Compte Rendu TP 1

Analyse spectrale d'un signal
Transformée de Fourier discrète

Réalisé par : Hamdaoui Khalil
Encadré par : Alae Ammour

EIDIA
École d'Ingénierie Digitale
et d'Intelligence Artificielle

UEMF

Introduction

Dans ce premier Tp on va faire la représentation temporelle et fréquentielle en utilisant la transformée de fourrier discrète sous le logiciel Matlab.
On va ensuite essayer de filtrer idéalement un signal bruité. (filtre passe-bas)

N.B :
- Dans le traitement du signal, la transformée de Fourier est utilisée pour analyser et comprendre le contenu en fréquence d'un signal.

- Le filtre passe-bas idéal supprime les fréquences supérieures à une certaine fréquence de coupure tandis que le filtre passe-haut idéal supprime les fréquences inférieures à une certaine fréquence de coupure.

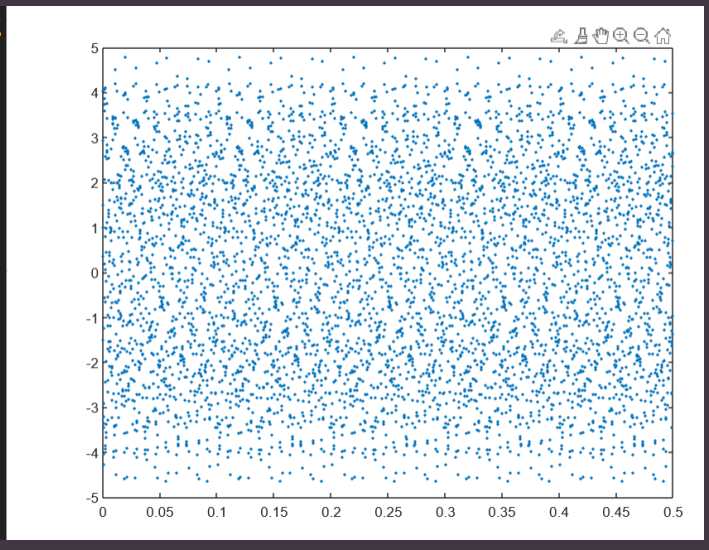
Représentation temporelle et fréquentielle

Considérons un signal périodique x(t) constitué d'une somme de trois sinusoïdes de fréquences 440Hz, 550Hz, 2500Hz.

$$x(t) = 1.2\cos(2pi440t + 1.2) + 3\cos(2pi550t) + 0.6\cos(2pi2500t)$$

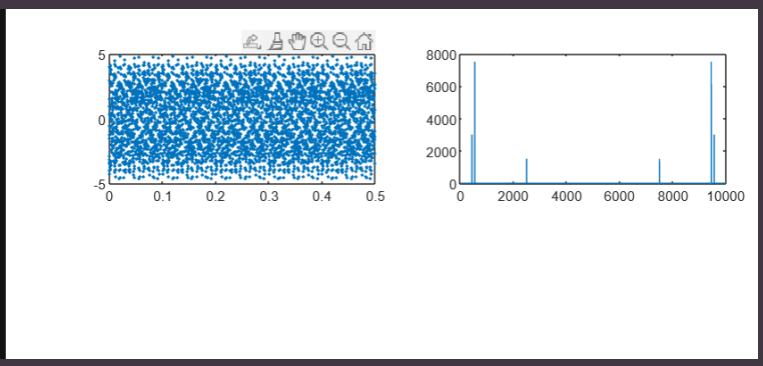
On génére un signal de 5000 échantillons

```
clear all
close all
clc

fe = 1e4;
Te = 1/fe;
N = 5000;

t = 0: Te: (N -1)*Te;
x = 1.2*cos(2*pi*440*t + 1.2) + 3*cos(2*pi*550*t) + 0.6*cos(2*pi*2500*t);
%subplot(3, 2, 1)
plot(t,x,".")

%fshift = 0: fe/N : (N -1)*(fe/N);
%x = 1.2*cos(2*pi*440*t + 1.2) + 3*cos(2*pi*550*t) + 0.6*cos(2*pi*2500*t);
%subplot(3, 2, 2)
%y= abs(fft(x));
%plot(fshift,y)
%
%subplot(3, 2, 3)
%xbruit = x + 2*randn(size(x));
%plot(t,2*randn(size(x)))
%
%subplot(3, 2, 4)
%ybruit = abs(fft(xbruit));
%plot(fshift,fftshift(ybruit))
%
%subplot(3, 2, 5)
%xbruit2 = x + 50*randn(size(x));
%plot(t,50*randn(size(x)))
```



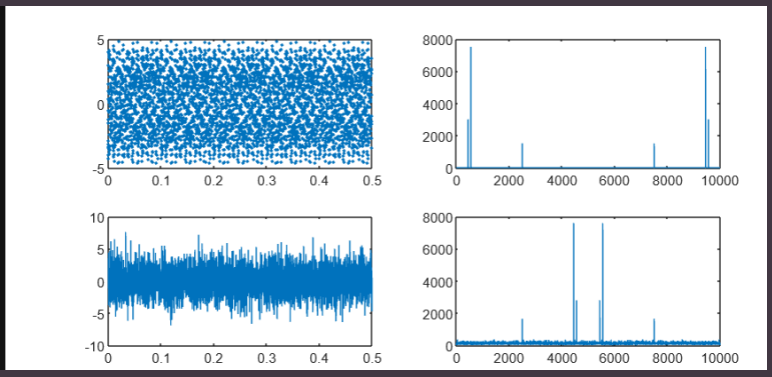
On applique la TFD qui nous génère un spectre qui est une fonction complexe qui contient une partie imaginaire et une partie réelle

```
clear all
close all
clc

fe = 1e4;
Te = 1/fe;
N = 5000;

t = 0: Te: (N -1)*Te;
x = 1.2*cos(2*pi*440*t + 1.2) + 3*cos(2*pi*550*t) + 0.6*cos(2*pi*2500*t);
subplot(3, 2, 1)
plot(t,x,".")

fshift = 0: fe/N : (N -1)*(fe/N);
x = 1.2*cos(2*pi*440*t + 1.2) + 3*cos(2*pi*550*t) + 0.6*cos(2*pi*2500*t);
subplot(3, 2, 2)
y= abs(fft(x));
plot(fshift,y)
```



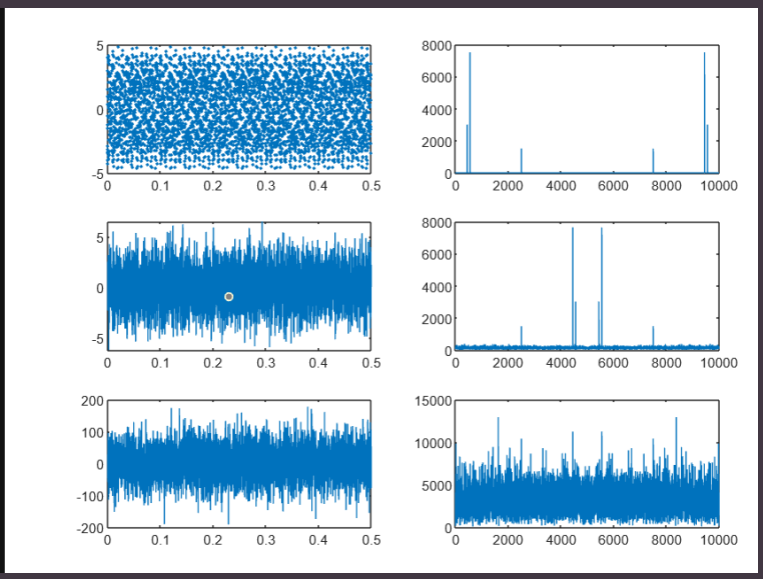
On injecte un bruit dans le signal

```
t = 0: Te: (N -1)*Te;
x = 1.2*cos(2*pi*440*t + 1.2) + 3*cos(2*pi*550*t) + 0.6*cos(2*pi*2500*t);
subplot(3, 2, 1)
plot(t,x,".")

fshift = 0: fe/N : (N -1)*(fe/N);
x = 1.2*cos(2*pi*440*t + 1.2) + 3*cos(2*pi*550*t) + 0.6*cos(2*pi*2500*t);
subplot(3, 2, 2)
y= abs(fft(x));
plot(fshift,y)

subplot(3, 2, 3)
xbruit = x + 2*randn(size(x));
plot(t,2*randn(size(x)))

subplot(3, 2, 4)
ybruit = abs(fft(xbruit));
plot(fshift,fftshift(ybruit))
```



On intensifie le bruit dans le signal en eugmentant le coefficient

```
t = 0: Te: (N -1)*Te;
x = 1.2*cos(2*pi*440*t + 1.2) + 3*cos(2*pi*550*t) + 0.6*cos(2*pi*2500*t);
subplot(3, 2, 1)
plot(t,x,".")

fshift = 0: fe/N : (N -1)*(fe/N);
x = 1.2*cos(2*pi*440*t + 1.2) + 3*cos(2*pi*550*t) + 0.6*cos(2*pi*2500*t);
subplot(3, 2, 2)
y= abs(fft(x));
plot(fshift,y)

subplot(3, 2, 3)
xbruit = x + 2*randn(size(x));
plot(t,2*randn(size(x)))

subplot(3, 2, 4)
ybruit = abs(fft(xbruit));
plot(fshift,fftshift(ybruit))

subplot(3, 2, 5)
xbruit2 = x + 50*randn(size(x));
plot(t,50*randn(size(x)))

subplot(3, 2, 6)
ybruit3 = abs(fft(xbruit2));
plot(fshift,fftshift(ybruit3))
```

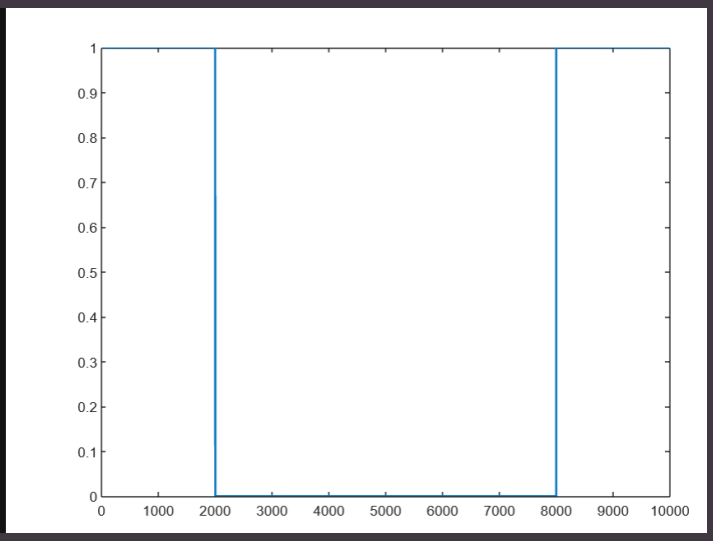


L'information est perdue il est donc difficile de faire une operation de filtrage

Réalisation un filtrage ideal fréquentiel

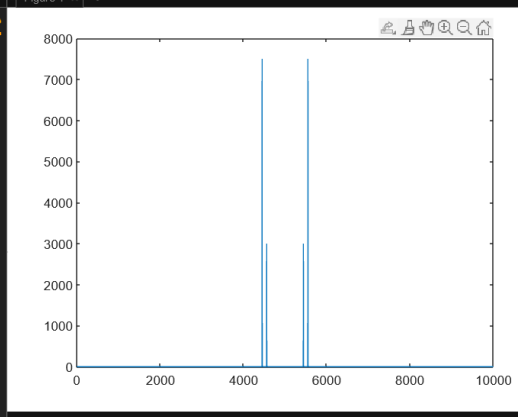
Conception du filtre pass bas

```
%
%subplot(3, 2, 4)
%ybruit = abs(fft(xbruit));
%plot(fshift,fftshift(ybruit))
%
%subplot(3, 2, 5)
%xbruit2 = x + 50*randn(size(x));
%plot(t,50*randn(size(x)))
%
%subplot(3, 2, 6)
%ybruit3 = abs(fft(xbruit2));
%plot(fshift,fftshift(ybruit3))

pass_bas = zeros(size(x));
fc=2000;
index_fc = ceil((fc*N/fe));

pass_bas(1:index_fc) = 1;
pass_bas(N-index_fc+1:N) = 1;
plot(fshift,pass_bas,"LineWidth",1.5);

%x_filter_freq = pass_bas.*y;
%x_filter_temp = ifft(x_filter_freq, "symetric");
% plot(fshift, fftshift(abs(fft(x_filter_temp))));
%
```



Application du filtre pass bas

```
%plot(t,2*randn(size(x)))

%subplot(3, 2, 4)
%ybruit = abs(fft(xbruit));
%plot(fshift,fftshift(ybruit))

%subplot(3, 2, 5)
%xbruit2 = x + 50*randn(size(x));
%plot(t,50*randn(size(x)))

%subplot(3, 2, 6)
%ybruit3 = abs(fft(xbruit2));
%plot(fshift,fftshift(ybruit3))

pass_bas = zeros(size(x));
fc=2000;
index_fc = ceil((fc*N/fe));

pass_bas(1:index_fc) = 1;
pass_bas(N-index_fc+1:N) = 1;
%plot(fshift,pass_bas,"LineWidth",1.5);

x_filter_freq = pass_bas.*abs(fft(x));
x_filter_temp = ifft(x_filter_freq, "symmetric");
plot(fshift, fftshift(abs(fft(x_filter_temp))));
```

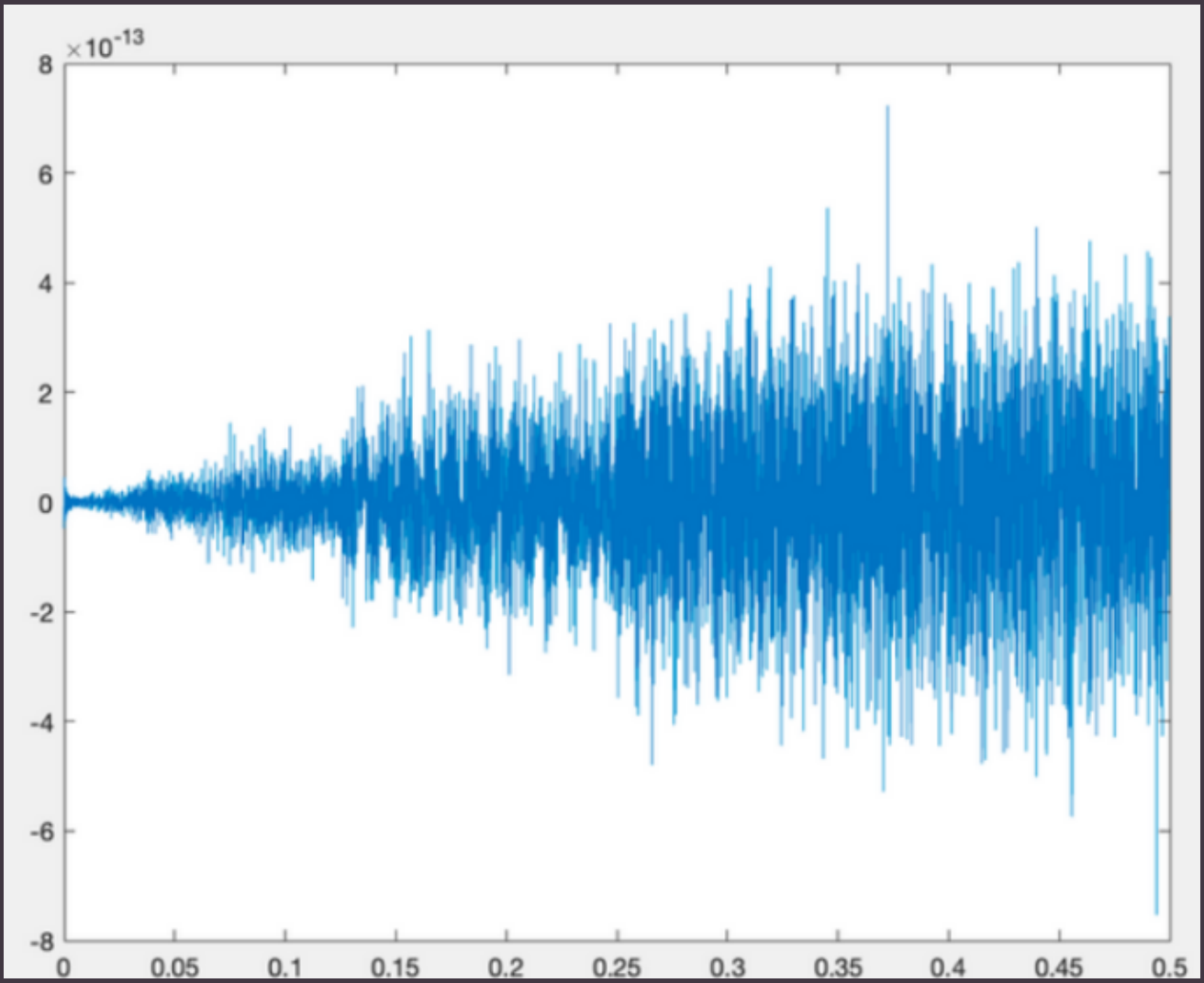


Workspace

Name	Value	Size	Class
fc	2000	1×1	double
fe	10000	1×1	double
fshift	1×5000 double	1×5000	double
index_fc	1000	1×1	double
N	5000	1×1	double
pass_bas	1×5000 double	1×5000	double
t	1×5000 double	1×5000	double
Te	1.0000e-04	1×1	double
x	1×5000 double	1×5000	double
x_filter_freq	1×5000 double	1×5000	double
x_filter_temp	1×5000 double	1×5000	double
y	1×5000 double	1×5000	double

On peut remarquer que les piques superieur à la fréquence de coupure ont été éliminés

Signal filtré



Conclusion

A travers ce premier TP j'ai appris comment représenter des signaux et leur TFD, et appliquer un filtrage idéal