École d'Ingénierie Digitale
et d'Intelligence Artificielle

الجامعة الأورومتوسطية بفاس
EURO-MEDITERRANEAN UNIVERSITY OF FES
UNIVERSITÉ EURO-MÉDITERRANÉENNE DE FÈS

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

Fait Par

BENDIDI OUMAIMA

Introduction

Lors de ce premier Tp on va représenter des signaux et l'application de la transformé de fourrier discrete.

On utilise la TFD lorsque l’on travaille avec des suites numériques sans lien avec un signal physique, pour définir une représentation de la suite sur une base de fonctions fréquentielles.

Ensuite,on va évaluer l'intérêt du passage du domaine temporel au domaine fréquentiel dans l'analyse et l'interpretation des signaux physiques réels .
Et on va essayer de filtrer idéalement un signal bruite.

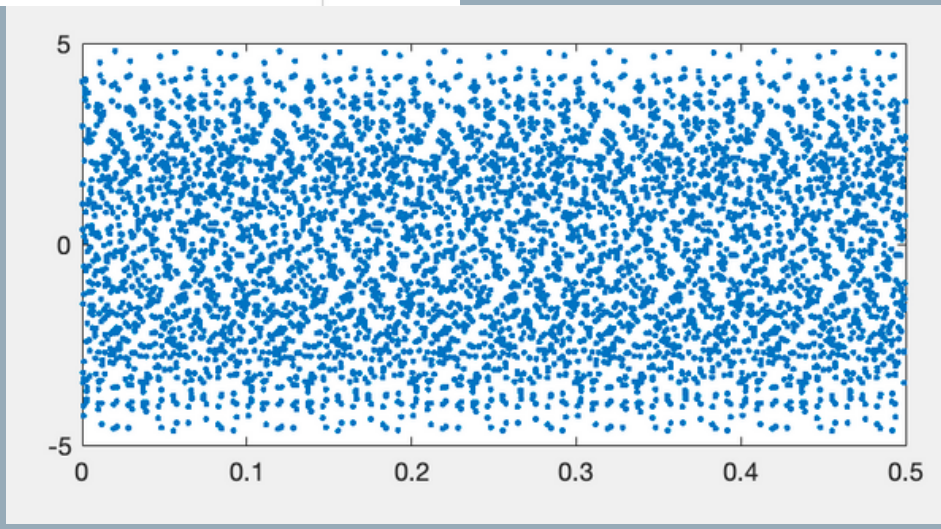
Représentation temporelle et fréquentielle

$$x(t) = 1.2\cos(2\pi 440t + 1.2) + 3\cos(2\pi 550t) + 0.6\cos(2\pi 2500t)$$

on Trace le signal x(t). avec une Fréquence d'échantillonnage : fe = 10000Hz, Intervalle : Nombre d'échantillons : N = 5000.

```
fe = 1e4;
Te = 1/fe;
N = 5000;
% generer un vecteur t qui contient des valeurs discretes et on applique le
% cosinus sur le vecteur t
  t = 0: Te: (N -1)*Te;
 %un signal enchantillone avec une frequence de 10^4
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,".")
```

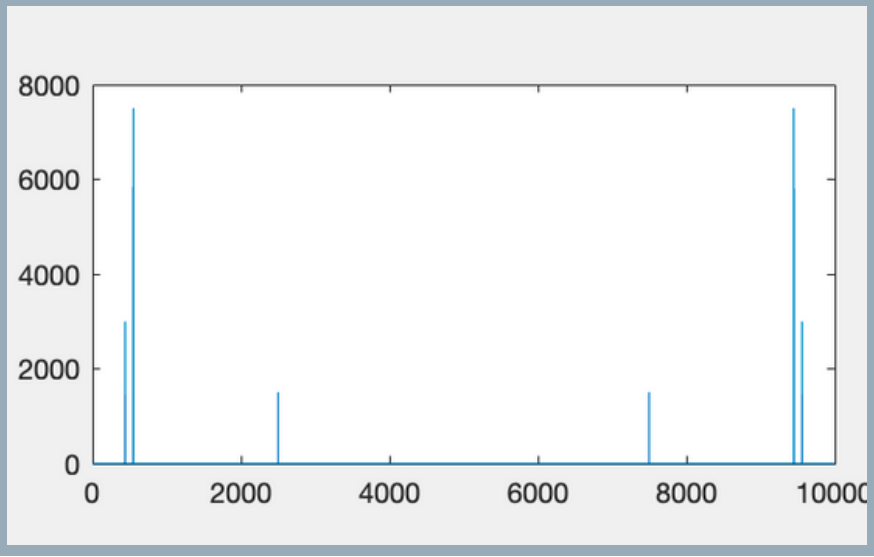
On a génére un total de 5000 échantillons



on va appliquer la TFD

l'application de tfd nous génère un spectre qui est une fonction complexe qui contient une partie imaginaire et une partie réelle .

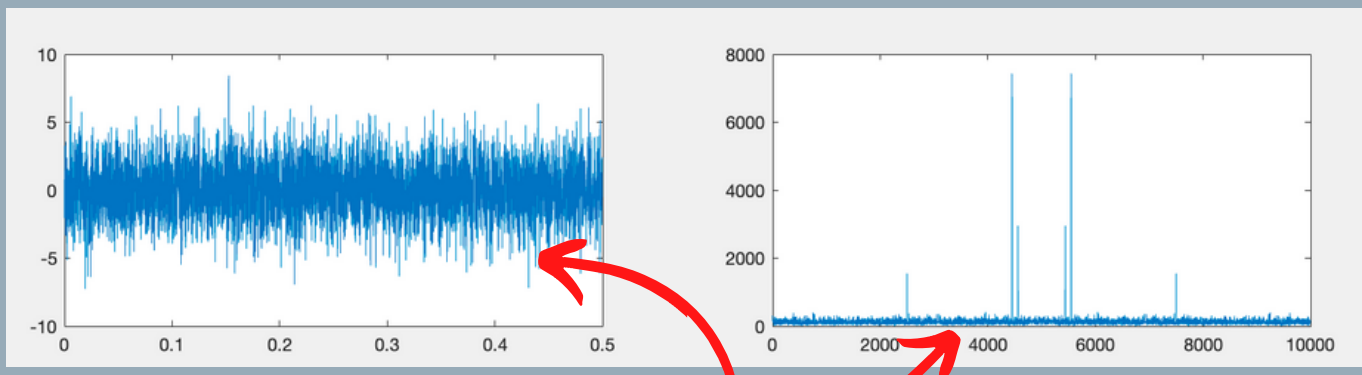
```
% % Application de la tfd
%on configure l'axe frequentiel avec un pas de descretisation de fe/N
fshift = 0: fe/N : (N -1)*(fe/N);
x = 1.2*cos(2*pi*440*t + 1.2) + 3*cos(2*pi*550*t) + 0.6*cos(2*pi*2500*t);
subplot(3, 2, 2)
%on represente le module du spectre d'amplitude
y= abs(fft(x));
plot(fshift,y)
```



On obtient 6 piques du spectre , le premier pique est à la fréquence 440 du premier cosinus qui constitue le signal x(t),le deuxième ets à 550 et le troisième est à 2500. les trois derniers piques sont symétriques par rapport à la fréquence fe/2.: **La symétrie conjuguée**

```
%on va injecter le bruit dans le signal
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))
```

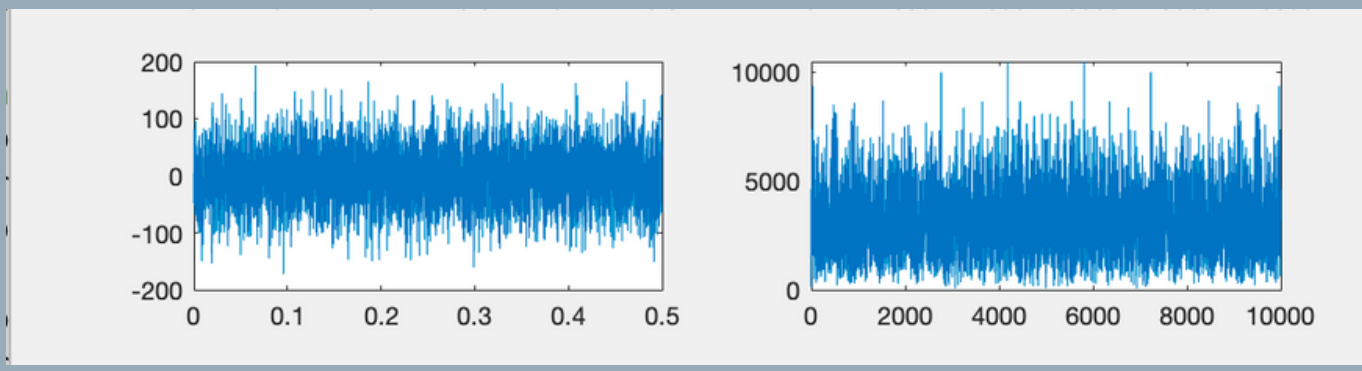


bruit

```
%on va intensifier le bruit dans le signal en augmentons le coefficients

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))
```



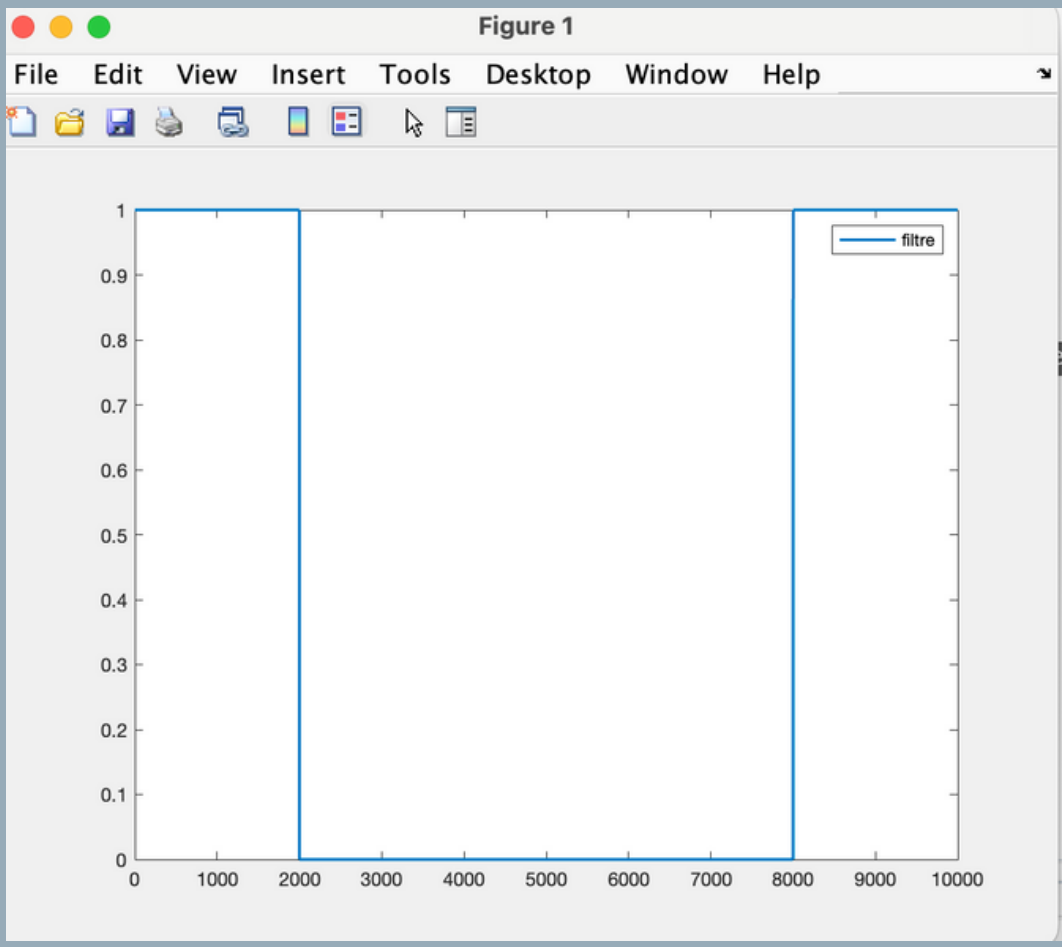
l'information est complètement perdu.il est difficile de faire une operation de filtrage.

Réaliser un filtrage ideal fréquentiel

On essaye d'éliminer une composante fréquentiel du signal. on a decider d'éliminer de le fréquence 2500 Hz. On va multiplier les piques moins de fréquence de coupure par 1 et au delà de la fréquence de coupure on multiplie par 0, par conséquent le pique sera éliminé.La fréquence de coupure qu'on va choisir entre 550 et 2500 Hz. le filtre qu'on va appliquer est un filtre passe-bas: il garde les basse fréquence et élimine les hautes fréquences.Physiquement on a aucun système passe-bas ou passe-haut c'est pour cela on appelle ce filtrage un filtrage ideal.

```
% Conception du filtre
  %initialisation du filtre
pass_bas = zeros(size(x));
 %definir la frequence de coupure
fc = 2000;
%l'index de la frequence de coupure

index_fc = ceil((fc*N)/fe);
%ceil Arrondit chaque élément de X à l'entier le plus proche, supérieur ou égal à cet élément.
pass_bas(1:index_fc) = 1;
pass_bas(N-index_fc+1:N) = 1;
plot(f,pass_bas,"linewidth",1.5)
legend("filtre")
```

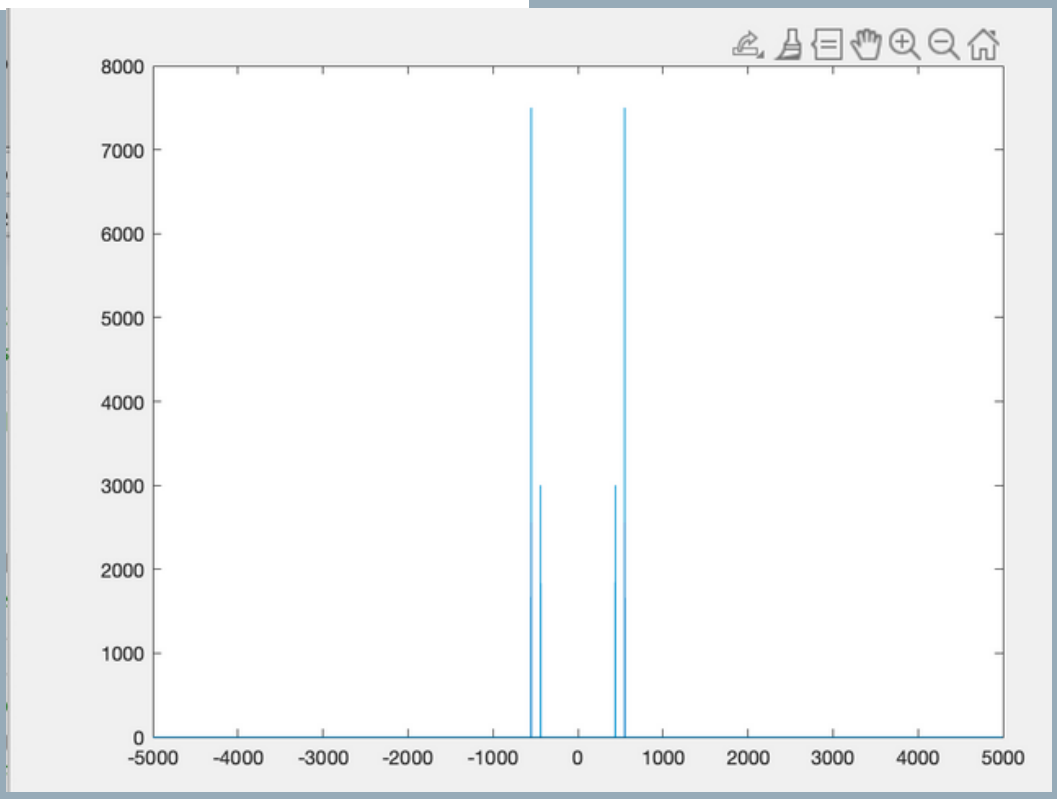


On peut remarquer que dans la conception du filtre on a pu rendre a 0 les valeurs qui sont plus grand que la fréquence de coupure

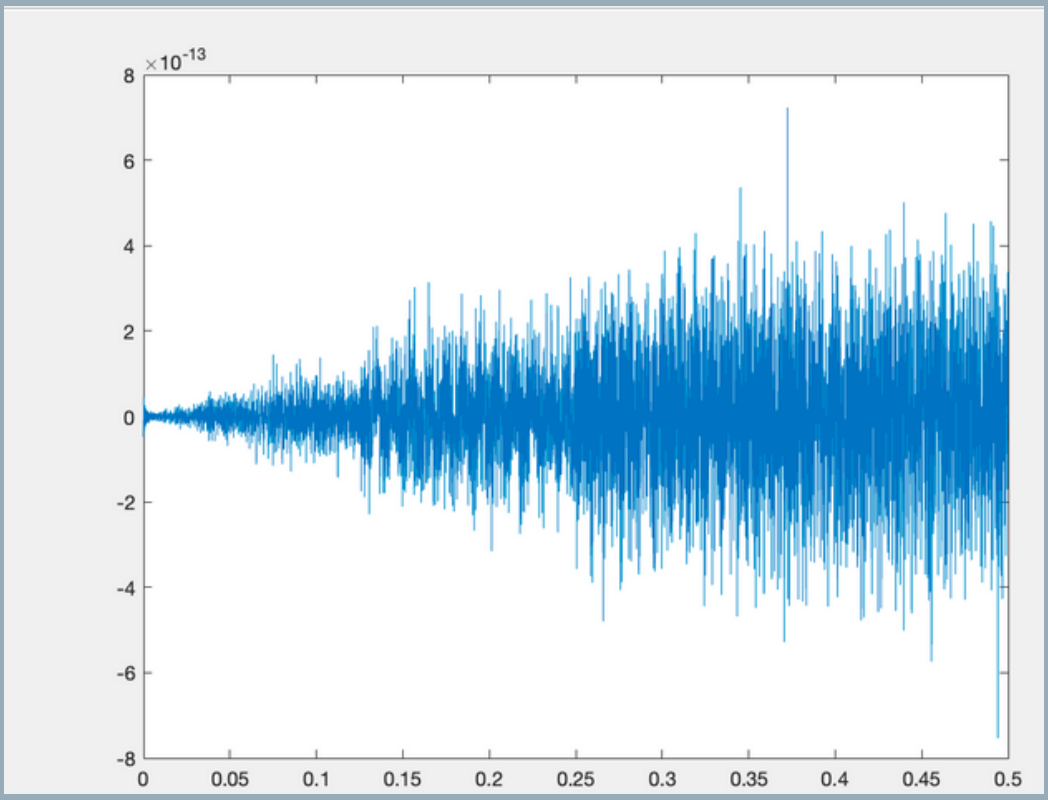
```
% Filtrage

x_filter_freq = pass_bas.*y;
x_filter_temp = ifft(x_filter_freq,"symmetric");

  plot(fshift, fftshift(abs(fft(x_filter_temp))));
```



On peut remarque que les piques superieur à la fréquence de coupure ont été éliminé

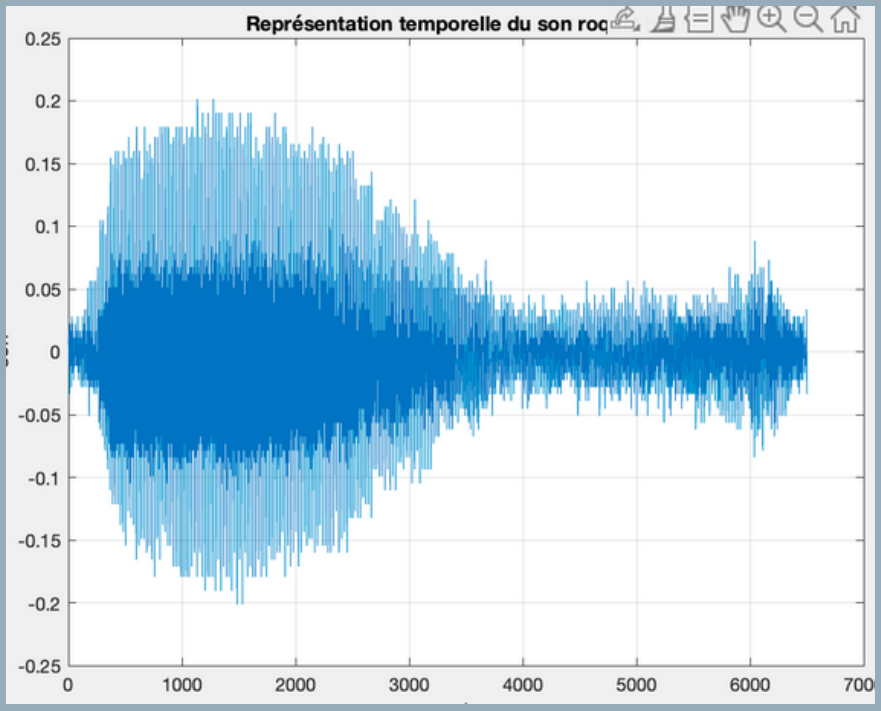


Analyse fréquentielle du chant du rorqual bleu

On affiche le signal du chant du roqal bleu

```
s='bluewhale.wav';
[son,Fe]=audioread(s);
sound(son,Fe);
son1 = son(2.45e4: 3.10e4);
sound(son,Fe);

plot(son1);
xlabel('t');
ylabel('son');
title('Représentation temporelle du son roqual bleu')
grid on
```



```
%%
    taille=size(data);
  %On applique la transformation de fourier rapide sur le chant
     Schant = fft(data);
    %Densité spectrale du Chant
    Densite_spectrale_chant = abs(Schant).^2/taille;
    f = (0:floor(taille/2))*(fs/taille)/10;
    plot(f,Densite_spectrale_chant(1:floor(taille/2)+1));
```

Ce code applique la transformée de Fourier rapide (FFT) aux données d'entrée et calcule la densité spectrale de puissance (des données. La variable "taille" est utilisée pour stocker la taille des données. La FFT est appliquée aux données et stockée dans la variable "Schant". La PSD est calculée en prenant la valeur absolue de la FFT et en divisant par la taille des données. La variable "f" est utilisée pour stocker les valeurs de fréquence pour l'axe des x du graphique, qui sont calculées en multipliant les valeurs d'index par la fréquence d'échantillonnage et en divisant par la taille des données.