

TP4 - Synthèse de filtres numériques

VERNOUX Thomas - ABID CHAREF Samy

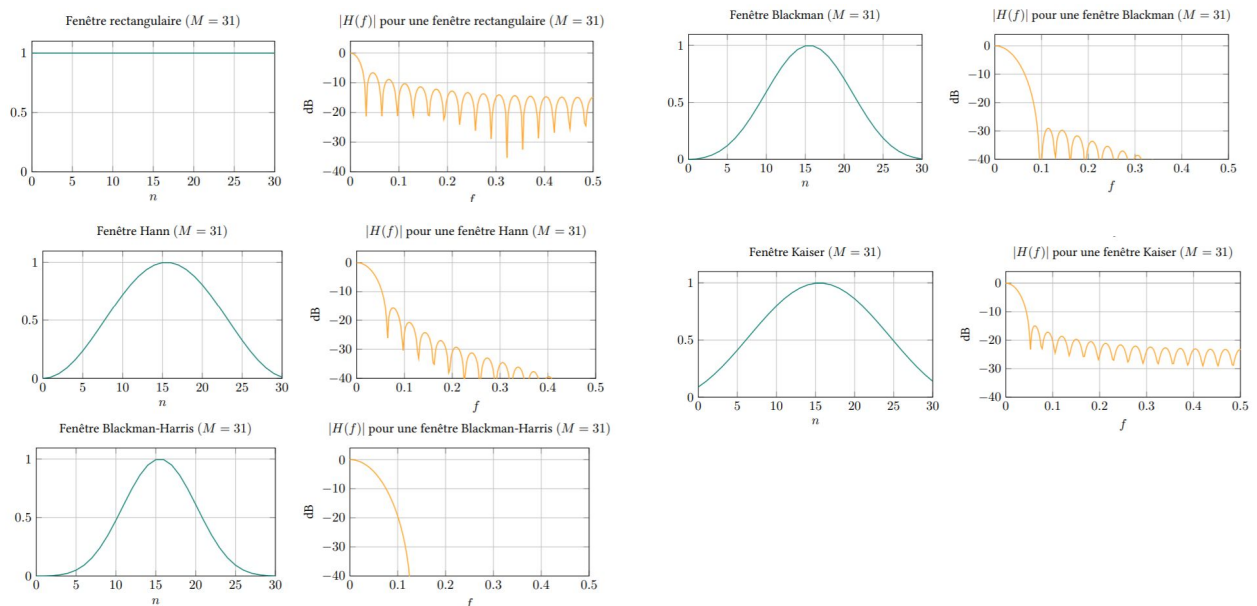
1- Préparation du TP

Spectrogramme

Un spectrogramme est une figure qui représente les différentes fréquences et leur amplitudes dans un signal en fonction du temps.

Méthode des fenêtres

La méthode des fenêtres est une méthode qui permet la synthèse de filtre RIF. Cette méthode consiste à partir du gain complexe $H_d(f)$ souhaité et de calculer sa TFTD inverse pour obtenir la réponse impulsionnelle idéale du filtre. On obtient cependant une RI infinie, ce qui est inenvisageable pour la conception d'un filtre. Il faut donc multiplier la RI par une porte ou un signal de durée limitée (ce qui revient à convoluer le gain complexe). On appelle cela la fenêtrage, il est possible de réaliser cette opération avec différentes fonctions:



Chacune de ces fonctions "sélectionne" plus ou moins finement la fréquence à garder, de même que l'atténuation de la bande coupée.

Étude de la fonction `fir1`

La fonction `fir1` est une fonction implémentée sous @Matlab qui crée un filtre à réponse impulsionnelle finie avec la méthode des fenêtres, notamment en utilisant une fenêtre de Hamming.

La syntaxe usuelle est : `b = fir1(n,Wn>window)`

En effet elle prend en entrée `n` qui correspond à la taille de la fenêtre de Hamming, `Wn` est la fréquence réduite de coupure et enfin `window` qui correspond à la fenêtre citée précédemment de longueur `n`.

Comme il s'agit d'un filtre RIF, ce dernier ne possède pas de pôles et de ce fait la fonction retourne `b` qui correspond aux coefficients polynomiaux du numérateur de la fonction de transfert dont les zéros sont les racines du polynôme engendré par `b`.

Conception d'un filtre RII à l'aide de la transformée bilinéaire

La première consiste à rapprocher le gabarit souhaité du gain du filtre en question par un filtre analogique. Puis il s'agit d'utiliser la transformée bilinéaire qui lie le domaine en z et le domaine de Laplace :

$$p = \frac{2}{T} \times \frac{(1-z^{-1})}{(1+z^{-1})}$$

Cette transformée permet d'assurer que l'axe imaginaire du plan de Laplace corresponde au cercle unité dans le plan de la TZ et que la partie gauche du plan complexe de Laplace (à partie réelle négative) corresponde à l'intérieur du cercle unité dans le plan de la TZ ce qui permet conservation de la stabilité

Principe du "prewarping"

La méthode du prewarping consiste à effectuer une correspondance entre la fréquence de coupure en analogique décrit dans le domaine de Laplace et la fréquence de coupure numérique dans le domaine en Z à l'aide de relations mathématiques qui constituent des propriétés de la transformée bilinéaire.

Ainsi en posant $z = e^{2i\pi f}$ et $p = 2i\pi v$ avec f la fréquence réduite numérique et v la fréquence analogique réelle il vient :

$$v = \frac{1}{T\pi} \tan(\pi f) \text{ et } f = \frac{1}{T} \arctan(\pi T v)$$

Étude de la fonction `ellip`

La fonction `ellip` est une fonction implémentée sous @Matlab qui crée une version numérique d'un filtre elliptique.

La syntaxe usuelle est : `[b,a] = ellip(n,Rp,Rs,Wp)`

Elle prend en entrée n qui correspond à l'ordre du filtre, ω_p correspond à la pulsation de coupure normalisé (pas d'unité), si on ne spécifie pas un autre argument s (en Hz), R_p constitue l'atténuation en dB à la fréquence de coupure et R_s la l'atténuation en dB en la fréquence marquant le début de la bande atténuée.

En sortie elle retourne a et b qui constituent respectivement les coefficients polynomiaux du dénominateur et du numérateur de la fonction de transfert d'un filtre elliptique.

Étude de la fonction `bilinear`

La fonction `bilinear` est une fonction implémentée sous @Matlab qui effectue la conversion d'un filtre analogique en un filtre numérique à l'aide de la transformée bilinéaire.

La syntaxe usuelle est : `[numd,dend] = bilinear(num,den,fs,fp)`

En entrée elle prend `num` et `den` qui correspondent aux coefficients polynomiaux du numérateur et du dénominateur de la fonction de transfert analogique, `fs` qui représente la fréquence d'échantillonnage en Hz et enfin `fp` qui correspond à la fréquence de coupure du filtre analogique (en Hz) que la fonction à l'aide du prewarping fera correspondre en numérique.

En sortie elle retourne `numd` et `dend` qui correspondent aux coefficients polynomiaux du numérateur et du dénominateur de la fonction de transfert du filtre numérique créé.

2 Analyse du signal et définition du gabarit idéal

2.1 Analyse du signal initial

On analyse le signal numéro 1 dont le spectrogramme est affiché ci-dessous :

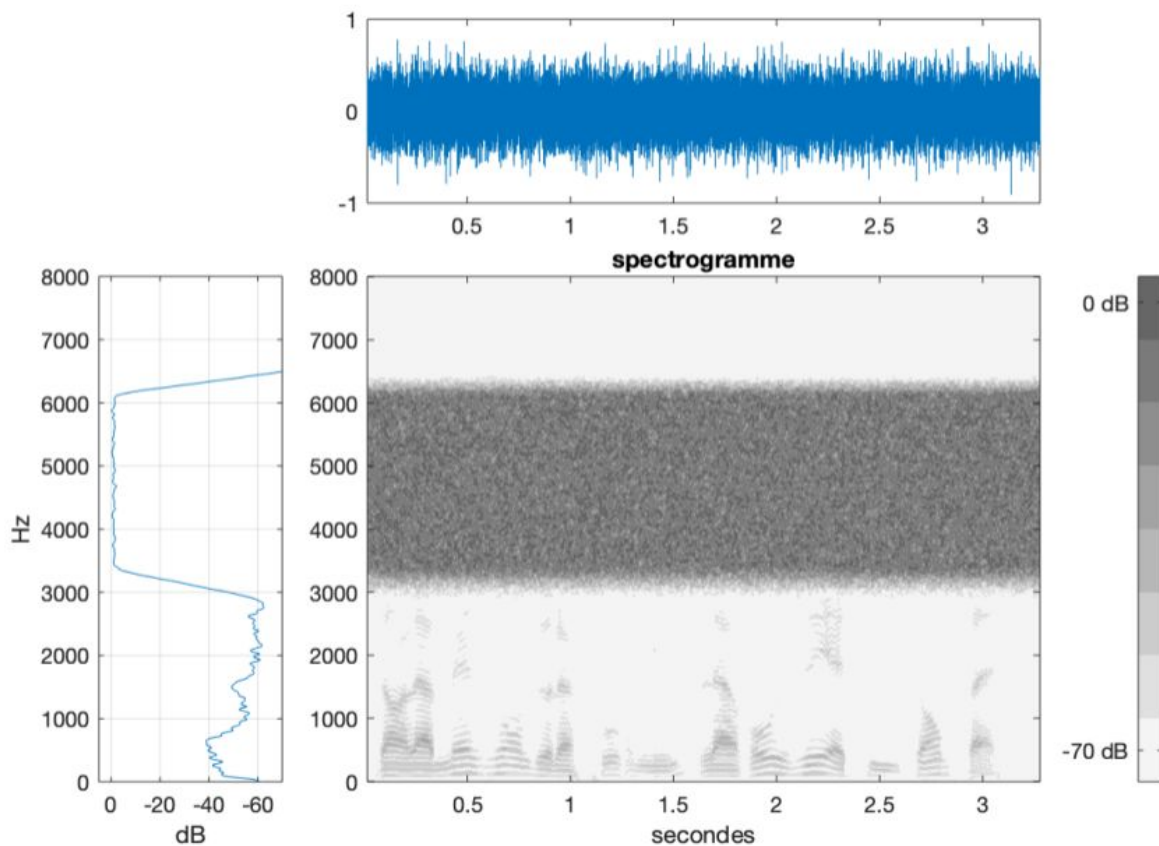


Figure 2.1.1 : Spectrogram signal n°1

On remarque sur cette figure la présence de deux signaux majeurs. De 0 à environ 2800 Hz on observe un faible signal plutôt irrégulier, tandis que de 3000 à 6300 Hz un signal de forte densité spectrale d'énergie est présent. Le signal de faible énergie correspond à une voix humaine tandis que l'autre est un bruitage.

Nous vérifions cette hypothèse à l'aide de la commande `sound` de matlab qui permet d'écouter le son. On entend bien des **voies graves** parler **doucement** alors qu'un **fort signal plus aigu** vient perturber notre écoute.

2.2 Définition du gabarit du filtre idéal

Notre filtre doit donc couper les fréquences à partir de 2800 Hz. Le gabarit de notre filtre doit donc être le suivant :

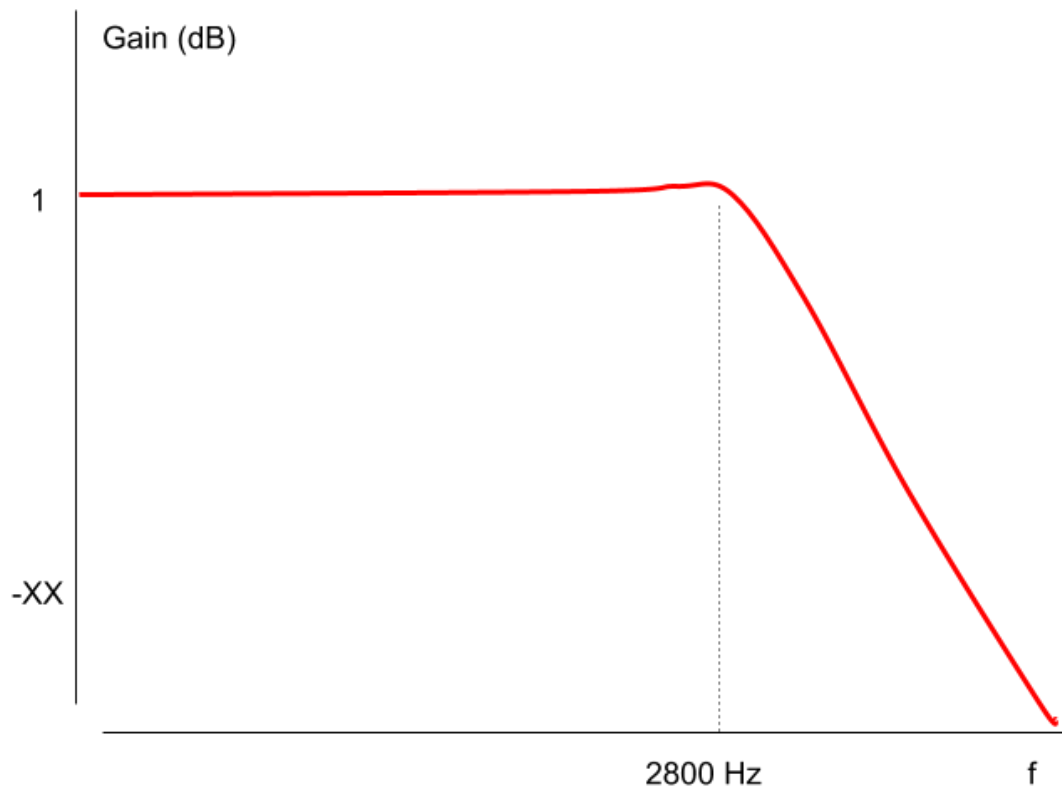


Figure 2.2.1 : Gabarit idéal du filtre

3 Synthèse de filtres RIF

3.1 Synthèse d'un filtre avec une fenêtre rectangulaire

Nous allons maintenant synthétiser un filtre avec la méthode des fenêtres. `fir1` est une fonction matlab qui va permettre de réaliser cette tâche. Cette fonction prend en entrée `n` l'ordre du filtre et `Wn` la liste des fréquences de coupure pour retourner les coefficients du filtre.

Nous souhaitons réaliser un filtre passe-bas il nous faut donc $H_r(0) \neq 0$ et $H_r(1/2) = 0$. La RI doit donc être symétrique et contenir un nombre pair de points. Nous allons donc prendre $n = 45$ pour avoir 46 points et satisfaire la condition précédente.

```
N = 45; # ordre du filtre, taille de la RI
window_1 = rectwin(N+1);

nuc = 2800; # fréquence de coupure en Hz
fc = 2*nuc/nue; # on passe en fréquences réduites
B_1 = fir1(N,fc,window_1);
```

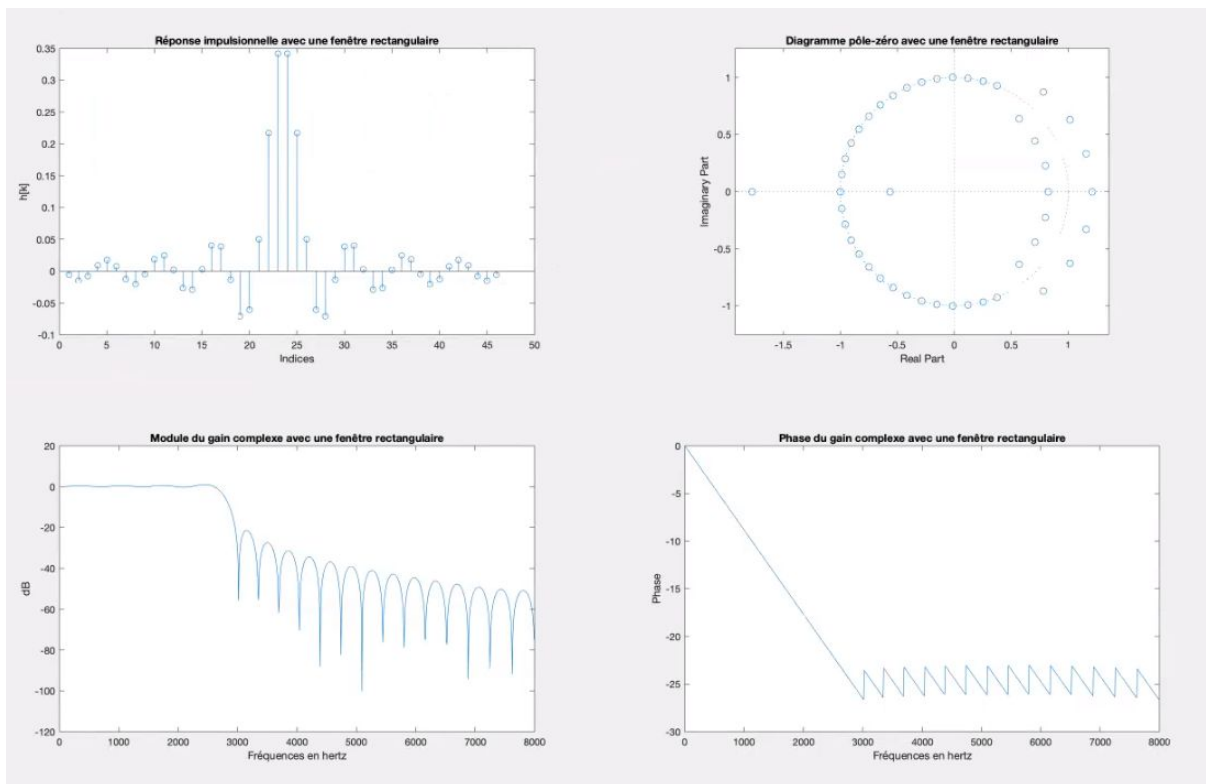


Figure 3.1.1 : Filtre fenêtre rectangulaire

Figure 3.1.1 :

- On remarque sur la première figure que la réponse impulsionnelle est bien symétrique et comporte bien 45 points. (conséquence du caractère fini de la réponse impulsionnelle)
- De manière général on peut mettre évidence la présence d'oscillations au niveau du gain complexe, en effet cela est la conséquence de la dégradation d'un signal par la convolution avec un sinus cardinal issu de la fenêtre rectangulaire.
- Le module du gain complexe a bien un gain de 0dB de 0 à 2800 Hz environ. Les fréquences au delà de 2800 Hz sont atténuées de -40dB environ.
- La phase du filtre est linéaire dans la bande passante. Dans la bande atténuée elle a un aspect en dent de scie car elle fait un "saut" de π à chaque fois que le gain change de signe (cet aspect aurait pu être évité en introduisant un module signé pouvant prendre des valeurs négatives, on aurait alors obtenu une droite affine de coefficient directeur α).

3.2 Synthèse avec une autre fenêtre (Hamming)

Nous allons maintenant synthétiser notre filtre avec une fenêtre de Hamming en utilisant la fonction matlab hamming. On crée la fenêtre de hamming à l'aide de la fonction hamming

```
window_2 = hamming(N+1);
```

```
B_2 = fir1(N,fc>window_2);
```

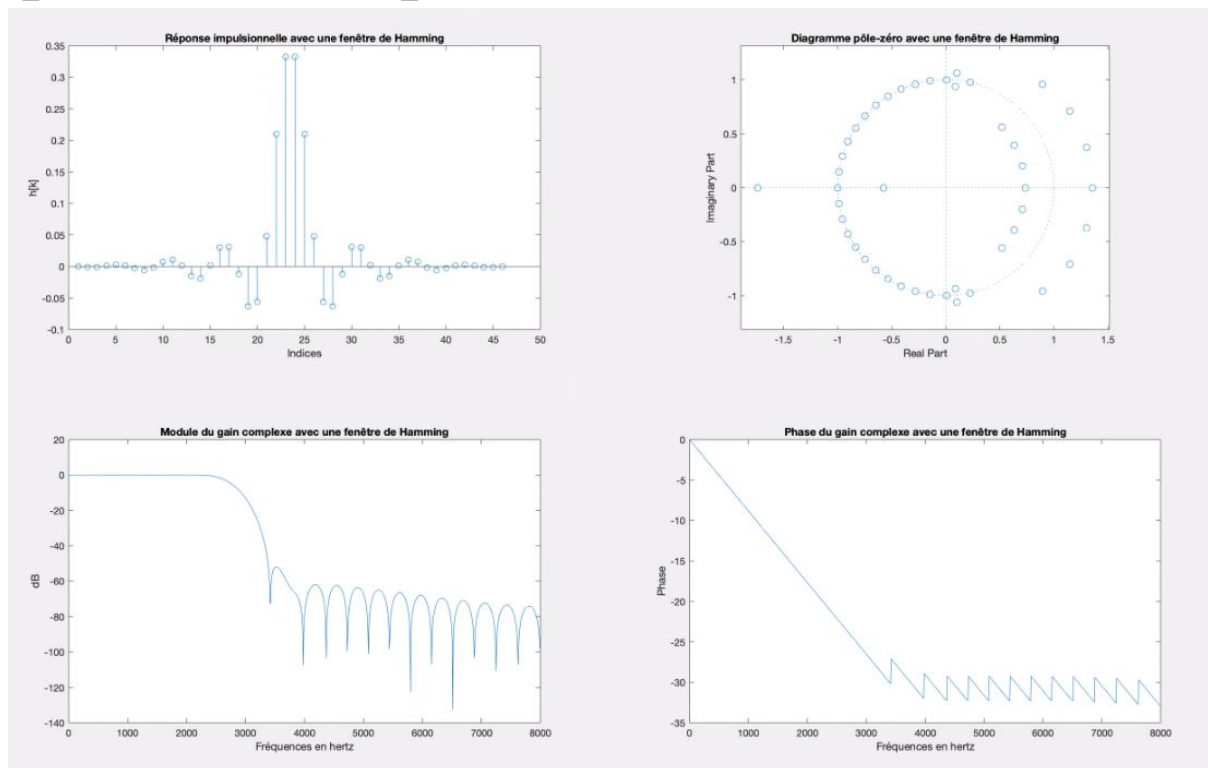


Figure 3.2.1 : Filtre Hamming

Filtre Hamming :

- Les observations sont similaires au filtre précédent.

3.3 Comparaison des caractéristiques des filtres

Nous allons maintenant mesurer les caractéristiques suivantes sur les courbes des filtres rectangulaires et de Hamming :

- Amplitude des oscillations dans les bandes passante et atténuées
- Largeur des oscillations

Fenêtre rectangulaire :

On mesure l'amplitude des oscillations dans la bande passante :

$$1 + \delta_f = 1,013 \text{ dB}$$

$$1 - \delta_f = -0,275 \text{ dB} \quad \text{L'amplitude est de } 1.3 \text{ dB}$$

On mesure l'amplitude des oscillations dans la bande atténuée :

$$\delta_a = -21,46 \text{ dB}$$

$$-\delta_a = -56,13 \text{ dB} \quad \text{L'amplitude est de } 77.7 \text{ dB}$$

Les fréquences d'oscillations est de :

$$f_a = 3016 \text{ Hz}$$

Le fréquence de coupure est à $f_p = 2516 \text{ Hz}$.

Fenêtre de Hamming

On réalise les mêmes mesure avec le filtre de Hamming :

$$1 + \delta_f = 0$$

$$1 + \delta_f = 0 \quad \text{Il n'y a pas d'oscillations dans la bande passante}$$

$$\delta_a = -62\text{dB}$$

$$-\delta_a = -107\text{dB} \quad \text{L'amplitude est de 169 dB}$$

$$f_p = 2500 \text{ Hz}$$

$$f_a = 3422 \text{ Hz}$$

Analyse

On remarque que le filtre Hamming ne présente pas d'oscillations dans la bande passante, ce qui est un atout. Dans la bande atténuée le filtre Hamming a des amplitudes de rebond bien plus élevées (c'est normale, le filtre rectangulaire répartissait les rebonds entre la bande passante et la bande atténuée).

La fréquence des oscillations du filtre Hamming est plus grande de 20%.

Les fréquences de coupures sont sensiblement les mêmes.

Le filtre Hamming présente donc une phase de transition plus grande mais n'a pas d'oscillations au niveau de la bande passante. Il s'agit donc d'un compromis.

3.4 Filtrage (Hamming)

Nous allons maintenant filtrer le signal brouillé avec notre filtre réalisé avec une fenêtre de Hamming.

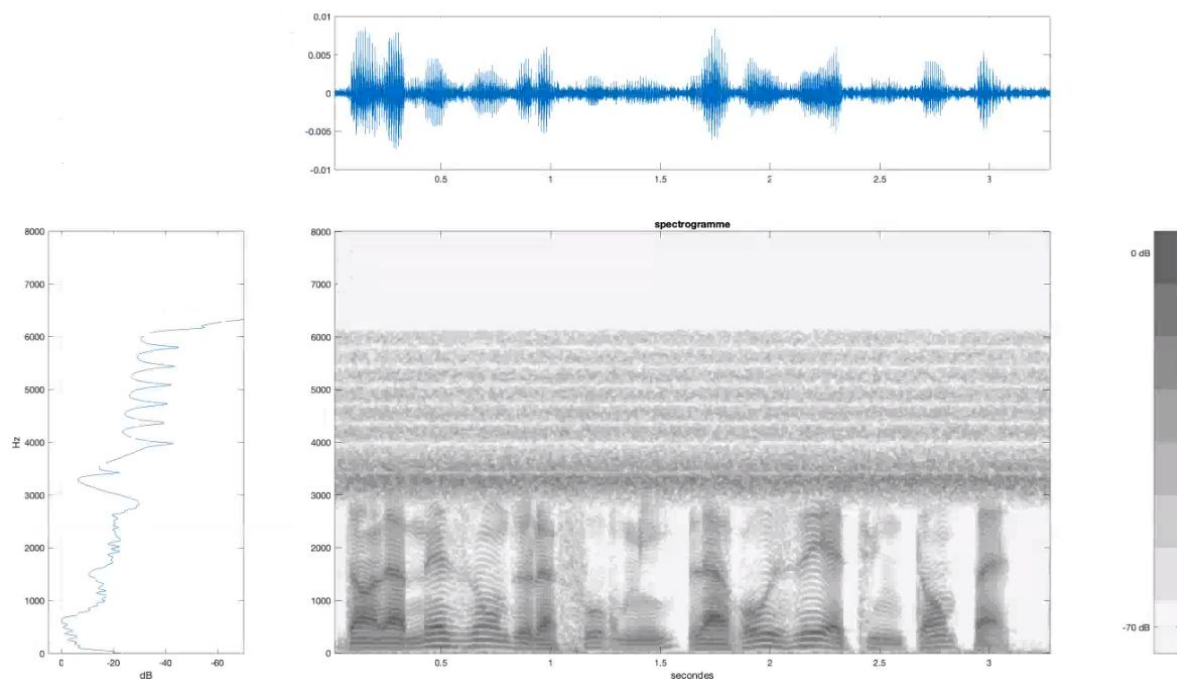


Figure 3.4.1 : Signal filtré (Hamming 46 points, $F_{\text{coupure}} = 2800 \text{ Hz}$)

On remarque sur la figure 3.4.1 que le bruit présent de 3000 à 6000 Hz a été atténué de 20 dB environ (comparaison avec la figure 2.1.1). L'aspect ondulatoire du gain complexe est présent sur ce spectrogramme, on le voit à la densité spectrale d'énergie qui ondule.

Le bruitage très imposant initialement a donc été réduit. Il est maintenant équivalent au signal dans la bande passante. Ce filtre ne permet donc pas de répondre au cahier des charges dans la mesure où le bruit est encore très présent. Nous allons donc l'améliorer.

3.5 Amélioration du filtrage

Il est possible de réaliser un meilleur filtrage en augmentant M , la largeur de la fenêtre. Ainsi, la réponse impulsionnelle sera plus grande et le filtre disposera de plus de zéro pour atténuer la bande coupée.

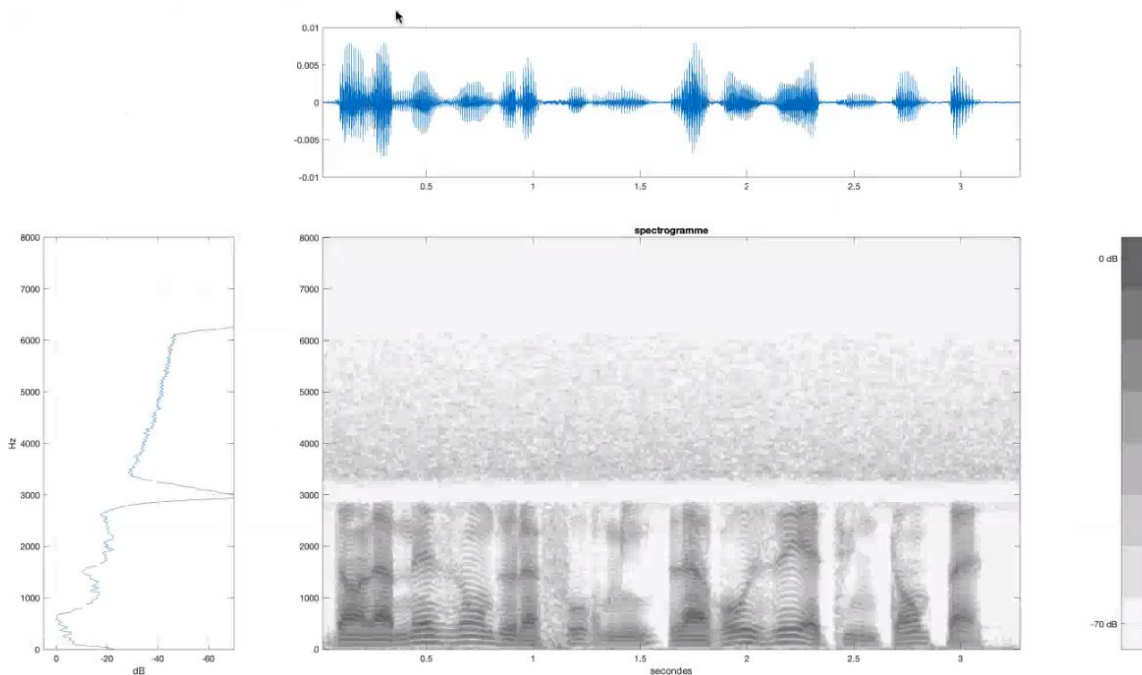


Figure 3.5.1 : Spectrogramme pour une fenêtre de 250 points

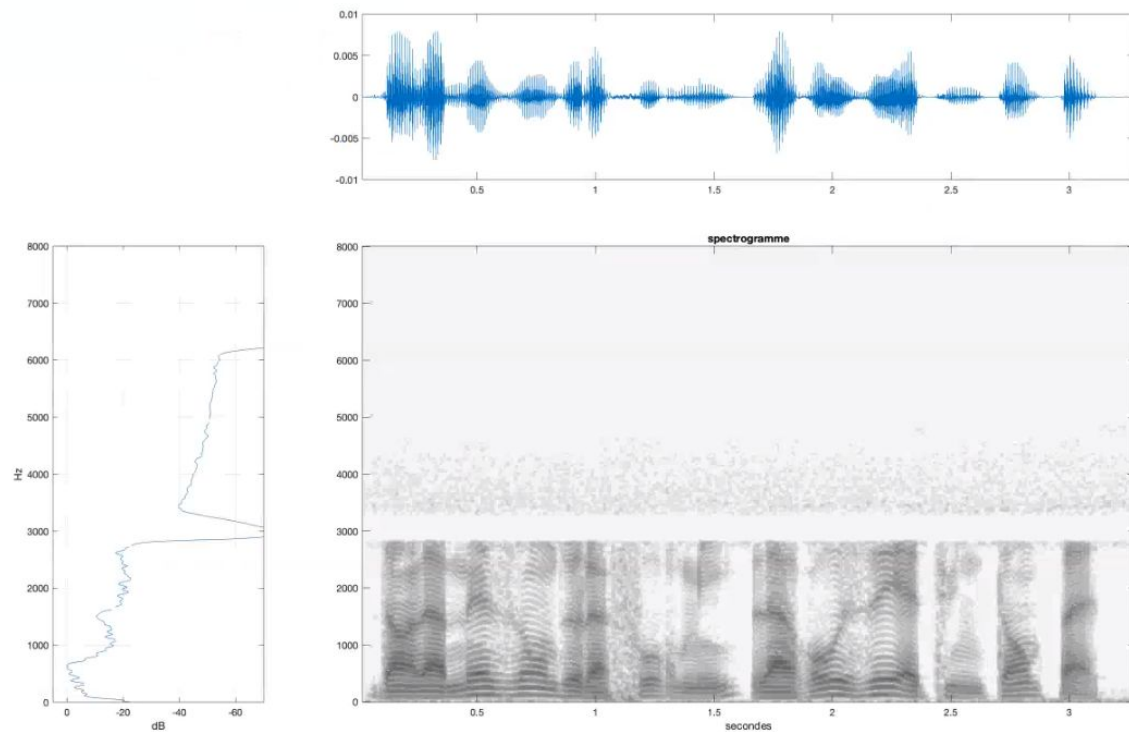


Figure 3.5.2 : Spectrogramme pour une fenêtre de 1000 points

On remarque sur les deux figures précédentes (3.5.1 et 3.5.2) que plus M la largeur de la fenêtre est grande, plus le filtrage est efficace.

On obtient ainsi, une atténuation de 40dB avec une fenêtre de 250 points et de 50 dB avec une fenêtre de 1000 points.

Pour une fenêtre de 1000 points on peut considérer que le cahier des charges est rempli.

Cette méthode permet d'augmenter l'efficacité du filtre mais est très chère à mettre en oeuvre puisqu'elle impose d'augmenter considérablement l'ordre du filtre.

Il est cependant possible de combiner cette technique avec une autre astuce. En effet il est aussi possible déplacer la fréquence de coupure du côté de la bande passante. En sacrifiant une petite partie du signal on repousse la gain de la bande coupée.

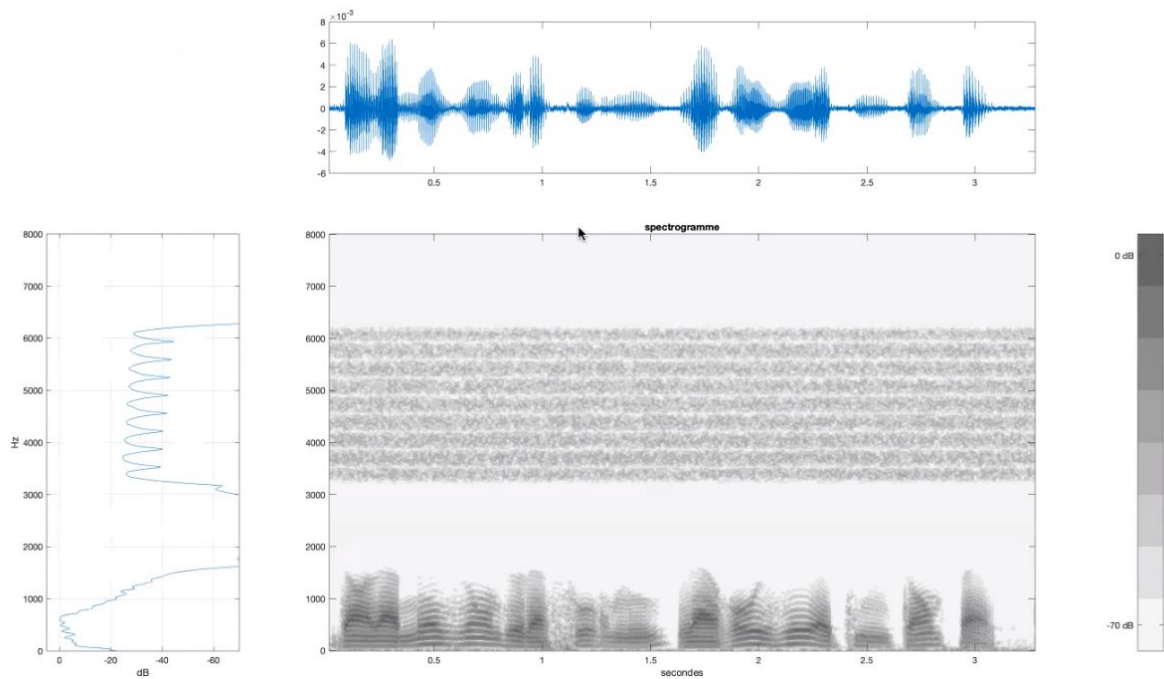


Figure 3.4.3 : Spectrogram (Hamming 46 points $F_{\text{coupure}} = 1000 \text{ Hz}$)

On remarque sur la figure 3.4.3 qu'il est possible de repousser la bande coupée à -20dB dès 3000 Hz avec une fenêtre de seulement 46 points, mais cela demande de perdre une grande partie du signal (de 1800 à 2800 Hz).

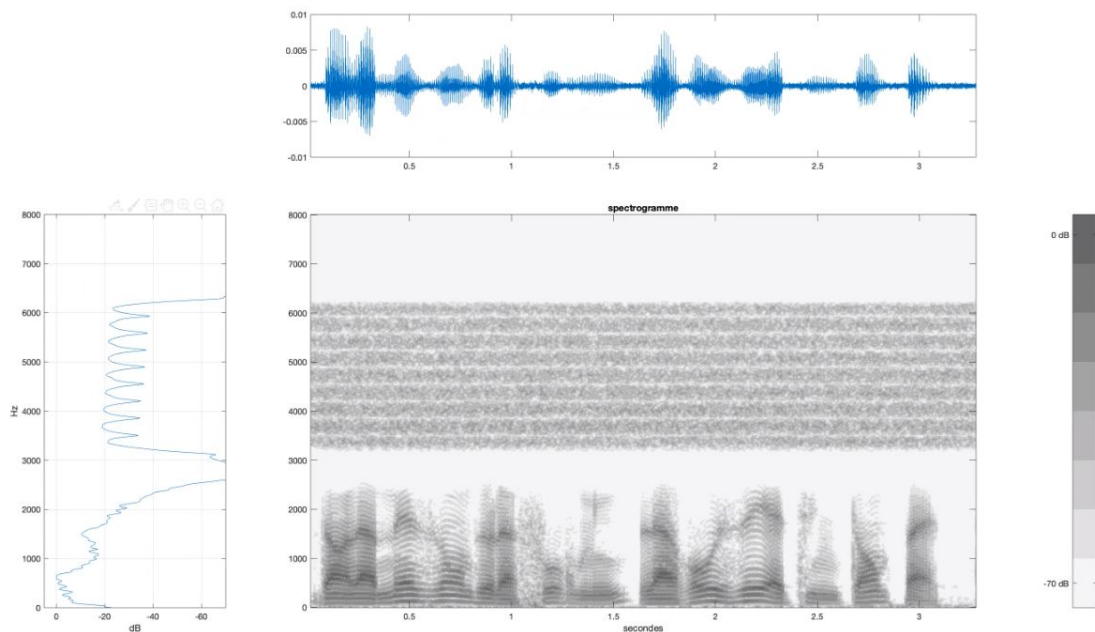


Figure 3.4.4 : Spectrogram (Hamming 46 points $F_{\text{coupure}} = 2800 \text{ Hz}$)

La figure 3.4.4 présente un compromis plus judicieux puisqu'avec une fréquence de coupure à 200 Hz et seulement 46 points pour la fenêtre on filtre le bruit dès 2500 Hz avec une bonne atténuation (-20dB).

4 Synthèse de filtres RII

Le but de cette partie est de filtrer le même signal que dans la partie précédente mais en utilisant un autre type de filtre, un filtre analogique.

Nous allons concevoir et utiliser un filtre à réponse impulsionnelle infinie de type elliptique / passe-bas.

4.1 Conception du filtre analogique

La syntaxe de la fonction `ellip` est la suivante, (sa description est rappelée en préparation) :

```
[b,a] = ellip(n,Rp,Rs,Wp)
```

On prendra $n = 7$, c'est l'ordre du filtre.

$R_p = 1$ correspond à l'atténuation à la fréquence de coupure.

$R_s = 150$ correspond à l'atténuation de la bande de coupure, elle ne doit pas être trop grande sous peine d'atténuer les performances du filtre.

W_p est la fréquence de coupure, on prendra $\omega_p = 2\pi 2800 \text{ rad/s}$.

```
w_nuc=2*pi*nuc; %frequence de coupure en rad/s
W=0:1:2*pi*nue;
Rp=1; %atténuation à la frequence de coupure
Rs=150; %atténuation au niveau de la bande atténuée
n=9; %ordre du filtre
[b,a]=ellip(n,Rp,Rs,w_nuc,'s');
[H_ana,nu_ana]=freqs(b,a,W2);
G_H_ana=20*log10(abs(H_ana)); % on calcule le gain
ang_H_ana = unwrap(angle(H_ana)); % la phase
```

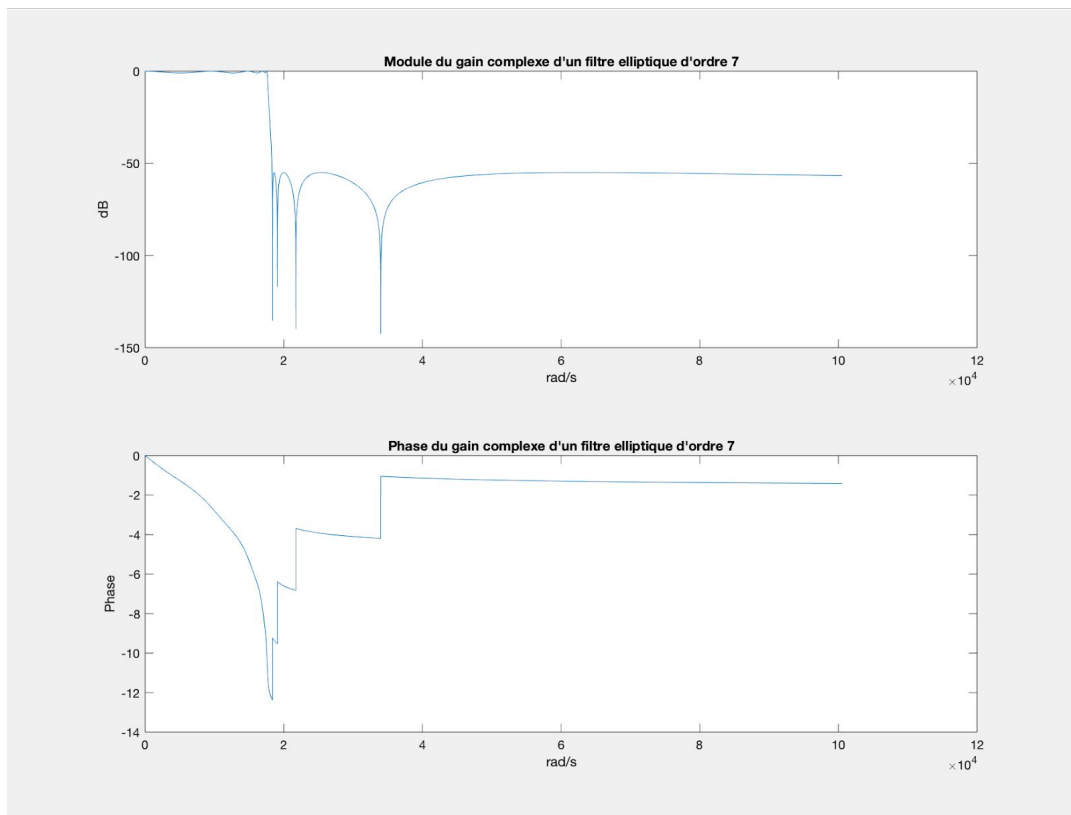


Figure 4.1.2 : Gain et phase du filtre analogique

- On remarque sur la figure 4.1.2 que le signal est très atténué à partir de la fréquence 20000 rad, ce qui correspond à une fréquence de 3000 Hz environ. Cela correspond au cahier des charges.
- On remarque que les rebonds sont répartis sur la bande passante et la bande atténuée, cela est caractéristique des filtres elliptiques.
- La bande de transition est très restreinte.
- On retrouve la stabilité du filtre elliptique.

4.2 Synthèse du Filtre numérique

Nous allons maintenant réaliser le filtre numérique correspondant à l'aide de la fonction `bilinear`, sa description a été rappelée dans la préparation.

```
[B_n,A_n] = bilinear(b,a,nue,nuc);
[H_num,nu_num] = freqz(B_n,A_n,N_suite,nue);
[h_num, k_num] = impz(B_n,A_n);
G_H_num = 20*log10(abs(H_num));
ang_H_num = unwrap(angle(H_num));
```

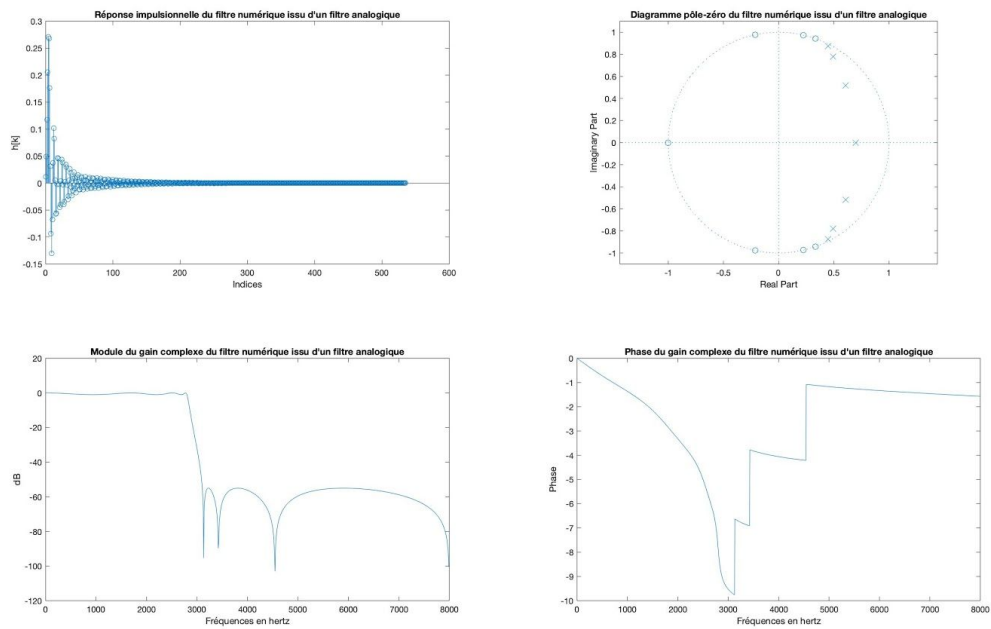


Figure 4.2.1 : Filtre numérique créé à partir d'un filtre analogique

De manière générale les caractéristiques du filtre numérique créé sont similaires à celles du filtre elliptique. En effet, si l'on observe les tracés de module et phase de gain complexe, on peut remarquer la forte ressemblance entre le filtre numérique et le filtre analogique. On observe une bande passante d'une largeur de 2800 Hz environ et une bande atténuée à -60dB.

Si l'on s'intéresse aux fréquences critiques f_a et f_c , on mesure $f_a = 3133 \text{ Hz}$ et $f_p = 2773 \text{ Hz}$. Ce qui donne une largeur de bande de transition à 360 Hz. Ces résultats très proches de ceux mesurés avec le filtre analogique.

De plus, on a $\delta_a = 48 \text{ dB}$ et $\delta_f = 0,9 \text{ dB}$, les oscillations en bande passante sont très faibles alors que les oscillations en bande atténuées sont assez importantes.

C'est ainsi qu'on retrouve les propriétés d'un filtre elliptique, c'est à dire une bande de transition assez faible, des oscillations en sortie importantes et la stabilité.

4.3 Filtrage

Nous allons maintenant filtrer notre signal à l'aide du filtre que nous venons de synthétiser.

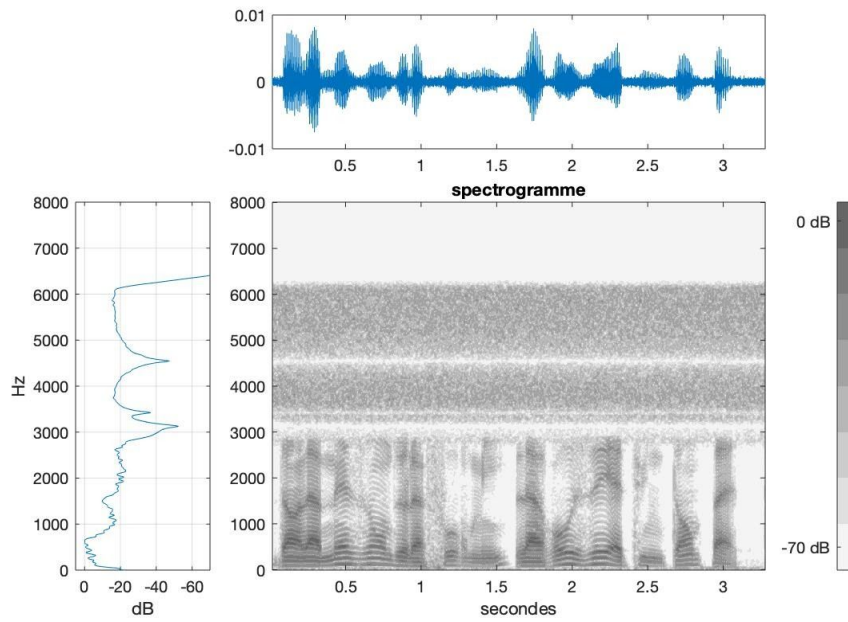


Figure 4.3.1 : Signal filtré

Après passage à travers notre filtre, on obtient le signal de la figure 4.3.1. Le bruit a été considérablement atténué.

4.4 Optimisation

Dans cette partie, nous allons faire varier l'ordre du filtre ainsi que l'atténuation de la bande coupée (R). Nous allons aussi faire varier la fréquence de coupure.

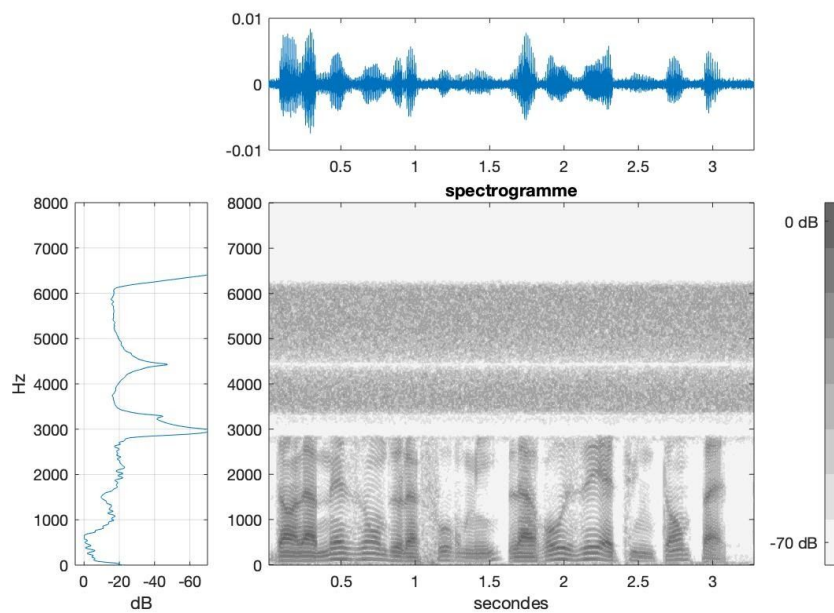


Figure 4.4.1 : $n = 8$

Sur la figure 4.1.1, on remarque que l'ordre $n=8$ ne suffit pas à filtrer correctement le signal. Le bruit est toujours très présent.

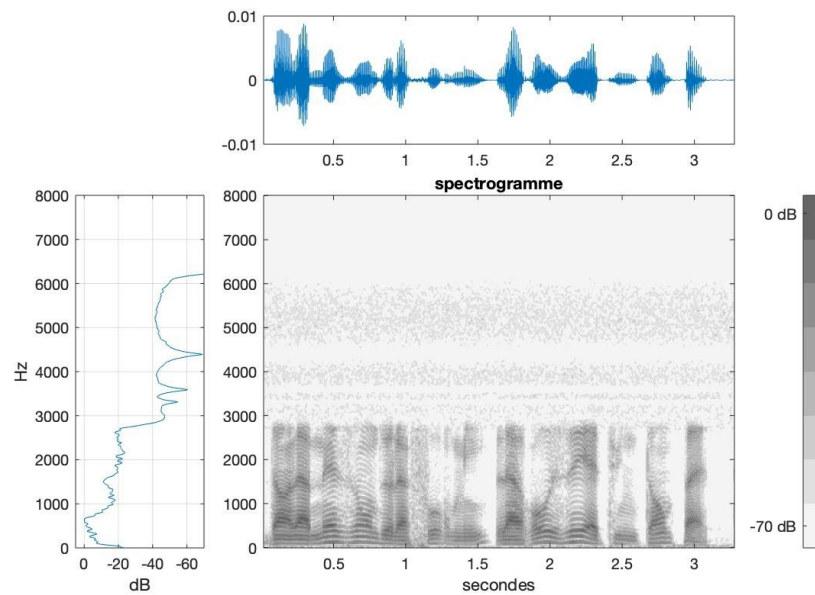


Figure 4.4.1 : $n=8$, $f=2720$ et $R_a = 80$ (initialisé à 50 au début)

Avec $n=8$ et R_a (atténuation en bande atténuée) = 80, on obtient un filtrage convenable, le bruit est environ 50 dB en dessous du signal que l'on souhaite conserver.

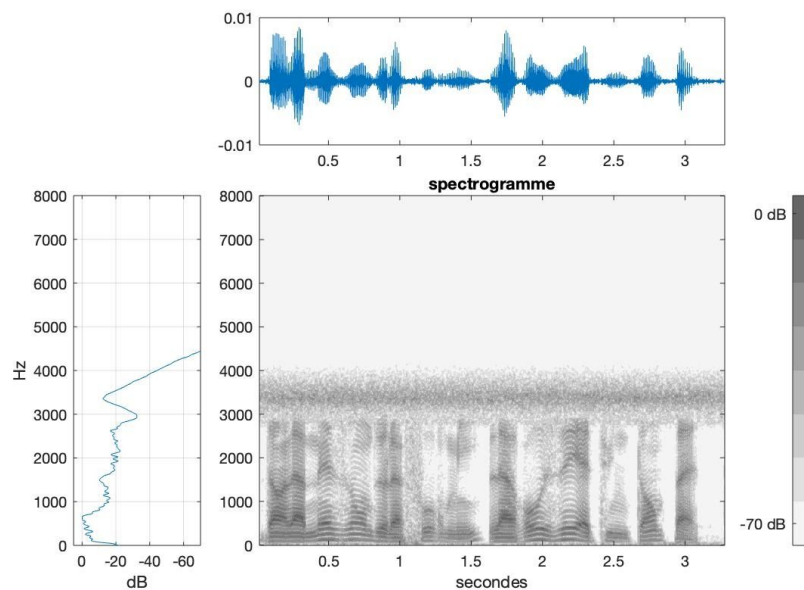


Figure 4.4.1 : $n = 9$ $R = 150$

Nous avons maintenant augmenté l'ordre du filtre ainsi que R . Bien qu'on puisse s'attendre à un meilleur résultat, l'augmentation de R se fait au prix d'une bande de transition plus large, il persiste donc un bruit important aux alentours des 3500 Hz.