 6 secondes :

- **Attack** : 10% — montée linéaire de l'amplitude de 0 à 1,
- **Decay** : 10% — diminution linéaire de l'amplitude de 1 à 0,6,
- **Sustain** : 70% — maintien constant de l'amplitude à 0,6,
- **Release** : 10% — descente linéaire de 0,6 à 0.

Cette modulation temporelle de l'amplitude permet de simuler le comportement dynamique d'un piano acoustique, où le son ne reste pas constant mais évolue naturellement après l'impact initial.

À l'écoute, le son obtenu paraît plus naturel. Il commence progressivement, atteint un pic d'intensité, puis décroît de manière fluide avant de s'éteindre. Comparé au signal synthétique constant de la question précédente, ce rendu améliore nettement la perception du timbre et renforce l'illusion d'un instrument réel.

Cette modulation temporelle de l'amplitude permet de simuler le comportement dynamique d'un piano acoustique, où le son ne reste pas constant mais évolue naturellement après l'impact initial.

À l'écoute, le son obtenu est plus proche du son réel. Le son commence de manière progressive, atteint un maximum, puis décroît graduellement jusqu'à disparaître. Contrairement au signal synthétique constant de la question précédente, ce rendu améliore nettement la perception du timbre.



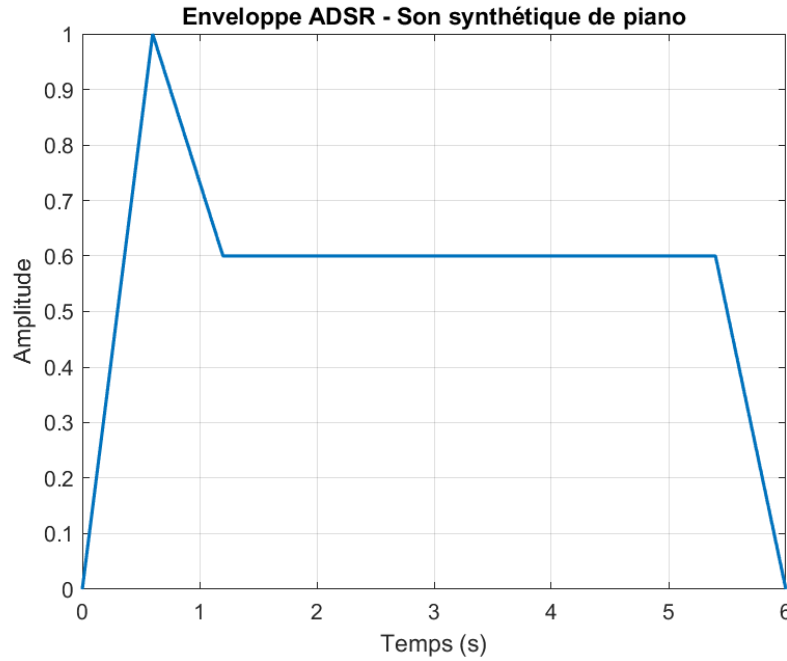


FIGURE 5 – Enveloppe ADSR appliquée au signal synthétique

## 2 Synthèse soustractive

Dans cette question, on cherche à déterminer le spectre d'un signal **carré centré** et d'un signal **dent de scie centré**, tous deux de période  $T$  et d'amplitude  $\pm 1$ . Nous procédons à une analyse en série de Fourier, puis vérifions numériquement nos résultats via la transformée de Fourier discrète (TFD).

### Question 2.1 :

**Définition :** Le signal carré est défini par :

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

C'est un signal périodique impair.

**Série de Fourier :** Comme  $x(t)$  est impair, seuls les termes en sinus apparaissent :

$$x(t) = \sum_{n=1}^{\infty} b_n \sin(2\pi n f_0 t), \quad \text{où } f_0 = \frac{1}{T}$$

**Calcul des coefficients :**

$$b_n = \frac{2}{T} \int_{-T/2}^{T/2} x(t) \sin(2\pi n f_0 t) dt = \frac{4}{T} \int_0^{T/2} \sin(2\pi n f_0 t) dt$$

$$b_n = \frac{4}{T} \cdot \left[ -\frac{\cos(2\pi n f_0 t)}{2\pi n f_0} \right]_0^{T/2} = \frac{2(1 - (-1)^n)}{\pi n}$$

Donc :

$$b_n = \begin{cases} \frac{4}{\pi n} & \text{si } n \text{ impair} \\ 0 & \text{si } n \text{ pair} \end{cases}$$

**Spectre :** Seules les harmoniques impaires sont présentes. Le signal s'écrit :

$$x(t) = \sum_{k=0}^{\infty} \frac{4}{\pi(2k+1)} \sin(2\pi(2k+1)f_0 t)$$

Fréquences :  $f_0, 3f_0, 5f_0, \dots$  Amplitudes :  $\frac{4}{\pi}, \frac{4}{3\pi}, \frac{4}{5\pi}, \dots$

**Définition :** Signal linéaire de  $-1$  à  $+1$  sur une période :

$$x(t) = 2 \left( \frac{t}{T} - \text{floor} \left( \frac{t}{T} + \frac{1}{2} \right) \right)$$

C'est un signal impair périodique.

**Série de Fourier :**

$$x(t) = \sum_{n=1}^{\infty} b_n \sin(2\pi n f_0 t)$$

**Calcul des coefficients :**

$$\begin{aligned} b_n &= \frac{2}{T} \int_{-T/2}^{T/2} x(t) \sin(2\pi n f_0 t) dt = \frac{4}{T} \int_0^{T/2} \left( \frac{2t}{T} - 1 \right) \sin(2\pi n f_0 t) dt \\ &= \frac{4}{T} \left[ \frac{2}{T} \int_0^{T/2} t \sin(\alpha t) dt - \int_0^{T/2} \sin(\alpha t) dt \right], \quad \alpha = 2\pi n f_0 \\ \int t \sin(\alpha t) dt &= -\frac{t \cos(\alpha t)}{\alpha} + \frac{\sin(\alpha t)}{\alpha^2} \\ b_n &= \frac{2(-1)^{n+1}}{\pi n} \end{aligned}$$

**Spectre :** Toutes les harmoniques sont présentes. Le signal s'écrit :

$$x(t) = \sum_{n=1}^{\infty} \frac{2(-1)^{n+1}}{\pi n} \sin(2\pi n f_0 t)$$

Fréquences :  $f_0, 2f_0, 3f_0, \dots$  Amplitudes :  $\frac{2}{\pi}, \frac{1}{\pi}, \frac{2}{3\pi}, \dots$

**Conclusion :** Le spectre du signal carré contient uniquement des harmoniques impaires, tandis que celui du signal dent de scie contient toutes les harmoniques. Ces propriétés sont fondamentales pour la synthèse sonore par filtrage (synthèse soustractive).

### Question 2.2 :

**Définition du filtre passe-bas d'ordre 1** Le filtre est défini par l'équation aux différences :

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

avec :

- $x(k)$  : signal d'entrée à l'instant  $k$ ,
- $y(k)$  : signal de sortie,
- $\frac{1}{2}$  : facteur de normalisation pour éviter l'amplification.

**Réponse impulsionnelle** La réponse impulsionnelle  $h(k)$  est la sortie du filtre pour une impulsion de Kronecker  $\delta(k)$ . On a :

$$h(k) = \begin{cases} \frac{1}{2} & \text{si } k = 0 \text{ ou } k = 1 \\ 0 & \text{sinon} \end{cases}$$

Forme compacte :

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

**Fonction de transfert en fréquence** La transformée de Fourier discrète (TFD) de  $h(k)$  donne :

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

Comme  $h(k)$  est nul sauf pour  $k = 0$  et  $k = 1$  :

$$H(f) = \frac{1}{2}e^{-j2\pi f \cdot 0 \cdot T} + \frac{1}{2}e^{-j2\pi f \cdot 1 \cdot T} = \frac{1}{2}(1 + e^{-j2\pi f T})$$

En utilisant l'identité d'Euler  $e^{-j\theta} = \cos(\theta) - j \sin(\theta)$  :

$$H(f) = \frac{1}{2}(1 + \cos(2\pi f T) - j \sin(2\pi f T))$$

**Gain en amplitude** Le module est :

$$\begin{aligned} |H(f)| &= \left| \frac{1}{2}(1 + e^{-j2\pi f T}) \right| = \sqrt{\left( \frac{1 + \cos(2\pi f T)}{2} \right)^2 + \left( \frac{\sin(2\pi f T)}{2} \right)^2} \\ &= \frac{1}{2} \sqrt{(1 + \cos(2\pi f T))^2 + \sin^2(2\pi f T)} \end{aligned}$$

En développant et utilisant  $\cos^2 \theta + \sin^2 \theta = 1$  :

$$|H(f)| = \frac{1}{2} \sqrt{2 + 2 \cos(2\pi f T)} = \sqrt{\frac{1 + \cos(2\pi f T)}{2}} = \cos(\pi f T)$$

### Interprétation :

- $f = 0$  :  $|H(0)| = \cos(0) = 1$  (gain maximal)
- $f = \frac{f_s}{2}$  :  $|H(f_s/2)| = \cos(\pi) = 0$  (atténuation totale)

Le filtre est donc un **passé-bas** qui atténue les hautes fréquences.

### Question 2.3 :

La synthèse additive permet de générer un son plus rond, avec une sensation de résonance accrue. En comparaison, le son obtenu par synthèse soustractive à partir d'un signal en dent de scie est plus clair, plus brillant, voire légèrement nasillard. À l'écoute, la distinction est nette : la synthèse soustractive évoque davantage le timbre d'un clavier électronique, tandis que la synthèse additive produit un son plus doux et organique.

Sur le plan spectral, le signal additif présente des harmoniques plus espacées mais de plus forte amplitude. À l'inverse, le signal obtenu par dent de scie contient davantage d'harmoniques intermédiaires, ce qui enrichit son contenu fréquentiel, mais avec des amplitudes généralement plus faibles.

## 3 Effets audio-numériques

### 3.1 Expression de la corrélation

Dans cette question, on s'intéresse à la manière dont on peut relier les fonctions de corrélation à la réponse impulsionnelle d'une pièce.

Le signal émis  $x(k)$  est modifié par les propriétés acoustiques de la pièce (réverbération, échos...), et ce signal transformé est capté sous la forme  $y(k)$ . Dans le cadre linéaire et invariant dans le temps (modèle LTI), la relation entre le signal d'entrée et celui de sortie est donnée par :

$$y(k) = (x * h)(k)$$

où  $h(k)$  désigne la réponse impulsionnelle de la pièce.

Nous voulons exprimer la fonction d'intercorrélation  $R_{yx}(u)$  entre  $y$  et  $x$  en fonction de  $h(k)$  et de la fonction d'autocorrélation  $R_{xx}(u)$ . L'intercorrélation est définie par :

$$R_{yx}(u) = y(u) * x(-u)$$

On remplace  $y(u)$  par son expression en convolution, soit  $y(u) = x(u) * h(u)$  :

$$R_{yx}(u) = (x(u) * h(u)) * x(-u)$$

Cette expression peut être réécrite en changeant l'ordre des opérations, puisque la convolution est associative :

$$R_{yx}(u) = x(u) * x(-u) * h(u)$$

Or, la convolution entre un signal et son opposé temporel  $x(u) * x(-u)$  correspond à l'autocorrélation de  $x$  :

$$x(u) * x(-u) = R_{xx}(u)$$

On en déduit donc :

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

Ce résultat montre que l'intercorrélation entre le signal émis et le signal capté est équivalente à la convolution de la réponse impulsionnelle avec l'autocorrélation du signal d'entrée.

Cette relation sera exploitée dans la suite du projet pour reconstruire ou estimer  $h(k)$  à partir de mesures effectuées sur  $x(k)$  et  $y(k)$ .

### 3.2 Méthode d'estimation de la réponse impulsionnelle

Nous avons donc montré que la fonction d'intercorrélation entre le signal de sortie  $y(k)$  et le signal d'entrée  $x(k)$  s'exprime comme la convolution suivante :

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

Cette équation est exploitable si l'on parvient à simplifier le rôle de  $R_{xx}(u)$  dans cette relation. Pour cela, nous allons exploiter un cas particulier : si  $x(k)$  est un signal dont l'autocorrélation  $R_{xx}(u)$  est proche de la fonction de Dirac  $\delta(u)$ , alors la convolution devient beaucoup plus simple à inverser.

Rappelons que la fonction de Dirac discrète est :

$$\delta(u) = \begin{cases} 1 & \text{si } u = 0 \\ 0 & \text{sinon} \end{cases}$$

En remplaçant  $R_{xx}(u)$  par  $\delta(u)$  dans la relation précédente, on obtient :

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

On en déduit une méthode directe pour estimer la réponse impulsionnelle d'un système acoustique : il suffit de calculer l'intercorrélation entre le signal d'entrée et le signal de sortie, à condition que le signal d'entrée ait une autocorrélation quasi nulle en dehors de  $u = 0$ .

Nous pouvons remplir cette condition par l'utilisation d'un bruit blanc, c'est-à-dire un signal aléatoire ayant une distribution de fréquence plate et une autocorrélation impulsionnelle.

La méthode consiste donc à :

1. Choisir un signal d'excitation  $x(k)$  ayant une autocorrélation proche d'une impulsion, par exemple du bruit blanc.
2. Simuler ou enregistrer la réponse du système (la pièce) en mesurant  $y(k)$ .
3. Calculer l'intercorrélation  $R_{yx}(u)$  à l'aide de la commande `xcorr` de Matlab.
4. Utiliser le résultat obtenu comme estimation de la réponse impulsionnelle  $h(u)$ .

### 3.3 Sélection du signal d'excitation

Le fichier `signal_excitation.mat` contient deux signaux candidats, `xe1` et `xe2`, tous deux échantillonnés à la fréquence  $f_e$ . L'objectif de cette question est de déterminer lequel des deux est le plus adapté pour estimer la réponse impulsionnelle d'une pièce.

Pour cela, nous avons analysé leurs autocorrélations respectives à l'aide de la fonction `xcorr`, qui calcule la corrélation. Les résultats sont présentés dans la figure ci-dessous.

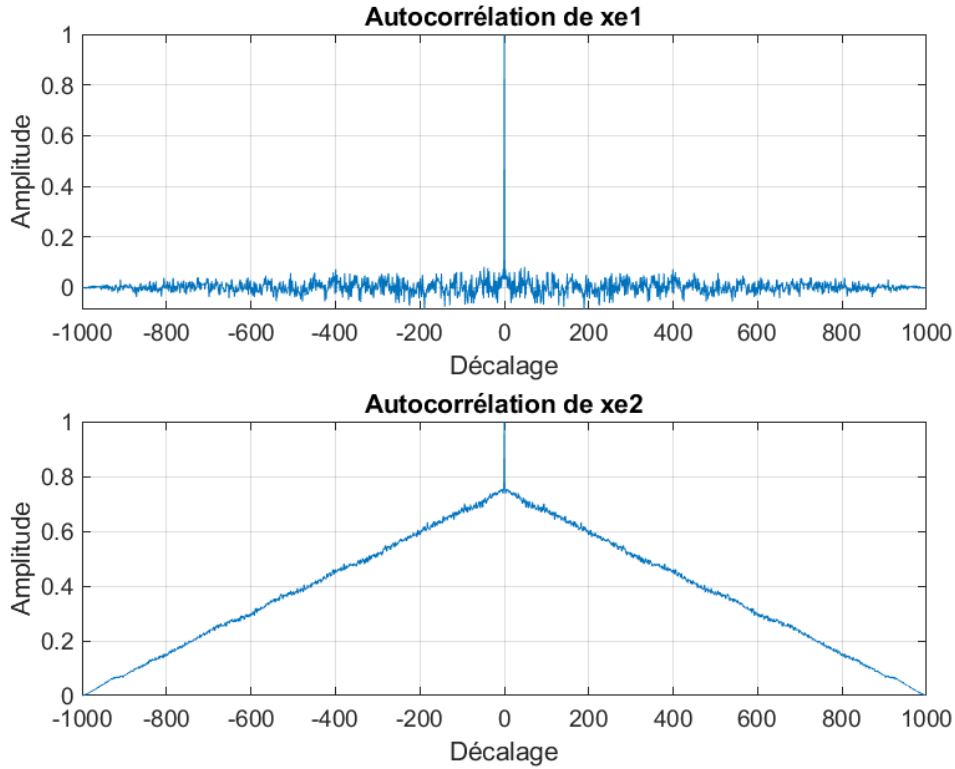


FIGURE 6 – Autocorrélations normalisées de **xe1** et **xe2** , affichées entre  $-1000$  et  $+1000$  échantillons

Visuellement, on constate que le signal **xe1** présente un pic très étroit centré en zéro, avec des lobes secondaires quasi inexistantes. À l'inverse, l'autocorrélation de **xe2** est plus étalée, avec un pic large et des valeurs non nulles sur une portion significative de la fenêtre.

Le signal **xe1** est donc nettement plus adapté et sera retenu pour la suite.

### 3.4 Estimation expérimentale de la réponse impulsionnelle

Dans cette question, nous mettons en œuvre la méthode d'estimation de la réponse impulsionnelle décrite précédemment, en utilisant le signal d'excitation **xe1** et la fonction `simule_piece`.

La procédure suivie est la suivante :

1. Le signal **xe1** est injecté dans le système modélisé par `simule_piece`, accompagné de la fréquence d'échantillonnage `fe`.
2. Le signal en sortie `y` est obtenu, représentant la réponse acoustique de la pièce au signal d'entrée.
3. L'intercorrélation `xcorr(y, xe1)` est calculée afin d'estimer la réponse impulsionnelle  $h(k)$  du système.
4. On repère le maximum de la corrélation pour isoler la partie causale de la réponse impulsionnelle (à partir de zéro).
5. L'axe temporel est converti en secondes pour une meilleure lecture.

La figure ci-dessous illustre la réponse impulsionnelle estimée :

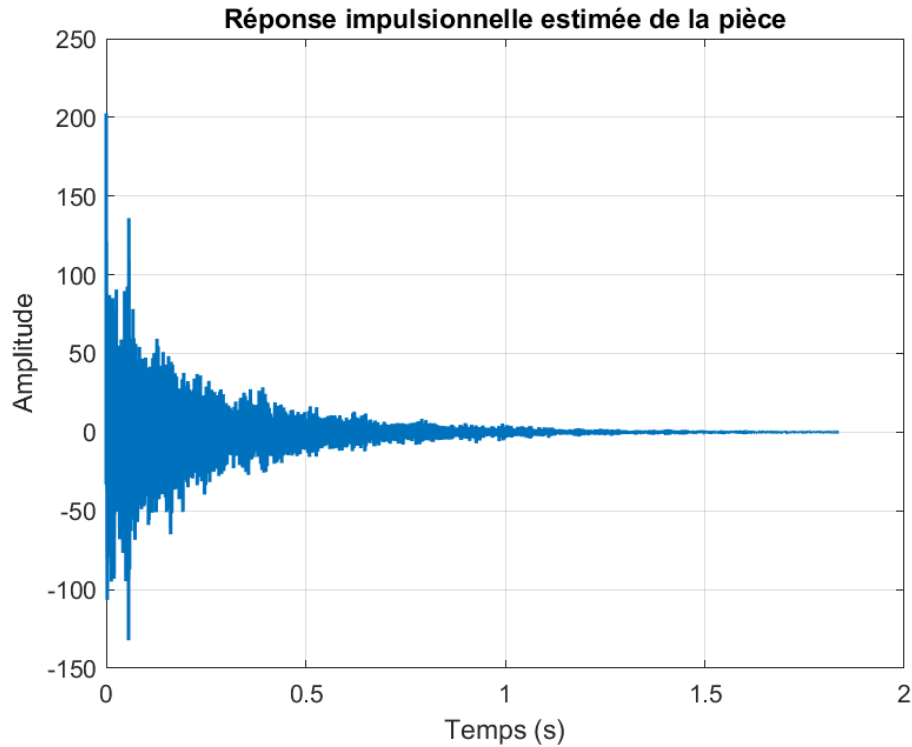


FIGURE 7 – Réponse impulsionnelle estimée de la pièce

On observe un pic initial très marqué qui correspond à l'arrivée directe du signal, suivi d'une décroissance progressive contenant des oscillations dues aux réflexions multiples dans la pièce. L'ensemble correspond à une réponse impulsionnelle typique d'un environnement réverbérant : forte énergie initiale, suivie d'une queue de réverbération qui diminue progressivement vers zéro.

Cette estimation sera utilisée dans les prochaines étapes pour appliquer un effet de réverbération réaliste à d'autres signaux.

### 3.5 Application de la réverbération par convolution classique

Nous avons codé la fonction suivante, notée `effet_reverb`, qui réalise la convolution classique entre un signal d'entrée et une réponse impulsionnelle :

Fonction `effet_reverb`

```
function y = effet_reverb(x, h)
    y = conv(x, h, 'same');
end
```

Cette fonction sera utilisée dans la question suivante pour traiter un signal audio et observer l'effet de la réverbération simulée par convolution temporelle.

### 3.6 Application de la réverbération par convolution directe

Maintenant nous voulons appliquer un effet de réverbération à un signal audio réel à l'aide de la fonction `effet_reverb` développée précédemment. L'objectif est de simuler un

rendu acoustique réaliste en convoluant le signal original avec une réponse impulsionnelle  $h(k)$ .

#### Méthodologie :

1. Le signal audio `single_tone_guitar_nylon_string_a2.wav` est chargé en mono.
2. La réponse impulsionnelle de la pièce est simulée à partir du signal d'excitation `xe1`, via la fonction `simule_piece`.
3. La convolution est effectuée à l'aide de la fonction :

```
y = effet_reverb(x, h);
```

4. Le signal résultant est normalisé pour éviter les effets de saturation lors de la sauvegarde :

```
y = y / max(abs(y));  
audiowrite('signal_reverbere.wav', y, fe);
```

5. Le temps de calcul est mesuré à l'aide des commandes `tic` et `toc`.
6. Un graphique compare les signaux original et réverbéré.

#### Temps de calcul mesuré :

Durée de la convolution classique  $\approx 1,396712$  secondes

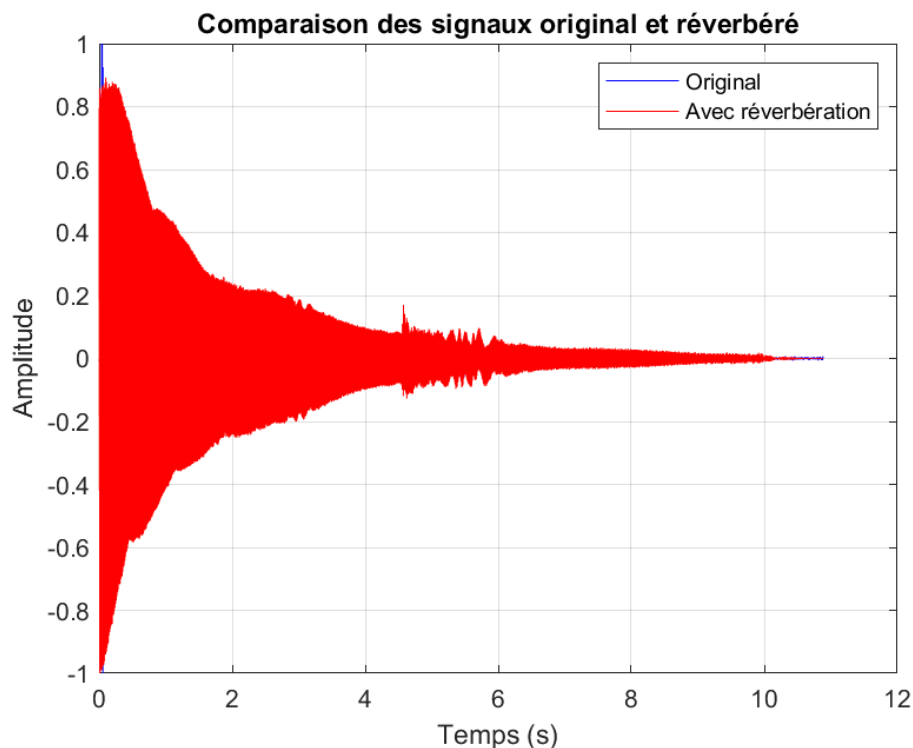


FIGURE 8 – Comparaison des signaux original et réverbéré d'un son d'une guitare, après convolution avec la réponse impulsionnelle  $h(k)$ .

On voit clairement que le signal réverbéré, en rouge ne s'arrête pas net comme le signal original que l'on ne voit même plus sur la figure vu que la réverbération est plus ample.



Il continue encore un moment, avec une sorte de traînée. C'est ce qu'on attend d'un effet de réverbération : le son met un peu de temps à s'éteindre, comme s'il résonnait dans une pièce.

À l'écoute, le signal paraît plus large, plus profond, comme si on jouait de la guitare dans une salle plutôt que dans un espace fermé ou neutre.

Par contre, cette méthode prend un peu de temps à calculer (environ 1,4 seconde dans notre cas), ce qui peut devenir un problème si on doit traiter des fichiers plus longs ou plusieurs pistes audio.

### 3.7 Application de la réverbération par convolution rapide

Maintenant, nous cherchons à avoir le même résultat mais en utilisant une méthode bien plus rapide : la convolution par transformée de Fourier (TFD).

La convolution dans le domaine temporel peut être remplacée par une multiplication dans le domaine fréquentiel. Cela nous permet un gain de temps, surtout visible pour des signaux longs. La méthode repose sur les étapes suivantes :

- Calcul de la FFT du signal d'entrée  $x(k)$  et de la réponse impulsionnelle  $h(k)$ ,
- Multiplication terme à terme dans le domaine fréquentiel,
- Retour dans le domaine temporel via TFI.

Voici la fonction utilisé :

#### Fonction effet\_reverb\_fft

```
function y = effet_reverb_fft(x, h)
    N = length(x) + length(h) - 1;
    X = fft(x, N);
    H = fft(h, N);
    Y = X .* H;
    y = ifft(Y, N);
end
```

Nous avons utilisé cette fonction pour traiter le fichier `single_tone_guitar_nylon_string_a2.wav`, avec la même réponse impulsionnelle que dans la question 3.6.

Temps de calcul (convolution TFD) $\approx 0,680972$ secondes
---

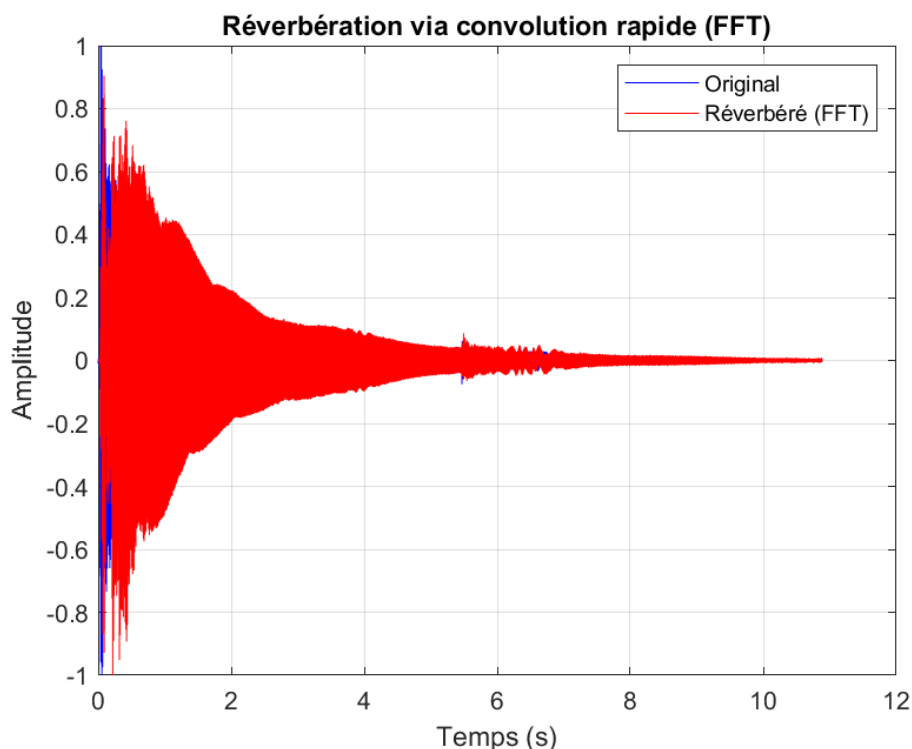


FIGURE 9 – Comparaison du signal original et du signal réverbéré par convolution rapide (TFD).

On retrouve la même réverbération que précédemment, avec une queue sonore qui s'étale bien dans le temps. Mais cette fois-ci, le traitement est presque deux fois plus rapide que la méthode classique. À l'écoute, le rendu est similaire, ce qui prouve que l'approche TFD est beaucoup plus efficace sans perte de qualité audible. Elle est donc mieux adaptée pour le traitement de signaux longs.

### 3.8 Comparaison des méthodes de convolution

Dans cette dernière question de cette sous-partie, nous comparons les deux approches utilisées pour appliquer la réverbération à un signal audio : la convolution classique, question 3.6 et la convolution rapide via TFD, question 3.7.

Les deux méthodes donnent un signal réverbéré très similaire, aussi bien visuellement que lors de l'écoute. Dans les deux cas, on retrouve une réverbération progressive, avec une queue sonore réaliste et une sensation d'espace clairement perceptible. Cela confirme que les deux méthodes sont mathématiquement équivalentes sur le plan du résultat.

Le principal écart se situe dans le temps d'exécution. Voici les durées mesurées :

- Convolution classique : **1,396712 sec**
- Convolution TFD : **0,680972 sec**

La méthode utilisant la TFD est donc environ deux fois plus rapide. Cet écart devient encore plus significatif quand la taille du signal augmente.

La convolution classique est facile à comprendre et rapide à mettre en œuvre, mais elle devient très lente avec des signaux longs. La convolution FFT est plus complexe à implémenter mais offre un gain de performance important. Elle est donc mieux adaptée pour les applications en temps réel ou pour traiter de longues séquences audio.

Les deux méthodes permettent de produire un effet de réverbération satisfaisant. Toutefois, dans une application réelle où la performance est un critère important, la méthode par TFD est préférable.

### 3.9 Démonstration par récurrence

On considère le système défini par la relation :

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

et on cherche sa réponse impulsionnelle  $h(k)$ , en supposant que l'entrée est une impulsion de Dirac :

$$x(k) = \delta(k) = \begin{cases} 1 & \text{si } k = 0 \\ 0 & \text{sinon} \end{cases}$$

On souhaite démontrer que :

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

#### Initialisation

Pour  $k = 0$  :

$$h(0) = x(0) - g \cdot y(-\tau) = \delta(0) - g \cdot 0 = 1 = (-g)^0$$

Pour  $1 \leq k < \tau$ , on a :

$$x(k) = 0, \quad k - \tau < 0 \Rightarrow y(k - \tau) = 0 \Rightarrow h(k) = 0$$

Donc la propriété est vraie jusqu'à  $k = \tau - 1$ .

#### Hérédité

On suppose la propriété suivante vraie pour un certain entier  $n \in \mathbb{N}$  :

$$H_n : \quad h(k) = \begin{cases} (-g)^n & \text{si } k = n\tau \\ 0 & \text{si } n\tau < k < (n+1)\tau \end{cases}$$

On veut montrer que cela implique que :

$$h(k) = \begin{cases} (-g)^{n+1} & \text{si } k = (n+1)\tau \\ 0 & \text{si } (n+1)\tau < k < (n+2)\tau \end{cases}$$

**Vérifions d'abord au rang  $k = (n+1)\tau$  :**

$$\begin{aligned} h((n+1)\tau) &= x((n+1)\tau) - g \cdot y((n+1)\tau - \tau) \\ &= 0 - g \cdot h(n\tau) = -g \cdot (-g)^n = (-g)^{n+1} \end{aligned}$$

**Pour les valeurs  $k \in (n\tau, (n+1)\tau)$  qui ne sont pas des multiples de  $\tau$  :**

$$h(k) = x(k) - g \cdot y(k - \tau) = 0 - g \cdot h(k - \tau)$$

Mais  $k - \tau \notin \tau \cdot \mathbb{N}$  donc, par hypothèse,  $h(k - \tau) = 0$ , ce qui donne :

$$h(k) = 0$$

## Conclusion

La propriété est vraie pour  $n = 0$ , et si elle est vraie pour un certain  $n$ , elle l'est aussi pour  $n + 1$ . Par le principe de récurrence, elle est donc vraie pour tout  $n \in \mathbb{N}$ , ce qui conclut la démonstration.

## 3.10 Condition de stabilité du filtre

On cherche ici à déterminer pour quelles valeurs de  $g \in \mathbb{R}$  le système défini par :

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

est **stable**.

On parle ici de stabilité au sens du traitement du signal, c'est-à-dire que la sortie  $y(k)$  doit rester bornée lorsque l'entrée  $x(k)$  est bornée.

### Réponse impulsionnelle :

D'après la question précédente, on sait que la réponse impulsionnelle du système est donnée par :

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

La question de la stabilité revient donc à se demander si la suite  $h(k)$  est **absolument sommable**, c'est-à-dire si :

$$\sum_{k=0}^{\infty} |h(k)| < +\infty$$

### Étude de la convergence :

Puisque  $h(k) = (-g)^n$  uniquement pour  $k = n\tau$ , on a :

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

Il s'agit d'une série géométrique de raison  $|g|$ . Elle converge si et seulement si  $|g| < 1$ .

### Conclusion :

Le filtre est stable si et seulement si la série  $\sum |h(k)|$  converge, c'est-à-dire :

$\text{Le filtre est stable si } |g| < 1$

Dans ce cas, l'effet des échos successifs décroît suffisamment vite pour que le signal de sortie reste borné.

### 3.11 Détermination des vecteurs **a** et **b**

À partir de l'équation du filtre delay :

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

on reconnaît une structure typique de filtre récursif de type RII (IIR), conforme à la forme générale vue en cours :

$$y(k) = \sum_{i=0}^M b_i x(k-i) - \sum_{j=1}^N a_j y(k-j)$$

**Vecteur **b** (partie directe) :** Le signal d'entrée  $x(k)$  intervient uniquement sans décalage :

$$\Rightarrow \mathbf{b} = [1]$$

**Vecteur **a** (partie réursive) :** Le signal de sortie est réinjecté avec un retard  $\tau$  et un coefficient  $g$ . Cela correspond à :

$$\mathbf{a} = [1, \underbrace{0, \dots, 0}_{\tau-1 \text{ zéros}}, g, 0, 0 \dots 0]$$

Ce vecteur **a** respecte la structure  $a_0 = 1$  imposée pour les filtres RII (normalisation du terme principal).

Cette représentation permet d'utiliser directement la fonction `filter(b, a, x)` de MATLAB pour simuler l'effet du filtre delay.

### 3.12 Obtention de la réponse impulsionnelle via `filter`

Nous simulons la réponse impulsionnelle du filtre défini par l'équation suivante :

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

en injectant un Dirac  $x(k) = \delta(k)$  en entrée du filtre.

On utilise pour cela la fonction `filter` de MATLAB, avec :

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

Le script `analyse_delay.m` a été exécuté avec les paramètres :  $\tau = 5$ ,  $g = 0,7$  et une durée de 100 échantillons.

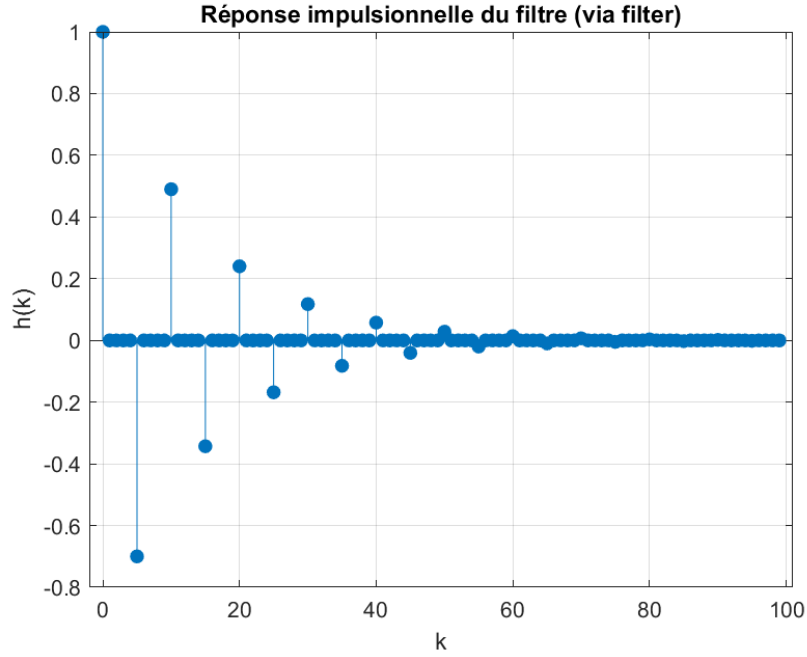


FIGURE 10 – Réponse impulsionnelle  $h(k)$  obtenue numériquement.

La figure ci-dessus montre bien les caractéristiques d'un filtre passe-bas RII comme expliqué dans le cours.

### 3.13 Réponse fréquentielle théorique du filtre à délai

Dans cette question, on souhaite déterminer la transformée de Fourier discrète de la réponse impulsionnelle  $h(k)$  du filtre défini par :

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

Nous avons vu précédemment que la réponse impulsionnelle de ce filtre est donnée par :

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

On souhaite donc calculer la transformée de Fourier réduite de  $h(k)$ , définie par :

$$\hat{h}(\nu) = \sum_{k=0}^{\infty} h(k) e^{-j2\pi\nu k}$$

Comme  $h(k)$  est non nul uniquement pour les indices  $k = n\tau$ , on remplace  $k$  par  $n\tau$  dans la somme :

$$\hat{h}(\nu) = \sum_{n=0}^{\infty} (-g)^n \cdot e^{-j2\pi\nu n\tau} = \sum_{n=0}^{\infty} (-g \cdot e^{-j2\pi\nu\tau})^n$$

Cette expression est une série géométrique de raison  $r = -ge^{-j2\pi\nu\tau}$ , de module  $|g| < 1$  (car filtre stable). On peut donc écrire :

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

On calcule maintenant le module  $|\hat{h}(\nu)|$  :

$$|\hat{h}(\nu)| = \left| \frac{1}{1 + ge^{-j2\pi\nu\tau}} \right| = \frac{1}{|1 + ge^{-j2\pi\nu\tau}|}$$

En posant  $e^{-j2\pi\nu\tau} = \cos(2\pi\nu\tau) - j \sin(2\pi\nu\tau)$ , on obtient :

$$|1 + ge^{-j2\pi\nu\tau}|^2 = (1 + g \cos(2\pi\nu\tau))^2 + (g \sin(2\pi\nu\tau))^2 = 1 + 2g \cos(2\pi\nu\tau) + g^2$$

Donc le module est :

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

La phase correspond à l'argument du dénominateur :

$$\arg(\hat{h}(\nu)) = -\arg(1 + ge^{-j2\pi\nu\tau})$$

On développe à nouveau  $e^{-j2\pi\nu\tau}$ , puis on écrit :

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

La réponse fréquentielle du filtre à délai est donc donnée par :

$$\hat{h}(\nu) = \frac{1}{1 + ge^{-j2\pi\nu\tau}}, \quad |\hat{h}(\nu)| = \frac{1}{\sqrt{1 + 2g \cos(2\pi\nu\tau) + g^2}}, \quad \arg(\hat{h}(\nu)) = -\arctan\left(\frac{g \sin(2\pi\nu\tau)}{1 + g \cos(2\pi\nu\tau)}\right)$$

Ces expressions serviront de base à la comparaison avec les résultats numériques (FFT) pour la question suivante.

### 3.14 Comparaison entre la réponse fréquentielle théorique et celle obtenue par FFT

Nous cherchons à vérifier que la réponse fréquentielle calculée numériquement à l'aide de la transformée de Fourier discrète correspond bien à l'expression théorique obtenue précédemment.

La réponse fréquentielle théorique du filtre à délai est la suivante :

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

où :

- $\nu \in [0, 0.5]$  est la fréquence
  - $g$  est le coefficient d'amortissement ici fixé à 0,7
  - $\tau$  est le retard en nombre d'échantillons, il sera ici à  $\tau = 5$
- Pour vérifier cette expression, nous avons procédé en deux étapes :

1. Nous avons d'abord calculé la réponse impulsionnelle du filtre avec la commande `filter`, en lui appliquant une impulsion de Dirac.
2. Ensuite, nous avons utilisé la fonction `fft` pour obtenir le spectre de cette réponse impulsionnelle.

De plus, nous avons tracé la courbe théorique en utilisant directement la formule précédente.

Les deux courbes ont ensuite été représentées sur le même graphique ci-dessous pour pouvoir les comparer :

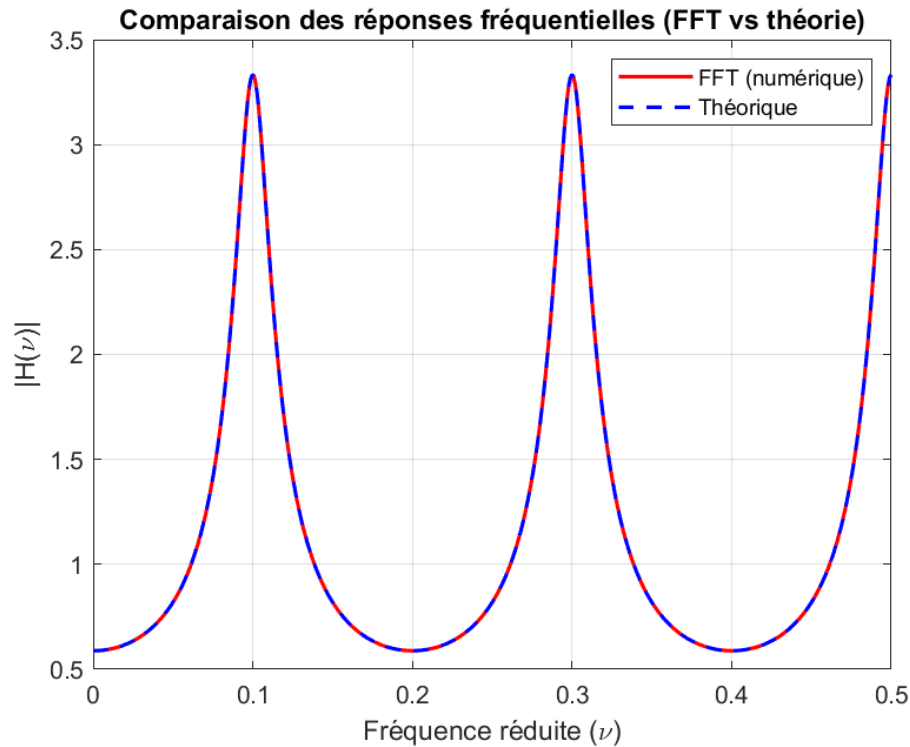


FIGURE 11 – Comparaison entre le module de la réponse fréquentielle obtenu par FFT et celui calculé théoriquement.

Les deux courbes se superposent parfaitement, ce qui confirme la cohérence entre le modèle analytique établi à la question précédente et l'implémentation numérique du filtre à l'aide de MATLAB.

Le motif périodique en dents de scie observé sur le graphique est typique d'un filtre en peigne. Il est caractérisé par la présence de pics réguliers, où le filtre laisse passer certaines fréquences et de creux où d'autres fréquences sont atténuées. Ces creux apparaissent toutes les  $\frac{1}{\tau}$  unités de fréquence réduite.

Cette comparaison confirme que notre filtre fonctionne aussi bien d'un point de vue théorique que numérique.

### 3.15 Implémentation de la fonction `effet_delay`

L'objectif de cette question est de programmer une fonction MATLAB nommée `effet_delay`, capable de simuler un effet de délai de type IIR selon l'équation suivante :



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

Cette fonction prend en paramètres :

- un signal audio d'entrée  $x$
- un délai exprimé en secondes `delay_sec`
- un coefficient d'amortissement  $g \in [-1, 1]$
- la fréquence d'échantillonnage  $F_e$

Le délai est d'abord converti en nombre d'échantillons :

$$\tau = \text{round}(\text{delay\_sec} \times F_e)$$

On utilise ensuite la fonction `filter` avec les coefficients suivants :

- le numérateur  $b = 1$
- le dénominateur  $a = [1, 0, \dots, 0, g, 0, 0, \dots, 0]$ , avec  $\tau - 1$  zéros entre 1 et  $g$

Cette structure permet de créer un filtre récursif simple, dans lequel chaque échantillon de sortie est influencé par une version retardée de la sortie précédente. Vous pouvez retrouver cette fonction dans le dossier source, sur gitlab.

### 3.16 Test de la fonction avec paramètres imposés

Nous avons testé la fonction `effet_delay` avec les paramètres suivants :

$$\text{delay\_sec} = 0,25 \text{ s} \quad \text{et} \quad g = 0,9$$

Nous avons donc :

$$\tau = 0,25 \times F_e = 11025 \text{ échantillons} \quad (\text{avec } F_e = 44100 \text{ Hz})$$

Le signal d'entrée est un accord de piano contenu dans le fichier `single_tone_piano1.wav`. La fonction `effet_delay`, que nous avons implémentée précédemment, a été appliquée à ce signal.

Pour compléter l'analyse, nous avons généré deux représentations temporelles : l'une pour le signal original, l'autre pour le signal traité.

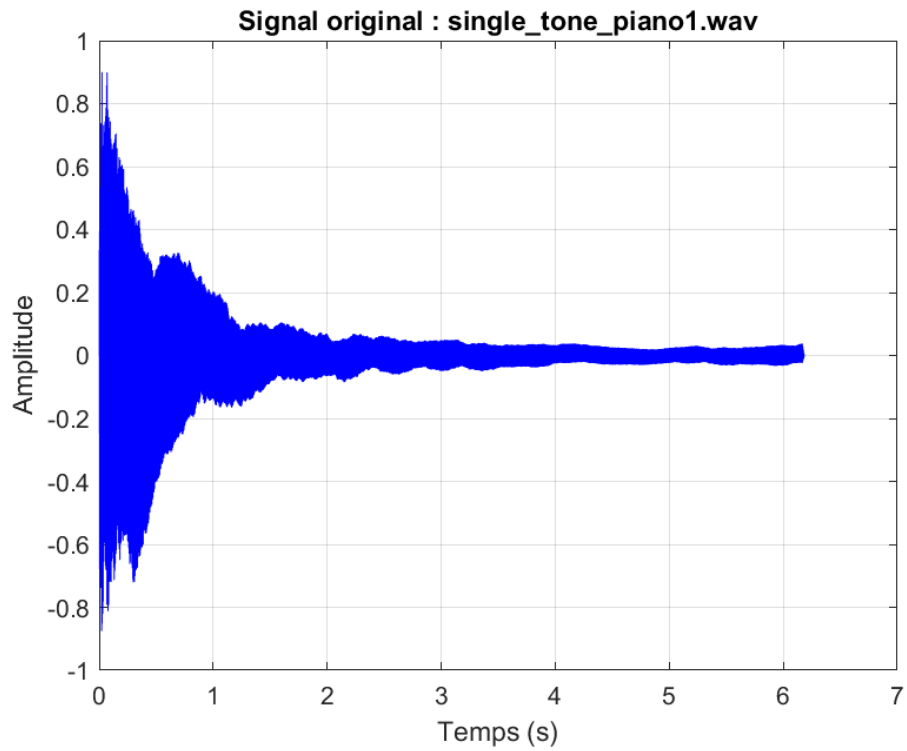


FIGURE 12 – Signal original : single\_tone\_piano1.wav.

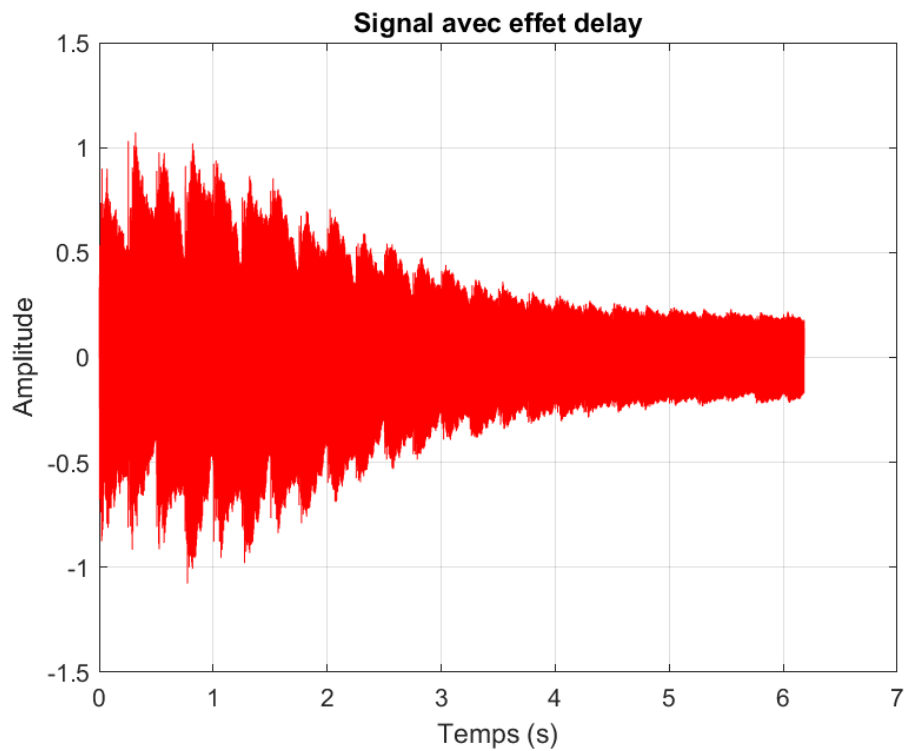


FIGURE 13 – Signal après application de l'effet delay

Le signal traité présente une succession d'échos bien visibles dans le domaine temporel, espacés de 250 ms. Ces répétitions se prolongent nettement, donnant un effet de

réverbération dense surement dû à un amortissement qui permet d' "allonger" le son.

L'effet delay est clairement identifiable dans le signal après application. Ce test confirme le bon comportement de notre fonction et permet de visualiser l'influence directe des paramètres  $\tau$  et  $g$  sur la persistance temporelle du son.

### 3.17 Implémentation de la fonction `effet_delay_filtre` A RE-VOIR

Dans cette question, nous avons implémenté une nouvelle fonction `effet_delay_filtre`, basée sur la modélisation canonique des filtres IIR. Cette fonction applique un effet de delay à un signal audio, en exploitant la commande `filter` de MATLAB.

L'équation du système considéré est la suivante :

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

Cette relation correspond à un filtre récursif (IIR), qui peut être formulé à l'aide des vecteurs  $a$  et  $b$ , selon la convention :

$$y(k) = \sum_i b_i \cdot x(k - i) - \sum_j a_j \cdot y(k - j)$$

Dans notre cas :

- Le vecteur  $b$  (numérateur) contient un seul coefficient non nul :

$$b = [1]$$

- Le vecteur  $a$  (dénumérateur) contient  $a_0 = 1$ , suivi de  $\tau - 1$  zéros, puis le coefficient de rétroaction :

$$a = [1, 0, \dots, 0, g]$$

La durée du delay (en secondes) est convertie en nombre d'échantillons selon :

$$\tau = \text{round}(\text{delay\_sec} \times F_e)$$

La fonction `effet_delay_filtre` prend donc en entrée :

- un signal  $x$ ,
- un delay `delay_sec` (en secondes),
- un coefficient d'amortissement  $g$ ,
- la fréquence d'échantillonnage  $F_e$ .

Et retourne le signal  $y$  traité, en appliquant :

$$y = \text{filter}(b, a, x)$$

### 3.18 Test de la fonction `effet_delay_filtre`

nous avons testé la fonction `effet_delay_filtre` que nous avons implémentée précédemment, en lui appliquant un effet de retard sur un signal réel.

Le signal utilisé est `single_tone_piano1.wav`, échantillonné à  $F_e = 44100$  Hz. Les paramètres sont :

$$\text{delay\_sec} = 0,25 \text{ s}, \quad g = 0,9, \quad K = 10$$

La conversion du délai en échantillons donne :

$$\tau = 0,25 \times F_e = 11025$$

Nous avons utilisé la fonction `effet_delay_filtre` pour appliquer cet effet au signal d'entrée. Le fichier résultant a été sauvegardé sous le nom :

`piano_delay_filtre.wav`

Les figures suivantes comparent le signal original et le signal traité.

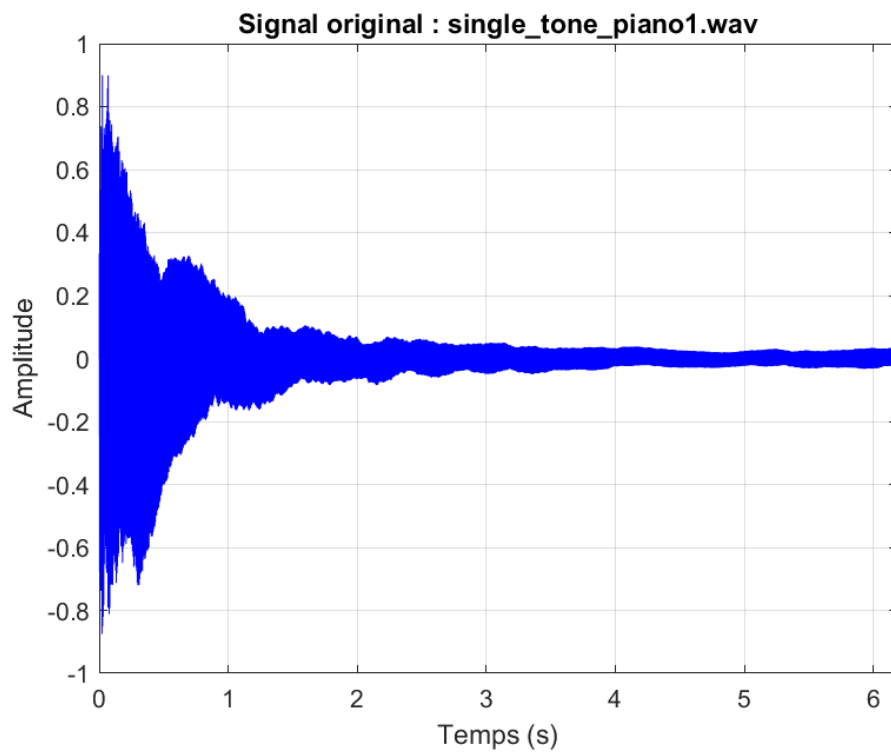


FIGURE 14 – Signal original : single\_tone\_piano1.wav

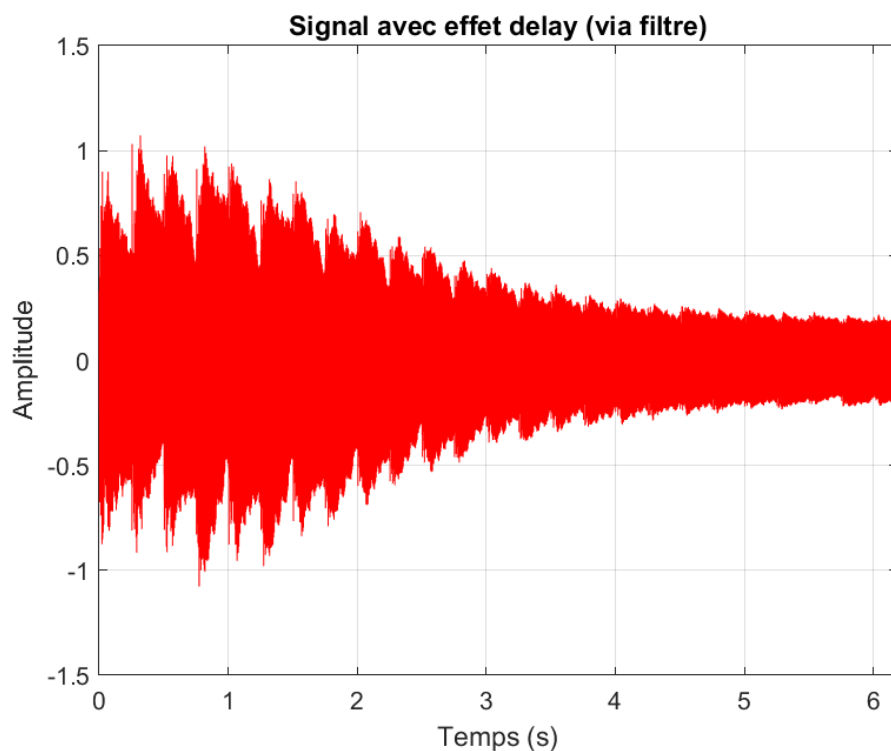


FIGURE 15 – Signal traité avec effet delay via filtre ( $g = 0,9$ ,  $\tau = 0,25 \cdot F_e$ ).

Une fois encore, le signal traité montre des échos réguliers qui s'atténuent lentement. Cela s'explique par le coefficient d'amortissement  $g = 0,9$ , qui autorise plusieurs répétitions successives du signal.

Ce test valide le bon fonctionnement de la fonction `effet_delay_filtre` et illustre clairement son impact temporel sur un son naturel.

### 3.19 Test de la fonction `effet_delay_filtre`

L'effet de filtrage observé dans `piano_delay_filtre.wav` est une conséquence directe de la réponse en fréquence du filtre delay. Les "coupures" de fréquence sont justifiées mathématiquement et sont bien visibles sur le spectre calculé.

## Conclusion

Ce projet nous a permis d'explorer de manière concrète les principales notions du traitement du signal appliquées à l'audio, à travers une série d'expériences de synthèse, d'analyse spectrale, de filtrage et d'effets audio-numériques.

Dans la première partie, la synthèse additive nous a appris à construire un son à partir de ses composantes fréquentielles. Nous avons constaté que le respect de la structure harmonique, associé à une enveloppe temporelle réaliste (type ADSR), permettait d'approcher le timbre d'un instrument acoustique.

La synthèse soustractive, quant à elle, a mis en évidence l'intérêt de partir d'un signal riche (comme une onde dent de scie ou carrée) et d'en filtrer le contenu spectral pour modéliser un timbre. L'expérimentation avec des filtres RIF et RII, de différents ordres et coupures, nous a permis de mieux comprendre l'impact de la conception du filtre sur la perception sonore finale.

Dans la deuxième partie, dédiée aux effets audio-numériques, nous avons implémenté l'effet de réverbération par convolution, puis optimisé son calcul par la FFT. Cette approche, fidèle à la réalité physique d'une pièce, nous a permis de manipuler des réponses impulsionnelles réelles pour recréer un environnement sonore immersif.

Enfin, dans la dernière section, nous avons analysé en profondeur l'effet de delay, d'abord en étudiant sa réponse impulsionnelle et sa réponse fréquentielle, puis en ajoutant un filtre de moyenne glissante dans la boucle de rétroaction. Nous avons démontré mathématiquement et expérimentalement que ce filtrage progressif atténue les hautes fréquences à chaque écho, ce qui explique la coloration plus sourde du son obtenu.

Au final, ce projet nous a permis non seulement d'appliquer concrètement les outils théoriques du traitement du signal (TFD, filtres IIR/FIR, convolution), mais aussi de développer une compréhension fine des mécanismes sonores, en liant perception auditive, analyse mathématique et implémentation numérique.