

## Rapport Tp1



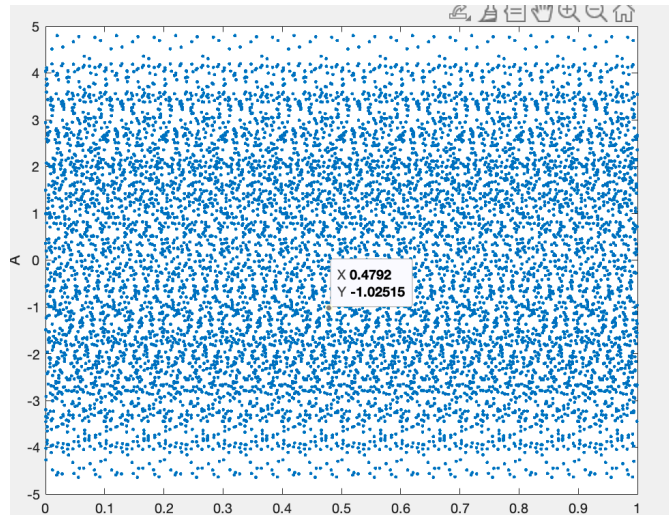
*Le traitement du signal est une discipline technique qui a pour objet la détection, l'élaboration et l'interprétation des signaux porteurs d'informations. Elle s'appuie sur la théorie du signal qui donne une description mathématique des signaux. Cette théorie fait essentiellement appel à l'algèbre linéaire, l'analyse fonctionnelle, l'électricité et l'étude des processus aléatoires.*

## Objectifs DU TP

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.

## Représentation temporelle et fréquentielle

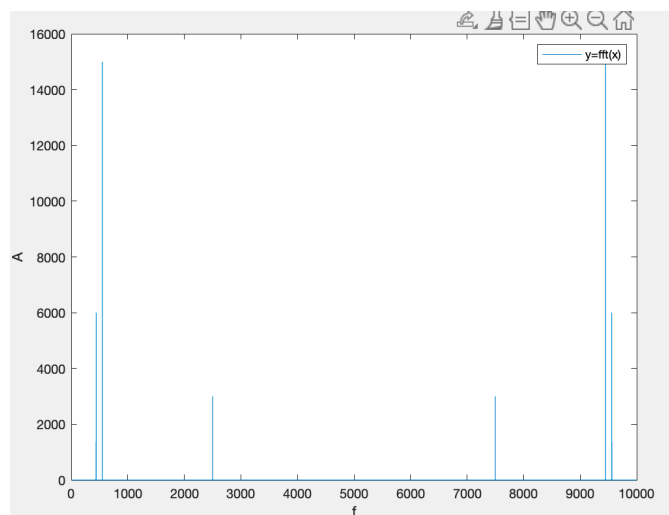
1) Tracer le signal  $x(t)$ . Fréquence d'échantillonnage :  $f_e = 10000\text{Hz}$ , Intervalle : Nombre d'échantillons :  $N = 5000$ .



Code :

```
fe = 1e4;
te = 1/fe;
N = 10000;
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);
f = (0:N-1)*(fe/N);
y = fft(x);
%Plot
plot(t, x, 'b')
```

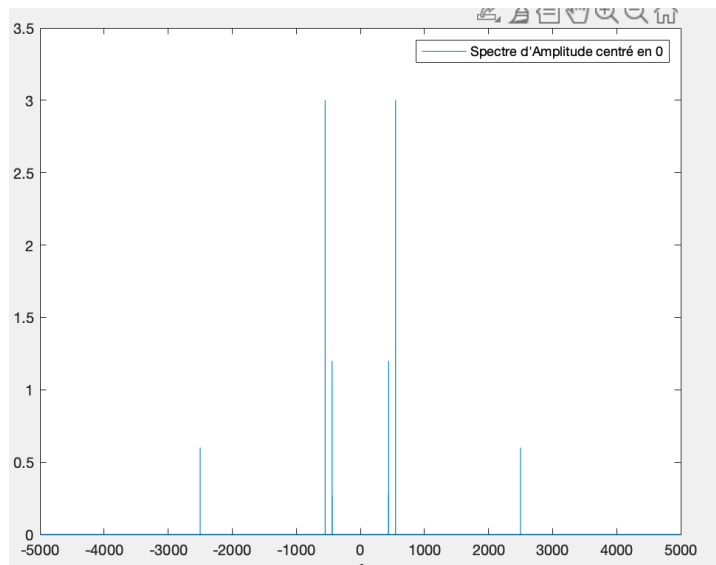
2) Calculer la TFD du signal  $x(t)$  en utilisant la commande `fft`, puis tracer son spectre en amplitude



**Code:**

```
%representation du spectre du signal x Transformé de Fourier
plot(f, abs(y))
legend("y=fft(x)");
xlabel("f");
ylabel("A");
```

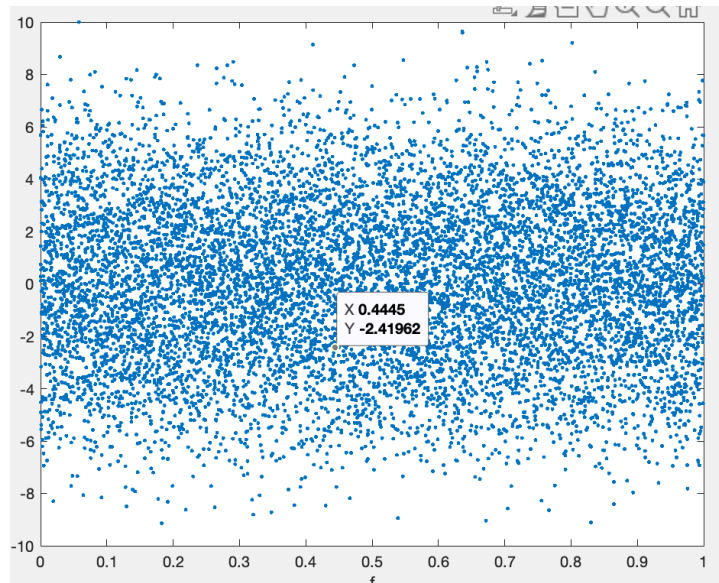
3) Pour mieux visualiser le contenu fréquentiel du signal, utiliser la fonction `fftshift`



**Code:**

```
fshift = (-N/2:N/2-1)*(fe/N);
z = fftshift(x);
%on represente le spectre d'amplitude centré à la fréquence zéro.
plot(fshift,fftshift(2*abs(y)/N));
legend("Spectre d'Amplitude centré en 0");
xlabel("f");
ylabel("A");
```

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



**Code:**

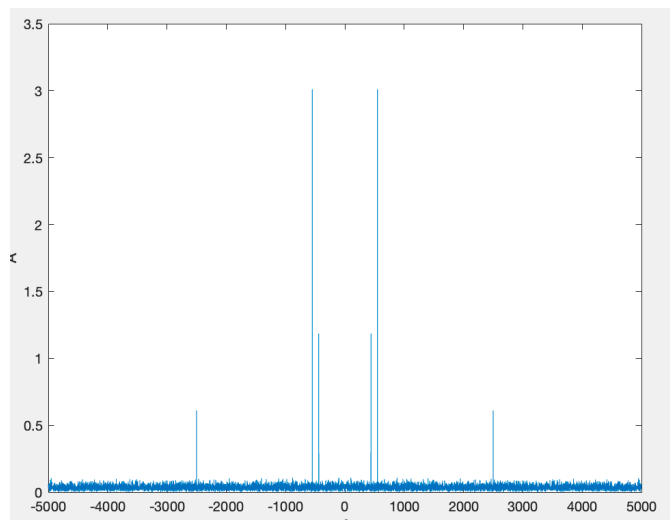
```
bruit = 2*randn(size(x)); %creation du bruit blanc gaussien de faible intensité
xbruit = x+bruit; %on l ajoute au signal x
plot(t,xbruit, 'b')
```

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

**Code :**

```
sound(xbruit,fe)
```

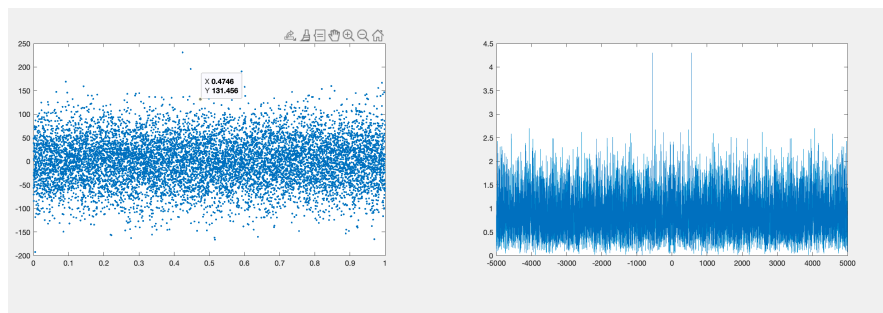
6) Calculez puis tracer le spectre de puissance du signal bruité centré à la fréquence zéro.



Code:

```
ybruit = fft(xbruit); %transformé de fourier du signal bruité
%on represente le spectre d'amplitude du signal bruité avec un bruit blanc gaussien de
faible intensité
plot(fshift,fftshift(2*abs(ybruit)/N))
xlabel("f");
ylabel("A");
```

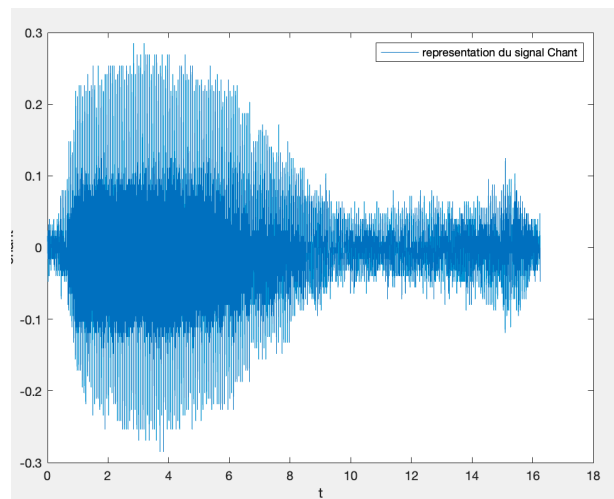
7) Augmenter l'intensité de bruit puis afficher le spectre



```
superbruit = 50*randn(size(x));
xsuperbruit = x+superbruit;
ysuperbruit = fft(xsuperbruit);
%on represente
subplot(2,2,1)
plot(t,xsuperbruit, '.')
subplot(2,2,2)
plot(fshift,fftshift(2*abs(ysuperbruit)/N))
```

## **Analyse fréquentielle du chant du rorqual bleu**

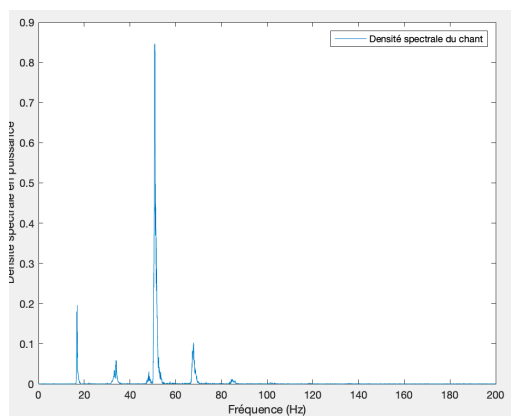
1) Chargez, depuis le fichier 'bluewhale.au', le sous-ensemble de données qui correspond au chant du rorqual bleu du Pacifique. En effet, les appels de rorqual bleu sont des sons à basse fréquence, ils sont à peine audibles pour les humains. Utiliser la commande `audioread` pour lire le fichier. Le son à récupérer correspond aux indices allant de  $2.45 \times 10^4$  à  $3.10 \times 10^4$ .



Code :

```
sound(chant,fs);  
plot(t,chant);  
legend("representation du signal Chant");  
xlabel("t");  
ylabel("chant");
```

2) Ecoutez ce signal en utilisant la commande `sound`, puis visualisez-le.

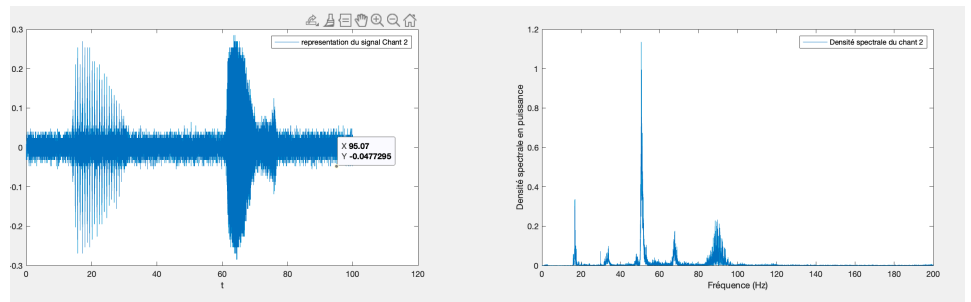


```

%On applique la transformation de fourier rapide sur le chant
Schant = fft(chant);
%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));
legend("Densité spectrale du chant");
xlabel("Fréquence (Hz)");
ylabel("Densité spectrale en puissance »);

```

3)Spécifiez une nouvelle longueur de signal qui sera une puissance de 2, puis tracer la densité spectrale de puissance du signal.



```

%On applique la transformation de fourier rapide sur le chant
Schant2 = fft(Chant2);
%Densité spectrale du Chant lorsqu on a augmenté la taille de l echantillon
Densite_spectrale_chant2 = abs(Schant2).^2/taille;

subplot(2,2,1)
sound(Chant2,fs);
plot(T,Chant2);
legend("representation du signal Chant 2");
xlabel("t");
ylabel("chant 2");

subplot(2,2,2)
f = (0:floor(Taille/2))*(fs/Taille)/10;
plot(f,Densite_spectrale_chant2(1:floor(Taille/2)+1));
legend("Densité spectrale du chant 2");
xlabel("Fréquence (Hz)");
ylabel("Densité spectrale en puissance");

```

