
Laboratoire 3

Table of Contents

Description des fichiers fournis	1
Commandes MATLAB possiblement utiles	1
Première partie (3.3 points)	2
Deuxième partie (1.45 point)	6
Signal audio avec bruit (2 points)	8
Réduction du nombre de coefficients de la transformée de Fourier (3.25 points)	12
Important	16

Fourier et échantillonnage compressé

Auteurs :

Antoine Langevin, Jean-Marc Lina, Ghyslain Gagnon

AUTOMNE 2015

Description des fichiers fournis

Fichiers audio pour la deuxième partie

- Durée : 15 secondes
- Fréquence d'échantillonnage : 44 100 kHz, stéréo
- Audio_original.wav : Signal audio original
- Audio_bruit.wav : Signal audio original avec interférence

Commandes MATLAB possiblement utiles

Lire l'aide de ces commandes:

semilogx, semilogy, loglog

fft, ifft

abs, log, log10

upsample, downsample

sort, find

fliplr, flipud

audioread, sound

numel, size, length

Première partie (3.3 points)

Pour cette première partie, vous devez utiliser la fonction A suivante :

$$A = 0.5 \cdot \cos(280\pi \cdot t) + 0.5 \cdot \sin(260\pi \cdot t) + 0.5 \cdot \cos(300\pi \cdot t)$$

a) Discrétiser la fonction à une fréquence d'échantillonnage de 1000 Hz pour une période totale de 100 ms. **(1 point)**

```
close all;
clc;

% Fréquence d'échantillonnage en Hertz
Fs = 1000;
% Période d'échantillonnage
T = 1 / Fs;
% Période totale
L = 100;
% Vecteur de temps
t = (0:L - 1) * T;
% Discrétisation de la fonction A
S = 0.5 * cos(280 * pi * t) + 0.5 * sin(260 * pi * t) + 0.5 * cos(300
    * pi * t);
% On sauve cette discrétisation dans X
X = S;
% Subplot X et Y pour diviser la figure en plusieurs graphes.
SubplotX = 4;
SubplotY = 1;
```

b) Quelle est la valeur de dt (Temps entre deux points) **(0.1 point)**.

```
% La distance entre deux (2) valeurs de temps est calculé avec les
    deux
% premiers éléments du vecteur.
dt = t(2) - t(1);
% On présente le résultat.
fprintf('dt = %g\n', dt)

dt = 0.001
```

c) Trouver les coefficients de la transformée de Fourier. **(0.5 point)**

```
% Les coefficients de la transformée de fourier sont calculés par
% l'algorithme de "Fast Fourier Transform" (FFT).
Y = fft(X);
% La précision de l'approximation dépendant du nombre N contrôlant les
% "tranches" de l'intégrale représentée, il y a trop de coefficients
    pour
% les présenter. Voici les 5 premiers:
disp('Les cinq (5) premiers coefficients de la transformée de
    Fourier');
```

```
disp(Y(1:5));
```

Les cinq (5) premiers coefficients de la transformée de Fourier

*1.0e-13 **

Columns 1 through 4

*-0.0799 + 0.0000i 0.0289 + 0.0833i -0.1715 + 0.0799i -0.0286 -
0.0311i*

Column 5

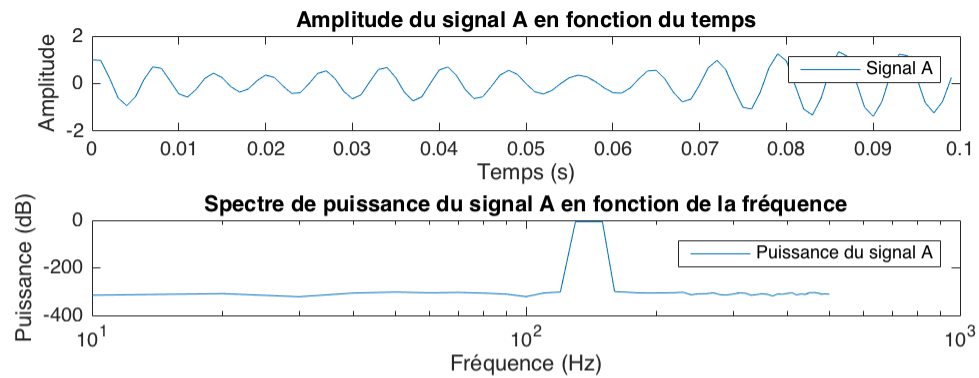
-0.1865 + 0.1676i

d) Dans un subplot, tracer le signal A en fonction du temps et tracer le spectre de puissance du signal A.
(0.25 point)

Les graphes des spectres de puissances doivent tous avoir la puissance en dB. De plus, vous devez seulement afficher les fréquences positives et mettre l'axe des fréquences en échelle logarithmique.

```
% On commence une nouvelle figure.
figure;
% Ce graphique représente le signal A en fonction du temps.
subplot(SubplotX,SubplotY,1);
plot(t, X);
% Les informations à afficher, associées à ce graphe.
title('Amplitude du signal A en fonction du temps');
xlabel('Temps (s)');
ylabel('Amplitude');
legend('Signal A');

% Spectre de puissance du signal A
subplot(SubplotX,SubplotY,2);
% On adapte les données pour l'affichage.
P2 = abs(Y/L);
P1 = P2(1:L/2+1);
P1(2:end-1) = 2 * P1(2:end-1);
% On calcule les fréquences correspondantes.
fax_Hz = Fs * (0:(L/2))/L;
% On trace sur un graphique où l'axe des X est d'échelle
logarithmique.
semilogx(fax_Hz, mag2db(P1));
% Les informations associées...
title('Spectre de puissance du signal A en fonction de la fréquence');
xlabel('Fréquence (Hz)');
ylabel('Puissance (dB)');
legend('Puissance du signal A');
```



e) Quelle est la fréquence de la raie du signal A? **(0.1 point)**

```
% Par observation du graphe, on tombe sur une fréquence de raie
% d'environ
% 140 Hz.
disp('La fréquence de la raie du signal A est d\'environ ~140 Hz.');
```

La fréquence de la raie du signal A est d'environ ~140 Hz.

f) Augmenter la résolution fréquentielle de la transformée de Fourier par un facteur 10. **(1 point)**

```
% Afin d'augmenter la résolution fréquentielle, on multiplie le nombre
% d'échantillons.
% Résolution fréquentielle = Fs/L

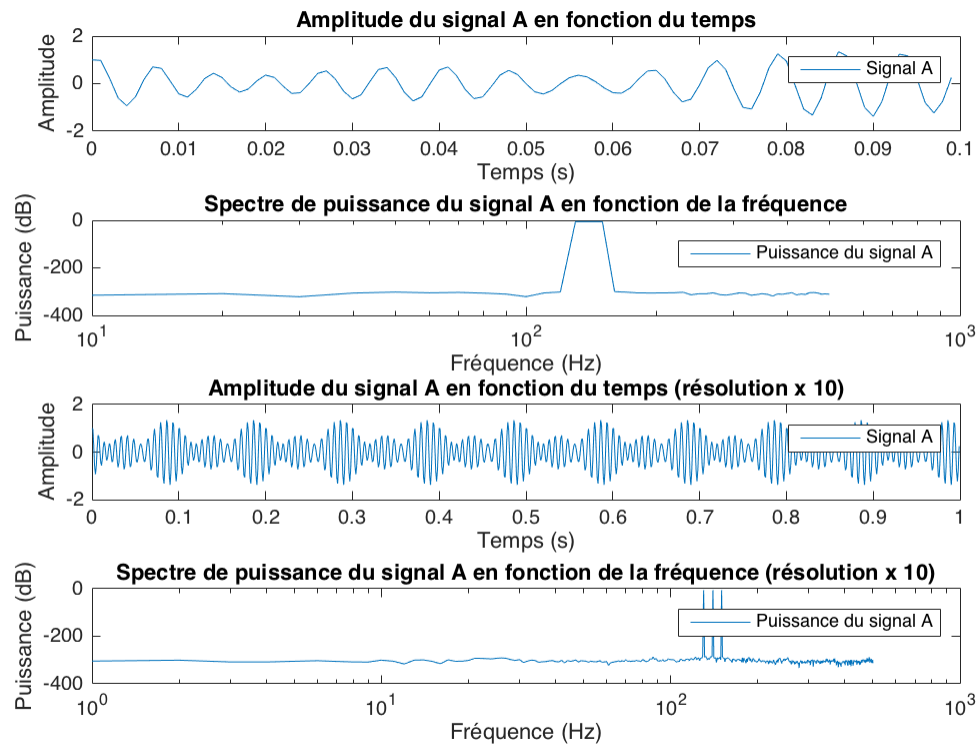
% Période totale
Fs = 1000;
T = 1 / Fs;
L = 100 * 10;
% Vecteur de temps
t = (0:L - 1) * T;
fprintf('dt = %g\n', t(2)-t(1));
% Discrétisation de la fonction A
S = 0.5 * cos(280 * pi * t) + 0.5 * sin(260 * pi * t) + 0.5 * cos(300
    * pi * t);

dt = 0.001
```

g) Dans un subplot, tracer le nouveau signal A en fonction du temps et tracer le spectre de puissance du nouveau signal A. **(0.25 point)**

```
% Signal A en fonction du temps
subplot(SubplotX,SubplotY,3);
X = S;
plot(t, X);
% Les informations du graphe
title('Amplitude du signal A en fonction du temps (résolution x 10)');
xlabel('Temps (s)');
ylabel('Amplitude');
legend('Signal A');

% Spectre de puissance du signal A
Y = fft(X);
P2 = abs(Y/L);
P1 = P2(1:L/2+1);
P1(2:end-1) = 2 * P1(2:end-1);
subplot(SubplotX,SubplotY,4);
% On convertit les différentes composantes en Hz et dB pour les
% afficher.
fax_Hz = Fs * (0:(L/2))/L;
semilogx(fax_Hz, mag2db(P1));
% Les informations du graphe
title('Spectre de puissance du signal A en fonction de la fréquence
(résolution x 10)');
xlabel('Fréquence (Hz)');
ylabel('Puissance (dB)');
legend('Puissance du signal A');
```



h) Quelles sont les fréquences des raies ? (0.1 point)

```
% Selon l'observation des graphes, on obtient trois raies à 130 Hz,
140Hz et
% 150 Hz.
```

```
disp('f1 = 130 Hz, f2 = 140 Hz et f3 = 150 Hz');
```

```
f1 = 130 Hz, f2 = 140 Hz et f3 = 150 Hz
```

Deuxième partie (1.45 point)

Modifier le signal audio 'Audio_original.wav' afin de créer un vecteur du signal sonore mono et à une fréquence d'échantillonnage de 22.05 kHz.

Utiliser la fonction "audioread" afin de lire le fichier audio. De plus, tout au long de ces manipulations, vous pouvez utiliser la fonction sound pour entendre le contenu audio des signaux.

a) Écrire une fonction permettant de faire la conversion d'un signal échantillonné à 44,1 kHz en stéréo à un signal audio mono échantillonné à 22,05 kHz. (0.5 point)

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% function [audio_22khz] = convert_44_22(audio_stereo_44)
%
% * audio_22khz : Vecteur contenant le signal audio échantillonné à
% 22 kHz.
% * audio_stereo_44 : Données que nous souhaitons convertir.
```

```
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Code de la fonction en commentaire
%
% function [audio_22khz] = convert_44_22(audio_stereo_44)
%     % On convertit le signal stereo en signal mono en faisant la
%     % moyenne
%     % des signaux de gauche et de droite.
%     mono44 = (audio_stereo_44(:,1) + audio_stereo_44(:,2)) / 2;
%     % On réduit le nombre d'échantillon par un facteur de 2.
%     audio_22khz = downsample(mono44, 2);
% end
```

b) Trouver le spectre de puissance du signal audio échantillonné à 22 kHz. **(0.5 point)**

```
% On lit le signal audio à partir du fichier wave.
% Y contient le signal de gauche et de droite et Fs la fréquence
% d'échantillonnage
[y, Fs] = audioread('Audio_original.wav');
mono22 = convert_44_22(y);
% On ajuste la fréquence d'échantillonnage post-réduction.
Fs = Fs/2;
% On trouve la fft du signal mono à 22 KHz.
X = mono22;
Y = fft(X);
spectre = mag2db(abs(Y));
```

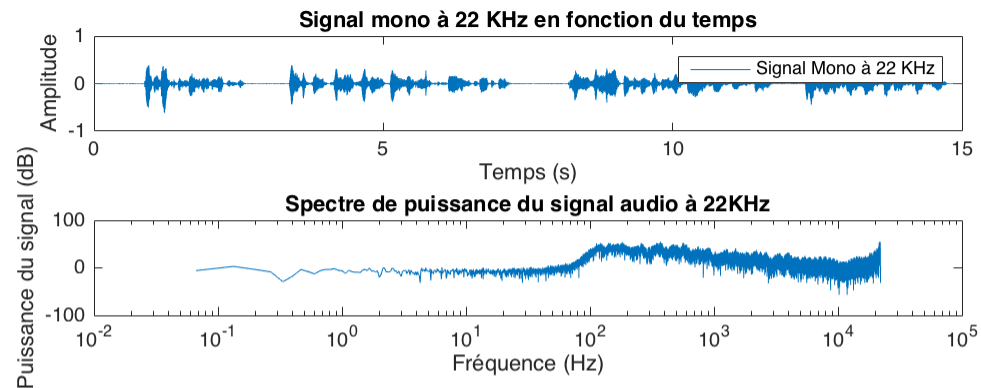
c) Dans un subplot, tracer le signal audio en fonction du temps et tracer le spectre de puissance. **(0.25 point)**

```
T = 1 / Fs;
% Période totale
L = length(Y);
% Vecteur de temps
t = (0:L-1) * T;

% Traçage d'une nouvelle figure
figure
subplot(SubplotX, SubplotY, 1);
plot(t, X);
% Les informations du graphe
title('Signal mono à 22 KHz en fonction du temps');
xlabel('Temps (s)');
ylabel('Amplitude');
legend('Signal Mono à 22 KHz');

% Conversion de bins à Hz selon la fréquence d'échantillonnage.
fax_bins = (0: L-1);
fax_Hz = fax_bins * Fs / L;
% Conversion de l'amplitude en dB
subplot(SubplotX, SubplotY, 2);
semilogx(fax_Hz, spectre);
title('Spectre de puissance du signal audio à 22KHz');
xlabel('Fréquence (Hz)');
```

```
ylabel('Puissance du signal (dB)');
```



d) Entre quel intervalle de fréquences la parole humaine peut-elle produire des sons ? **(0.1 point)**

```
disp('Selon les résultats, la parole humaine semble produire des sons  
entre 100 Hz et environ 10 KHz.')
```

```
disp('La majorité semble être entre 100 Hz et 500 Hz.');
```

```
% e) Que remarquez-vous sur le graphique du spectre de puissance ?  
*(0.1 point)*
```

```
%  
disp('Le graphique du spectre de puissance présente des anomalies pour  
fréquence > 10 KHz');
```

Selon les résultats, la parole humaine semble produire des sons entre 100 Hz et environ 10 KHz.

La majorité semble être entre 100 Hz et 500 Hz.

Le graphique du spectre de puissance présente des anomalies pour fréquence > 10 KHz

Signal audio avec bruit (2 points)

Le fichier Audio_bruit.wav contient le même contenu que le fichier Audio_original, mais 4 signaux ont été ajoutés. Un premier à une fréquence fondamentale f_0 ainsi que 3 harmoniques de cette fréquence f_0 .

a) À l'aide de la transformée de Fourier, trouver visuellement la fréquence de ces 4 signaux ajoutés. **(0.5 point)**

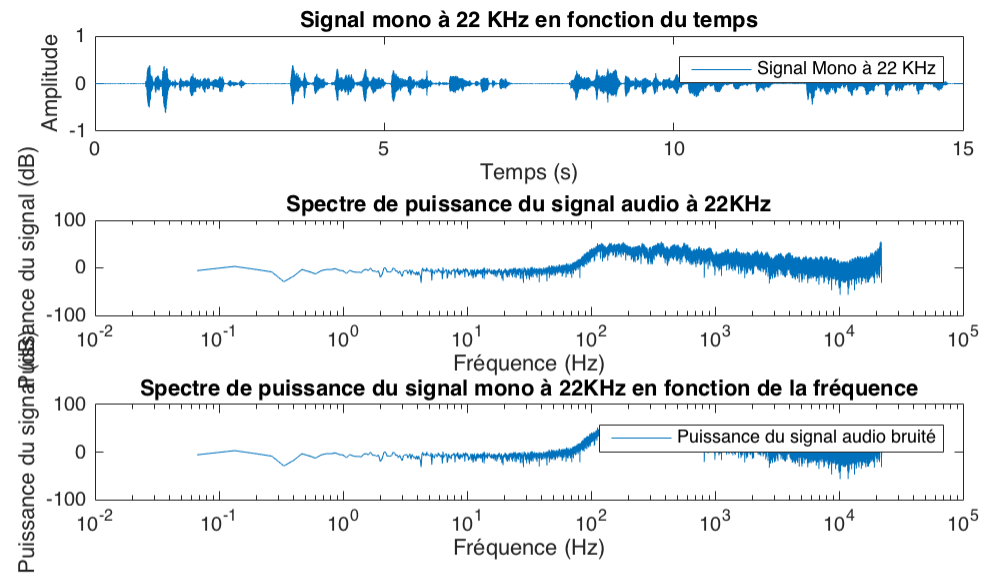

```
% On extrait le signal avec bruit du fichier.
[y, Fs] = audioread('Audio_bruit.wav');

% On convertit le signal stéréo en signal mono à une fréquence
% d'échantillonnage multipliée par un facteur 1/2.
mono22 = convert_44_22(y);
Fs = Fs / 2;
X = mono22;
% On effectue la transformée de Fourier à l'aide de fft
Y = fft(X);
% Obtenir le nombre de valeurs possibles (bins)
L = length(X);
T = 1 / Fs;
% Période totale
L = length(Y);
% Vecteur de temps
t = (0:L-1) * T;
% On calcule le spectre de puissance du signal pour chacune des
% fréquences.
spectre = mag2db(abs(Y));
% Générer un vecteur de L nombre de 0 à L-1
fax_bins = (0:L-1);
% Convertir les valeurs bins en fréquence
fax_Hz = fax_bins * Fs / L;

% On trace le graphique
subplot(SubplotX, SubplotY, 3);
semilogx(fax_Hz, signal);
title('Spectre de puissance du signal mono à 22KHz en fonction de la
    fréquence');
xlabel('Fréquence (Hz)');
ylabel('Puissance du signal (dB)');
legend('Puissance du signal audio bruité');

% Par observation des graphes, on trouve 4 signaux ajoutés à 60, 120,
% 180
% et 240 Hz.
disp('Les fréquences des 4 signaux sont 60, 120, 180 et 240 Hz.');
```

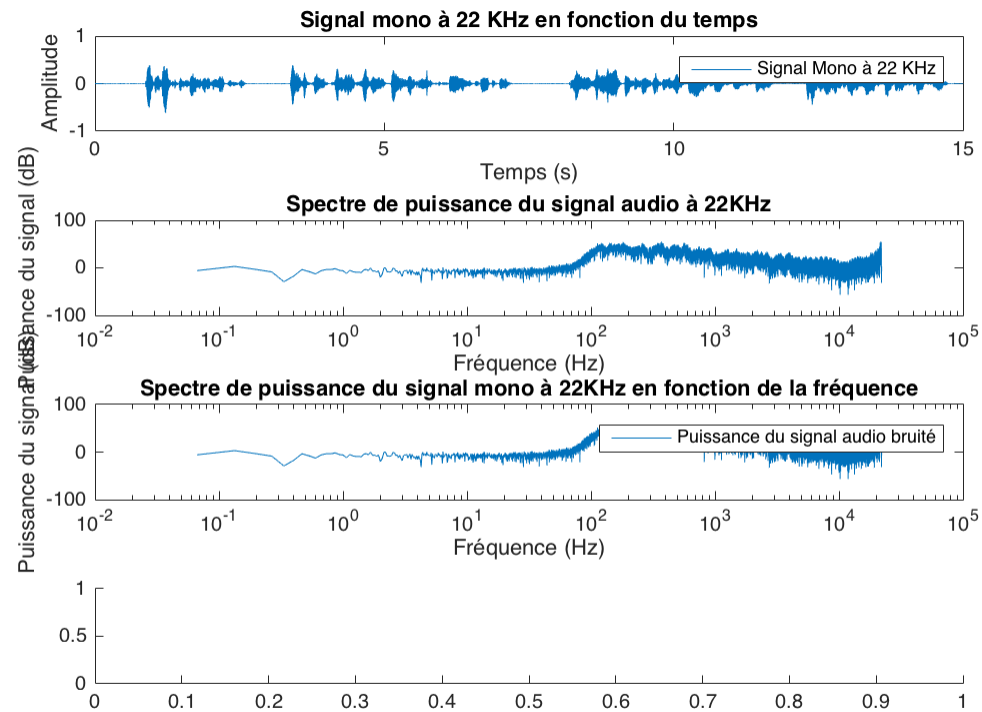
Les fréquences des 4 signaux sont 60, 120, 180 et 240 Hz.



b) Une fois que vous avez localisé les 4 fréquences de bruits, retirer les signaux de bruits en effectuant des opérations sur les coefficients de la transformée de Fourier. **(1 point)**

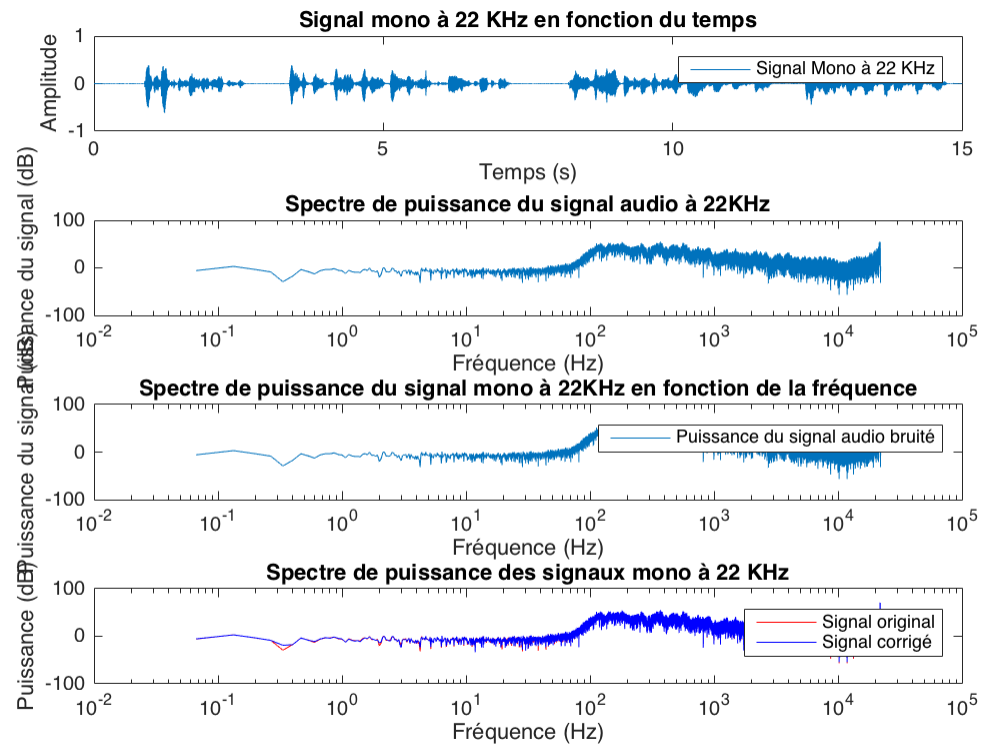
```
subplot(SubplotX,SubplotY,4);
Y = fft(mono22);

% On calcule le bin correspondant à chacune des fréquence identifiées
% On annule le signal à ces points.
freqs = [60, 120, 180, 240];
for freq = freqs
    index = (freq * L/Fs) + 1;
    Y(index) = 0;
end
```



c) Tracer sur un même graphique le spectre de puissance avec le bruit et sans le bruit. (0.25 point)

```
% On trace le signal corrigé au dessus du signal original.
signal2 = mag2db(abs(Y));
semilogx(fax_Hz, signal, 'r', fax_Hz, signal2, 'b');
title('Spectre de puissance des signaux mono à 22 KHz');
xlabel('Fréquence (Hz)');
ylabel('Puissance (dB)');
legend('Signal original', 'Signal corrigé')
```



d) Faire la transformée inverse afin de retrouver le signal audio. (0.25 point)

```
% La transformée inverse à l'aide de ifft (inverse fast fourier
transform).
inverse = ifft(Y);
```

Réduction du nombre de coefficients de la transformée de Fourier (3.25 points)

On désire garder seulement une fraction des coefficients de la transformée de Fourier. Afin de réduire le nombre de coefficients, vous devez éliminer les coefficients ayant les plus petites puissances en premier.

a) Calculer le seuil de la puissance correspondant à la coupure si nous désirons garder 95% des coefficients. Tous les coefficients qui seront sous ce seuil devront être mis à 0. (0.5 point)

Utiliser le fichier audio Audio_original.wav

```
% On lit le signal audio original.
[y, Fs] = audioread('Audio_original.wav');
mono22 = convert_44_22(y);
X = mono22;
% On ajuste la fréquence d'échantillonnage post-réduction.
Fs = Fs/2;

% Code de la transformée de Fourier
```

```
% On trouve la fft du signal mono à 22 Khz.
Y = fft(X);
Y95 = Y;

% Code pour trouver le seuil
L = length(Y);
% On tri en decroissant.
sortedY = sort(Y, 'descend');
% On regarde la valeur correspondant au seuil 95%
indexSeuil = ceil(0.95 * L);
seuil = Y(indexSeuil);

fprintf('Le seuil est de %d dB pour conserver 95%% des coefficients.
\n', seuil);

Le seuil est de -1.228789e+01 dB pour conserver 95% des coefficients.
```

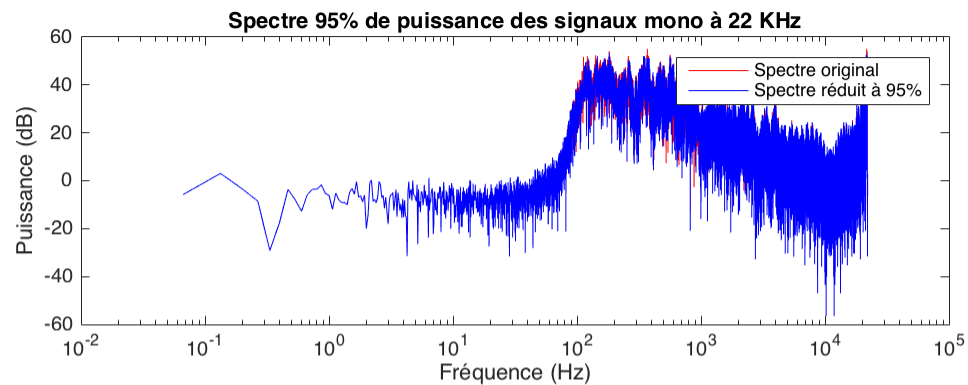
b) Retirer tous les coefficients qui sont sous le seuil trouvé. **(0.75 point)**

```
% Code pour retirer les coefficients
% On itère sur les valeurs, annulant celles qui tombe sous le seuil.
elimine = 0;
for idx = 1:numel(Y95)
    valeur = Y95(idx);
    if valeur <= seuil
        Y95(idx) = 0;
        elimine = elimine + 1;
    end
end
fprintf('Enlever %d %% des valeurs.\n', elimine / L);

Enlever 5.761149e-02 % des valeurs.
```

c) Tracer sur un même graphique le spectre de puissance du signal audio original et celui du signal modifié. **(0.25 point)**

```
figure
subplot(2,1,1);
signal = mag2db(abs(Y));
signal2 = mag2db(abs(Y95));
semilogx(fax_Hz, signal, 'r', fax_Hz, signal2, 'b');
title('Spectre 95% de puissance des signaux mono à 22 KHz');
xlabel('Fréquence (Hz)');
ylabel('Puissance (dB)');
legend('Spectre original', 'Spectre réduit à 95%')
```



d) Faire la transformée inverse afin de retrouver le signal audio. **(0.25 point)**

```
signalAudio = ifft(Y95);
```

e) Refaire les mêmes opération pour un seuil permettant de conserver 50% des coefficients. Tracer sur un même graphique les spectres de puissance du signal original et du signal compressé. **(0.75 point)**

```
Y = fft(X);
Y50 = Y;

% Code pour trouver le seuil
L = length(Y);
% On tri en decroissant.
sortedY = sort(Y, 'descend');
% On regarde la valeur correspondant au seuil 95%
indexSeuil = ceil(0.50 * L);
seuil = Y(indexSeuil);

fprintf('Le seuil est de %d dB pour conserver 95%% des coefficients.\n', seuil);

% On itère sur les valeurs, annulant celles qui tombe sous le seuil.
elimine = 0;
for idx = 1:numel(Y50)
    valeur = Y50(idx);
    if valeur <= seuil
        Y50(idx) = 0;
    end
end
```

```

        elimine = elimine + 1;
    end
end
fprintf('Enlever %d %% des valeurs.\n', elimine / L);

```

Le seuil est de $-3.300152e-01$ dB pour conserver 95% des coefficients.
Enlever $4.507271e-01$ % des valeurs.

f) Indiquer le seuil de coupure si nous désirons conserver 50% des coefficients. (0.25 point)

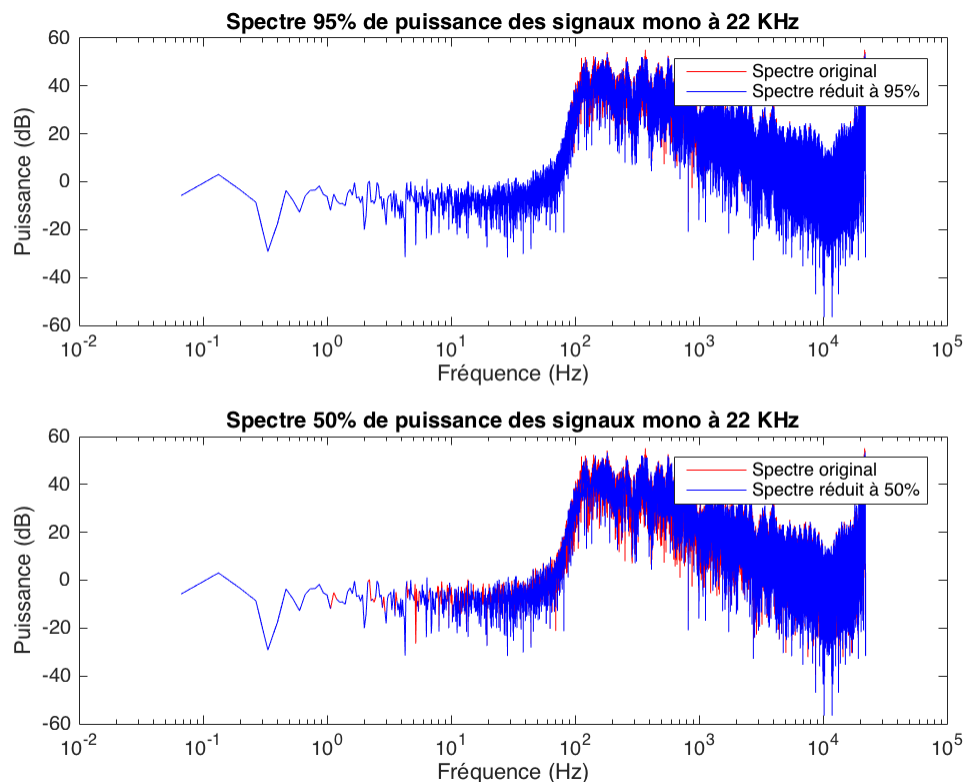
```

subplot(2,1,2);
signal = mag2db(abs(Y));
signal2 = mag2db(abs(Y50));
semilogx(fax_Hz, signal, 'r', fax_Hz, signal2, 'b');
title('Spectre 50% de puissance des signaux mono à 22 KHz');
xlabel('Fréquence (Hz)');
ylabel('Puissance (dB)');
legend('Spectre original', 'Spectre réduit à 50%')

fprintf('Le seuil est de %d dB pour conserver 50% des coefficients.',
    seuil);

```

Le seuil est de $-3.300152e-01$ dB pour conserver 50



g) Que remarquez-vous lorsque vous écoutez les signaux modifiés ? (0.25 point)

% On entend une forme de bruit sinusoïdal.

```
disp('Le bruit augmente de plus en plus.');
```

*% h) Pourquoi ? *(0.25 point)**

```
disp('Le filtre utilisé est une forme de filtre qui favorise le  
passage du bruit moins perceptible pour l''oreille mais qui est  
fortement présent\n');
```

*Le bruit augmente de plus en plus.
Le filtre utilisé est une forme de filtre qui favorise le passage du
bruit moins perceptible pour l'oreille mais qui est fortement présent
\n*

Important

- Les graphiques doivent tous avoir un titre, les axes doivent être identifiés et il doit y avoir une légende pour chaque fonction du graphique.
- Les fonctions doivent toutes avoir une description. De plus, il faut identifier les arguments et les variables que retourne la fonction.
- Le code doit être commenté.

Vous devez remettre un fichier .zip contenant le fichier PDF publish et tous les fichiers Matlab afin de pouvoir exécuter le script lors de la correction.

Le fichier zip doit être nommé ainsi : Labo3_XXXX#####_XXXX#####.zip où XXXX##### doit être remplacé par votre code permanent.

Published with MATLAB® R2015b