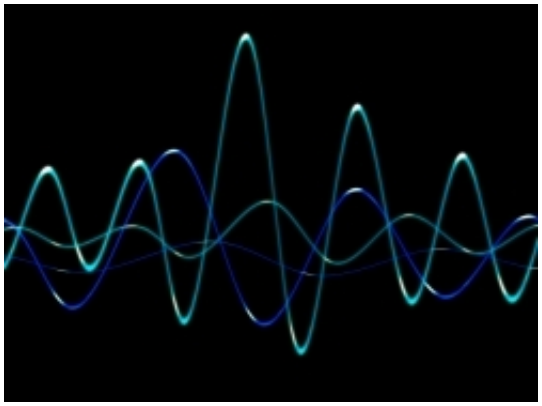
ES101 - Rapport de projet

Hubert Bastien - Mathis Le Bail

27 octobre 2023

Table des matières



1 Introduction

Dans le cadre du cours ES101, nous avons réalisés trois séances de travaux pratiques, portant chacun sur un sujet, permettant de travailler sur plusieurs projets de traitement du signal. Nous avons utilisés le logiciel Matlab pour l'ensemble de nos travaux.

2 Sujet 1

2.1 Premier projet - Filtrage

Nous disposions d'un enregistrement audio bruité par des fréquences parasites que nous devions filtrer pour retrouver l'enregistrement initial. Pour ce faire, nous avons commencé par tracer le spectre du signal afin de repérer le bruit :

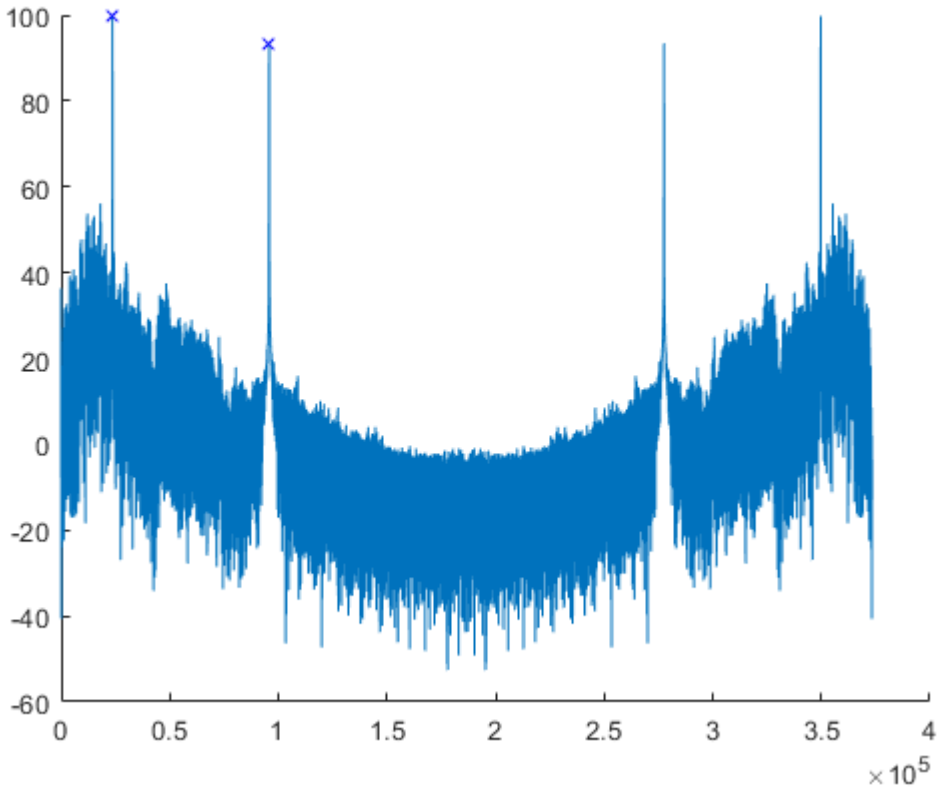


FIGURE 1 – Spectre du signal non filtré

Sur le spectre on remarque la présence de deux pics de fréquences (par symétrie du spectre réel autour de $f = \frac{F_e}{2}$, marqués par des croix sur la figure ci-dessus : ce sont les fréquences parasites qui bruitent le signal.

Nous implémentons donc un filtre réjecteur de type RII coupant les deux fréquences parasites de la forme :

$$H(\,Z\,) := \frac{(Z - Z_0)\,(Z - Z_1)}{(Z - P_0)\,(Z - P_1)} \quad \text{avec } Z_k := e^{2i\,\pi\,f_{par_k}} \text{ et } P_k := 0.95\,Z_k$$

En effet, la transformée en Z donne directement les valeurs de zéros de la fonction de transfert, qui atteint son efficacité maximale quand chaque pôle a le même argument que son zéro associé et en est proche en norme. Pour que le filtre soit en plus stable, il est nécessaire que les pôles se situent à l'intérieur du cercle unité : on fixe donc à 0.95 le rapport des normes.

On obtient alors le spectre filtré ci-dessous, qui est bien débarrassé des fréquences parasites qu'on cherchait à éliminer :

On aboutit donc au code Matlab suivant :

```
clear all;
close all
clc;
hold on;

[x, Fe] = audioread( "Mo11.wav" ); % lecture du fichier
N = length( x );

X = abs( fft( x ) ); % calcul de la transformee de Fourier
plot( 20 * log10( X ) );

[S, I] = sort( X( 1 : N / 2 ), "descend" ); % recuperation des frequences
    parasites
f_par = ( I( 1 : 2 ) - 1 ) / N;
plot( I( 1 : 2 ), 20 * log10( S( 1 : 2 ) ), "bx" );

Z = exp( 2i * pi * f_par ); % creation du filtre coupe-bande
P = 0.95 * Z;
Num = conv( conv( [1 -Z( 1 )], [1 -Z( 2 )] ), conv( [1 -conj( Z( 1 ) )],
    [1 -conj( Z( 2 ) )] ) );
Den = conv( conv( [1 -P( 1 )], [1 -P( 2 )] ), conv( [1 -conj( P( 1 ) )],
    [1 -conj( P( 2 ) )] ) );

```

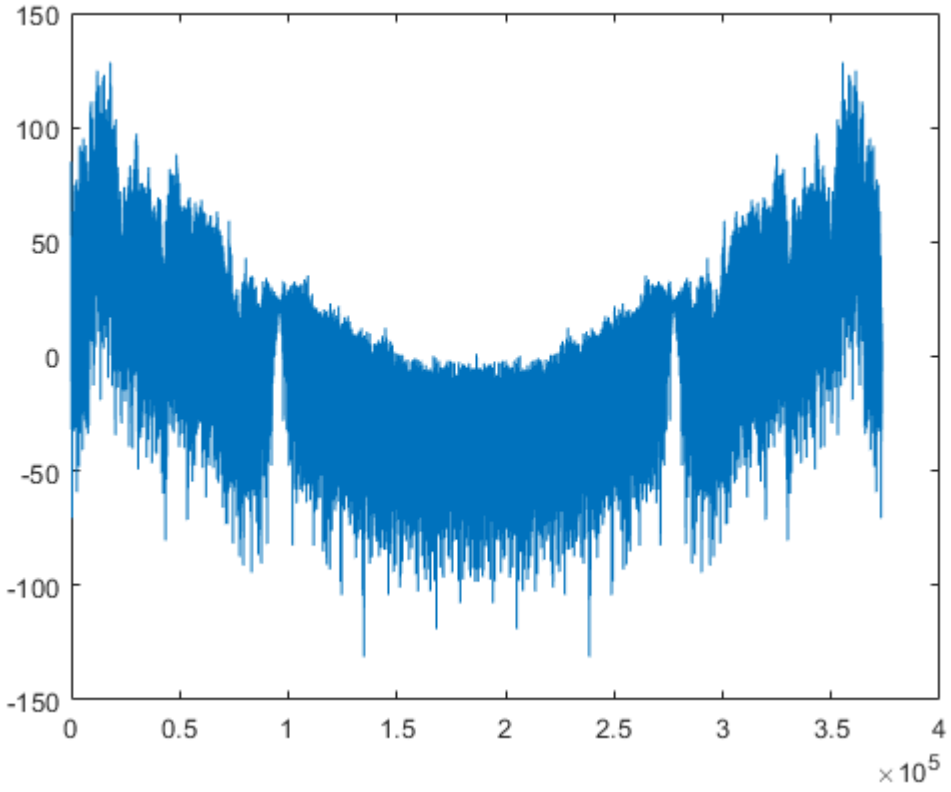


FIGURE 2 – Spectre du signal filtré

```
figure( 2 ); % filtrage du signal
y = filter( Num, Den, x );
Y = abs( fft( y ) );
plot( 20 * log( Y ) );

audiowrite( "son.wav", y, Fe ); % enregistrement du fichier audio filtre
```

2.2 deuxième projet - Son tournant

Dans ce projet, on dispose d'un enregistrement audio mono que l'on souhaite transformer en enregistrement stéréo avec effet de mouvement de la source. On utilise la formule approximée suivante pour créer un retard :

$$x_2(\, nT_e \,) \approx x_1 \left(\left(n - \left\lfloor \frac{\tau}{T_e} \right\rfloor \right) T_e \right)$$

De plus, la trigonométrie nous montre que, pour un angle θ, le retard du son est donné par la formule $\tau = \frac{d \sin(\theta)}{v}$ avec $d = 15\,cm$ la largeur moyenne du visage et $v = 340\,m/s$ la vitesse du son dans l'air. Sous Matlab, le code permettant d'obtenir cet effet est donc le suivant :

```
clear all;
close all
clc;

[x1, Fe] = audioread( "Erlk.wav" ); % lecture du fichier
N = length( x1 );

teta = [0 : pi / ( N - 1 ) : pi]; % creation de l'effet stereo
tau = ( 0.15 * sin( teta ) ) / 340;
x2 = zeros (N ,1) ;
for i = 1 : N
    if round( tau(i) * Fe ) < i
        x2(i) = x1( i - round( tau(i) * Fe ) );
    else
        x2(i) = 0;
    end
end
y = [x1 x2];
audiowrite( "resultat.wav", y, Fe ); % enregistrement du fichier stereo
```

3 Sujet 2

3.1 Premier projet - Echo

Le signal récupéré en sortie x(n) = s(n) + α s (n - p_0) présente un echo. Pour pouvoir corriger celui-ci, il faut d'abord à estimer le paramètre d'atténuation α et celui de retard p_0.

Le signal récupéré présente un pic principal correspondant au signal d'origine et un pic dû à l'echo. Le calcul de la focntion d'autocorrélation du signal x permet de fournir α et p_0. En effet, la fonction d'autocorélation présente 3 pics. Le pic central, le plus important qui correspond à la superposition exacte du signal x avec lui-même, il est atteint en 0 et sa hauteur est donc égale à $r_{xx}(0)$. Les deux autres pics sont obtenus lorsque le pic d'echo coincide avec le pic principal associé au signal d'origine lors du calcul de la fonction d'autocorrélation, cela est le cas par deux fois.

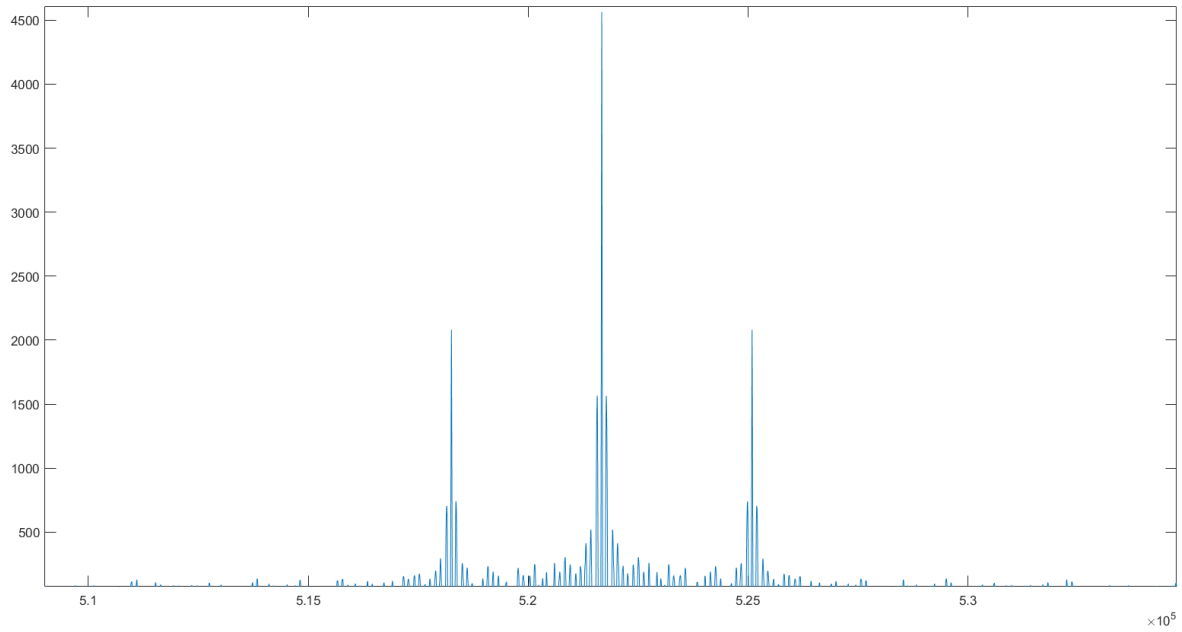


FIGURE 3 – Fonction d'autocorrélation du signal : le pic central est bien supérieur aux deux autres pics et correspond à $r_{xx}(0)$

La recherche du maximum de xcorr nous donne l'indice n_0 qui fait correspondre xcorr(n_0) = $r_{xx}(0)$. Puis, en déterminant un intervalle adéquat grâce à la lecture graphique, on détermine l'indice n_1 qui correpond à l'indice du pic secondaire à droite : xcorr(n_1) = $r_{xx}(p_0)$. Ainsi p_0 = | n_1 - n_0 |. Il reste à trouver α.

On a :

$$\begin{aligned} r_{xx}(0) &= E[x(n)^2] = E[(s(n) + \alpha s(n - p_0))^2] \\ &= E[s(n)^2] + \alpha^2 E[s(n - p_0)^2] + 2\alpha E[s(n)s(n - p_0)] \end{aligned} \tag{1}$$

$$\begin{aligned} r_{xx}(p_0) &= E[x(n)x(n - p_0)] = E[(s(n) + \alpha s(n - p_0))(s(n - p_0) + \alpha s(n - 2p_0))] \\ &= E[s(n)s(n - p_0)] + \alpha E[s(n - 2p_0)s(n)] + \alpha E[s(n - p_0)^2] + \alpha^2 E[s(n - p_0)s(n - 2p_0)] \end{aligned} \tag{2}$$

Or on a fait l'hypothèse que le signal d'origine s(n) était blanc, ainsi il est décoréllé et centré donc :

$$E[s(n)s(n - p_0)] = E[s(n)]E[s(n - p_0)] = 0 \tag{3}$$

car E[s(n)] = 0
Egalement, avec le même raisonnement, on a :

$$E[s(n)s(n-2p_0)] = E[s(n-p_0)s(n-2p_0)] = 0 \tag{4}$$

s(n) est également WSS donc :

$$E[s(n)^2] = E[s(n-p_0)^2] \tag{5}$$

Finalement :

$$\frac{r_{xx}(0)}{r_{xx}(p_0)} = \frac{E[s(n)^2] + \alpha^2 E[s(n-p_0)^2]}{\alpha E[s(n-p_0)^2]} = \frac{1+\alpha^2}{\alpha} \tag{6}$$

D'où :

$$\alpha^2 - \frac{r_{xx}(0)}{r_{xx}(p_0)}\alpha + 1 = 0 \tag{7}$$

Cette équation du second ordre donne deux valeurs de alpha possibles, mais la condition $\alpha < 1$ impose de choisir la plus petite des racines. On obtient alors la valeur de alpha. Il est alors possible de modéliser le filtre connaissant α et p_0.

$$S(Z) = \frac{1}{1+\alpha Z^{-p_0}} X(Z) \tag{8}$$

La fonction d'autocorrélation du signal récupéré en sortie du filtre est représentée par la figure ci-dessous :

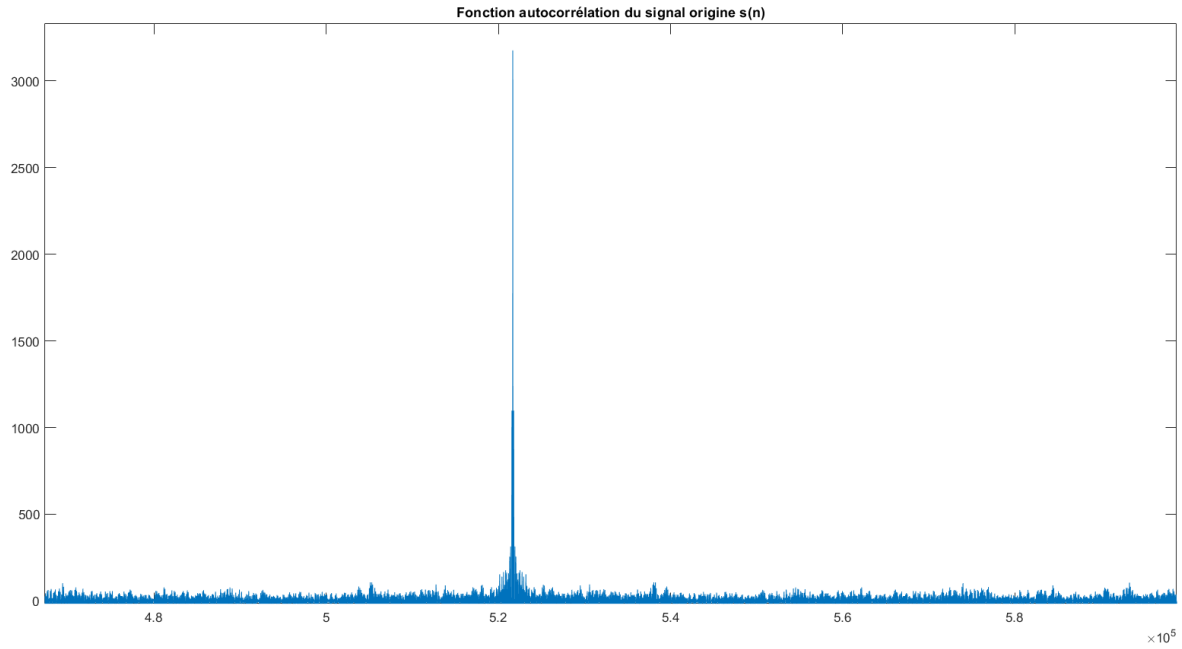


FIGURE 4 – Fonction d'autocorrélation du signal après traitement : il ne reste plus que le pic central en $r_{xx}(0)$

Il ne reste plus que le pic central, l'echo a bien été supprimé.

3.2 Deuxième projet - Inversion de puissance

Pour récupérer au mieux le signal s(n) d'origine grâce à $x_1(n)$ et $x_2(n)$, on combine linéairement ceux-ci sous la forme $\epsilon(n) = x_1(n) - \rho x_2(n)$. Cherchons le ρ qui minimise $E[\epsilon(n)^2]$.

On a :

$$E[\epsilon(n)^2] = E[x_1^2] - 2\rho E[x_1 x_2] + \rho^2 E[x_2^2] \tag{9}$$

Donc le ρ qui minimise notre expression est égal à $\frac{E[x_1 x_2]}{E[x_2^2]}$. Si on dessine $E[\epsilon(n)^2]$ en fonction de ρ on obtient la courbe suivante :

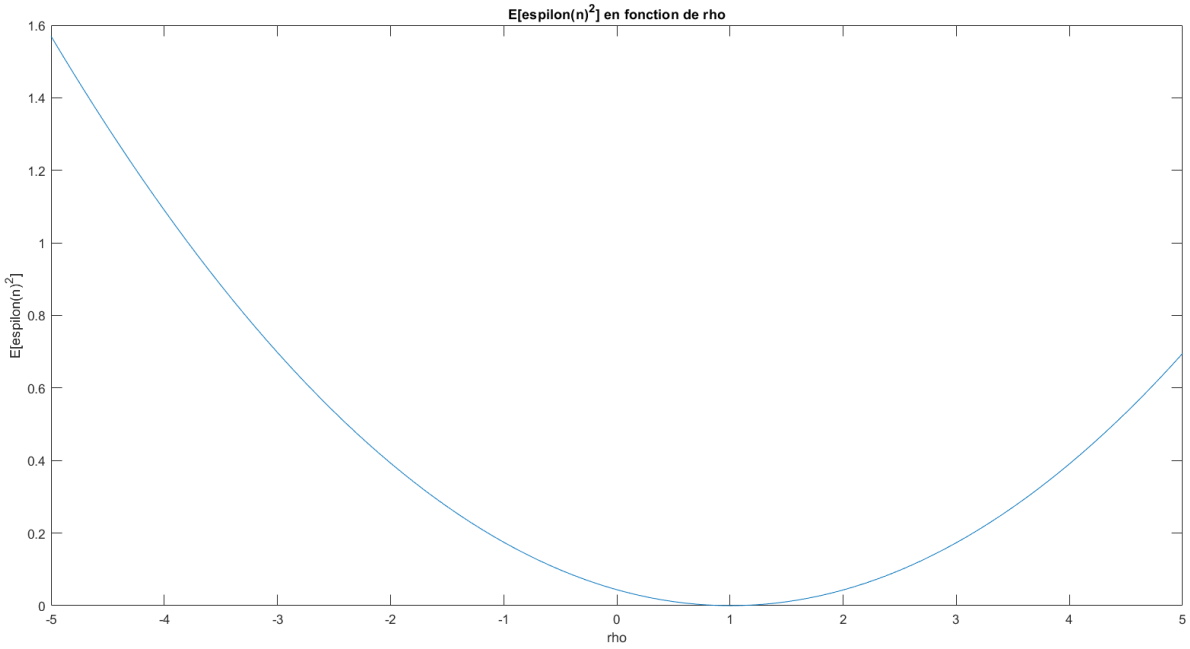


FIGURE 5 – Courbe représentant $E[\epsilon(n)^2]$ en fonction de ρ

D'après celle-ci, $E[\epsilon(n)^2]$ admet bien un minimum proche de 1. C'est effectivement ce que l'on trouve en calculant $\frac{E[x_1x_2]}{E[x_2^2]} = 0.9949$.

En choisissant ce $\rho_0 = \frac{E[x_1x_2]}{E[x_2^2]}$, $\epsilon(n)$ et $x_2(n)$ sont décorrélés. En effet, le ρ choisi vérifie $\frac{dE[\epsilon(n)^2]}{d\rho} = 0$. D'où :

$$\frac{dE[\epsilon(n)^2]}{d\rho}(\rho_0) = -2E[x_1x_2] + 2\rho_0 E[x_2^2] = 0 \Leftrightarrow E[x_1x_2] - \rho_0 E[x_2^2] = 0 \tag{10}$$

Ainsi

$$\begin{aligned} E[\epsilon(n)x_2(n)] &= E[x_1(n)x_2(n) - \rho_0 x_2(n)^2] \\ &= E[x_1(n)x_2(n)] - \rho_0 E[x_2(n)^2] \\ &= 0 \end{aligned} \tag{11}$$

$\epsilon(n)$ et $x_2(n)$ sont bien décorrélés.

Pour montrer l'inversion du rapport signal sur bruit entre $x_2(n)$ et $\epsilon(n)$, on suppose pour simplifier que $E[s(n)^2] = 1$ et $E[w(n)^2] = 1$ et on écrit $\epsilon(n) = a_3 s(n) + b_3 w(n)$. En utilisant que $\epsilon(n)$ et $x_2(n)$ sont décorrélés, on a :

$$
\begin{aligned}
& E[\epsilon(n)x_2(n)] = 0 \\
\Leftrightarrow\ & E[(a_3 s(n) + b_3 w(n))x_2(n)] = 0 \\
\Leftrightarrow\ & E[a_3 s(n)x_2(n) + b_3 w(n)x_2(n)] = 0 \\
\Leftrightarrow\ & E[a_3 a_2 s(n)^2 + a_3 b_2 s(n)w(n) + b_3 a_2 s(n)w(n) + b_3 b_2 w(n)^2] = 0 \\
\Leftrightarrow\ & a_3 a_2 E[s(n)^2] + a_3 b_2 E[s(n)w(n)] + b_3 a_2 E[s(n)w(n)] + b_3 b_2 E[w(n)^2] = 0 \\
\Leftrightarrow\ & a_3 a_2 + b_3 b_2 = 0
\end{aligned}
\tag{12}
$$

En utilisant que $E[s(n)^2] = E[w(n)^2] = 1$ et $E[s(n)w(n)] = E[s(n)]E[w(n)] = 0$ car s(n) et w(n) sont indépendants et E[w(n)] = 0 car w(n) est un bruit blanc.

Le résultat obtenu se traduit par $\mid a_3 a_2 \mid = \mid b_3 b_2 \mid$. D'où $\frac{a_3^2}{b_3^2} = \frac{b_2^2}{a_2^2}$. Or le rapport signal sur bruit du signal $x_2(n)$: $\mathrm{SNR}(x_2) = \frac{E[|a_2 s(n)|^2]}{E[|b_2 w(n)|^2]} = \frac{a_2^2}{b_2^2}$ et celui du signal $\epsilon(n)$: $\mathrm{SNR}(\epsilon(n)) = \frac{E[|a_3 s(n)|^2]}{E[|b_3 w(n)|^2]} = \frac{a_3^2}{b_3^2}$.

On retrouve donc bien $\mathrm{SNR}(\epsilon(n)) = \frac{1}{SNR(x_2)}$. L'inversion de puissance est bien réalisée. De plus, la relation $b_2 \gg a_2$ donne que $a_3 \gg b_3$: $\epsilon(n) = a_3 s(n) + b_3 w(n) \approx a_3 s(n)$. Le signal de sortie comporte essentiellement le signal d'origine s(n).

3.3 Troisième projet - Canal +

Pour récupérer le signal d'origine, on permute les hautes et les basses fréquences de la Transformée de Fourier du signal dans le sens inverse. La Transformée de Fourier du signal canal.wav donne la figure ci-dessous :

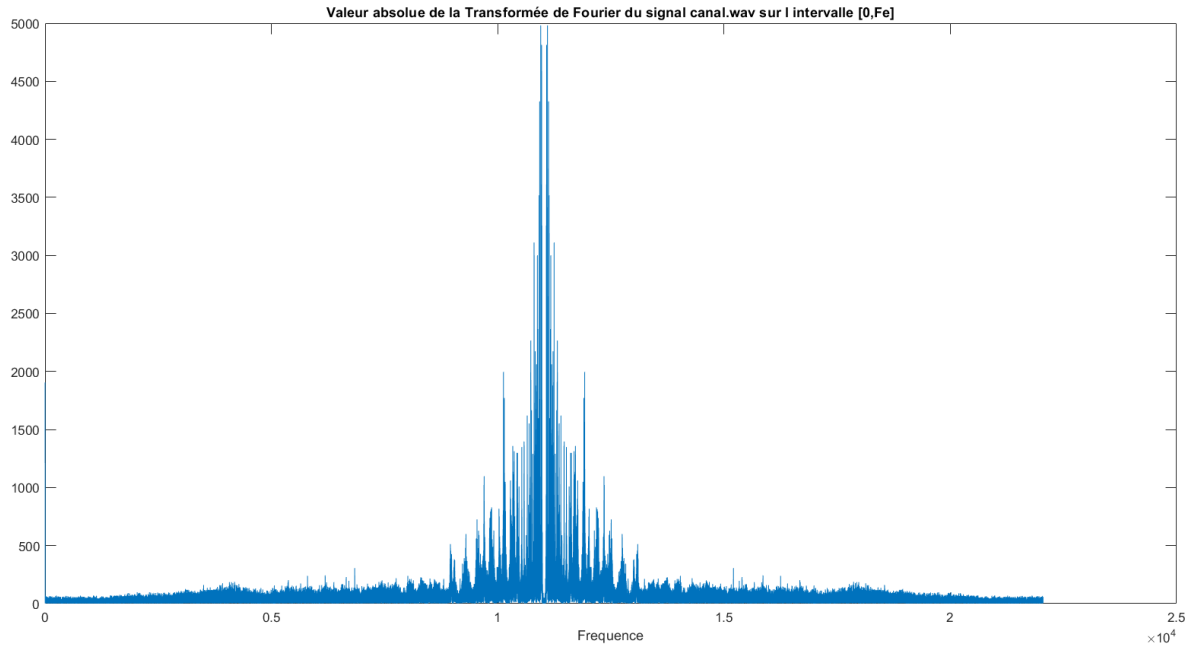


FIGURE 6 – Valeur absolue de la Transformée de Fourier du signal canal.wav sur l'intervalle [0,Fe]

Après avoir obtenu la Transformée de Fourier du signal, on découpe celle-ci en deux échantillons, l'un allant de $\frac{Fe}{N}$ à $Fe(\frac{1}{2} - \frac{1}{N})$ et l'autre allant de $Fe(\frac{1}{2} + \frac{1}{N})$ à $Fe(1 - \frac{1}{N})$. Puis on permute chacun de ces échantillons grâce à la fonction **flipud** car les valeurs discrètes de la Transformée de Fourier sont stockées dans un vecteur colonne.

On replace les valeurs de fréquences 0, $\frac{Fe}{2}$ et Fe qui n'étaient pas permutées tout en concaténant les deux échantillons qui ont été permutés séparément. Le résultat est la Transformée de Fourier suivante :

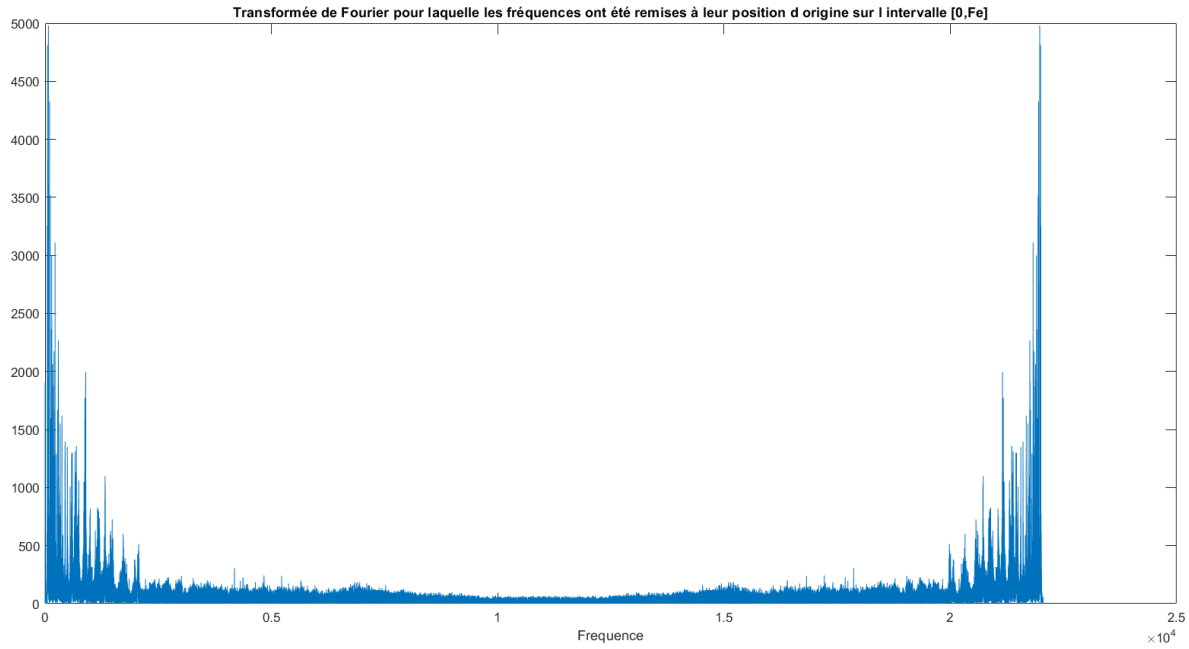


FIGURE 7 – Transformée de Fourier du signal après avoir permuté les fréquences sur l'intervalle [0,Fe]

Le code associé à l'explication plus haut est le suivant :

```
1
2
3  clear all;
4  close all
5  clc;
6
7  [x,Fe]=audioread('canal.wav') ; %%vecteur colonne
8
9  N = length(x);
10
11 X=fft(x); %%vecteur colonne
12
13
14 te = 1/Fe;
15
16 t = (0:N-1)*te;
17
18 f = [0:Fe/N:Fe*(1-1/N)];
19
20
21 X_head = X(2:(N/2)-1);
```

```
22
23 X_tail = X((N/2)+1:N-1);
24
25 Y_head = flipud(X_head);
26 Y_tail = flipud(X_tail);
27
28 Y_head = [ X(1) ; Y_head ];
29 Y_tail = [ X(N/2) ; Y_tail ;X(N)];
30 Y= [  Y_head ; Y_tail ];
31
32
33 figure();
34
35
36 y = ifft(Y,'symmetric');
37
38
39 y = y/max(abs(y));
40
41 filename = "result_canal_plus.wav";
42
43 audiowrite(filename,y,Fe);
44
45
46 end
47
```

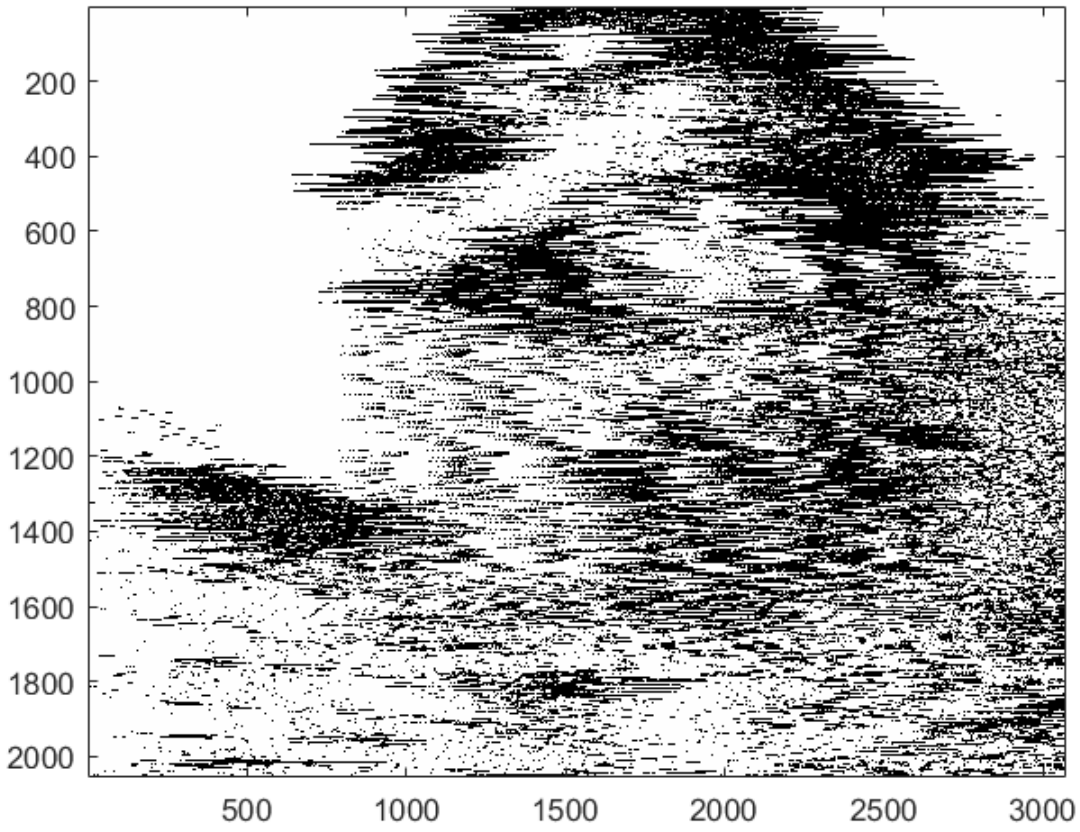
On retrouve bien un signal périodique avec des pics en 0 et en Fe. Le signal audio récupéré est également décodé.

4 Sujet 3

4.1 Premier projet - projet image

L'objectif est de récupérer l'image d'origine avant le décalage des lignes. Le code ci-dessous permet d'afficher, l'image brouillée :

```
1
2     B=imread('image.bmp','bmp');
3     B=255*B;
4     image(B);
5     colormap(gray);
6
7
```



Pour obtenir le bon décalage de correction pour chaque ligne de l'image, l'idée consiste à prendre la première ligne comme ligne de référence et considérer que chaque pixel de celle-ci est bien positionné. Le spectre d'une image est continue. Ainsi, pour la ligne suivante, on fait en sorte de maximiser la continuité avec la ligne précédente. Pour cela il faut maximiser l'autocorrélation entre les deux lignes. On effectue donc le décalage de pixels sur la nouvelle ligne qui maximise la corrélation. Pour cela on utilise la fonction **circshift** pour effectuer une permutation circulaire des pixels sur chaque ligne comme le montre le code suivant :

```
1
2  colonnes = length(B(1, : ));
3  lignes = length(B(:,1));
4
5  vecteur = [2 : 1 : lignes ];
6  result = zeros(lignes,colonnes);
7
8  result(1,:) = B(1,:);
9
10 for i = vecteur
```

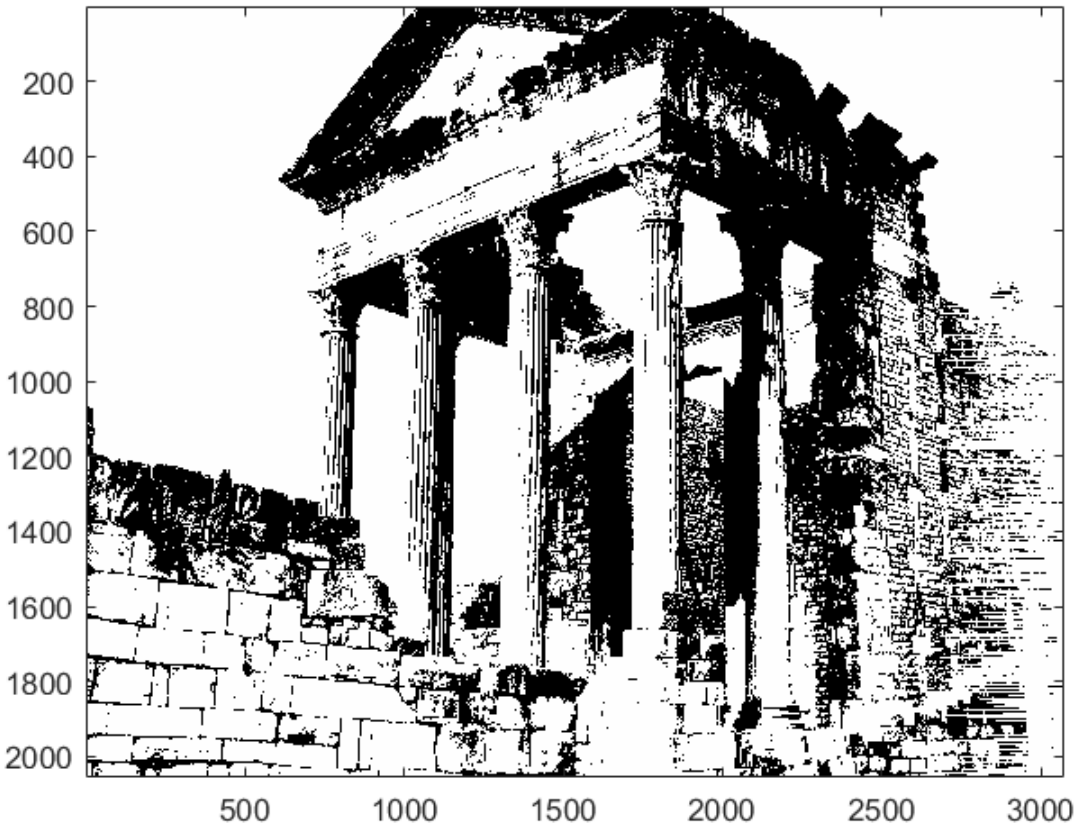
```
    u = result(i-1,:);
    v = B(i,:);

    r=real(ifft(fft(u).*conj(fft(v))));
    [val,ind]=max(r);
    vnew=circshift(v,(ind-1));
    result(i,:) = vnew;


end

```

L'image obtenue après traitement est affichée ci-dessous :



On observe encore des résidus à droite. Il serait possible de traiter ce problème en rajoutant des 255 aux points situés à droite qui portent des informations de la partie

gauche de l'image car la permutation est circulaire.

4.2 Deuxième projet - Encodage du signal

Ce dernier tp proposait d'analyser le contenu d'un signal inconnu, dont le spectre a été tracé ci-dessous :

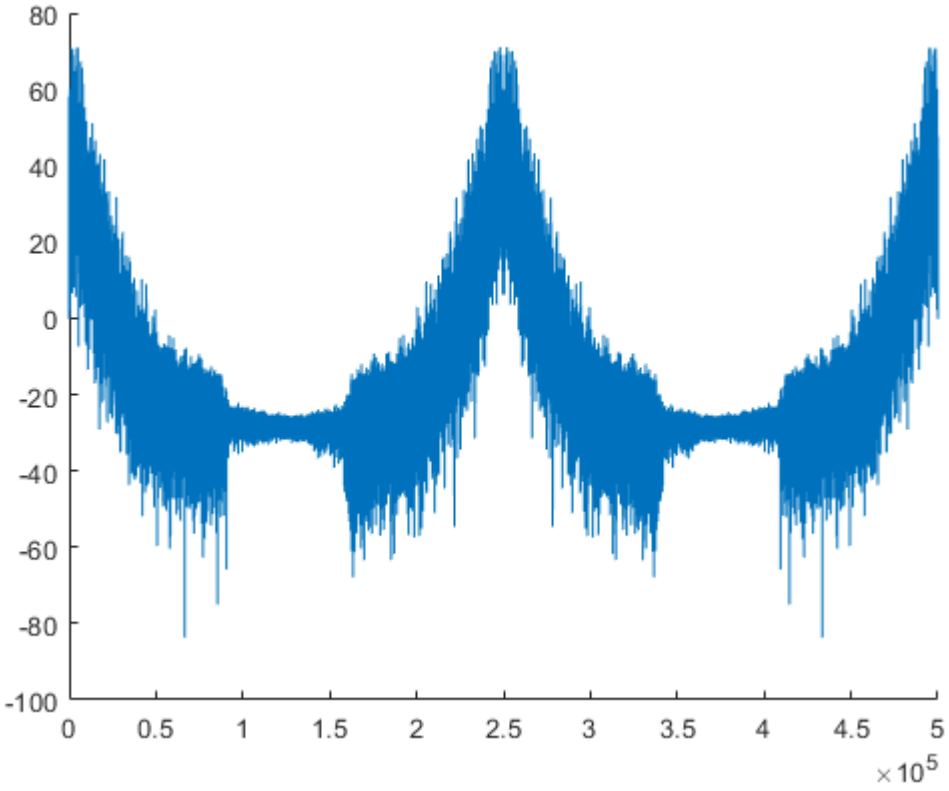


FIGURE 8 – Spectre du signal à étudier

On remarque que le spectre présente un pic symétrique à la moitié du spectre, ce qui laisse penser qu'il a été ré-échantillonné à 2 fois sa fréquence initiale d'échantillonnage.

En réalité, ce signal contient deux signaux échantillonnés à la fréquence F_e, qu'on a artificiellement fait passer à 2 F_e en insérant des 0 aux fréquences non-échantillonnées (décalées pour l'un des deux signaux), puis en faisant la somme des deux nouveaux signaux :

$$s_f = s_1 + s_2 \text{ avec } s_1 = \left[x_1^1, 0, x_1^2, 0, \ldots, x_1^N, 0\right] \text{ et } s_2 = \left[0, x_2^1, 0, x_2^2, \ldots, 0, x_2^N\right]$$

5 Conclusion

Au cours de ces trois séances de travaux pratiques, nous avons pu réaliser plusieurs projets de traitement du signal, que se soit sur des signaux sonores ou sur une image numérique. Nous avons ainsi pu mettre en application les connaissances théoriques acquises lors du cours ES101 telles que la réalisation de filtrage RIF et RII, l'utilisation de la fonction d'autocorrélation ou encore la notion de puissance d'un signal.