

TP1- Analyse spectrale d'un signal Transformée de Fourier discrète

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

- **Objectifs :**
- Représentation de signaux et applications de la transformée de Fourier discrète (TFD) sous Matlab.
- Evaluation de l'intérêt du passage du domaine temporel au domaine fréquentiel dans l'analyse et l'interprétation des signaux physiques réels.

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.

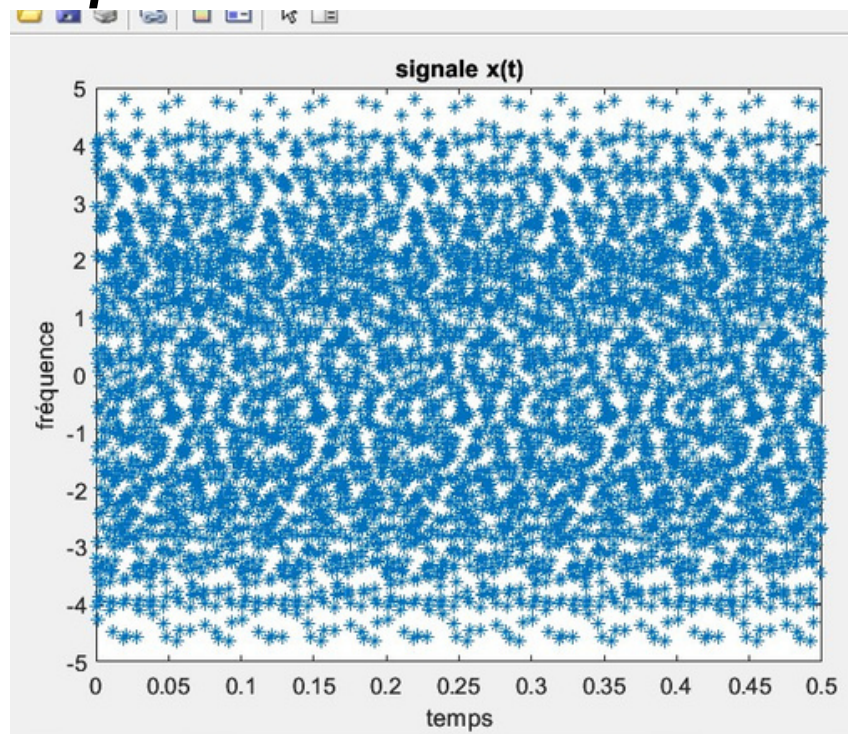
=> Représentation temporelle et fréquentielle :

1- Tracer le signal $x(t)$. Fréquence d'échantillonnage : $f_e = 10000\text{Hz}$

=>code

```
clear all
clc
close all
fe = 1e4 ;
te = 1/fe ;
N = 5000 ;
t = (0: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);
plot(t,x,'*');
title('signale x(t)');
xlabel('temps');
ylabel('fréquence');
```

=>représentation graphique

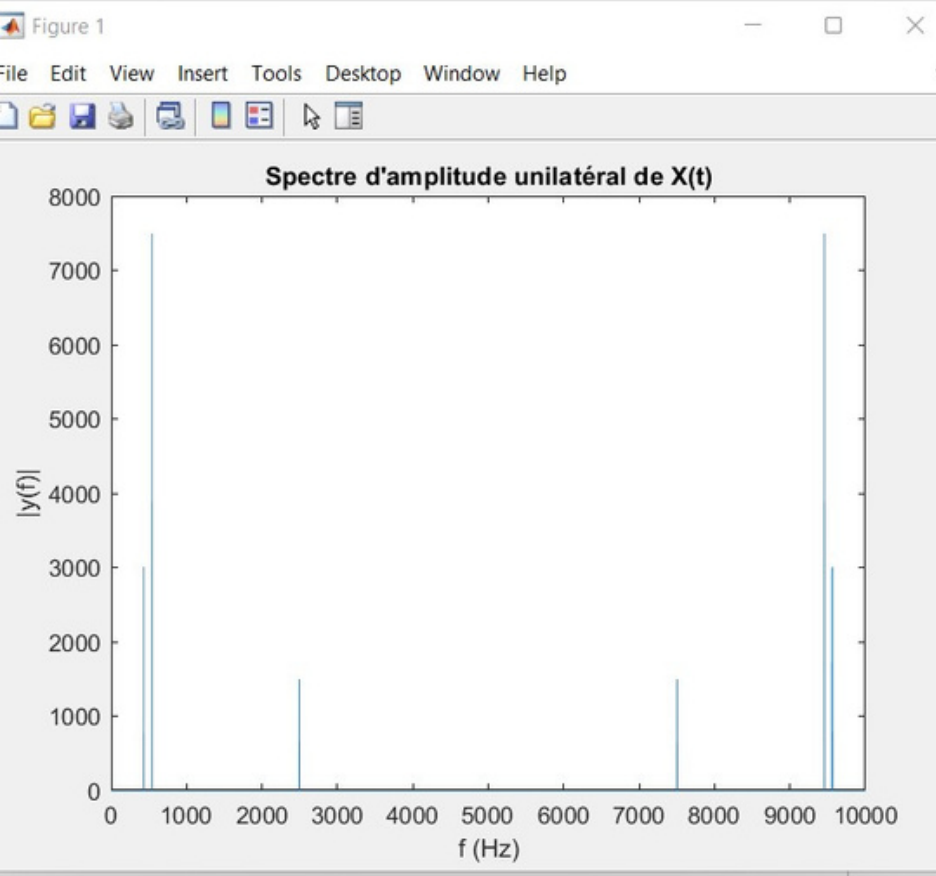


2- Calculer la TFD du signal $x(t)$ en utilisant la commande `fft` puis tracer son spectre en amplitude

=>code

```
clear all
clc
close all
fe = 1e4 ;
te = 1/fe ;
N = 5000 ;
t = (0: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);
y = fft(x);
f = fe*(0:(N-1))/N;
plot(f,abs(y))
title("Spectre d'amplitude unilatéral de X(t)")
xlabel("f (Hz)");
ylabel("|y(f)|");
```

=>représentation graphique



3. Pour mieux visualiser le contenu fréquentiel du signal, utiliser la fonction `fftshift`, qui effectue un décalage circulaire centré sur zéro du spectre en amplitude obtenu par la commande `fft`.

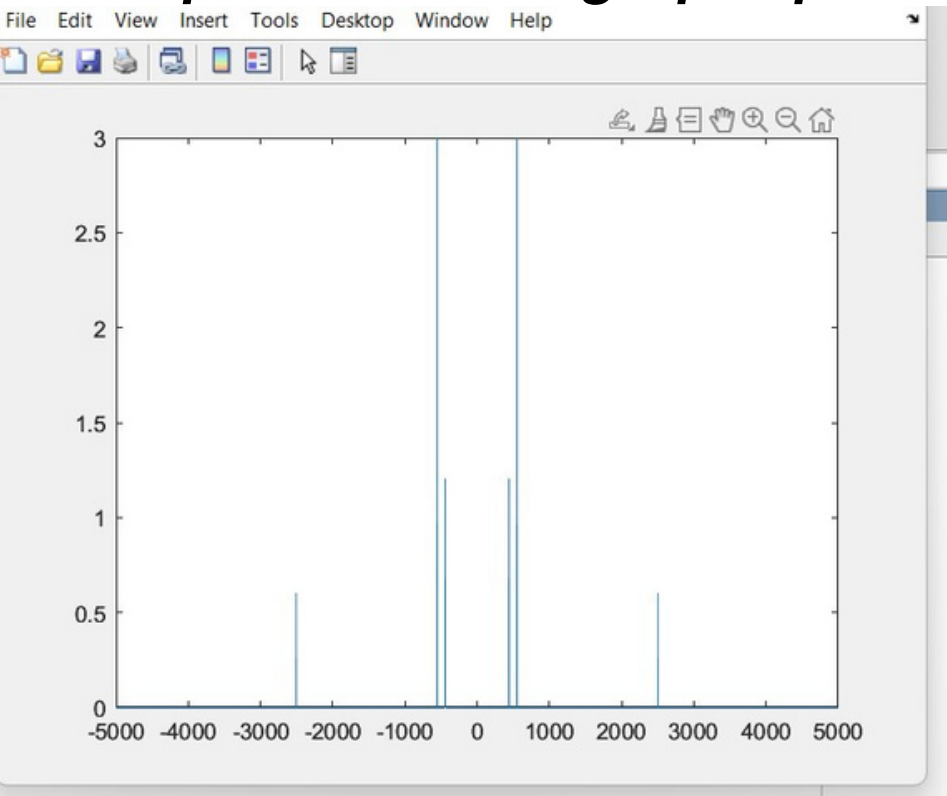
=>code

```

clear all
clc
close all
fe = 1e4 ;
te = 1/fe ;
N = 5000 ;
t = (0: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);
y = fft(x);
f = fe*(0:(N-1))/N;
y2 = fftshift(2*abs(y)/N)
fshift = (-N/2: (N/2)-1)*(fe/N);
plot(fshift , y2)

```

=>représentation graphique



4:Créer un nouveau signal xnoise, en introduisant un bruit blanc gaussien dans le signal d'origine $x(t)$, puis visualisez le

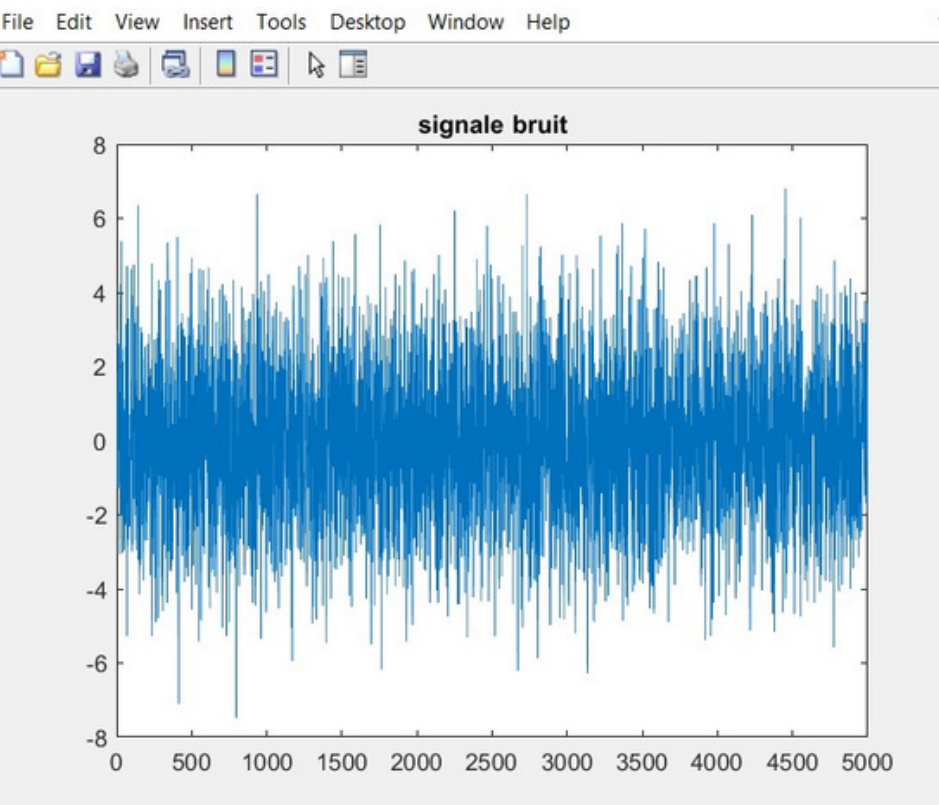
=>code

```

editor - C:\Users\fatim\OneDrive\Desktop\3eme annee\s5\signal\tp3\TP1.m
lab1.m x TP1.m x lab2.m x lab3.m x untitled x untitled2 * x +
1 clear all
2 clc
3 close all
4 fe = 1e4 ;
5 te = 1/fe ;
6 N = 5000 ;
7 t = (0:N-1)*te ;
8 x = 1.2*cos(2*pi*440*t+1.2)+3*cos(2*pi*550*t)+0.6*cos(2*pi*2500*t);
9
10 noise = 2*randn(size(x))
11 plot(noise)
12 title('signale bruit')

```

=>représentation graphique



5 : Utiliser la commande sound pour écouter le signal et puis le signal bruité.

=>code

```
Editor - C:\Users\fatim\OneDrive\Desktop\3eme année\s5\signal\tp3\TP1.m
TP1.m  x lab2.m  x lab3.m  x untitled2  x +
clear all
clc
close all

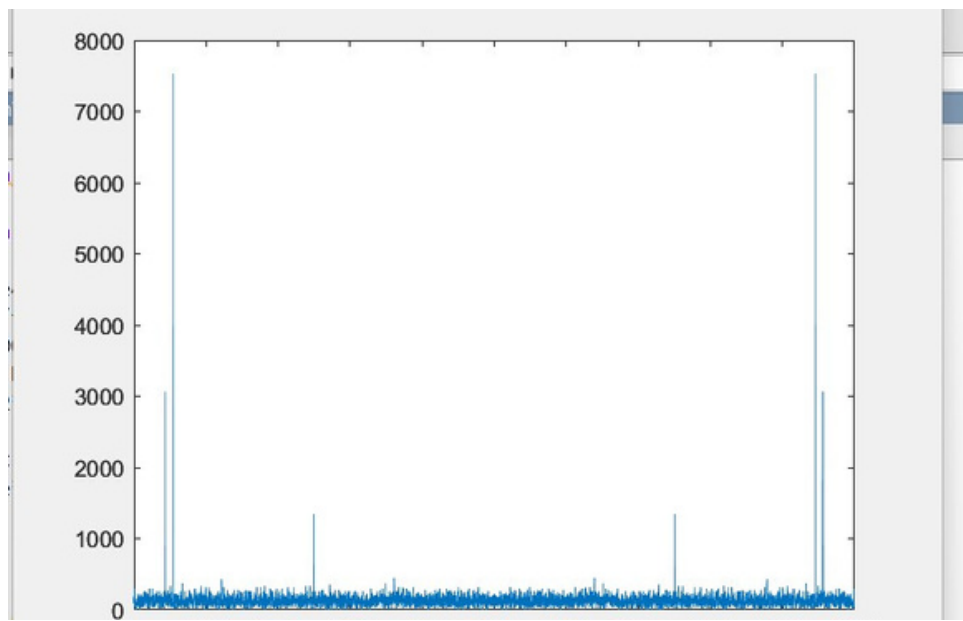
fe = 1e4 ;
te = 1/fe ;
N = 5000 ;
t = (0:N-1)*te ;
x = 1.2*cos(2*pi*440*t+1.2)+3*cos(2*pi*550*t)+0.6*cos(2*pi*2500*t);

y = fft(x);
f = fe*(0:(N-1))/N;

noise = 2*randn(size(x))
|
xnoise = noise + x ;

TFxnoise = fft(xnoise)
plot(f, abs(TFxnoise ))
```

=>représentation graphique

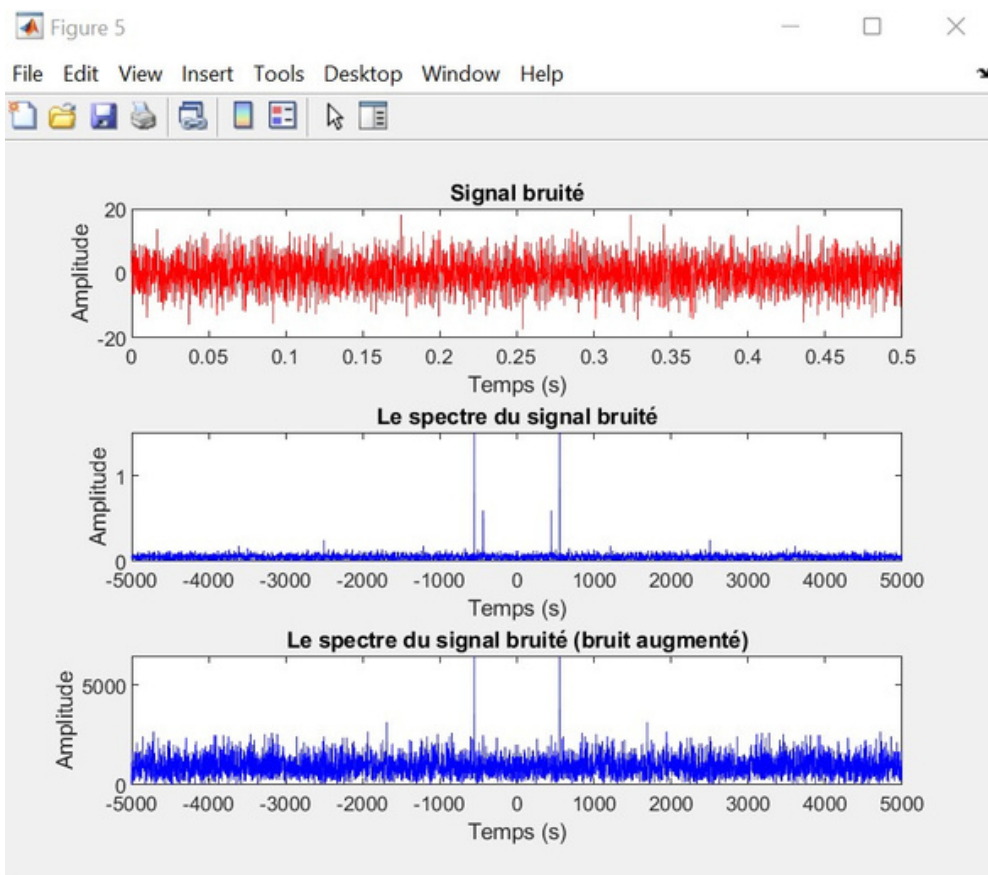


l'analyse de signale bruitée

=>code

```
Filtered_Spectrum=fftshift(Filter_Passe_Bas.*x_Spectrum);  
Figure  
subplot(3,1,1)  
plot(x_Spectrum)  
xlabel('Fréquence (Hz)')  
ylabel('Amplitude')  
title('Spectre du signal')  
subplot(3,1,2)  
plot(Filter_Passe_Bas)  
xlabel('Fréquence (Hz)')  
ylabel('Amplitude')  
title('Le filtre pass-bas')  
subplot(3,1,3)  
plot(Filtered_Spectrum)  
xlabel('Fréquence (Hz)')  
ylabel('Amplitude')
```

=>représentation graphique



2:Analyse fréquentielle du chant du rorqual bleu.

=>code

```
Editor - C:\Users\fatim\OneDrive\Documents\MATLAB\untitled3.m
untitled3.m x +
1
2 [signal,fs]=audioread('bluewhale.au');
3
4 Bluewhale_Sound = signal( 2.45e4:3.10e4 );
5
6 tt=( 0:length(Bluewhale_Sound)-1 )*( 10/fs );
7
8 n = 2^nextpow2(length(Bluewhale_Sound));
9
10 Bluewhale_Sound_Spectrum=fftshift( abs(fft(Bluewhale_Sound,n)).^2 / (n*fs) );
11
12 freq = fs/2*linspace(-0.5,0.5,n);
13
14 figure
15 subplot(2,1,1)
16 plot(tt, Bluewhale_Sound,'r')
17 xlabel('Temps (s)')
18 ylabel('Amplitude')
19 title('rorqual bleu')
20 subplot(2,1,2)
21 plot(freq, Bluewhale_Sound_Spectrum,'g')
22 xlabel('Temps (s)')
23 ylabel('Amplitude')
24 title('Le spectre du signal')
25
26 % sound(Bluewhale_Sound, fs)
27
28
29
30
```

=>représentation graphique

