UEMF
الجامعة الأورومتوسطية بفاس
EUROMED UNIVERSITY OF FES
UNIVERSITÉ EUROMED DE FÈS

EIDIA
École d'Ingénierie Digitale
et d'Intelligence Artificielle

TP4 :
Filtrage Analogique

Réalisé par : IKEN Fatima
Encadre par : Mr AMMOUR ALAE
Filière : big data
Année universitaire : 2022/2023

Objectifs :

Appliquer un filtre réel pour supprimer les composantes indésirables d'un signal. Améliorer la qualité de filtrage en augmentant l'ordre du filtre

Travail demandé :

un script Matlab commenté contenant le travail réalisé et des commentaires sur ce que vous avez compris et pas compris, ou sur ce qui vous a semblé intéressant ou pas, bref tout commentaire pertinent sur le TP.

Filtrage et diagramme de Bode

Sur le réseau électrique, un utilisateur a branché une prise CPL (Courant Porteur en Ligne), les signaux utiles sont de fréquences élevées. Le réseau électrique a cependant sa propre fréquence (50 hz). Le boiter de réception doit donc pouvoir filtrer les basses fréquences pour s'attaquer ensuite à la démodulation du signal utile.

Schématiquement :

Mathématiquement, un tel filtre fournit un signal de sortie en convoluant le signal d'entrée par la réponse temporelle du filtre :

$$y(t) = x(t) * h(t)$$

Nous souhaitons appliquer un filtre passe-haut pour supprimer la composante à 50 Hz. Soit notre signal d'entrée :

$$x(t) = \sin(2.pi.f_1.t) + \sin(2.pi.f_2.t) + \sin(2.pi.f_3.t)$$

1. Définir le signal x(t) sur t = [0 5] avec Te = 0,0001 s.

```
clear all
close all
clc
[music, fs] = audioread('test.wav');
save('testt.mat', 'y', 'fs');
load('testt.mat');
music = music';
```

2Tracer le signal x(t) et sa transformé de Fourrier Essayez de tracer avec Te = 0,0005 s.

==>le code

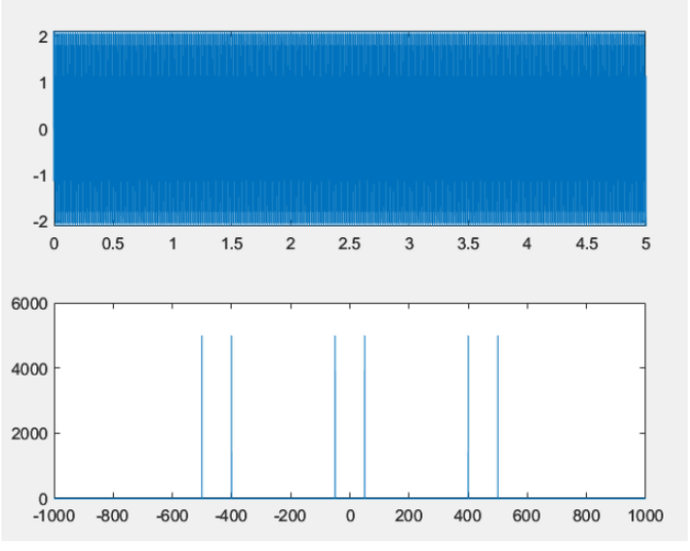
==>représentation graphique

```
clear all
close all
clc

[music, fs] = audioread('test.wav');
save('testt.mat', 'y', 'fs');
load('testt.mat');
music = music';
N=length(music);
te = 1/fs;
t = (0:N-1)*te;

f = (0:N-1)*(fs/N);
fshift = (-N/2:N/2-1)*(fs/N);

y_trans = fft(music);
subplot(33,1,1)
plot(t,y)
subplot(3,1,2)
 plot(fshift,fftshift(abs(y_trans)))
```



la transmittance complex

La fonction H(f) (transmittance complexe) du filtre passe haut de premier ordre est donnée par :

$$H(f) = (K.j.w/w_c) / (1 + j.\ w/w_c)$$

```
k = 1;
fc = 5000;
%la transmitance complexe
h =k./(1+1j*(f/fc).^1000);
h_filter = [h(1:floor(N/2)), flip(h(1:floor(N/2)))];

semilogx(f(1:floor(N/2)),abs( h(1:floor(N/2))),'linewidth',1.5)
```

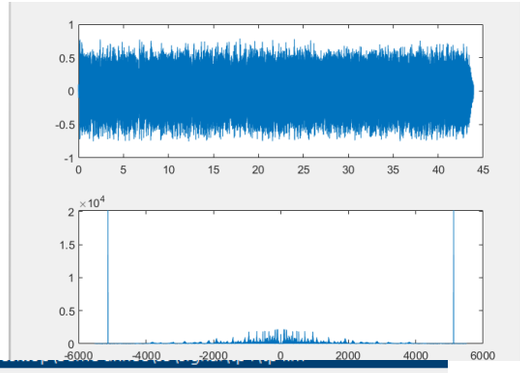
2:Dé-bruitage d'un signal sonore

Dans son petit studio du CROUS, un futur ingénieur a enregistré une musique en « .wav » avec un très vieux micro. Le résultat est peu concluant, un bruit strident s'est ajouté à sa musique. Heureusement son voisin, expert en traitement du signal est là pour le secourir : « C'est un bruit très haute fréquence, il suffit de le supprimer. »

1. Proposer une méthode pour supprimer ce bruit sur le signal.

le signal et sa transformée de fourrier :

==>représentation graphique



==>le code

```
clear all
close all
clc
[music, fs] = audioread('test.wav');
music = music';
N=length(music);
te = 1/fs;
fs=fe
t = (0:N-1)*te;
f = (0:N-1)*(fs/N);
fshift = (-N/2:N/2-1)*(fs/N);
y_trans = fft(music);
subplot(3,1,1)
plot(t,y)
subplot(3,1,2)
 plot(fshift,fftshift(abs(y_trans)))

k = 1;
fc = 5000;
%la transmitance complexe
h =k./(1+1j*(f/fc).^100);
h_filter = [h(1:floor(N/2)), flip(h(1:floor(N/2)))];
semilogx(f(1:floor(N/2)),abs( h(1:floor(N/2))),'linewidth',1.5)

y_filtr = y_trans(1:end-1).*h_filter;
sig_filtred= ifft(y_filtr,"symmetric");
plot(fshift(1:end-1),fftshift(abs(fft(sig_filtred))))
```

Spectre du signal après le filtrage :

==>le code

```
 plot(fshift,fftshift(abs(y_trans)))

k = 1;
fc = 5000;
%la transmitance complexe
h =k./(1+1j*(f/fc).^100);
h_filter = [h(1:floor(N/2)), flip(h(1:floor(N/2)))];
semilogx(f(1:floor(N/2)),abs( h(1:floor(N/2))),'linewidth',1.5)

y_filtr = y_trans(1:end-1).*h_filter;
sig_filtred= ifft(y_filtr,"symmetric");
plot(fshift(1:end-1),fftshift(abs(fft(sig_filtred))))
plot(sig_filtred,t)
```

==>représentation graphique