

IMT Atlantique

Technopôle de Brest-Iroise - CS 83818

29238 Brest Cedex 3

Téléphone : +33 (0)2 29 00 13 04

Télécopie : +33 (0)2 29 00 10 12

URL : www.imt-atlantique.fr

**Compte rendu**

SAR Traitements audio : partie signal

Encadrants :

GUILLOUD Frédéric

SOCHELEAU François-Xavier

Etudiants :

RABIER Valentin

BAYLE SEI

Date d'édition : 21 mai 2025

**IMT Atlantique**

Bretagne-Pays de la Loire

École Mines-Télécom

Sommaire

1. Introduction	2
2. Partie 1 : Synthèse additive	3
2.1. Analyse d'un son harmonique	3
2.2. Synthèse	7
3. Partie 2 : Synthèse soustractive	9
4. Partie 3 : Effets audio-numériques	13
4.1. Effets de réverbération	13
4.1.1. Mesure de réponse impulsionnelle	13
4.1.2. Convolution classique	14
4.1.3. Convolution rapide	15
4.2. Effet de retard	15
5. Pour aller plus loin : synthèse Karplus-Strong	20
5.1. Analyse de l'algorithme de Karplus-Strong	20
5.2. Formule de récurrence	20
5.3. Test avec un bruit blanc	21
5.4. Conclusion	21

1. Introduction

Ce projet de traitement audio a pour objectif d'explorer et de mettre en pratique plusieurs techniques fondamentales du traitement du signal appliqué à l'audio. À travers différentes étapes, nous analyserons et modifierons des sons numériques en utilisant des outils mathématiques tels que la transformée de Fourier discrète (FFT), des filtres numériques, ainsi que des modèles de synthèse sonore.

Le projet se divise en plusieurs parties : l'analyse fréquentielle de sons réels pour en extraire les composantes harmoniques ; la synthèse additive et soustractive pour générer des sons à partir de modèles mathématiques ; l'implémentation d'effets audio numériques comme la réverbération ou le délai ; et enfin l'étude d'un modèle de synthèse avancé, Karplus-Strong, inspiré des cordes pincées.

Tous ces travaux sont réalisés dans l'environnement Matlab, afin de mieux comprendre les propriétés des signaux sonores, les manipulations possibles en numérique, et leurs impacts perceptifs dans un contexte musical ou acoustique. Ce projet vise ainsi à développer une compréhension concrète du lien entre les outils du traitement du signal et les sons que nous percevons.

2. Partie 1 : Synthèse additive

Dans cette première partie, nous analysons le contenu fréquentiel de sons d'instruments à l'aide de la transformée de Fourier discrète. Cette étude permet de visualiser le spectre d'amplitude, d'identifier la fréquence fondamentale et de comparer les caractéristiques harmoniques d'instruments à cordes avec d'autres types d'instruments.

Après cette courte analyse, nous essayerons ensuite de synthétiser ce son à l'aide de l'outil MATLAB.

2.1. Analyse d'un son harmonique

Nous avons à disposition différents sons d'instruments à corde, dont le son de piano que nous avons choisi pour cette première analyse [figure1] : voici ci dessous la transformée de fourier centrée d'un son de piano. On peut trouver en ordonnée l'amplitude en dB en fonction de la fréquence centrée autour de 0.

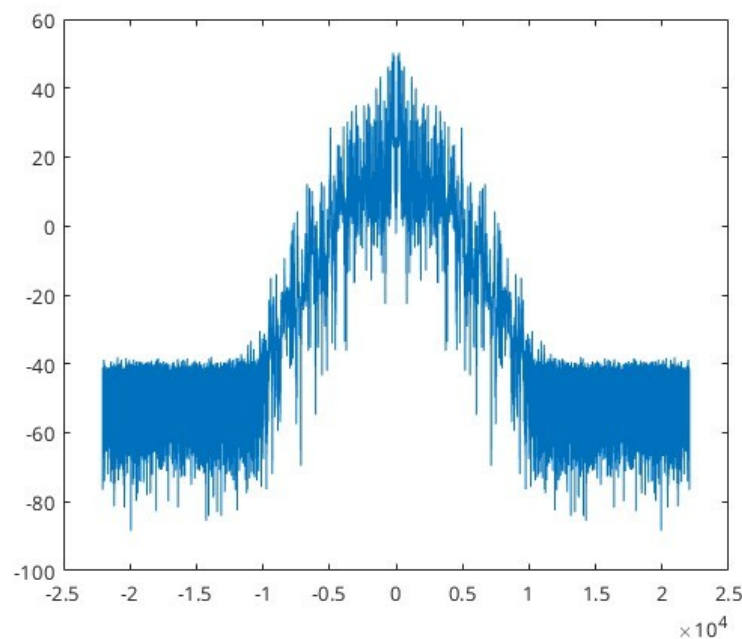


FIGURE 1 – FTT de piano 1

Le spectre présente une structure symétrique autour de zéro, ce qui est attendu pour le spectre d'un signal réel. On observe un pic central important, qui correspond à la fréquence fondamentale du son de piano. Autour de cette fréquence, plusieurs bandes visibles indiquent la présence d'harmoniques, typiques des sons musicaux riches tels que celui d'un piano.

Autour de la bande central du spectre, on observe un niveau de bruit plus faible, relativement constant, qui peut provenir de bruits ambiants, de l'instrument lui-même ou de limitations du système d'enregistrement.

Cette analyse permet donc d'illustrer la richesse fréquentielle d'un son de piano et met en évidence ses composantes principales ainsi que le bruit de fond.

Nous pouvons faire de même pour un nouvel instrument à corde : le violon. Nous obtenons la transformée de Fourier ci-dessous [figure2].

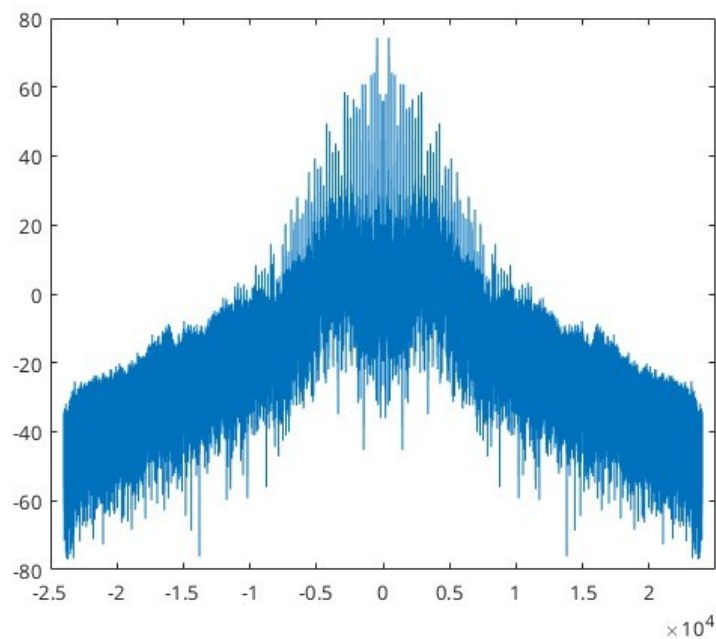


FIGURE 2 – FTT de violin.a3

Les spectres en fréquence centrés des sons de piano et de violon présentent des caractéristiques communes, mais aussi des différences qui reflètent la nature propre de chaque instrument.

Le spectre du violon, comparé à celui du piano, est plus dense et continu, avec une répartition énergétique plus large autour du centre. On observe une forme plus en pente, sans creux aussi nets, ce qui pourrait représenter la nature soutenue du son du violon, produit par les frottements de la baguette. Les harmoniques sont toujours présentes, mais moins isolées que pour le piano, traduisant un timbre plus homogène.

Finalement, le spectre du piano est plus marqué par des pics bien distincts, tandis que celui du violon est plus enveloppant et régulier, représentant la sonorité continue et expressive du son à l'archet.

La relation harmonique idéale des instruments à cordes s'exprime par :

$$f_n = n \cdot f_1$$

où f_1 est la fréquence fondamentale et n un entier naturel. Toutefois, en pratique, l'inharmonicité est présente et peut être quantifiée en cents par la formule

$$\xi = 1200 \times (\log_2(\hat{f}_n) - \log_2(n f_1))$$

On se propose donc d'étudier l'inharmonicité sur 2 sons de piano différent, piano.1 et piano.2, et de retrouver à partir de cette étude lequel est le plus harmonieux. Pour utiliser la formule de l'inharmonicité sur les 2 sons de piano, il nous faut trouver les fréquences fondamentales des deux pianos. Pour cela, nous allons nous pencher de plus près sur les ftt obtenues pour les deux notes, trouver le premier pic significatif d'amplitude et reporter la fréquence à laquelle ce pic occure.

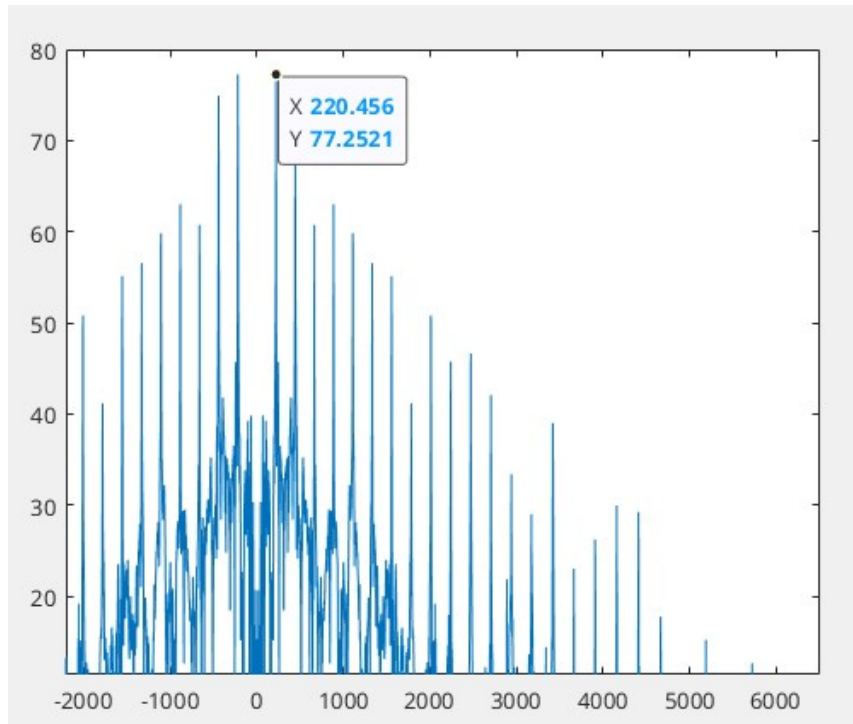


FIGURE 3 – Fe de piano.1

Voici le résultat obtenue pour la note piano.1 [figure 3]. On retrouve une fréquence fondamentale d'environ 220 Hz. En raisonnant de la meme manière sur le second son, nous trouvons une fréquence fondamentale semblable de 218 Hz.

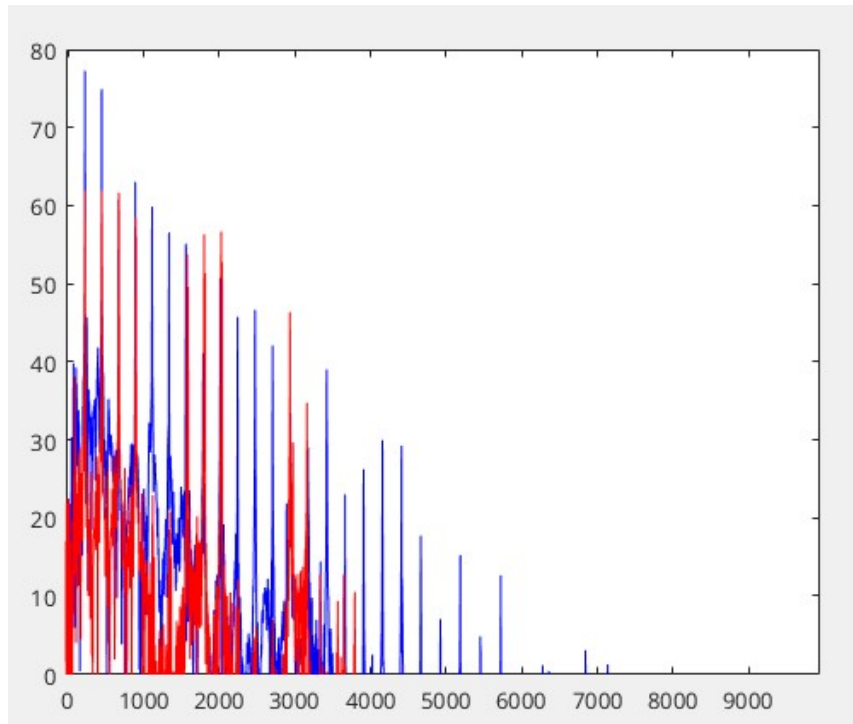


FIGURE 4 – Superposition du spectre piano.1 (bleu) et piano.2(rouge)

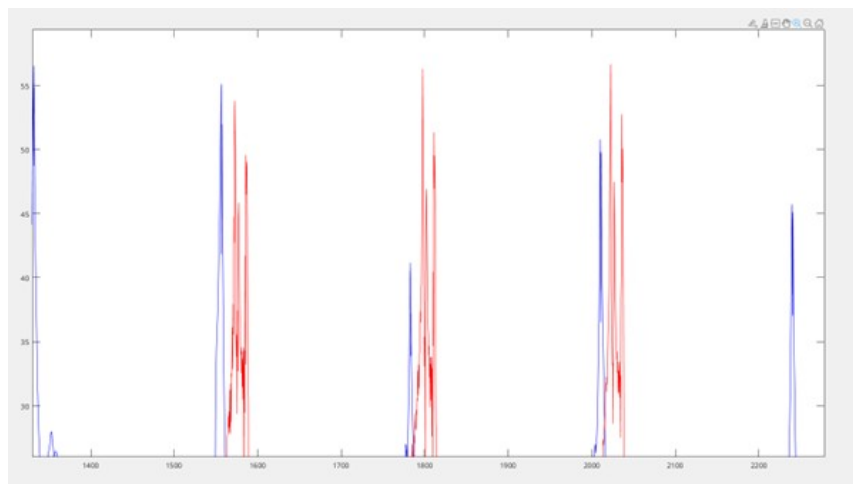


FIGURE 5 – Différence de d'unicité des harmoniques

En superposant les deux transformées de Fourier, une première étude qualitative nous donne l'intuition que le piano.1, en bleu, est plus harmonieux que le second. En effet, toutes ses harmoniques sont présentes et intenses, mais surtout, quand nous regardons de plus près, nous remarquons que lorsque les harmoniques du piano.1 sont uniques, celles du piano.2 sont souvent multiples, ce qui entraîne une superposition de fréquences proches, générant des battements et une sensation de flou acoustique.

Ce phénomène peut être interprété comme un signe d'inharmonicité accrue chez le piano.2. Ce désaccord subtil entre les fréquences partiels peut affecter la perception globale du timbre, rendant le son du piano.2 plus métallique, tandis que le piano.1 conserve une plus grande clarté.

Une étude quantitative à l'aide de la notion d'inharmonicité peut confirmer cette première intuition :

Piano.1 (Bleu)(Hz)	f_2	f_3	f_4	f_5	f_6	f_7
Mesurées	442	663	886	1108	1332	1556
Théoriques	440	660	880	1100	1320	1540
Inharmonicité	7,85	7,85	11,76	12,54	15,67	17,89

Piano.2 (Rouge)(Hz)	f_2	f_3	f_4	f_5	f_6	f_7
Mesurées	443	668	893	1118	1347	1572
Théoriques	436	654	872	1090	1308	1526
Inharmonicité	27,60	36,67	41,19	43,91	50,86	51,42

A la vue des inharmonicités obtenues, on constate une différence nette entre les deux instruments. Le piano.1 présente des valeurs d'inharmonicité relativement faibles et croissantes de manière modérée. En revanche, le piano.2 affiche des inharmonicités bien plus élevées, dès les premières harmoniques, avec une progression rapide, traduisant un écart significatif par rapport aux fréquences théoriques attendues.

Ces écarts plus importants pourrait s'expliquer par des facteurs mécaniques comme la tension ou la rigidité des cordes, ou encore l'usure de certains éléments du piano. Quoi qu'il en soit, ces valeurs confirment objectivement l'intuition formulée plus tôt : le piano.1 est plus harmonieux que le piano.2, tant sur le plan perceptif que sur le plan mathématique.

2.2. Synthèse

Grâce à notre analyse en première partie, nous avons montré que le son du piano.1 était plus harmonique que le second. L'objectif de cette sous-partie est de synthétiser ce son. Pour cela nous commençons par récupérer les amplitudes des harmoniques (nous nous limiterons aux 8 premières) et générons un son à l'aide de matlab en superposant les ondes sinusoïdales aux fréquences et amplitudes correspondantes. Voici par exemple une mesure sur la première fréquence :

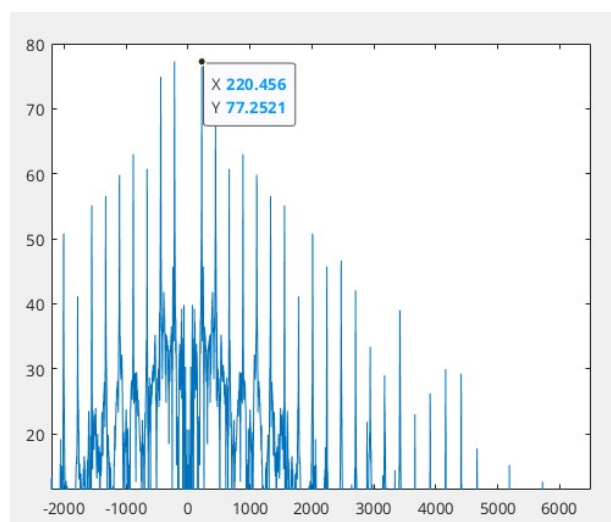


FIGURE 6 – Mesure des amplitudes des fondamentales du piano.1

Nous relevons les 8 premières amplitudes suivantes :

Harmonique	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8
Amplitude	77.2	74.9	60.7	63.0	59.8	56.5	55.1	41.1

Nous obtenons un premier son à partir de ces données très uniformes, plutôt éloigné de ce à quoi nous nous attendions. En effet, cette première synthèse repose sur une enveloppe temporelle constante (amplitude fixe), ce qui donne un son très régulier et artificiel.

Dans un second temps, nous appliquons sur ce son une enveloppe ADSR (Attack, Decay, Sustain et Release) qui permet d'obtenir un son plus proche de la réalité et plus proche du son d'origine piano.¹ Ce type d'enveloppe permet de moduler l'amplitude du signal au cours du temps, de manière à simuler la manière dont un piano "réagit" lorsqu'une touche est frappée : une montée rapide de l'intensité (attaque), suivie d'un déclin rapide (decay).

L'ajout de cette enveloppe a significativement amélioré le réalisme du son synthétique, en reproduisant le comportement dynamique du piano. On perçoit mieux l'attaque caractéristique de la note et la décroissance naturelle de l'énergie sonore.

Enfin, en partant des mêmes fréquences et amplitudes, nous essayons une dernière synthèse en passant par la transformée de Fourier discrète inverse.

Le son obtenu est proche de celui généré par la méthode directe avec superposition de sinusoides, bien que de légères différences puissent être perçues en raison de la discrétisation du spectre et de l'échantillonnage. En particulier, selon la manière dont la phase est gérée dans la transformée de Fourier discrète inverse, des interférences subtiles peuvent apparaître.

En conclusion, la synthèse directe par superposition d'ondes et la synthèse par transformée de Fourier discrète inverse donnent des résultats similaires dans ce cas simple, surtout si les composantes sont bien définies. Toutefois, l'ajout de l'enveloppe ADSR reste indispensable pour atteindre un résultat convaincant et musicalement réaliste.

3. Partie 2 : Synthèse soustractive

Nous allons maintenant aborder une approche complémentaire : la synthèse soustractive. Contrairement à la synthèse additive, cette méthode consiste à partir d'un signal périodique riche en harmoniques, comme une onde carrée ou une onde en dent de scie, puis à le filtrer pour sculpter le spectre du son souhaité. Ce procédé est à la base de nombreux synthétiseurs analogiques, notamment le célèbre TB-303, qui a fortement marqué l'univers de la musique électronique des années 1980.

Avant de filtrer ces signaux, il est essentiel de comprendre leur contenu spectral. Nous allons donc, dans un premier temps, calculer analytiquement la série de Fourier de deux signaux typiques – le signal carré et le signal en dent de scie – puis vérifier ces résultats numériquement à l'aide de la transformée de Fourier discrète (DFT). Cette analyse nous permettra d'identifier les composantes fréquentielles sur lesquelles agir par filtrage.

Pour commencer, il nous faut donc son développement en série de Fourier, soit trouver les b_n correspondant dans la série ci dessous :

$$s(t) = \sum_{n=1}^{\infty} b_n \sin\left(\frac{2\pi n t}{T}\right)$$

Les coefficients b_n sont donnés par :

$$b_n = \frac{2}{T} \int_0^T s(t) \sin\left(\frac{2\pi n t}{T}\right) dt$$

On peut montrer par symétrie et périodicité que ces coefficients deviennent :

$$b_n = \frac{4}{T} \int_0^{\frac{T}{2}} \sin\left(\frac{2\pi n t}{T}\right) dt$$

En calculant cette intégrale, nous obtenons :

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

Finalement :

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

Autrement dit, seules les harmoniques impaires restent présentes.

Pour vérifier ces résultats, on se propose d'utiliser une méthode numérique. Pour cela, nous souhaitons utiliser la transformée de Fourier sur ces deux signaux. Pour cela nous commençons par générer un signal carré et un signal en dent de scie, avant d'appliquer la transformée de Fourier sur ces deux derniers. En affichant le résultat obtenu à l'aide de Matlab, il en sort :

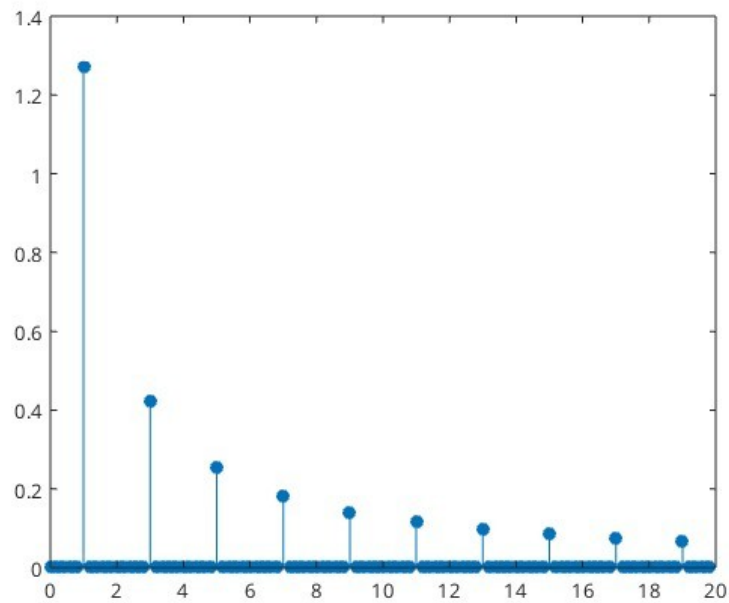


FIGURE 7 – TF du signal carré

Le problème de cette technique peut être le fenêtrage, lorsqu'il est mal adapté (comme une fenêtre rectangulaire), peut provoquer des discontinuités aux bords du signal si celui-ci n'est pas parfaitement périodique dans la fenêtre d'observation. Cela engendre un phénomène appelé leakage, où l'énergie d'un pic fréquentiel s'étale sur les fréquences voisines, rendant le spectre moins lisible et les pics moins nets.

On peut appliquer le même raisonnement sur le signal dent de scie.

On cherche les coefficients de la série de Fourier pour un signal impair défini sur une période T , ce qui implique qu'il n'y a que des sinus dans la décomposition (pas de cosinus ni de terme constant) :

$$b_n = \frac{2}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} s(t) \sin\left(\frac{2\pi nt}{T}\right) dt$$

Si $s(t) = \frac{2t}{T}$ (fonction linéaire sur la période), alors :

$$b_n = \frac{2}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} \frac{2t}{T} \sin\left(\frac{2\pi nt}{T}\right) dt$$

On pose :

$$u = t \quad \text{et} \quad dv = \sin\left(\frac{2\pi nt}{T}\right) dt$$

$$du = dt \quad \text{et} \quad v = -\frac{T}{2\pi n} \cos\left(\frac{2\pi nt}{T}\right)$$

Par intégration par parties :

$$\int t \sin\left(\frac{2\pi nt}{T}\right) dt = -\frac{T}{2\pi n} t \cos\left(\frac{2\pi nt}{T}\right) + \frac{T}{2\pi n} \int \cos\left(\frac{2\pi nt}{T}\right) dt$$

Après évaluation de l'intégrale sur l'intervalle $[-\frac{T}{2}, \frac{T}{2}]$, on obtient :

$$b_n = \frac{4}{n\pi}(-1)^{n+1}$$

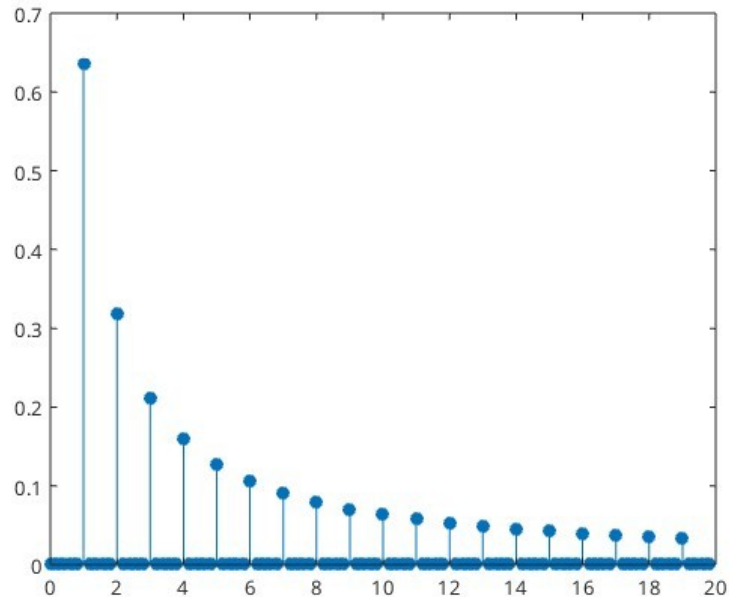


FIGURE 8 – TF du signal dent de scie

Dans le but final de réaliser une synthèses soustractive avec ces signaux, on cherche à contrôler leurs spectres qui sont très riches en harmoniques. On choisit d'introduire un filtre passe bas qui va laisser passer les basses fréquences (les harmoniques les plus graves) et atténue ou supprime les hautes fréquences (les harmoniques aiguës). Cela va permettre d'adoucir le timbre du son, en supprimant la brillance ou l'agressivité dues aux hautes harmoniques. On introduit le filtre passe bas d'ordre 1 suivant :

$$y(k) = 0,5(x(k) + x(k-1))$$

On se propose de calculer son spectre théorique de sortie :

$$\begin{aligned}
 |\hat{y}(f)|^2 &= \frac{1}{4} |\hat{x}(f) + \hat{x}(f-1)|^2 \\
 &= \frac{1}{4} |\hat{x}(f)|^2 |1 + e^{-2i\pi f}|^2 \\
 &= \frac{1}{4} |\hat{x}(f)|^2 |e^{+i\pi f} + e^{-i\pi f}|^2 \\
 &= \underline{|\hat{x}(f)|^2 \cos^2(\pi f/2)}
 \end{aligned}$$

FIGURE 9 – Calcul du spectre

À l'aide de la fonction **filter** de MatLab, on peut facilement vérifier ce résultat avec une simulation :

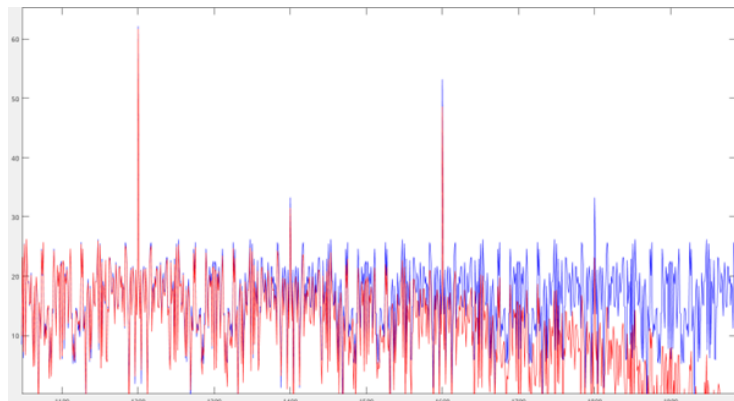


FIGURE 10 – Spectre de sortie d'un signal carré filtré

Ici, nous avons le gain en décibels du signal carré en bleu et le gain en décibels du signal carré filtré en rouge. Comme espéré, il s'agit bien d'un filtre passe bas, qui atténue fortement les fréquences aiguës et laisse passer les fréquences plus graves.

Nous avons ensuite exploré l'impact du signal source dans le cadre de la synthèse soustractive, en gardant constante l'enveloppe d'amplitude utilisée (ADSR).

À l'écoute, la synthèse additive produit un son relativement doux, mais limité en expressivité sonore car il est souvent restreint à un petit nombre d'harmoniques pour des raisons de calcul ou de clarté. En comparaison, la synthèse soustractive à partir d'un signal carré ou en dent de scie, combinée à un filtre adapté, permet d'obtenir des sons plus variés et plus expressifs, en modulant directement le contenu fréquentiel par le filtrage !

Afin d'améliorer le rendu sonore obtenu par synthèse soustractive, nous avons utilisé la fonction `designfilt` de Matlab pour créer des filtres passe-bas plus performants que le simple filtre d'ordre 1 précédemment utilisé. Par exemple, la commande suivante :

```
d = designfilt('lowpassfir','FilterOrder',2,'HalfPowerFrequency',0.25);
```

nous permet de spécifier un filtre RIF (Réponse Impulsionnelle Finie) d'ordre 2, avec une fréquence de coupure réduite à 0.25. À l'aide de la fonction `filterAnalyzer(d)`, nous avons pu analyser précisément la réponse fréquentielle du filtre et ajuster ses paramètres.

En jouant sur plusieurs aspects tel que la fréquence de coupure, l'ordre du filtre, et la nature du filtre (RIF ou RII), nous avons affiné la forme du spectre de sortie pour obtenir des timbres plus riches ou plus doux, selon le réglage choisi. Par exemple, un filtre d'ordre élevé permet une coupure plus nette des hautes fréquences, donnant un son plus rond et moins agressif, tandis qu'un filtre à coupure plus haute laisse davantage d'harmoniques et donc plus de brillance.

En conclusion, cette étape a mis en évidence le rôle fondamental du filtre dans la synthèse soustractive, qui permet à partir d'un signal de base très riche, de sculpter un timbre sonore selon les besoins. L'approche reste complémentaire à la synthèse additive, mais elle offre une plus grande souplesse créative, notamment en musique électronique.

4. Partie 3 : Effets audio-numériques

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

4.1. Effets de réverbération

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

Pour simuler les effets de réverbération, nous allons utiliser une approche par convolution en commençant par mesurer sa réponse impulsionnelle puis en la convoluant avec un autre son.

4.1.1. Mesure de réponse impulsionnelle

Soit $x(k)$ le signal d'excitation, $h(k)$ la réponse impulsionnelle et $y(k)$ le signal capté par un microphone, nous pouvons exprimer la fonction l'intercorrrelation $R_{yx}(u)$ en fonction de $h(k)$.

En supposant $R_{yx}(u) = d(u)$, on prouve ci dessous que la fonction l'intercorrrelation des signaux $x(k)$ et $y(k)$ conduisent directement à l'estimation de la réponse impulsionnelle.

3) On a : $y(k) = h(k) * x(k)$

Par définition : $R_{yx}(k) = y(k) * x^*(-k)$

$= h(k) * \underbrace{x(k) * x^*(-k)}_{R_{xx}(k)}$

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

2) Si $R_{xx}(u) \approx d(u)$

On a $R_{yx}(u) \approx h(u) * d(u)$

Soit $R_{yx}(u) \approx h(u)$

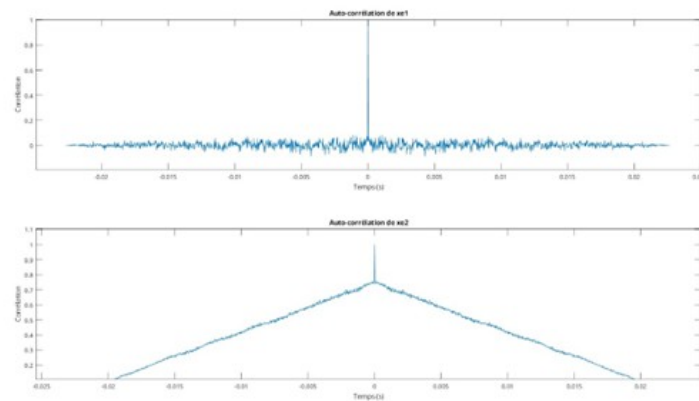
L'intercorrrelation nous donne directement la réponse impulsionnelle avec cette estimation.

FIGURE 11 – Réponses questions 3.1 et 3.2

Deux signaux $xe1$ et $xe2$ nous sont proposés : à l'aide de la fonction MatLab `xcorr`, nous pouvons retenir le signal plus adapté comme signal d'excitation pour mesurer la réponse impulsionnelle de la pièce.

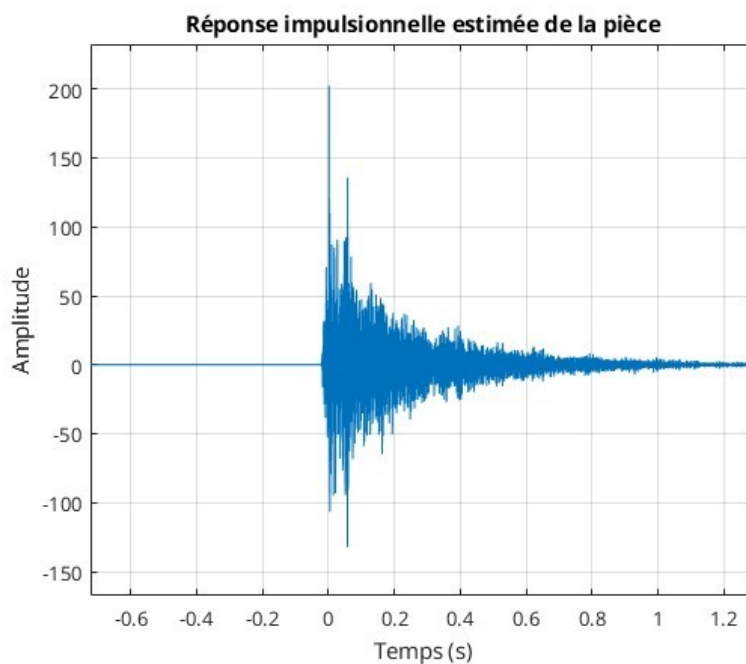
Pour déterminer lequel de ces deux signaux est le plus adapté, on calcule l'autocorrélation de chacun des signaux. Celui qui sera le plus adapté pour mesurer la réponse impulsionnelle est celui qui aura un pic unique et élevé. En effet, grâce à la question précédente, on a pu voir que si notre signal est un Dirac (donc

un seul pic très élevé), on obtient directement notre réponse impulsionnelle en calculant l'intercorrrelation. Ainsi, on peut tracer les autocorrélations des signaux x_{e1} et x_{e2} :



On en conclut alors que le signal x_{e1} est le plus adapté pour réaliser notre mesure.

Comme vu ci-dessus, en calculant l'intercorrrelation entre le signal d'entrée et le signal de sortie, affecté par la pièce, on obtient la réponse impulsionnelle de la pièce. Ci-dessous, nous avons sa représentation graphique. Nous avons donc un signal causal, qui s'atténue de manière exponentielle. On peut de plus observer des effets de réflexions et de réverbérations, qui proviennent des murs ou des plafonds. C'est donc plutôt cohérent dans notre cadre d'étude.



4.1.2. Convolution classique

L'application de l'effet de réverbération s'obtient en convoluant la réponse impulsionnelle estimée précédemment avec un signal source c-a-d $y(k) = x(k) * h(k)$. L'effet de réverbération pourra donc être implémentée sous Matlab via la fonction `filter`.

Après la création d'une fonction `effet-reverb`, nous pouvons tester celle ci sur un son de guitare. Nous obtenons effectivement le son de guitare modifié par les réflexions subies. Cependant, l'exécution de

cette fonction est relativement longue : 0.4045 secondes. Il est donc intéressant de raccourcir ce temps ce qui fait l'objet de la prochaine partie.

4.1.3. Convolution rapide

Pour diminuer ce temps de calcul, on se propose d'implémenter l'effet de réverbération dans le domaine fréquentiel en utilisant la FFT, en écrivant la sortie du filtre comme :

$$y_2(k) = \text{IFFT}(\text{FTT}(h(k)) \cdot \text{FTT}(x(k)))$$

Avec cette nouvelle méthode, nous trouvons maintenant un temps de calcul de 0.0665 secondes, soit un résultat six fois plus rapide ! Cette technique est très efficace dans notre contexte.

Les deux méthodes sont équivalentes d'un point de vue mathématiques, mais il faut veiller à bien respecter la taille des échantillons et bien définir la variable NFFT car sinon il peut y avoir des effets d'artefacts ou des tronquages. On remarque par ailleurs que cette méthode est beaucoup plus rapide que la convolution classique car la convolution classique calcule chaque terme donc a une complexité en $N \cdot M$ avec N et M la taille des deux échantillons, alors que la méthode IFFT a une complexité en $N \cdot \log(N)$.

4.2. Effet de retard

Les effets de retard (delay) sont très utilisés en musique. Ces effets permettent d'ajouter de l'espace en simulant les réflexions des ondes acoustiques dans un espace clos. Pour les simuler, nous utiliserons la relation de récurrence suivante :

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

En appliquant une impulsion unitaire en entrée, on obtient alors la réponse impulsionnelle en sortie, d'où :

la relation de récurrence conduit à :

$$\begin{aligned} h(0) &= \delta(0) - g h(-\tau) = \delta(0) = 1 \\ h(1) &= -g h(1-\tau) = 0 \\ h(2) &= -g h(2-\tau) = 0 \\ &\vdots \\ h(\tau) &= -g h(0) = -g \\ &\vdots \\ h(2\tau) &= -g h(\tau) = g^2 \\ \Rightarrow h(k) &= \begin{cases} (-g)^n & \text{si } k = n\tau, n \in \mathbb{N} \\ 0 & \text{sinon} \end{cases} \end{aligned}$$

Nous reconnaissons ici une suite géométrique qui conduit à la conditions de stabilité suivante pour que le filtre soit stable : $|g| < 1$.

Nous voulons maintenant représenter la réponse impulsionnelle grâce à matlab. Pour cela nous allons utiliser la fonction `filter` comme vu un peu plus haut. Nous allons donc chercher les deux vecteurs a et b nécessaire pour calculer la réponse impulsionnelle.

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

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

$$a_0.y(k) + a_\tau.y(k - \tau) = b_0.x(k)$$

Où $a_0 = b_0 = 1$ et $a_\tau = g$.

On a donc les vecteurs de taille τ , $a = (1, 0, \dots, 0, g)$ et $b = (1, 0, \dots, 0, 0)$ que nous utilisons pour générer la réponse impulsionnelle. Nous obtenons alors :

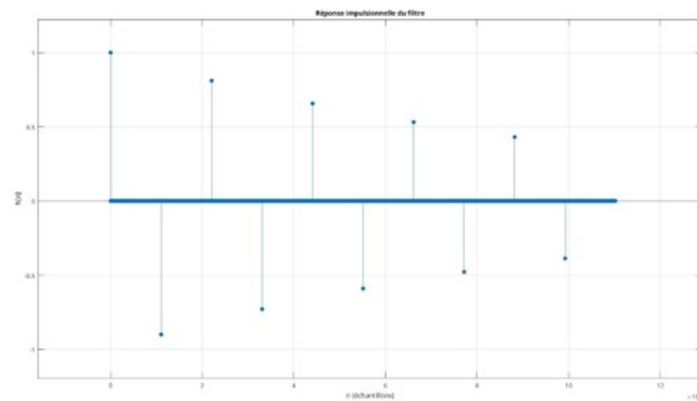


FIGURE 12 – Vérification de la réponse impulsionnelle sur matlab

Nous obtenons le résultat attendu : une réponse impulsionnelle qui décroît en valeur absolue, qui est positive pour les entiers pairs et négatifs pour les entiers impairs.

La réponse impulsionnelle décrit le comportement du filtre dans le domaine temporel. Pour mieux comprendre le comportement du filtre, il est parfois plus simple de raisonner à partir de sa réponse fréquentielle. Cette réponse décrit la modification (module, phase) de chaque composante fréquentielle entre l'entrée et la sortie du filtre.

$$\begin{aligned}
 \hat{h}(\omega) &= \sum_{k \in \mathbb{Z}} h(k) e^{-2i\pi k \omega} \\
 &= \sum_{k=0}^{+\infty} h(k) e^{-2i\pi k \omega} \\
 &= 1 - g e^{-2i\pi \tau \omega} + g^2 e^{-4i\pi \tau \omega} - g^3 e^{-6i\pi \tau \omega} + \dots \\
 &= (-g e^{-2i\pi \tau \omega})^0 + (-g e^{-2i\pi \tau \omega})^1 + (-g e^{-2i\pi \tau \omega})^2 + \dots \\
 &= \sum_{k=0}^{+\infty} (-g e^{-2i\pi \tau \omega})^k
 \end{aligned}$$

Puisque $|-g e^{-2i\pi \tau \omega}| = |g| < 1$, on a une série convergente et :

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

FIGURE 13 – Preuve de l'expression de la réponse en fréquence

On obtient alors son module et sa phase :

$$\begin{aligned}
 |\hat{h}(\nu)| &= \frac{1}{|1 + g \cdot e^{-2i\pi \tau \nu}|} \\
 |\hat{h}(\nu)| &= \frac{1}{\sqrt{1 + g^2 + 2 \cdot g \cdot \cos(\omega \tau)}} \\
 \phi &= \arg\left(\frac{1}{1 + g \cdot e^{-2i\pi \tau \nu}}\right) \\
 \phi &= -\arctan\left(\frac{-g \cdot \sin(\omega \tau)}{1 + g \cdot \cos(\omega \tau)}\right) \\
 \phi &= \arctan\left(\frac{g \cdot \sin(\omega \tau)}{1 + g \cdot \cos(\omega \tau)}\right)
 \end{aligned}$$

Car arctan est une fonction impaire.

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

Nous le vérifions en traçant sur un même graphique le module de la réponse en fréquence obtenu théoriquement et celui obtenu numériquement à l'aide de la transformée de Fourier discrète de la réponse impulsionnelle.

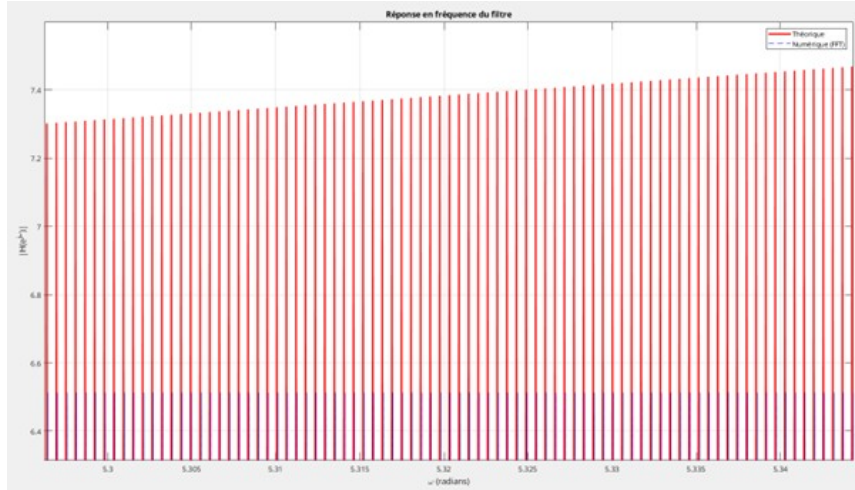


FIGURE 14 – Écart entre réponse en fréquence théorique et numérique

Le tracé rouge représente le module de la réponse théorique, qui représente donc des pics, ce qui correspond avec l'effet d'un filtre en peigne. On a pu vérifier ce tracé en regardant les maxima, égal à 10 pour $g = 0,9$. Le tracé bleu représente le module de la réponse calculée de manière numérique. On remarque alors qu'il y a une petite différence entre les deux tracés. En effet, cela peut être dû au fait que dans notre modèle numérique, la réponse a une taille finie, déterminée par L , qu'on a pris égal à $10 \cdot \tau$, alors qu'en théorie la réponse est de taille infinie. Ainsi, on constate que plus on augmente cette taille grand L , plus le tracé bleu se rapproche du tracé rouge.

Nous allons à présent programmer l'effet de delay sous forme d'une fonction Matlab. Pour cela, nous codons une fonction `effet-delay` fonction de g , du temps de delay, de la fréquence d'échantillonnage et un signal x . Cette dernière renvoie le signal modifié par le filtre présenté plus tôt.

Nous testons notre programme sur un accord de piano en choisissant les paramètres $\tau = 11025s$ et $g = 0,9$.

A l'écoute, nous observons que cet effet produit des sonorités "synthétiques". Pour rendre l'effet plus naturel, nous allons modifier le son après réflexion en filtrant le signal dans la boucle de retour, en choisissant un filtre dont la réponse impulsionnelle est donnée par :

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

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

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

Pour visualiser ce nouvel effet, nous avons codé une nouvelle fonction `effet-delay-filtre` qui renverra en sortie le signal x modifié par le filtre présenté ci-dessus. Nous utilisons à nouveau la fonction `filter` pour laquelle nous avons calculé cette fois-ci les vecteurs $a = (1, 0, \dots, 0, \frac{g}{K}, \dots, \frac{g}{K})$, avec $\tau - 1$ (0) d'affilé et K ($\frac{g}{K}$), et $b = (1, 0, \dots, 0)$.

En testant notre fonction, l'écoute du résultat montre que le filtre $h_r(k)$ semble "couper" progressivement les hautes fréquences du signal.

Pour vérifier ceci, nous traçons numériquement la réponse en fréquence de l'effet de retard filtré. Nous obtenons ce magnifique graphique :

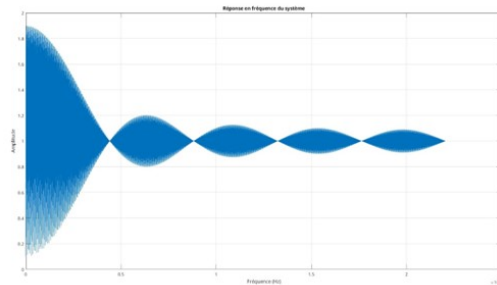


FIGURE 15 – Réponse en fréquence de l'effet de retard filtré

En présence d'un filtre moyennneur, nous nous attendons à un rapport de sinus en réponse en fréquence, semblable à un sinus cardinal. Nous vérifions en effet l'obtention d'une telle figure en le simulant numériquement. De plus l'atténuation de l'amplitude avec l'augmentation des fréquences traduit bien la présence d'un filtre passe bas !

5. Pour aller plus loin : synthèse Karplus-Strong

Pour améliorer notre synthèse, on se propose de prendre en compte l'énergie perdue pour gagner en réalité.

5.1. Analyse de l'algorithme de Karplus-Strong

Le principe de l'algorithme Karplus-Strong avec une perte d'énergie peut être représenté avec l'algorithme suivant :

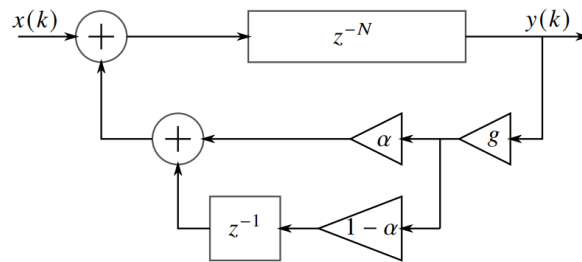


FIGURE 16 – Algorithme de Karplus Strong

Un signal d'entrée $x(k)$ est choisi en entrée. Celui ci pourra être choisi de deux manières possibles : par un signal aléatoire blanc gaussien (fonction matlab `randn`) ou par un son d'une note jouée par un instrument (guitare,...). Suite à quoi l'algorithme présente plusieurs fonctions :

- Une fonction de retard : le signal est d'abord retardé de N échantillons. Avec la fréquence d'échantillonnage choisie à $F_e = 44100\text{Hz}$, la relation $f_1 = \frac{F_e}{N}$ fixe la fréquence fondamentale du son synthétisé f_1 . **Ce retard simule la longueur d'une corde vibrante, et sa valeur conditionne donc la hauteur perçue du son.**
- Une fonction d'amortissement : le signal obtenue est ensuite amorti : un facteur d'amortissement g , strictement inférieur à 1 pour réduire l'énergie, est appliqué au signal, ce qui permet de simuler la perte progressive d'énergie au fil du temps.
- Un filtrage passe-bas : le signal retardé est ensuite filtré à l'aide d'un filtre passe-bas de la forme $H(z) = \alpha + (1 - \alpha).z^{-1}$. En choisissant $\alpha = 0,5$, ce filtre permet de faire la moyenne entre l'échantillon actuel et le précédent. Remarquons que si $\alpha = 1$, on s'attend à obtenir un gain unitaire et que si $\alpha = 0$, le filtre ne garde que l'échantillon précédent.

5.2. Formule de récurrence

Pour synthétisé le son, nous allons suivre la même technique que précédemment. Pour cela, commençant par établir la relation de récurrence entre l'entrée $x(k)$ et la sortie $y(k)$:

En sortie du premier additionneur, notons $a(k)$ et $b(k)$ les signaux tel que :

$$a(k) = x(k) + b(k)$$

Où :

$$b(k) = \alpha.g.y(k) + (1 - \alpha).g.y(k - 1)$$

Ce qui conduit à :

$$h_r(k) = \begin{cases} y(k) = a(k - N) \\ a(k) = x(k) + \alpha \cdot g \cdot y(k) + (1 - \alpha) \cdot g \cdot y(k - 1) \end{cases}$$

D'où la relation de récurrence finale :

$$y(k) = x(k) + \alpha \cdot g \cdot y(k - N) + (1 - \alpha) \cdot g \cdot y(k - N - 1)$$

En mettant sous la forme de filtre IIR, on obtient facilement les vecteurs **a** et **b** pour la fonction `filter` :

$$\begin{aligned} a &= (1, 0, \dots, 0, -\alpha \cdot g, -(1 - \alpha) \cdot g) & \text{taille : } (N + 2) \\ b &= (1, 0, \dots, 0, 0) \end{aligned}$$

5.3. Test avec un bruit blanc

Pour la synthèse, on choisit un signal aléatoire blanc gaussien à l'aide de la fonction matlab `randn`. On code la fonction avec les paramètres $g = 0,95$, $N = 5$ et $\alpha = 0,5$. Ceci correspond à une fréquence fondamentale élevée (autour de 8820 Hz), ce qui donne un son percussif, presque métallique — comme un petit claquement ou un pincement bref de corde.

Bien que celui ci ne soit pas très beau, le résultat est surprenant par sa richesse, surtout si l'on considère la simplicité du système mis en œuvre. La combinaison du bruit initial, du retard et du filtrage forme une onde qui se stabilise rapidement. L'algorithme Karplus-Strong ne reproduit pas seulement la hauteur du son : il en simule aussi l'évolution dans le temps et sa dynamique, tout ce qui donne au son un caractère vivant !

5.4. Conclusion

Cette modélisation permet une synthèse très efficace du son d'une corde pincée. La richesse spectrale du son est contrôlée par la valeur de α , qui règle la proportion entre les deux échantillons passés. Lorsque $\alpha = 0,5$, on obtient une moyenne simple entre les deux, ce qui permet d'atténuer les hautes fréquences et de simuler un amortissement doux et naturel. En diminuant α , le filtre devient plus agressif sur les hautes fréquences, ce qui renforce l'effet d'amortissement, tandis que des valeurs proches de 1 conservent davantage les harmoniques aiguës.

L'intérêt principal de cette méthode par rapport à la synthèse soustractive classique est sa capacité à produire des sons réalistes et dynamiques avec très peu de paramètres, et sans devoir manipuler directement des signaux complexes ou des spectres de Fourier. Elle est également rapide à exécuter ce qui en fait une technique de choix pour les synthétiseurs.

Ainsi, la méthode de Karplus-Strong constitue une approche efficace pour la synthèse de sons de type cordes pincées, tout en offrant une grande flexibilité dans le choix du timbre, via la nature du signal d'entrée, la longueur du retard N , le facteur d'amortissement g et le coefficient de filtrage α .

OUR WORLDWIDE PARTNERS UNIVERSITIES - DOUBLE DEGREE AGREEMENTS



3 CAMPUS



IMT Atlantique Bretagne-Pays de la Loire – <http://www.imt-atlantique.fr/>

Campus de Brest

Technopôle Brest-Iroise
CS 83818
29238 Brest Cedex 3
France
T +33 (0)2 29 00 11 11
F +33 (0)2 29 00 10 00

Campus de Nantes

4, rue Alfred Kastler
CS 20722
44307 Nantes Cedex 3
France
T +33 (0)2 51 85 81 00
F +33 (0)2 99 12 70 08

Campus de Rennes

2, rue de la Châtaigneraie
CS 17607
35576 Cesson Sévigné Cedex
France
T +33 (0)2 99 12 70 00
F +33 (0)2 51 85 81 99



IMT Atlantique

Bretagne-Pays de la Loire
École Mines-Télécom