



28 mai 2025

Tandeau de Marsac Gautier, Essaleh Achraf, Bonduelle Louis

Classe B1 3/4

SAR Audio

Votre mail stipulait que
votre dépôt était celui de
M. Tandeau, mais il était vide.
J'ai donc pris celui de M. Essaleh,
le seul avec du contenu.



IMT Atlantique

Bretagne-Pays de la Loire
École Mines-Télécom

1 Introduction

Ce document rend compte de l'état final de notre projet en SAR Audio de la partie signal sur MATLAB.

Ce TP est divisé en 3 parties pour aborder différentes techniques de traitement de signal : la synthèse additive, la synthèse soustractive, et les effets audio-numériques.

Les expériences menées permettent aussi de montrer les différences potentielles entre les instruments de musique et leur équivalent numérique.

2 Synthèse additive

Question 1.1

Grâce au code MathLab présent sur le git, on obtient ces représentations spectrales pour le Cello-a3, la nylon-guitar et le Sax-alto qui lui n'est pas un instrument à corde.

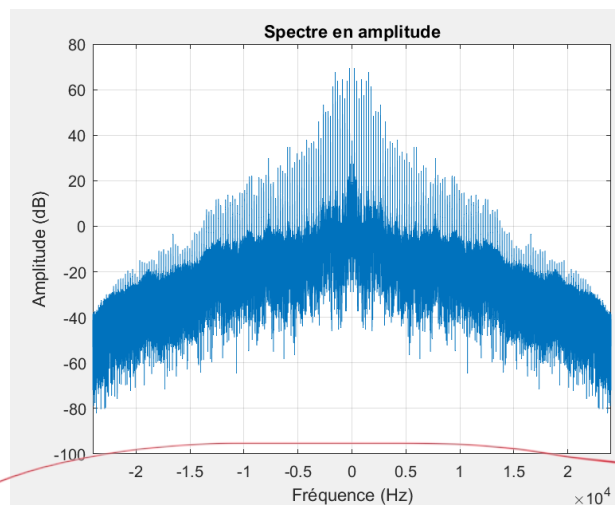


FIGURE 1 – Représentation spectrale de Cello-a3

dans votre code, l'échelle des fréquences n'est pas correcte :
linspace $(-\frac{F_e}{2}, \frac{F_e}{2}, N)$ ne donne pas un écart fréquentiel de $\frac{F_e}{N}$ mais de $\frac{F_e}{N-1}$;
cela donne des erreurs conséquentes si N n'est pas très grand

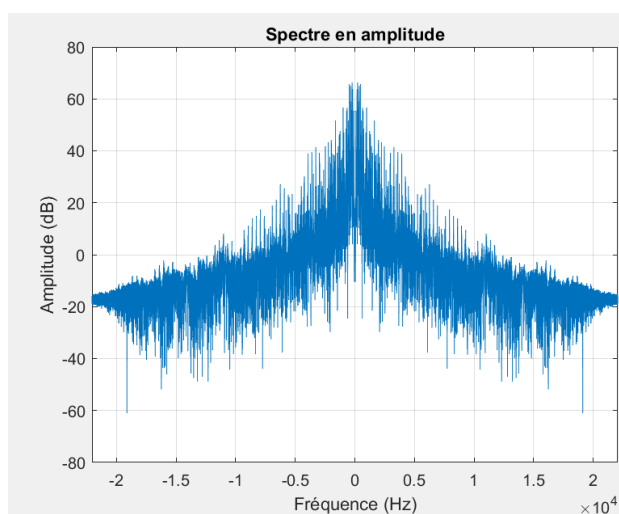


FIGURE 2 – Représentation spectrale de Nylon guitar

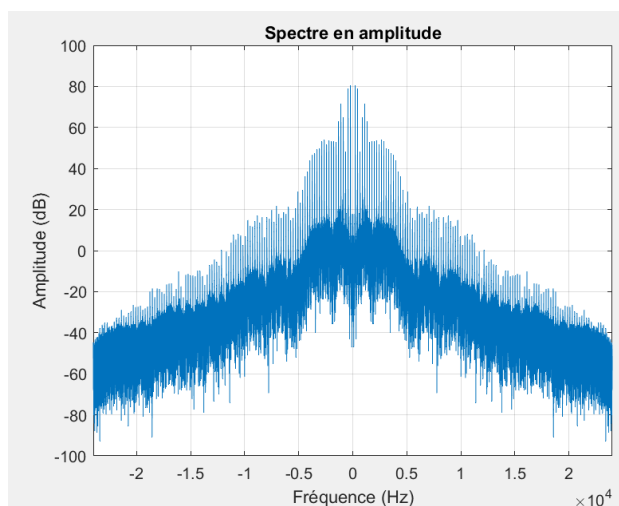


FIGURE 3 – Représentation spectrale de Sax-alto

Les 3 figures ont toutes une allure globale similaire et le calcul des fréquences fondamentales de ces instruments donne respectivement 220,9 Hz, 232,6 Hz et 220,7 Hz.

On peut aussi remarquer que pour le Cello-a3 et le saxo-alto, les raies sont moins régulières en amplitude que les instruments à corde (par exemple la nylon-guitar) ce qui traduit une inharmonicité plus grande.

Question 1.2

La représentation spectrale des 2 pianos piano1.wav et piano2.wav est la suivante :

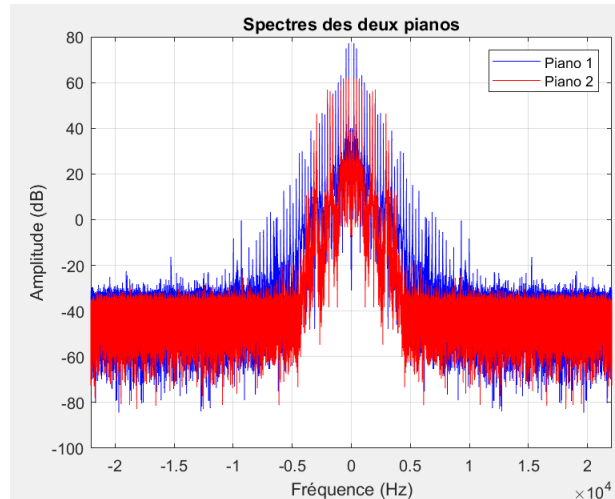


FIGURE 4 – Représentation spectrale des sons de 2 pianos

Pour comparer au mieux ces 2 pianos, on peut construire le tableau des harmoniques de chaque piano pour en déduire une inharmonicité. Grâce au code permettant de trouver les différentes valeurs des harmoniques, on obtient :

	f2	f3	f4	f5	f6	f7
Théorique	440,75	661,12	881,50	1101,87	1322,24	1542,62
Mesurées	442,04	663,22	885,54	1108,34	1331,46	1552,48
Inharmonicité	5,07	5,50	7,92	10,13	12,02	11,03

FIGURE 5 – Tableau du piano 1

	f2	f3	f4	f5	f6	f7
Théorique	435,43	653,14	870,95	1088,56	1306,28	1523,99
Mesurées	442,78	662,23	880,81	1095,92	1306,71	1531,78
Inharmonicité	29,01	23,93	19,68	11,66	0,57	8,83

FIGURE 6 – Tableau du piano 2

On en déduit que l'inharmonicité moyenne du piano1 est 8,61 cents alors que celle du piano2 est 15,61 cents.

Le piano1 est donc plus harmonique que le piano2.

ces tableaux ne sont pas cohérents avec votre code, dans lequel vous essayez de trouver automatiquement les coordonnées des harmoniques, ce qui n'est pas facile, mais qui n'était pas demandé.

Question 1.3

Le code "Generation-son-q13" sur le git permet de générer le son demandé en se limitant aux 8 premières harmoniques.

Il faut donc travailler sur le son du piano 1 qui est le plus harmonieux.

Question 1.4

Pour mieux approcher le son du piano qui n'a pas une enveloppe constante en fonction du temps, il faut utiliser la modification d'enveloppe de type ADSR (attack, decay, sustain, release).

Chaque terme de ce type de méthode décrit une partie précise de celle-ci. Il faut donc implémenter des paramètres spécifiques puis coder ce que va faire chaque partie.

Le code modifié est sur le git : "Modification-ADSR-q14".

Avec l'enveloppe ADSR appliquée ($T_A=30\text{ms}$, $T_D=120\text{ms}$, Niveau sustain = 0.7, $T_R=300\text{ms}$), l'attaque rapide puis décroissante se rapproche du comportement énergétique d'une corde frappée. Le son est donc nettement plus réel qu'à la question précédente.

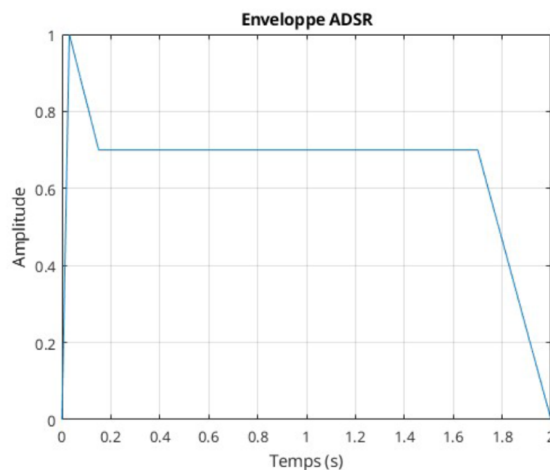


FIGURE 7 – Enveloppe ADSR

Question 1.5

L'erreur maximale entre la somme des sinusoïdes et l'IFFT est très minime voire négligeable. En réalité ces deux méthodes sont équivalentes d'un point de vue numérique et une différence existe et provient du nombre d'harmoniques ou de l'arrondi.

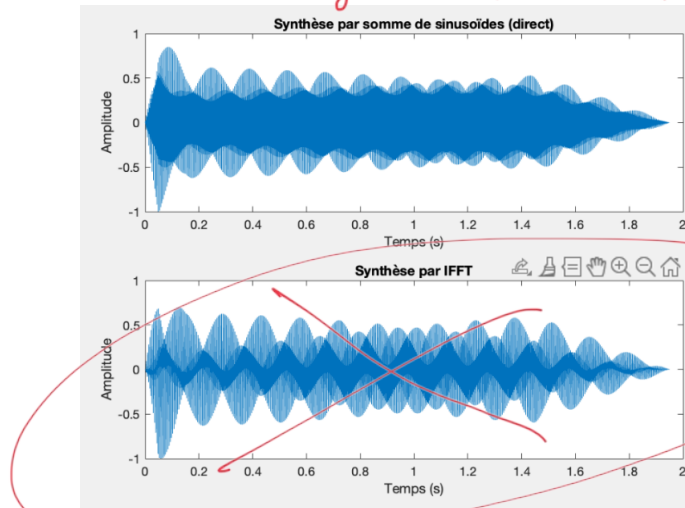


FIGURE 8 – Comparaison somme de Q1.3 et IFFT

Ainsi, l'IFFT est efficace si le nombre d'harmonique est grand, alors que la somme temporelle a une souplesse (ajout / retrait dynamique de partiels)

Au bon ? comment
pensez vous
dire
cela ?
manifeste, les 2 plots sont
assez différents !
et en fréquentiel ?

ça ne
correspond
pas
à votre
code
(n'a une
synthèse par IFFT)

3 Synthèse soustractive

Question 2.1

Pour l'analyse théorique du signal carré centré, sa série de Fourier est :
Décroissance en $1/n$ avec seulement les harmoniques impaires.

$$s_{\text{carré}}(t) = \sum_{n \text{ impairs}} \frac{4}{n\pi} \sin\left(\frac{2\pi n}{T}t\right)$$

FIGURE 9 – Fourier du carré

Pour l'analyse théorique du signal carré centré, sa série de Fourier est :
Décroissance en $1/n$ avec toutes les harmoniques.

$$s_{\text{dent}}(t) = \sum_{n=1}^{\infty} \frac{-2}{n\pi} (-1)^n \sin\left(\frac{2\pi n}{T}t\right)$$

FIGURE 10 – Fourier du signal en dent de scie

3. Synthèse soustractive

Validation numérique :

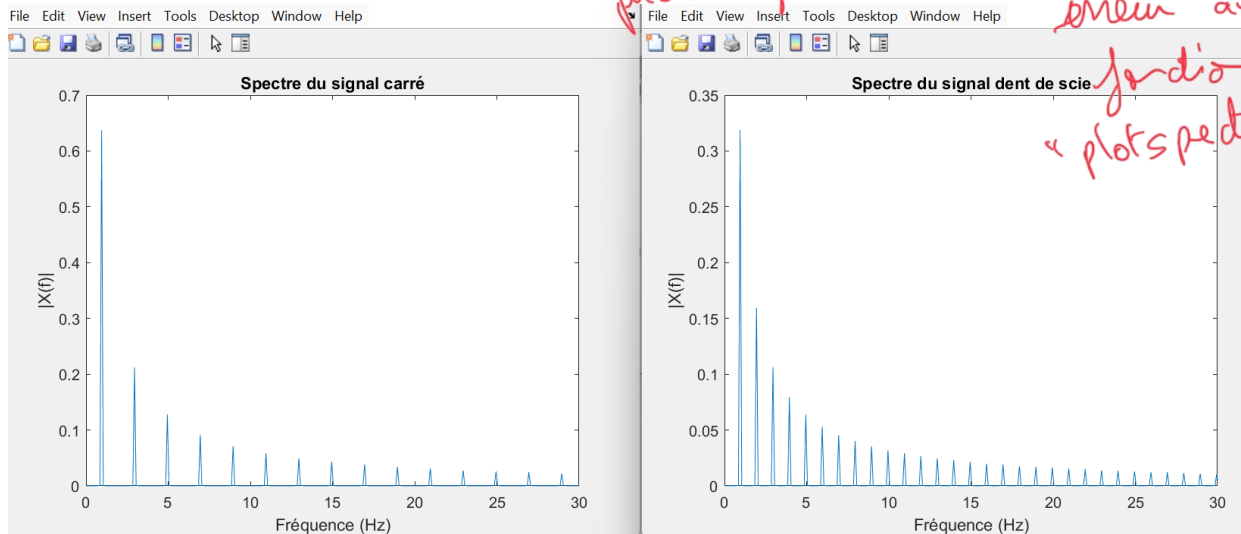


FIGURE 11 – Spectres du signal carré et du signal en dent de scie

On retrouve bien les caractéristiques énoncées plus haut. Le modèle numérique est validé. *→ No pas vraiment !*

Cependant, il y a des limites à la méthode numérique qui sont :

- phénomène de repliement spectral (effet de bord) *→ je ne le vois pas sur votre figure.*
- la FFT suppose que le signal soit périodique *et en superposant les 2 ?*

→ NON!!
Je pense comprendre ce que vous souhaitez dire mais ça c'est très faux !

Question 2.2

Nous avons le filtre passe bas défini par :

$$y(k) = 1/2(x(k) + x(k-1))$$

Ce filtre correspond à une convolution avec :

$$h(k) = 1/2\delta(k) + 1/2\delta(k-1)$$

La réponse en fréquence $H(f)$ est la TFD de $h(k)$:

$$H(f) = 1/2 + 1/2 \exp(-j2\pi fT_e)$$

Le gain du filtre en fréquence est donc :

$$|H(f)| = |\cos(\pi fT_e)|$$

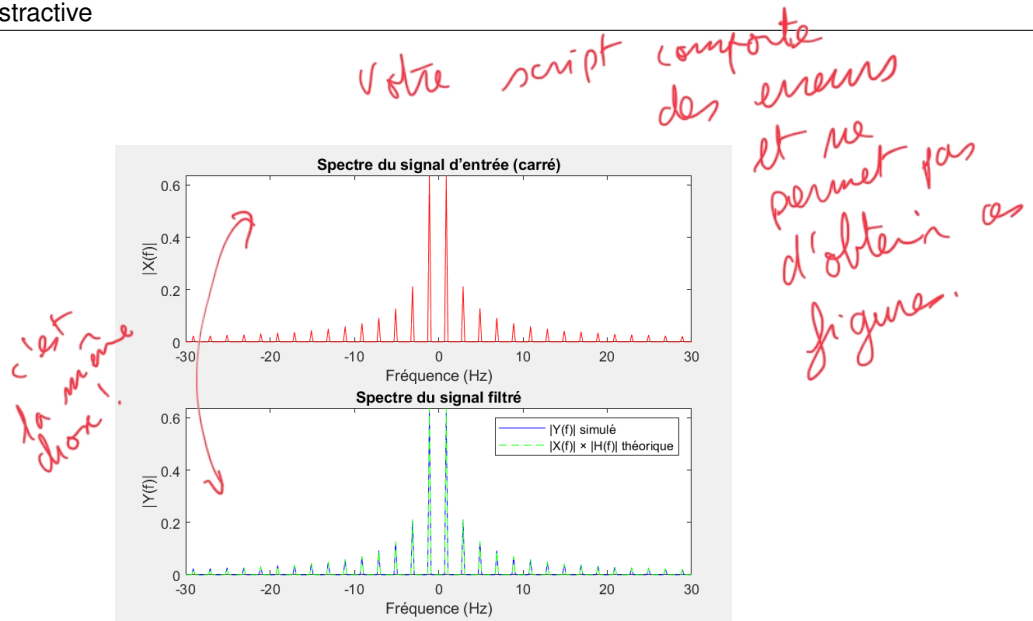


FIGURE 12 – Spectre de sortie du signal carré

Les 2 signaux, simulé et théorique, sont confondus. On peut donc valider la simulation.

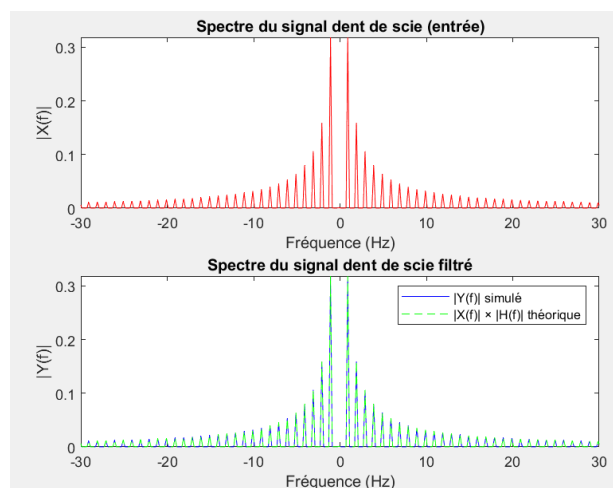


FIGURE 13 – Spectre de sortie du signal en dent de scie

De même avec le signal en dent de scie : les signaux théoriques et simulés sont confondus d'où la validation de la simulation.

Question 2.3

Tout d'abord, rappelons que pour la synthèse soustractive on part d'un signal riche en harmoniques (dent de scie, carré...) et on enlève des composantes avec un filtrage (souvent passe-bas). Alors que pour la synthèse additive on somme directement des sinusoïdes (harmoniques) avec différentes fréquences et amplitudes.

3. Synthèse soustractive

On garde la même enveloppe, on teste plusieurs signaux de base en synthèse soustractive (carré, dent de scie et triangulaire) et on le fait aussi pour la synthèse additive.

Après écoute des sons générés, on peut conclure :

En comparant les signaux obtenus avec une enveloppe commune, on observe que la forme initiale de l'onde a un impact fort sur le timbre final en synthèse soustractive.

La dent de scie, très riche en harmoniques, donne un son plus "brillant", tandis que l'onde triangulaire, plus pauvre, offre un son plus doux.

La synthèse additive, en revanche, permet un contrôle très précis du contenu spectral, mais peut manquer de la "chaleur" ou de la "vie" que la soustractive apporte naturellement par son imperfection.

Question 2.4

Pour améliorer la qualité sonore obtenue par synthèse soustractive, il peut être judicieux de changer de filtre. On peut utiliser la commande *designfilt* pour créer un filtre RIF.

Grâce au tableau ci-dessous, on peut voir qu'augmenter l'ordre du filtre rend la courbe plus raide donc permet de supprimer les harmoniques non souhaitées.

On remarque aussi qu'ajuster la fréquence de coupure permet un meilleur contrôle des harmoniques.

Et enfin, utiliser un filtre RII donne un meilleur effet de résonance.

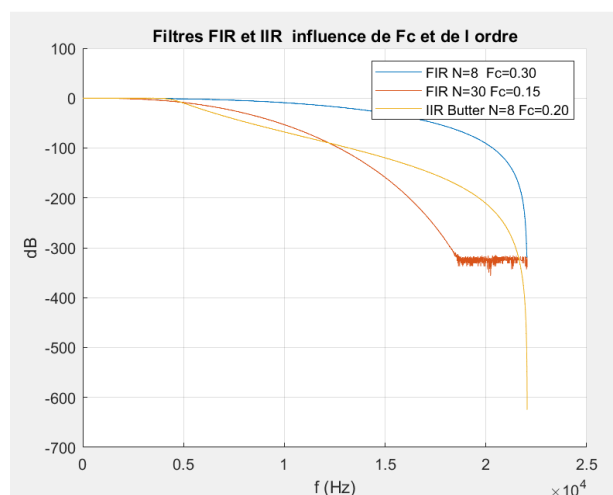


FIGURE 14 – Influence de f_c et de l'ordre

4 effets audio-numériques

4.1 effets de réverbération (3.1-3.8)

! dans votre script partie 3. ~ dans la section relative à Q 3.1, vous y faites des choses incompréhensibles ?

$$R_{yx} = h * R_{xx} \quad (1)$$

En particulier, si $R_{xx} \approx d$, on a $R_{yx} \approx h$ donc on peut estimer h en calculant différentes valeurs de R_{yx} .

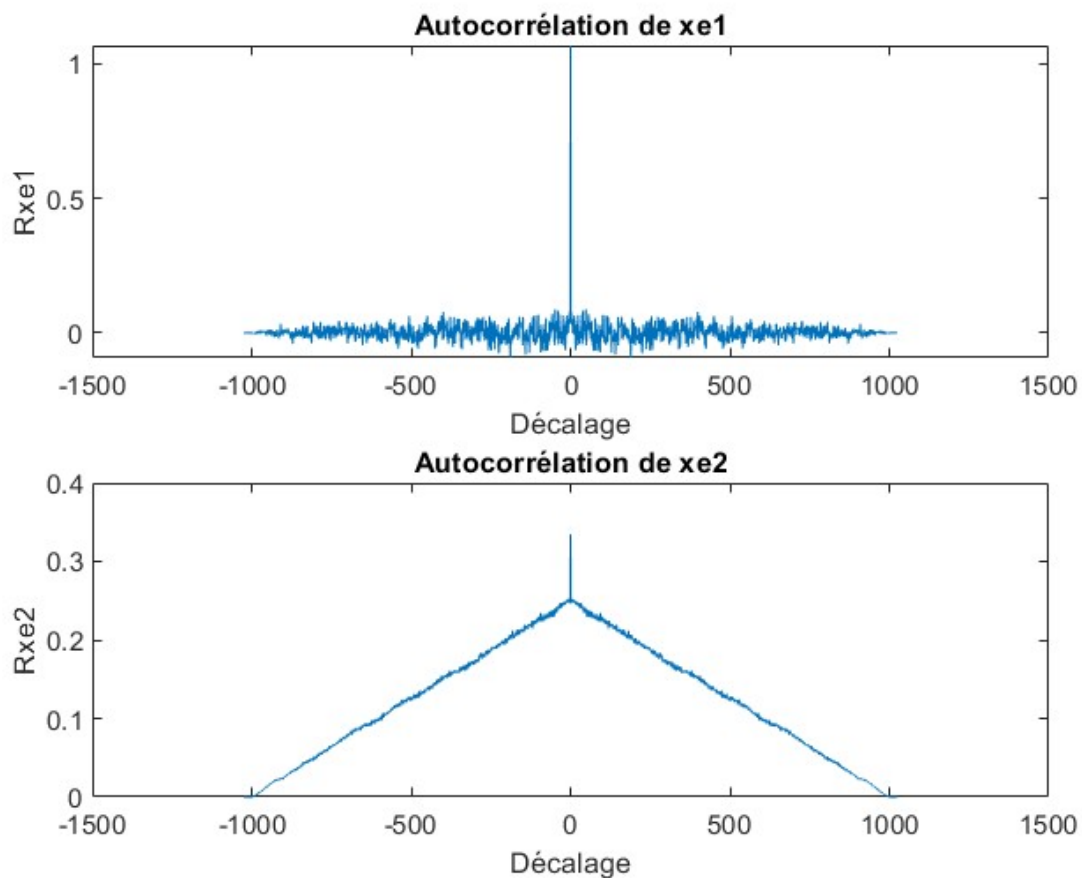


FIGURE 15 – Autocorrélation des signaux xe

On constate que l'autocorrélation de xe1 est bien plus proche d'un échelon unité que celle de xe2. C'est donc xe1 qu'il faut choisir pour estimer la réponse impulsionnelle. Voici ce que nous obtenons pour l'estimation de la réponse impulsionnelle :

l'utilisation de votre paramètre "leg" est strange dans votre script : pour quelle raison avez-vous fait ça ?

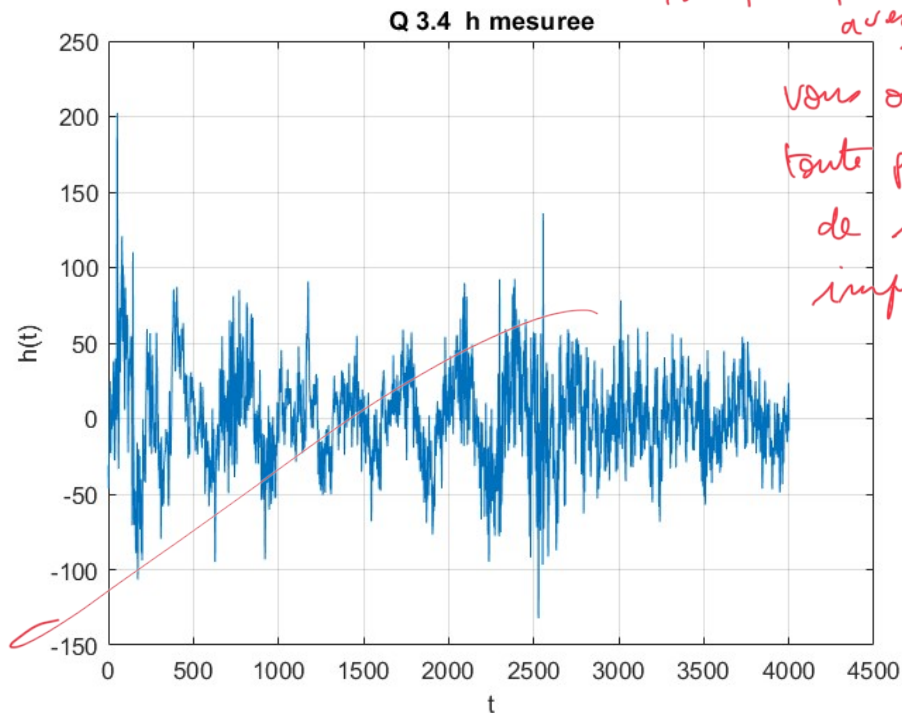


FIGURE 16 – Réponse impulsionnelle

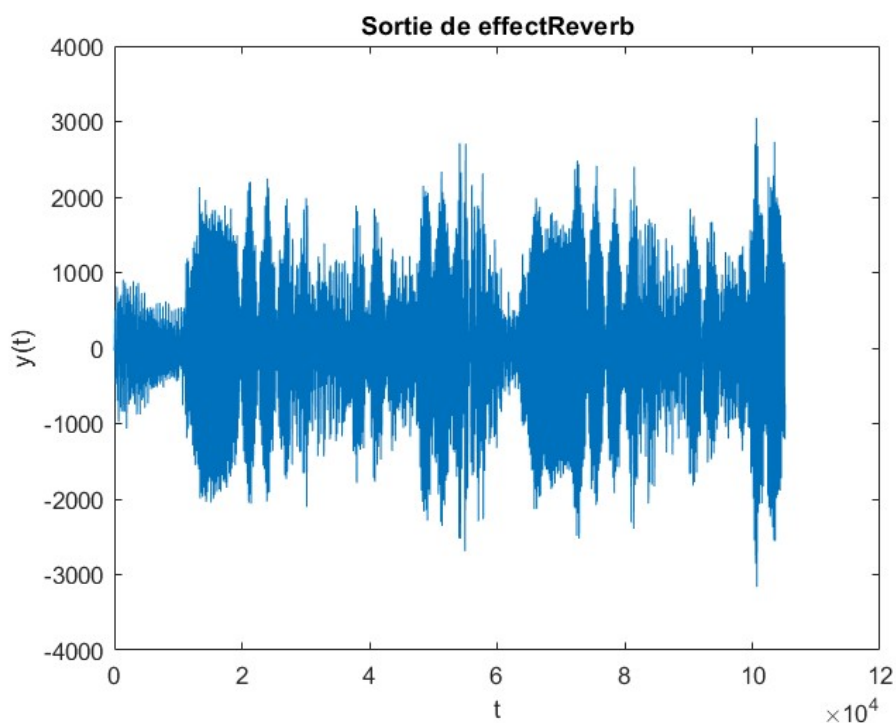


FIGURE 17 – Réponse d'un sample de guitare avec effectreverb

Le temps de calcul nécessaire à l'exécution de la fonction effectreverb sur un son de guitare est 0.021s.

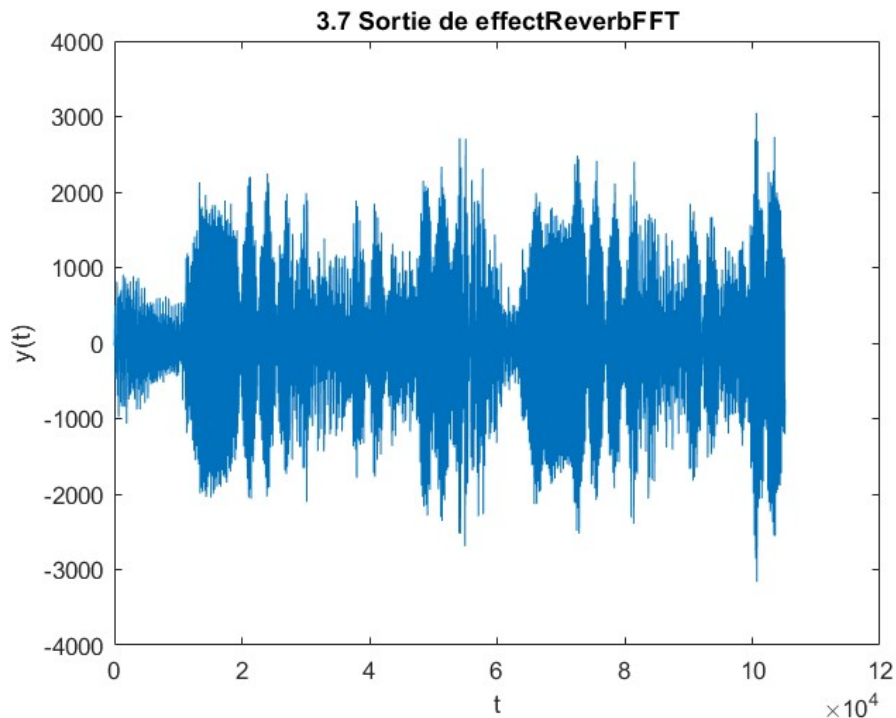


FIGURE 18 – Réponse d'un sample de guitare avec effectreverbFFT

Le temps de calcul nécessaire à l'exécution de la fonction effectreverbFFT sur un son de guitare est 0.003s. C'est sept fois moins long qu'avec une convolution classique.

Une FFT de taille N calcule en fait une convolution circulaire de longueur N .

Pour retrouver la convolution linéaire (de longueur $L_h + L_x - 1$), il faut zero-pad h et x à une même taille

$$N \geq L_h + L_x - 1$$

Si on oublie ce padding, on aura de la récurrence (« aliasing ») sur les indices, et le résultat ne sera pas celui de la convolution linéaire souhaitée.

4.2 effet de retard (3.9-3.19)

3.9 – Réponse impulsionnelle théorique du système à délai

On sait que le système à délai est définie par :

$$y(k) := x(k) - g \cdot y(k - \tau) \quad (2)$$

On considère un système caractérisé par deux paramètres : τ , un entier strictement positif représentant un retard exprimé en nombre d'échantillons, et g , un coefficient réel d'atténuation.

On détermine la **réponse impulsionnelle** $h(k)$, c'est-à-dire la sortie $y(k)$ du système lorsque l'entrée est une impulsion unité.

$$x(k) = \delta(k)$$

On résout :

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

On fait une Induction simple :

► Pour $k = 0$:

$$h(0) = 1$$

► Pour $k = \tau$:

$$h(\tau) = -g$$

► Pour $k = 2\tau$:

$$h(2\tau) = g^2$$

► Pour $k = 3\tau$:

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

► Par induction simple on obtient :

$$h(k) = (-g)^{\lfloor k/\tau \rfloor} \cdot \delta_{k \bmod \tau = 0}$$

où $\delta_{k \bmod \tau = 0}$ vaut 1 si k est un multiple de τ , et 0 sinon.

Ce système génère une réponse impulsionnelle constituée de valeurs non nulles espacées régulièrement de τ échantillons. L'amplitude de ces valeurs diminue selon une loi exponentielle en $-g$, traduisant un effet de réverbération simulée, où chaque écho est atténué de manière répétée après un intervalle constant.

3.10 – Condition de stabilité du filtre de delay

La réponse impulsionnel du systeme suivant :

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

peut s'écrire sous la forme :

$$h(k) = (-g)^{\lfloor k/\tau \rfloor} \cdot \delta_{k \bmod \tau = 0}$$


où $\delta_{k \bmod \tau = 0}$ vaut 1 si k est un multiple de τ , et 0 sinon.

Pour que le filtre soit stable pour toute entrée bornée (i.e. la sortie reste également bornée), il faut que la série suivante converge i.e. :

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

Cela revient à étudier la convergence de la série :

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

Cette série géométrique est convergente ssi $|g| < 1$. 

3.11 – Expression des coefficients du filtre à retard

La sortie du système :

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

peut se réarranger sous la forme classique d'un filtre récursif LTI :

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

Cette forme met en évidence les coefficients des polynômes du filtre utilisés dans MATLAB via `filter(b, a, x)`.

Les coefficients du vecteur b : La partie d'entrée correspond simplement à :

$$b = [1]$$

Les coefficients du vecteur a : Le polynôme associé à la sortie inclut les termes :

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

Ce filtre modélise un délai avec rétroaction, où la sortie est influencée par une version retardée et atténuée du signal.

3.12 – Simulation de la réponse impulsionnelle du filtre à retard

Pour étudier le comportement du filtre à retard, on simule sa réponse impulsionnelle en appliquant une impulsion unité en entrée.

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

La simulation utilise la fonction `filter` avec une entrée impulsionnelle $x(k) = \delta(k)$.

- $g = -0.6$ valeur négative
- $\tau = 2000$ échantillons, soit environ 45,4 ms à $F_e = 44100$ Hz
- Fréquence d'échantillonnage $F_e = 44100$ Hz

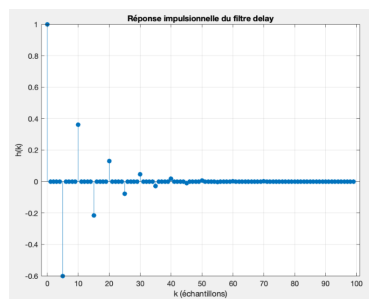


FIGURE 19 – réponse impulsionnelle avec l'effet à délai

Observations : On remarque une série d'impulsions espacées de τ échantillons, dont l'amplitude suit la loi $(-g)^n$. L'alternance des signes reflète la valeur négative de g , tandis que la décroissance exponentielle de l'amplitude confirme la modélisation théorique.

CC : La simulation confirme l'expression analytique de la réponse impulsionnelle vue en question 3.9 :

$$h(k) = (-g)^{\lfloor k/\tau \rfloor} \cdot \delta_{k \bmod \tau = 0}$$

où $\delta_{k \bmod \tau = 0}$ vaut 1 si k est un multiple de τ , et 0 sinon.

3.13 Analyse fréquentielle du filtre à retard

En appliquant la transformée de Fourier discrète aux deux membres, et en utilisant les propriétés de linéarité ainsi que le décalage dans le temps sur l'équation :

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

on obtient :

$$Y(e^{j2\pi\nu}) = X(e^{j2\pi\nu}) - g Y(e^{j2\pi\nu})e^{-j2\pi\nu\tau}$$

En isolant $Y(e^{j2\pi\nu})$, la relation devient :

$$Y(e^{j2\pi\nu})(1 + ge^{-j2\pi\nu\tau}) = X(e^{j2\pi\nu})$$

Le rapport des sorties aux entrées en fréquence, soit la fonction de transfert, est donc donné par :

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

Le module de cette réponse fréquentielle s'écrit :

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

Les pics d'amplitude apparaissent lorsque le dénominateur est minimal, ce qui correspond à :

$$\cos(2\pi\nu\tau) = -1 \quad \Rightarrow \quad 2\pi\nu\tau = (2k + 1)\pi \quad \Rightarrow \quad \nu_k = \frac{2k + 1}{2\tau}, \quad k \in \mathbb{Z}$$

Pour ces fréquences, l'amplitude maximale est :

$$|H(e^{j2\pi\nu_k})| = \frac{1}{1 - g}$$

L'argument complexe est donné par :

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

la réponse fréquentielle du filtre est entièrement caractérisée par le gain g , le retard τ , et la fréquence normalisée ν .

CC : La fonction de transfert correspond à un *filtre en peigne*, présentant un motif périodique dans le domaine fréquentiel. Ce filtre amplifie certaines fréquences spécifiques (ν_k) tout en atténuant d'autres, ce qui est typique du comportement des systèmes à retard.

3.14 réponse fréquentielle analytique vs estimation numérique

On veut évaluer la cohérence entre la réponse en fréquence calculée analytiquement à partir de la transformée de Fourier du filtre à retard et celle estimée numériquement via la DFT appliquée à une réponse impulsionnelle simulée.

On a utilisé la configuration suivante :

- Gain du filtre : $g = 0,8$
- Retard : $\tau = 441$ échantillons (correspondant à 10 ms pour une fréquence d'échantillonnage de $F_e = 44100$ Hz)
- Longueur de la DFT : $N = 8192$

La figure 20 ci-dessous illustre la comparaison entre le module de la réponse en fréquence théorique et celui obtenu numériquement.

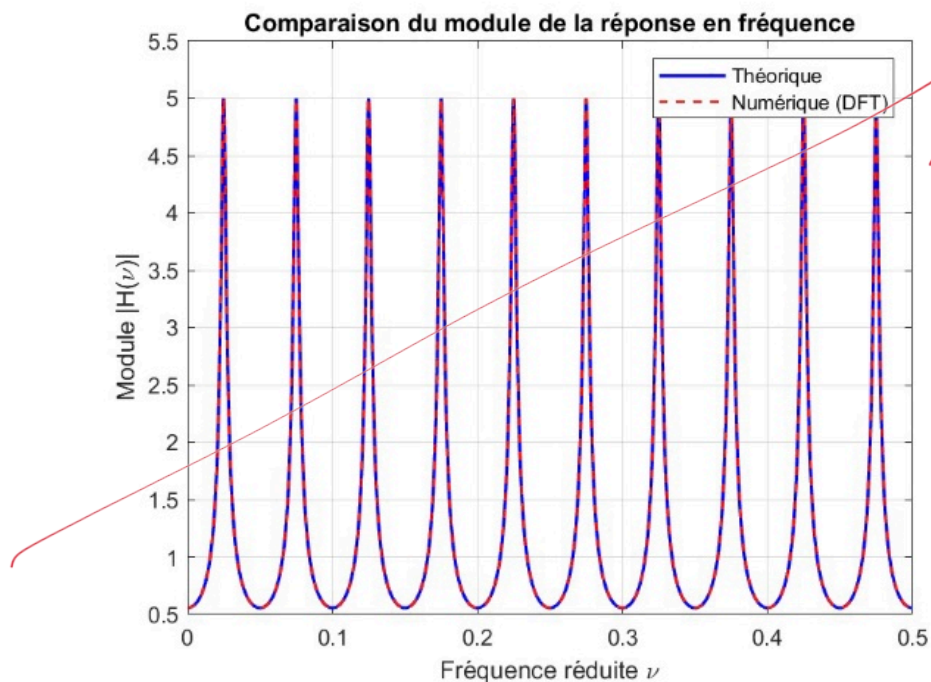


FIGURE 20 – comparaison entre le module analytique et estimé DFT de la réponse en fréquence du filtre à retard.

La formule analytique du module s'écrit :

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

Ce comportement typique d'un filtre en peigne se traduit par des maxima périodiques, localisés aux fréquences :

$$\nu_k = \frac{2k+1}{2\tau}, \quad \text{lorsque} \quad \cos(2\pi\nu\tau) = -1$$

Pour ces valeurs, l'amplitude atteint un maximum :

$$|\hat{h}(\nu_k)| = \frac{1}{1-g} = \frac{1}{0,2} = 5$$

CC : On retrouve sur la courbe numérique (DFT) les mêmes pics que sur la courbe théorique, aux mêmes fréquences et avec la même amplitude maximale : 5 .

Cela confirme que la simulation suit bien le comportement attendu. Les petites différences visibles entre les deux courbes viennent du fait que la réponse impulsionnelle utilisée pour le calcul numérique est tronquée (i.e. de durée finie), ce qui peut légèrement lisser certains pics.

3.15 Réalisation d'un delay simple par filtrage récursif (IIR)

Le but est de mettre en œuvre un effet de delay simple à l'aide d'un filtre IIR, selon l'équation suivante :

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

Ce filtre introduit un retour contrôlé : la sortie actuelle dépend du signal d'entrée mais aussi d'une version retardée de la sortie, atténuée par un gain g . Cela produit une série d'échos espacés, avec une intensité qui diminue progressivement.

On a codé une fonction qui prend comme arguments le signal, la durée du delay en secondes, le coefficient g , et la fréquence d'échantillonnage F_e cette fonction nous a permis d'appliquer cet effet à certains types de signaux

Ce traitement est souvent utilisé dans les effets sonores pour simuler un espace clos . Toutefois, il ne reproduit pas fidèlement les phénomènes acoustiques complexes (réverbération naturelle, diffraction ou encore absorption).

3.16 Vérification expérimentale du delay simple

Il s'agit ici de tester la fonction qu'on a codé en 3.15 sur un signal audio correspondant à un accord de piano. Les paramètres retenus pour cette expérience sont :

- Coefficient d'atténuation : $g = 0,9$
- Délai : $\tau = 0,25 \cdot F_e = 11025$ échantillons, avec une fréquence d'échantillonnage $F_e = 44100$ Hz

Le fichier `single_tone_piano1.wav` a été importé et traité avec l'effet de delay implémenté. Après écoute, le résultat a été exporté sous le nom `q.16piano1_delay.wav`.

À l'écoute, on perçoit clairement une série d'échos espacés de 250 ms. Ces répétitions s'atténuent progressivement, ce qui est cohérent avec un facteur g proche de 1. L'effet est net, sans coloration du son ni modification du contenu fréquentiel : il s'agit bien d'un delay simple, au rendu régulier et synthétique.

Le traitement a bien été appliqué. Le rendu audio est conforme au comportement attendu pour ce type de filtre. La durée du fichier reste inchangée d'environ 5 secondes, ce qui confirme que seul le contenu interne du signal a été modifié. Le code ayant permis ce test figure en annexe.

3.17 Delay avec filtrage dans la boucle de rétroaction

4.2.0.1 Objectif : L'idée est de simuler une atténuation plus réaliste des hautes fréquences, comme dans un espace réel.

On utilise un filtre passe-bas simple, sous la forme d'une moyenne glissante de longueur K . Sa réponse impulsionnelle est donnée par :

$$h_r(k) = \frac{1}{K} \cdot \mathbf{1}_{[0, K)}(k)$$

où $\mathbf{1}_{[0, K)}(k)$ vaut 1 si $0 \leq k < K$, et 0 sinon.

L'équation du filtre récursif devient :

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

Ce modèle introduit un retour atténué et filtré, ce qui permet de lisser les répétitions dans le temps et de mieux contrôler la richesse fréquentielle du signal traité.

La fonction `effectDelayFiltre` applique cette formule sur un signal mono ou stéréo. Elle prend en entrée : le signal, la durée du delay, le coefficient g , la taille de la fenêtre K et la fréquence d'échantillonnage. Le code source est inclus en annexe.

Ce type d'effet donne un résultat plus progressif que le delay simple. Le filtrage dans la boucle de rétroaction permet d'obtenir des échos moins marqués, ce qui rapproche le rendu d'une réverbération naturelle.

*il faut utiliser la
fo filter avec les
vecteurs a et b prenant
en compte le nouveau
filtre moyenné.
(la boucle for et
source d'input)*

3.18 Test de l'effet de delay avec filtrage

L'objectif est d'évaluer la fonction `effectDelayFiltre`, développée à la question précédente, en l'appliquant au même signal audio que précédemment. Les paramètres du test sont :

- $\tau = 0,25 \cdot F_e$
- $g = 0,9$
- $K = 10$

Le fichier `single_tone_piano1.wav` a été traité à l'aide de la fonction `effectDelayFiltre`. Le signal obtenu a été écouté, puis enregistré sous le nom `piano_delay_filtre.wav`.

Par rapport à la version sans filtrage, l'effet obtenu est plus claire. Les répétitions espacées de 250 ms restent audibles, mais leur attaque est plus arrondie, avec une décroissance plus naturelle. L'atténuation progressive des hautes fréquences rend l'effet plus réaliste et moins métallique.

Le filtrage intégré dans la boucle de retour permet d'améliorer la qualité sonore de l'effet de delay. Le traitement produit est en accord avec les attentes, et la fonction `effectDelayFiltre` est considérée comme validée. Le code et le fichier audio correspondant sont fournis en annexe.

3.19 Justification de l'atténuation des hautes fréquences par le filtre

$$h_r(k)$$

Lors de l'écoute du signal traité par la fonction `effectDelayFiltre`, on remarque que les répétitions deviennent de plus en plus sourdes. Les hautes fréquences s'atténuent progressivement, ce qui donne à l'effet une sonorité plus naturelle par rapport au delay simple.

5. Conclusion

Le filtre $h_r(k)$ utilisé dans la boucle de retour est une moyenne glissante de longueur K . Sa réponse impulsionnelle est définie comme :

$$h_r(k) = \frac{1}{K} \cdot \mathbf{1}_{[0, K)}(k)$$

où $\mathbf{1}_{[0, K)}(k)$ vaut 1 si $0 \leq k < K$, et 0 sinon.

Ce filtre est un filtre RIF passe-bas. En effectuant une moyenne temporelle, il agit comme un lissage du signal, ce qui a pour conséquence de **réduire les hautes fréquences**.

Pour vérifier ce comportement, nous avons calculé et tracé la réponse en fréquence du filtre via une FFT. Le module de cette réponse est illustré dans la figure 21 ci-dessous.

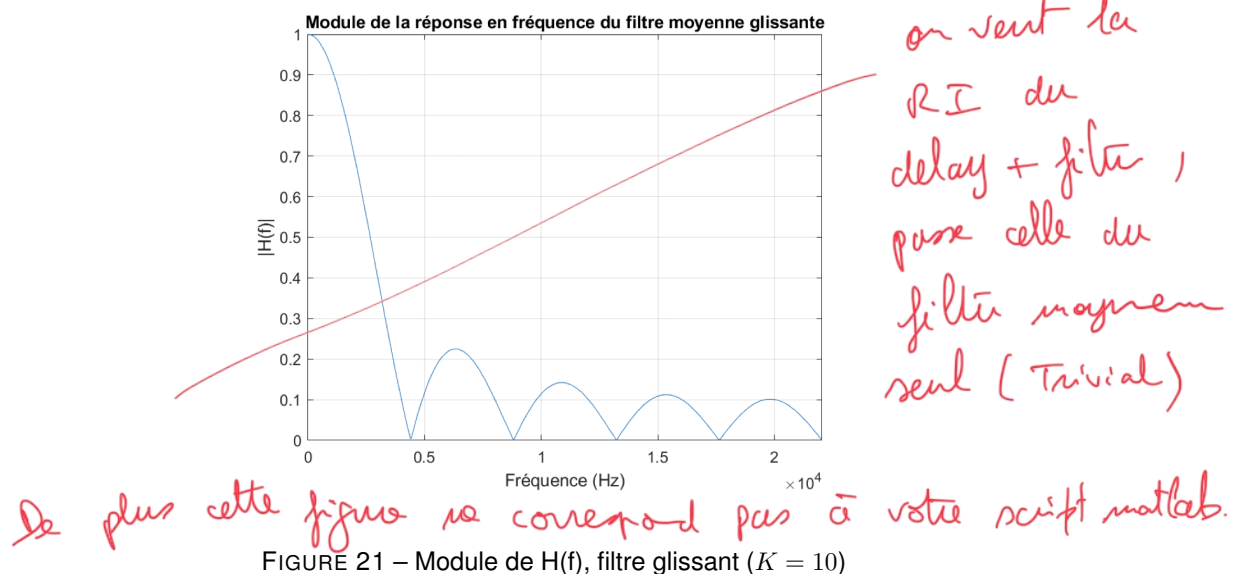


FIGURE 21 – Module de $H(f)$, filtre glissant ($K = 10$)

On observe une forte atténuation à partir de quelques kHz, ce qui confirme que le filtre coupe efficacement les hautes fréquences. Les oscillations visibles sont dues à la nature rectangulaire de la fenêtre temporelle phénomène de Gibbs, mais l'atténuation globale est nette. Le comportement observé est donc en parfaite adéquation avec la sensation d'amortissement sonore perçue à l'écoute.

CC : Le filtrage introduit dans la boucle de feedback agit bien comme un filtre passe-bas. Cela explique la perte progressive des aigus dans les répétitions, rendant l'effet de delay plus réaliste. Le traitement numérique confirme parfaitement les observations .

5 Conclusion

Ce projet nous a permis d'appliquer concrètement les notions de traitement du signal vues en cours. À travers les trois parties abordées, nous avons manipulé des signaux audio, observé leurs spectres et implémenté divers effets numériques.

Les résultats obtenus confirment l'intérêt de Matlab pour simuler et analyser des systèmes audio.

5. Conclusion

Ce travail nous a aussi permis de mieux comprendre le lien entre théorie mathématique et perception sonore. Enfin, la démarche expérimentale adoptée a renforcé notre compréhension des outils fréquentiels et temporels.