



ÉCOLE NATIONALE D'INGÉNIEURS DE TUNIS

Département Génie Industriel

Mini-Projet MATLAB
TRAITEMENT NUMERIQUE DU SIGNAL

Compression de la parole par codage en sous-bandes

Elaboré par :

Balti Chaima

Encadré par :

Mme. Soumaya MEHERZI

1^{ère} Année MIndS

Année universitaire : 2022/2023

Remerciements

Je souhaite exprimer ma sincère reconnaissance à ma professeure, Madame Soumaya Meherzi, pour son encadrement dévoué et son soutien précieux tout au long de ce mini-projet. Sa compétence, sa disponibilité et ses conseils éclairés.

Table des matières

Table des figures	3
0.1 Phase d'analyse	8
0.1.1 Banc de filtres d'analyse	8
0.1.2 Décimation	13
0.1.3 Quantification/Codage	14
0.2 Phase de synthèse	16
0.2.1 Interpolation	16
0.2.2 Filtres de synthèse	16

Table des figures

1	Phase d'analyse	5
2	Phase de synthèse	6
3	Réponse fréquentielle de filtre passe-bas avec fenêtres rectangulaire et de Hamming	8
4	Réponses fréquentielles des filtres passe-bas pour différents ordres .	9
5	Conception du filtre passe-haut	10
6	Réponses fréquentielles des filtres passe-haut	10
7	Comparaison des signaux s et sr avec filtre RIF	11
8	Comparaison des signaux s et sr avec filtre RII	12
9	script MATLAB de sr avec filtre RII	12
10	Spectres des signaux sous-échantillonnés s1d et s2d.	13
11	La fonction Unifquant	14
12	RSB de sd1 et sd2	15
13	sd1 et sd2 pour différentes valeurs de l1 et l2	15
14	Spectres des signaux sd1 et sd2 après un suréchantillonnage d'un facteur $M = 2$	16
15	Spectres des signaux sd1 et sd2 après un suréchantillonnage et après éliminer l'effet miroir	17
16	Figure du décalage temporel entre les signaux s et ss	18
17	Script MATLAB de calcule du décalage temporelle	18

Introduction

La compression de la parole par codage en sous-bandes est une technique largement utilisée dans le domaine de la compression audio. Elle permet de subdiviser le signal en différentes sous-bandes fréquentielles et d'appliquer des techniques de codage adaptées à chaque sous-bande, réduisant ainsi le débit tout en préservant une qualité audio satisfaisante. Cette approche trouve des applications dans des domaines tels que les communications vocales haute qualité, le codage audio et la compression d'images.

Ce projet vise à mettre en œuvre cette technique de codage en sous-bandes en utilisant MATLAB, en adoptant une architecture simple et en la testant sur un exemple audio. Il exploite les notions théoriques fondamentales telles que le filtrage, la décimation, l'interpolation et la quantification, offrant ainsi une illustration pratique et concrète de ces concepts tout en renforçant la maîtrise de MATLAB.

Dans ce projet, nous présenterons le principe du codage en sous-bandes, en soulignant ses avantages et ses applications. Ensuite, nous décrirons les différentes étapes de réalisation du projet, notamment l'analyse du signal, la conception des filtres, la décimation, la quantification/codage, l'interpolation et les filtres de synthèse. Nous discuterons également des critères d'évaluation de la qualité et du débit binaire, ainsi que du choix des paramètres optimaux pour obtenir une compression efficace.

Principe

L'objectif principal de cette technique est de réaliser un codage de haute qualité avec un débit minimal pour un signal donné. Pour atteindre cet objectif, le principe de base consiste à subdiviser le spectre du signal en sous-bandes et à appliquer une technique de compression spécifique à chaque sous-bande. Le choix des sous-bandes se fait généralement en tenant compte de leur importance perceptuelle, qu'elle soit basée sur l'énergie ou l'entropie.

Ce processus de subdivision du signal en sous-bandes comprend deux phases principales :

Phase d'analyse :

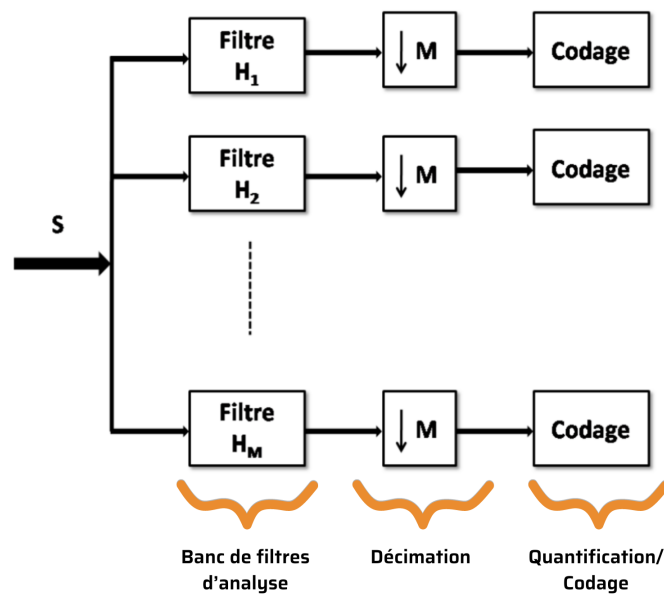


FIGURE 1 – Phase d'analyse

- ⌘ *Banc de filtres d'analyse* : Cette étape permet de diviser le signal en sous-bandes en utilisant des filtres passe-bande appropriés. Chaque filtre sélectionne une bande spécifique du spectre et isole les fréquences correspondantes.
- ⌘ *Décimation* : Après l'analyse, la fréquence d'échantillonnage de chaque sous-bande est réduite pour minimiser le nombre d'échantillons nécessaires à la représentation du signal.
- ⌘ *Quantification/Codage* : Les échantillons de chaque sous-bande sont ensuite représentés par des codes binaires, ce qui permet de compresser les données en réduisant la quantité d'informations requises pour les stocker.

Phase de synthèse :

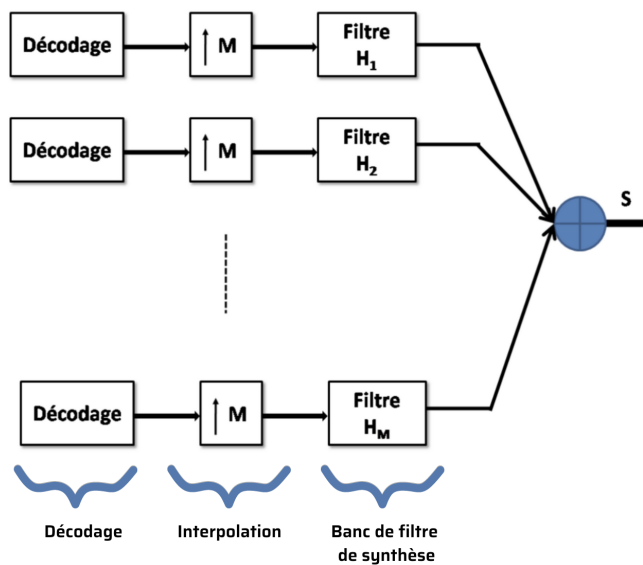


FIGURE 2 – Phase de synthèse

- ⌘ *Décodage* : Les codes binaires sont décodés pour retrouver les échantillons originaux de chaque sous-bande.
- ⌘ *Interpolation* : Pour restaurer la fréquence d'échantillonnage d'origine, une interpolation est réalisée pour chaque sous-bande afin de reconstituer les informations fréquentielles manquantes.
- ⌘ *Banc de filtres de synthèse* : Enfin, chaque composante est renvoyée dans sa bande de départ en utilisant des filtres de synthèse complémentaires aux filtres d'analyse, permettant ainsi de reconstituer le signal audio original.

Cette approche de subdivision du spectre en sous-bandes offre de nombreux avantages, notamment une compression efficace du signal tout en préservant une qualité audio élevée. En divisant le spectre en sous-bandes, on peut appliquer des techniques de compression adaptées à chaque bande, en mettant l'accent sur les parties les plus importantes pour l'audition humaine. Ainsi, cette méthode permet d'optimiser la représentation et la transmission des signaux sonores avec une utilisation minimale des ressources.

Réalisation et Résultats

0.1 Phase d'analyse

0.1.1 Banc de filtres d'analyse

Dans cette partie, nous avons utilisé des filtres passe-bas et passe-haut sur un signal audio. Nous avons exploré les effets des différentes fenêtres utilisées pour la conception des filtres et nous avons analysé les réponses fréquentielles ainsi que les signaux filtrés. Les résultats sont présentés à travers des figures illustrant les modifications apportées au signal audio.

Comparaison des fenêtres de filtrage : Rectangulaire vs. Hamming

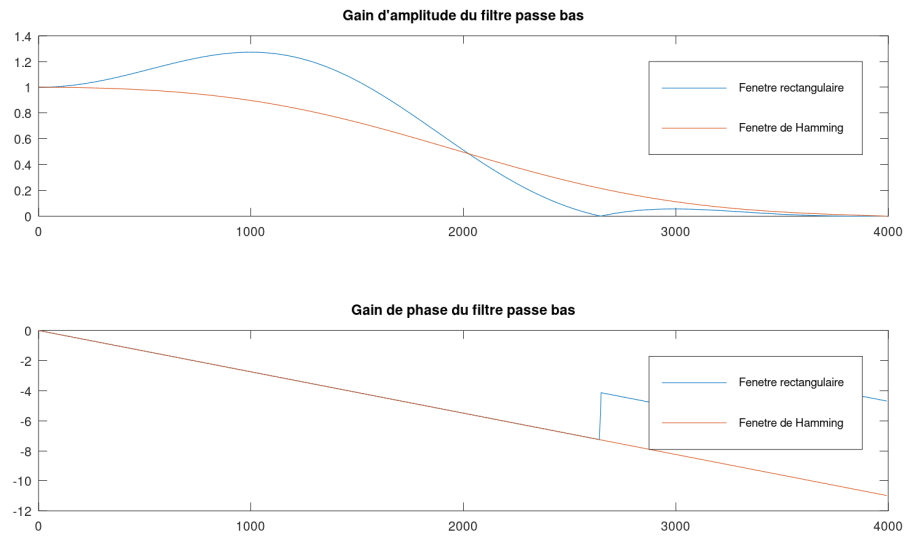


FIGURE 3 – Réponse fréquentielle de filtre passe-bas avec fenêtres rectangulaire et de Hamming

En comparant les réponses fréquentielles de filtre avec une fenêtre rectangulaire et une fenêtre de Hamming dans la Figure 3, on constate que la fenêtre de Hamming offre une meilleure sélectivité en atténuant davantage les fréquences indésirables. Les courbes d'amplitude montrent une atténuation plus prononcée aux hautes fréquences, ce qui indique une meilleure capacité à supprimer le bruit indésirable. Après avoir comparé les réponses fréquentielles des filtres avec les fenêtres rectangulaire et de Hamming, nous avons choisi de travailler avec la fenêtre de Hamming. Cette décision s'appuie sur les avantages de la fenêtre de Hamming, tels qu'une meilleure réduction du lobage principal, une sélectivité fréquentielle améliorée et une préservation des caractéristiques temporelles du signal. En optant pour la fenêtre de Hamming, nous visons à obtenir des résultats de filtrage plus précis et de meilleure qualité.

Impact de l'ordre du filtre sur la réponse fréquentielle

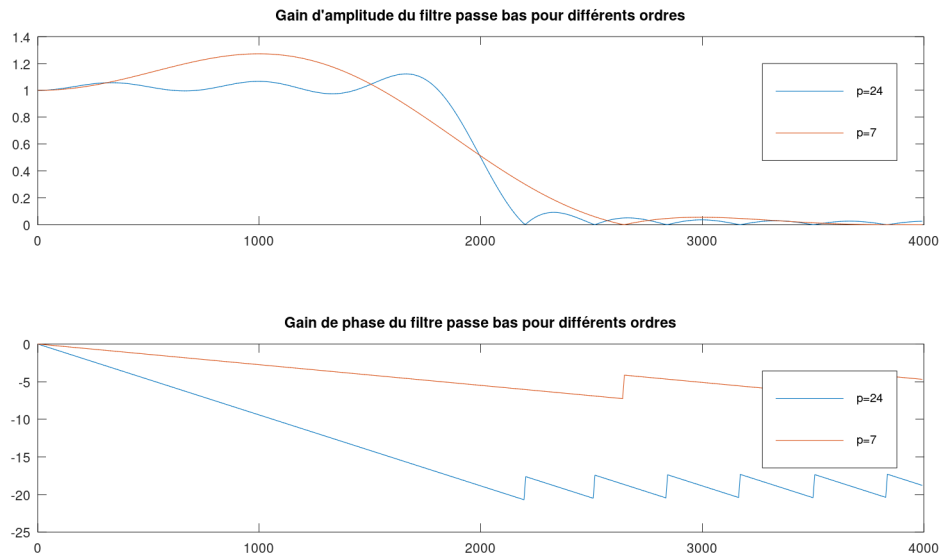


FIGURE 4 – Réponses fréquentielles des filtres passe-bas pour différents ordres

Lorsque l'ordre du filtre passe-bas est augmenté à $p = 24$, la réponse fréquentielle devient plus précise et étroite. Cela signifie que le filtre est plus sélectif et atténue davantage les fréquences situées en dehors de la bande passante souhaitée.

En augmentant l'ordre du filtre, on augmente le nombre de coefficients utilisés pour la conception du filtre. Cela permet d'affiner la réponse fréquentielle en ajustant plus finement la courbe d'amplitude du filtre.

Conception du filtre passe-haut pour la récupération de la bande HF

Voici le script MATLAB utilisé qui propose un filtre passe-haut permettant de récupérer la bande de fréquences élevées (bande HF) du signal. Voici le script MATLAB utilisé :

```
k=0:p;
HH24=((-1).^k).*HL24;
figure(4)
[Hi,Wi]=freqz(HH24,1);
subplot(2,1,1)
plot(Fs*Wi/(2*pi),abs(Hi));
title("Gain d'amplitude du filtre passe haut")
hold on
subplot(2,1,2)
plot(Fs*Wi/(2*pi),unwrap(angle((Hi)),pi));
title("Gain de phase du filtre passe haut")
hold on
```

FIGURE 5 – Conception du filtre passe-haut

Dans ce programme, après utiliser la fonction `fir1` pour concevoir le filtre passe-bas `HL24`, on a utilisé une approche spécifique pour obtenir le filtre passe-haut `HH24` à partir du filtre passe-bas. Cela est réalisé en utilisant une alternance de signes dans les coefficients du filtre passe-bas.

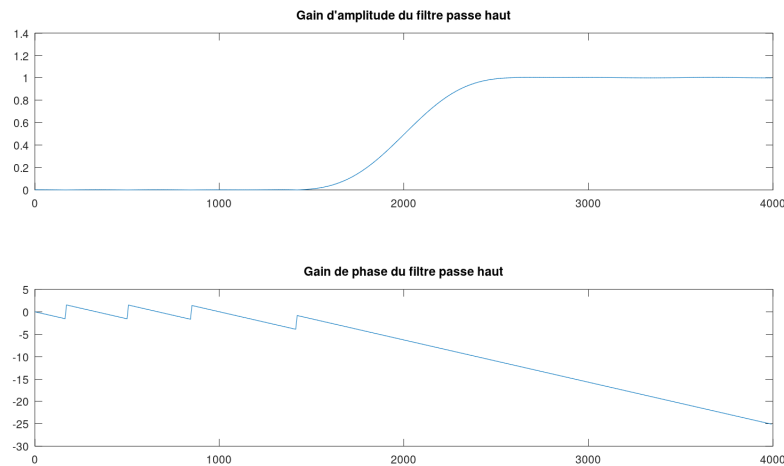


FIGURE 6 – Réponses fréquentielles des filtres passe-haut

Reconstitution du spectre du signal après sous-échantillonnage

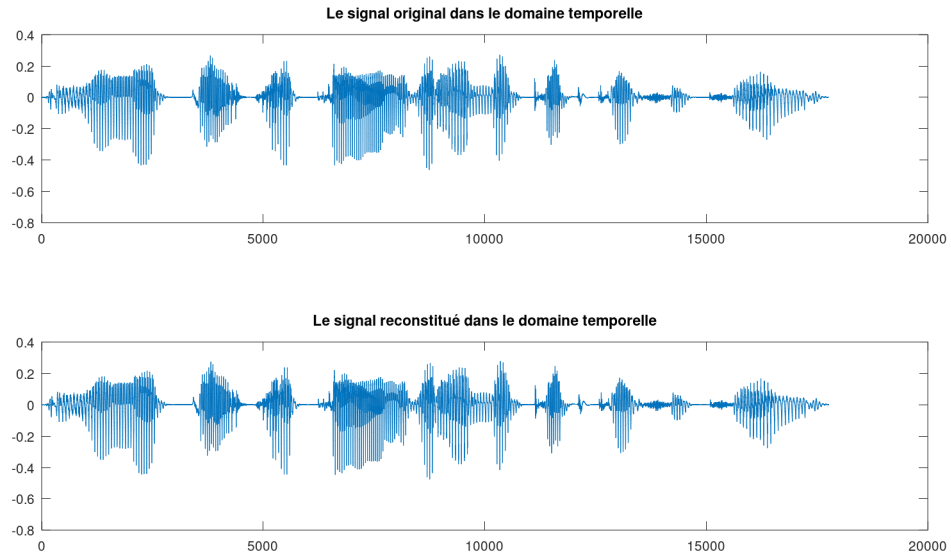


FIGURE 7 – Comparaison des signaux s et s_r avec filtre RIF

En analysant le signal reconstitué s_r (figure 7) , on peut observer que les composantes du signal original ont été préservées. Il est possible alors de conclure quant au choix des deux filtres. le signal reconstitué s_r présente une bonne fidélité par rapport au signal original, avec une faible distorsion et un niveau de bruit acceptable, cela indique que les filtres utilisés (s_1 et s_2) ont réussi à préserver les caractéristiques souhaitées du signal. Donc le choix des deux filtres peut être considéré comme approprié.

Comparaison des filtres RII et RIF pour le traitement du signal

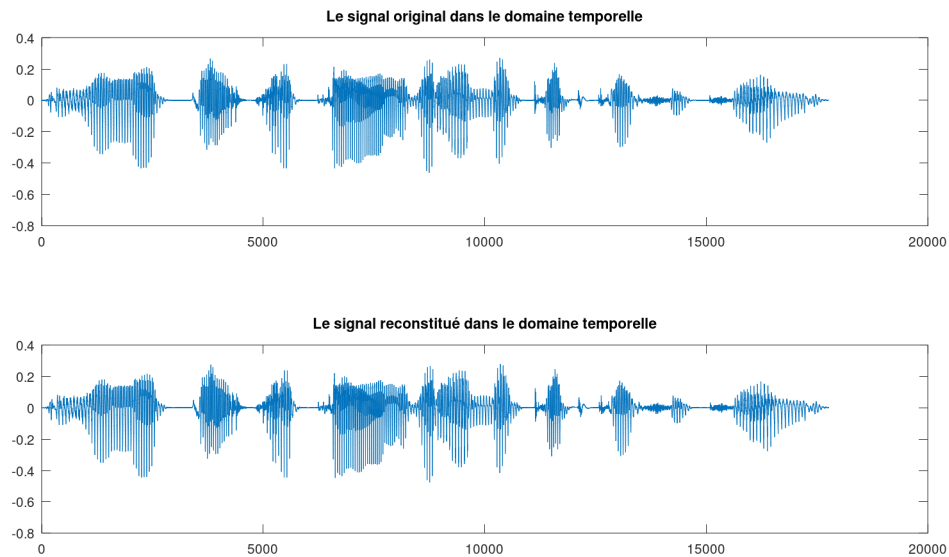


FIGURE 8 – Comparaison des signaux s et sr avec filtre RII

les filtres RIF (Réponse Impulsionnelle Finie) et RII (Réponse Impulsionnelle Infinie) ont été évalués pour déterminer celui qui convient le mieux à l'application. Les deux approches ont été testées, en utilisant des filtres RII tels que les filtres Butterworth conçus avec la fonction "butter".

```
%11) les filtres RII avec la fonction butter
[BH, AH] = butter (p, 2*fc/Fe, 'high' );
[BL, AL] = butter (p, 2*fc/Fe, 'low' );
s1i=filter(BL ,AL,s);
s2i=filter(BH ,AH,s);
s_ri=s1i+s2i;

figure(7)
subplot(2,1,1)
plot(s);
title("Le signal original dans le domaine temporelle")
subplot(2,1,2)
plot(s_ri);
title("Le signal reconstitué dans le domaine temporelle")
```

FIGURE 9 – script MATLAB de sr avec filtre RII

Plusieurs facteurs ont été pris en compte lors du choix, notamment :

- *Complexité de conception* : Les filtres RII, tels que les filtres Butterworth, Chebyshev ou elliptiques, peuvent nécessiter une conception plus complexe et une rétroaction. En revanche, les filtres RIF offrent une approche plus directe et plus simple pour la conception et l'implémentation des filtres.
- *Performance satisfaisante* : Les filtres RIF ont démontré une bonne performance dans cette application spécifique de filtrage du signal audio. Bien que les filtres RII offrent généralement une sélectivité fréquentielle accrue et des caractéristiques de réjection plus prononcées, aucune différence notable n'a été observée dans ce cas d'étude particulier.

Sur la base de ces considérations, *le choix s'est porté sur l'utilisation de filtres RIF* dans ce projet.

0.1.2 Décimation

La partie de décimation aborde le processus de réduction du taux d'échantillonnage d'un signal. La décimation est une étape importante dans le traitement des signaux, permettant de réduire la taille des données tout en préservant les informations essentielles.

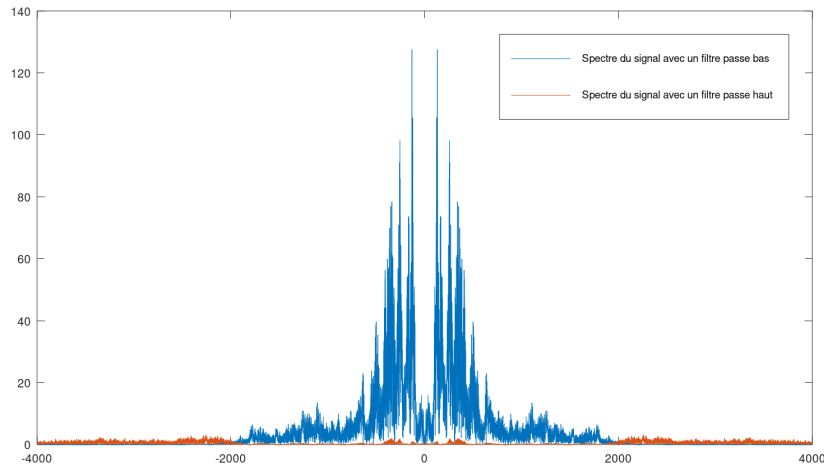


FIGURE 10 – Spectres des signaux sous-échantillonnés s1d et s2d.

La Figure 10 présente les signaux sous-échantillonnés s1d et s2d après l'application des filtres passe-bas et passe-haut respectivement sur le signal d'origine s. Le signal s1d est représenté en couleur bleue, tandis que le signal s2d est représenté en couleur rouge.

0.1.3 Quantification/Codage

Dans la partie Quantification/Codage, on aborde la conversion du signal analogique en signal numérique. On explore les concepts de quantification et de codage, ainsi que leur impact sur la qualité du signal audio et l'efficacité de la transmission des données.

La fonction "unifquant"

La fonction "unifquant" en Matlab permet de quantifier uniformément un signal donné. Elle prend en compte les bornes du signal, le nombre de bits utilisés et renvoie le signal quantifié ainsi que la valeur du RSB correspondant.

```
function [xq,RSB] = unifquant (x,l,Amin=min(x),Amax=max(x))
q =(Amax - Amin)/(2^l);
xq=zeros(size(x));
for n=1:size(x)(1)
    if (floor(x(n,l)/q)*q <= x(n,l)) && (x(n,l)< ((floor(x(n,l)/q) + 1)*q))
        xq(n,l) = floor(x(n,l)/q)*q + q/2;
    endif
endfor
RSB = 10*log10(mean(x.^2)/((q^2)/12));
endfunction
```

FIGURE 11 – La fonction Unifquant

Choix de l1 et l2

Valeur de l	RSB1	RSB2
2	3.5209	-3.0908
3	9.5415	2.9298
4	15.562	8.9504
5	21.583	14.971
6	27.603	20.992
7	33.624	27.012
8	39.645	33.033
9	45.665	39.053
10	51.686	45.074
11	57.706	51.095
12	63.727	57.115

FIGURE 12 – RSB de sd1 et sd2

En évaluant les valeurs de RSB pour différentes valeurs de l (de 2 à 12) et en considérant qu'un RSB supérieur ou égal à 30 dB est généralement associé à une qualité acceptable de sons. Dans le but d'optimiser le stockage, on a sélectionné les valeurs l1=7 et l2=8 qui représentent les plus petites valeurs permettant d'atteindre un RSB supérieur ou égal à 30 dB pour les signaux quantifiés s1d et s2d. Par ailleurs, il convient de noter que le filtre passe-haut nécessite une valeur de l légèrement plus grande que le filtre passe-bas pour obtenir un RSB équivalent. Cela s'explique par les différences dans les caractéristiques de filtrage et les effets de la quantification sur le signal.

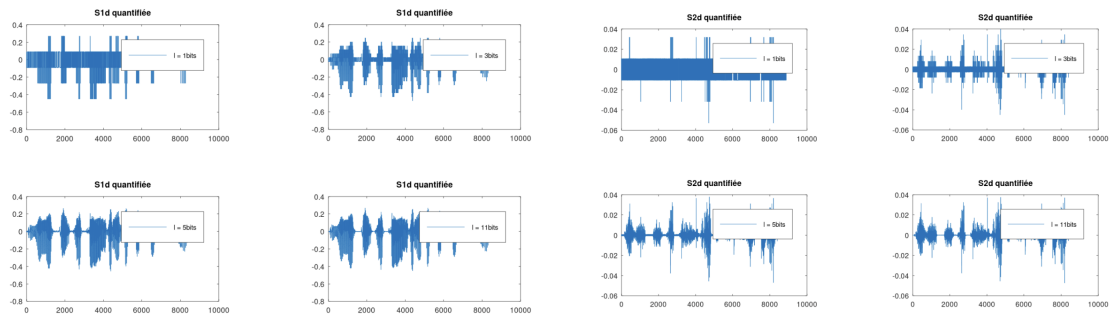


FIGURE 13 – sd1 et sd2 pour différentes valeurs de l1 et l2

0.2 Phase de synthèse

0.2.1 Interpolation

Dans cette partie consacrée à l'interpolation, nous abordons la phase de synthèse du signal. Après le processus de décodage, le signal obtenu dans chaque sous-bande contient $F_e' = F_e/2$ échantillons par seconde

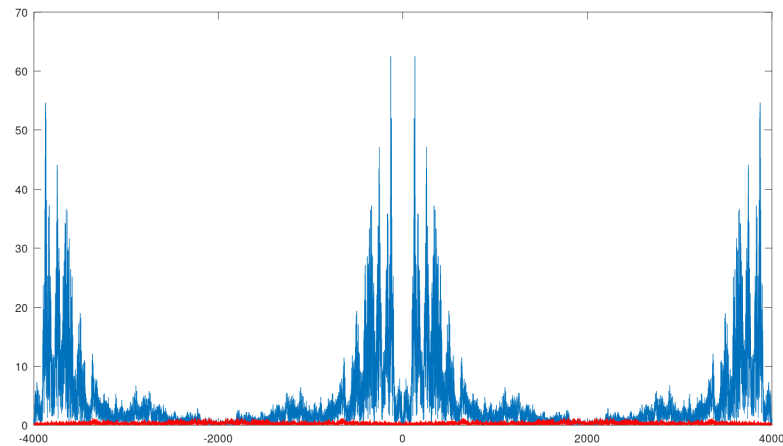


FIGURE 14 – Spectres des signaux sd1 et sd2 après un suréchantillonnage d'un facteur $M = 2$.

La figure présente les spectres des signaux sd1 et sd2 après avoir été soumis à un suréchantillonnage d'un facteur $M = 2$. Le suréchantillonnage consiste à insérer des zéros entre les échantillons d'origine pour augmenter la fréquence d'échantillonnage. On peut observer un effet miroir dans les spectres des signaux sd1 et sd2. Cet effet est dû à la réplication périodique des spectres résultant du suréchantillonnage. On peut remarquer des composantes spectrales apparaissant symétriquement de part et d'autre de la moitié de la fréquence d'échantillonnage. Ces composantes spectrales miroir ne correspondent pas aux informations du signal d'origine, mais sont une conséquence du suréchantillonnage.

0.2.2 Filtres de synthèse

Pour compenser l'effet miroir et préserver les caractéristiques du signal d'origine, on a appliqué des filtres passe-haut et passe-bas sur les signaux sd1 et sd2 après le surechantillonnage.

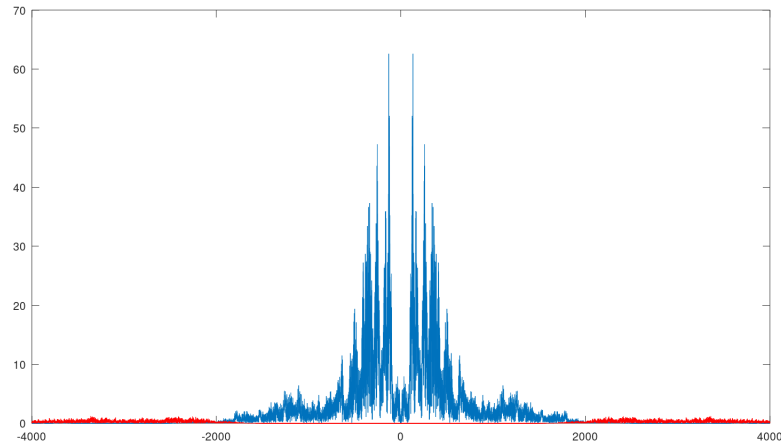


FIGURE 15 – Spectres des signaux sd1 et sd2 après un suréchantillonnage et après éliminer l'effet miroir

Le filtre passe-bas permet de supprimer les composantes spectrales miroir situées au-dessus de la moitié de la fréquence d'échantillonnage. Cela permet de récupérer la bande de fréquences d'intérêt du signal d'origine et de limiter les distorsions indésirables causées par l'effet miroir.

Le filtre passe-haut, quant à lui, permet de supprimer les composantes spectrales miroir situées en dessous de la moitié de la fréquence d'échantillonnage. Cela permet de rejeter les fréquences indésirables qui pourraient perturber la qualité du signal reconstruit.

En appliquant ces filtres, on préserve les caractéristiques du signal d'origine tout en éliminant les distorsions causées par l'effet miroir. Cela contribue à obtenir une reconstruction plus fidèle et de meilleure qualité du signal de départ.

Synthèse du signal ss et détermination du décalage temporel

Dans cette partie, nous allons synthétiser le signal ss à partir des sous-bandes filtrées et effectuer une analyse du décalage temporel résultant. La figure ci-dessous présente le décalage temporel entre le signal d'origine s et le signal synthétisé ss. Dans notre analyse du décalage entre les signaux s et ss, on a utilisé la méthode de corrélation croisée pour estimer précisément le décalage temporel. Cette méthode est couramment utilisée pour mesurer les différences de phase et de temps entre deux signaux.

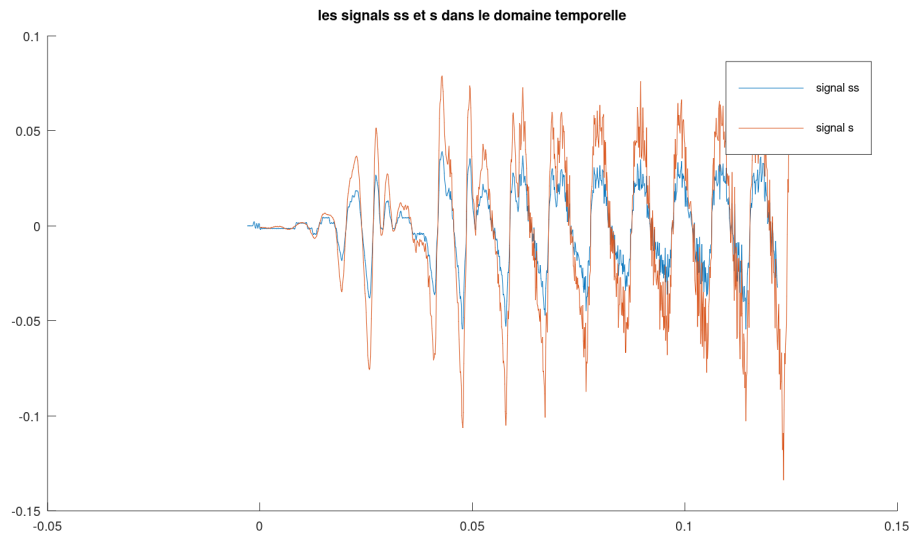


FIGURE 16 – Figure du décalage temporel entre les signaux s et ss

Le script MATLAB suivant illustre l'utilisation de la corrélation croisée pour calculer le décalage temporel :

```
% Calcul de la corrélation croisée entre s et ss
corr= xcorr(ss, s);
% Recherche de l'indice du maximum de la corrélation croisée
[max_corr, max_index] = max(corr);
%decalage temporel
decalage_temporel = (max_index-size(s)(1))/Fe;
%decalage temporel par bit
decalage_temporel_par_bit =decalage_temporel*(11+12)*Fe;
%Représentation
figure(13)
hold on
plot(t(1:1000)-decalage_temporel ,ss(1:1000),";signal ss;");
plot(t(1:1000),s(1:1000),";signal s;");
title('les signaux ss et s dans le domaine temporelle');
```

FIGURE 17 – Script MATLAB de calcul du décalage temporel

Dans ce script, s représente le signal d'origine et ss est le signal synthétisé. La fonction xcorr est utilisée pour calculer la corrélation croisée entre ces deux signaux. Ensuite, l'indice correspondant au pic de corrélation maximale est extrait à l'aide de la fonction max. Ce pic correspond au décalage temporel entre les signaux. Le calcul révèle un décalage temporel de 0.003 unités de temps (360 par bit) . Ce décalage correspond à une différence de phase entre les signaux s et ss.

Conclusion

En conclusion, on a mené ce projet dans le but d'étudier et d'analyser le traitement du signal audio à travers différentes étapes. On a examiné l'impact de chaque étape sur la qualité du signal et pris des décisions en fonction des résultats obtenus.

Au cours de ce projet, on a utilisé des outils et des techniques de traitement du signal pour analyser et manipuler les signaux audio. On a exploré les concepts de l'échantillonnage, de la quantification, de la filtration et de l'interpolation. On a évalué l'effet de chaque étape sur les caractéristiques du signal et pris des mesures pour améliorer sa qualité.

On a constaté que les choix des paramètres, tels que l'ordre des filtres et le nombre de bits de quantification, peuvent avoir un impact significatif sur la qualité du signal reconstruit. En utilisant des critères comme le rapport signal sur bruit (RSB), on a pu évaluer objectivement la performance des différentes configurations et ajuster les paramètres pour obtenir les meilleurs résultats.

L'interpolation s'est avérée être une étape cruciale dans la phase de synthèse, permettant de ramener les composantes du signal vers la bande spectrale initiale. On a observé les effets du suréchantillonnage et l'utilisation de filtres passe-haut et passe-bas pour atteindre la bande spectrale souhaitée et minimiser les distorsions.

Enfin, on a utilisé la technique de corrélation croisée pour estimer avec précision les décalages temporels entre les signaux synthétisés et les signaux d'origine. Cela nous a permis d'évaluer la synchronisation et de vérifier la fidélité de la reconstruction.

En somme, ce projet a permis de mieux comprendre les différentes étapes du traitement du signal audio et leur impact sur la qualité du signal. On a pu mettre en pratique des techniques et des méthodes de traitement du signal pour améliorer la représentation et la restitution des signaux audio.