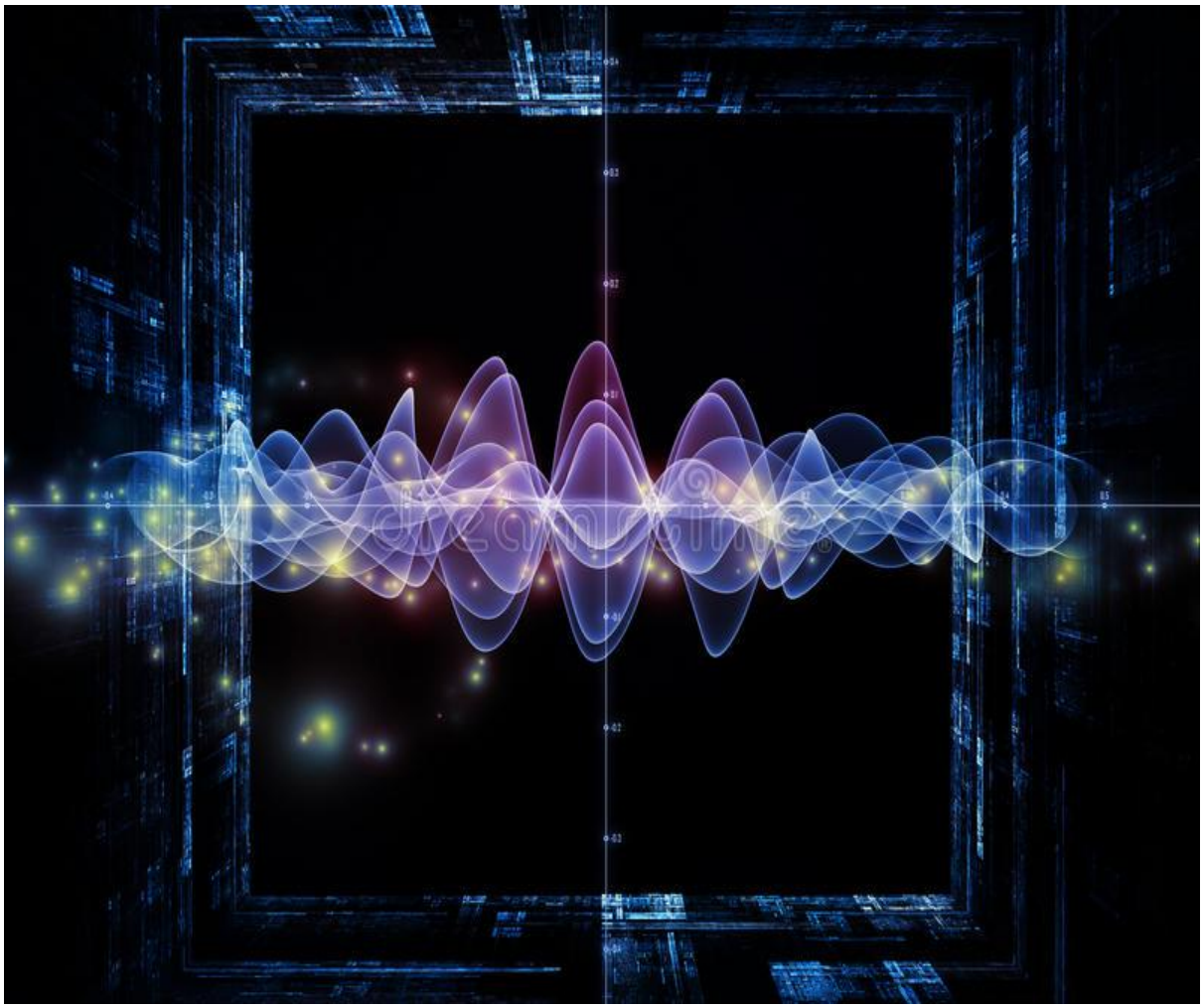
TP1- Analyse spectrale d'un signal



Réaliser par : Youssef laamari

Module : Traitement de signal

Prof : AMMOUR Alae

Objectifs :

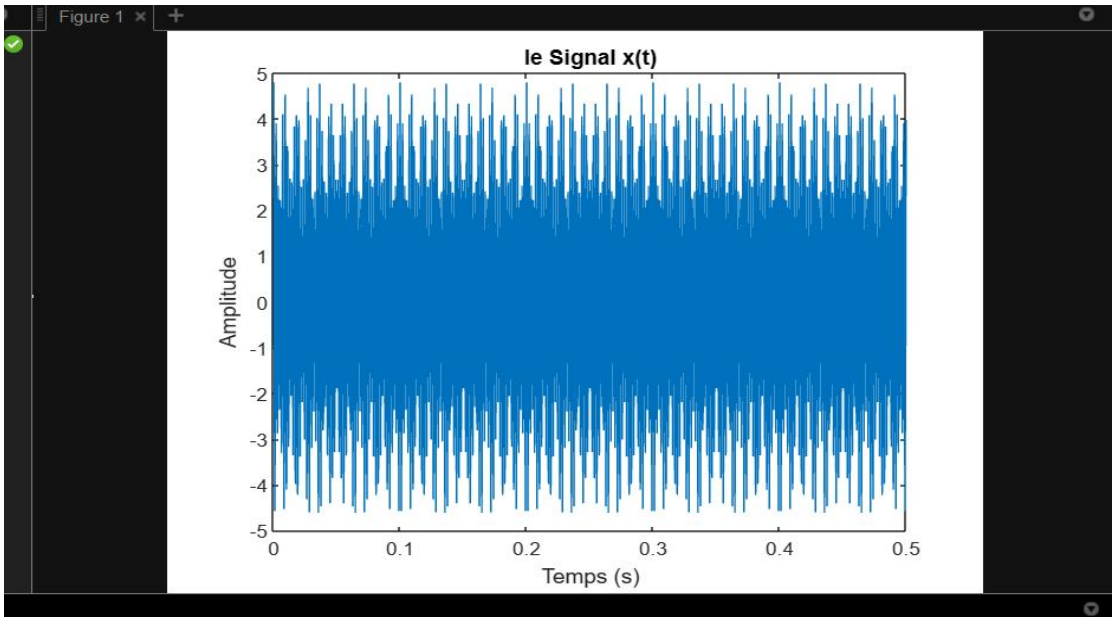
*** Représentation de signaux et applications de la transformée de Fourier discrète (TFD) sous Matlab.**
*** Evaluation de l'intérêt du passage du domaine temporel au domaine fréquentiel dans l'analyse et l'interprétation des signaux physiques réels. Commentaires : Il est à remarquer que ce TP traite en principe des signaux continus. Or, l'utilisation de Matlab suppose l'échantillonnage du signal. Il faudra donc être vigilant par rapport aux différences de traitement entre le temps continu et le temps discret. Tracé des figures : toutes les figures devront être tracées avec les axes et les légendes des axes appropriés. 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 :

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

1) Tracer le signal x(t). Fréquence d'échantillonnage : fe = 10000Hz, Intervalle : Nombre d'échantillons : N = 5000.

```
clear all
close all
clc
%****************** 1 **************
% Définition des paramètres du signal x(t)
fe = 10000; % fréquence d'échantillonnage
T = 1/fe; % pas d'échantillonnage
N = 5000; % nombre d'échantillons
t = 0:T:(N-1)*T; % vecteur temps

% Calcul de x(t)
x = 1.2*cos(2*pi*440*t + 1.2) + 3*cos(2*pi*550*t) + 0.6*cos(2*pi*2500*t);

% Tracé de x(t)
figure (1);
plot(t, x);
xlabel('Temps (s)');
ylabel('Amplitude');
title('le signal x(t)')
```

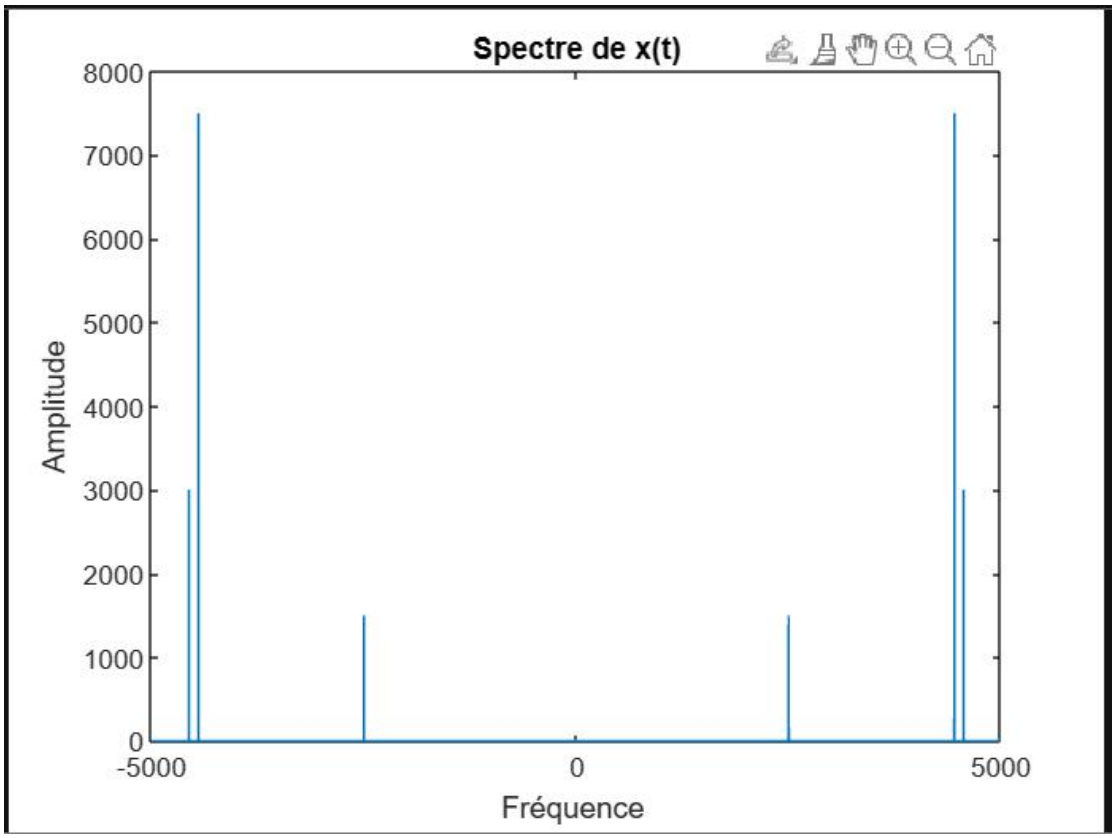


Pour approximer la TF continue d'un signal x(t), représenté suivant un pas Te, on utilise les deux commandes : fft et fftshif.

***On remarquera que la TF est une fonction complexe et que la fonction ainsi obtenue décrit la TF de x(t) entre –1/(2Te) et 1/(2Te) par pas de 1/(nTe) où n est le nombre de points constituant le signal x(t).**
*** La commande fft codant les fréquences positives sur les n/2 premières valeurs du signal et les valeurs négatives entre n/2+1 et n, la commande fftshift permet de les inverser.**

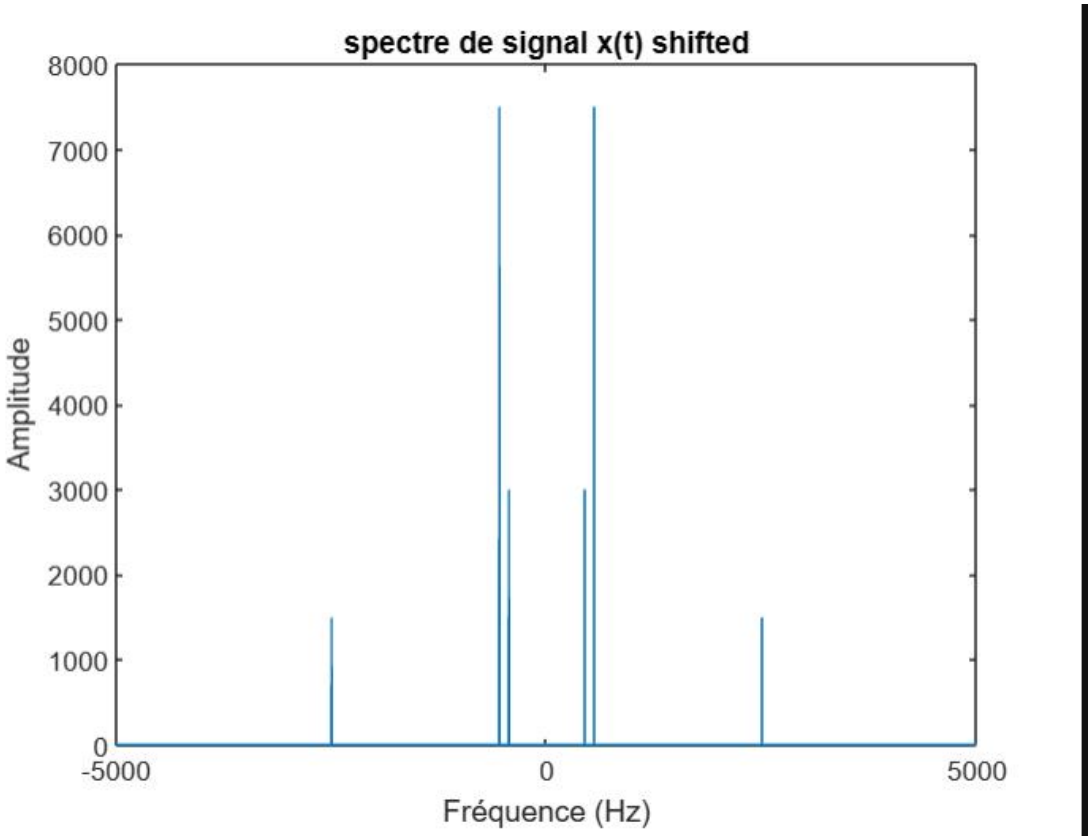
2) - Calculer la TFD du signal x(t) en utilisant la commande fft, puis tracer son spectre en amplitude après avoir créé le vecteur f qui correspond à l'échantillonnage du signal dans l'espace fréquentiel. Utiliser la commande abs pour afficher le spectre d'amplitude

```
%***************** 2 **************
% Calcul de la TFD avec la fonction fft
X = fft(x);

% Calcul de l'échantillonnage fréquentiel
f = linspace(-fe/2, fe/2, N);

% Tracé du spectre en amplitude
figure (2);
plot(f, abs(X));
xlabel('Fréquence (Hz)');
ylabel('Amplitude');
title('spectre de signal x(t)')
```



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

```
%****************** 3 ***************
% Utilisation de la fonction fftshift pour centrer le spectre en fréquence
autour de 0 Hz
X_shift = fftshift(X);

% Tracé du spectre en amplitude centré en fréquence autour de 0 Hz
figure (3);
plot(f, abs(X_shift));
xlabel('Fréquence (Hz)');
ylabel('Amplitude');
title('spectre de signal x(t) shifted')
```



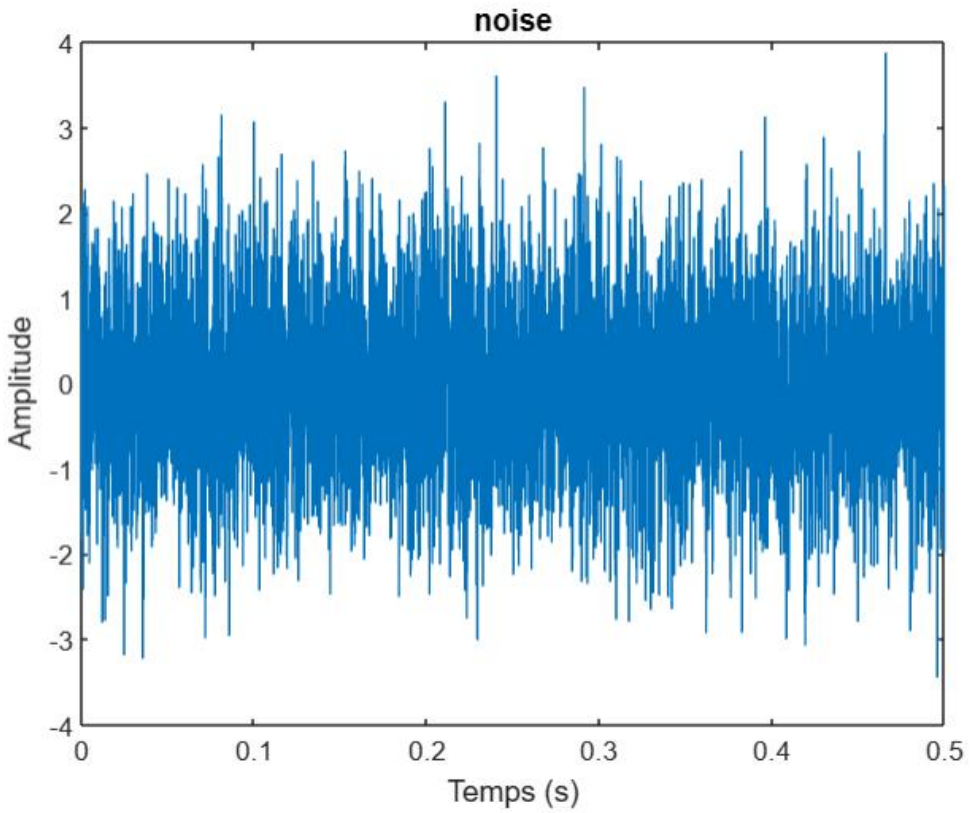
Commentaire :

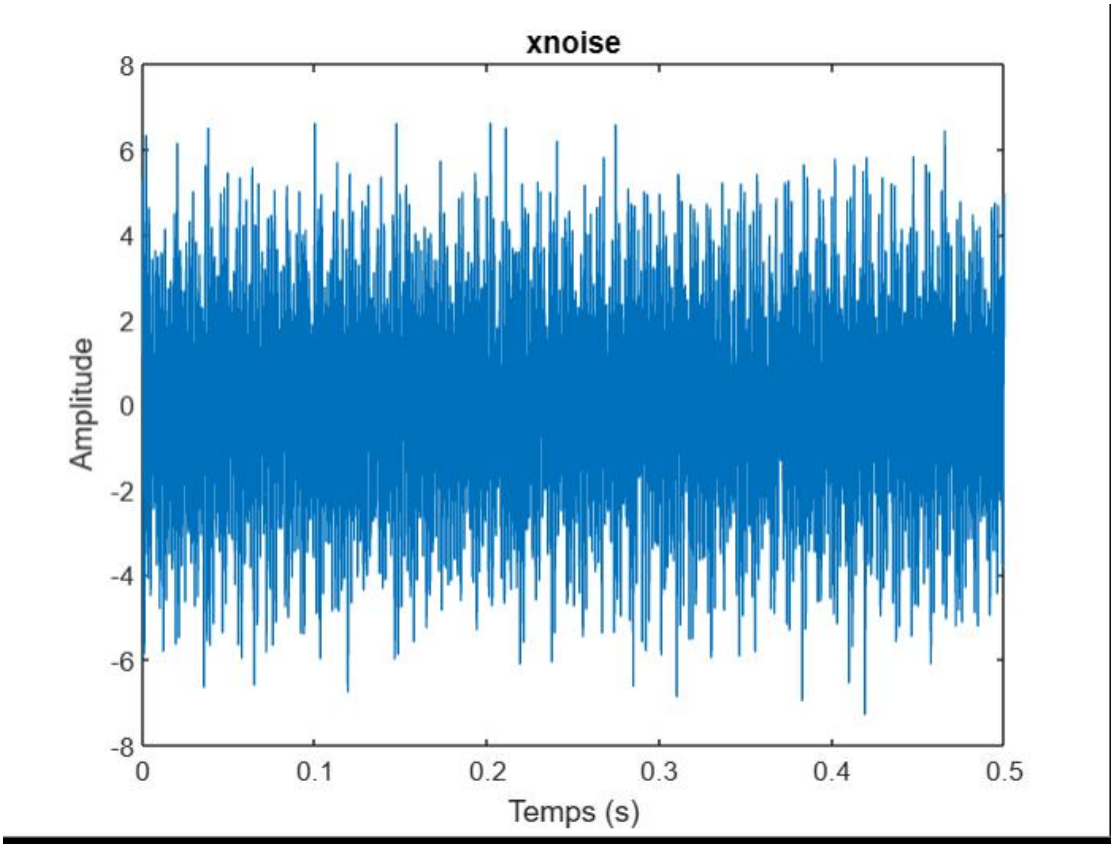
**** Le code commence par calculer la TFD de x(t) en utilisant la fonction fft(). Ensuite, il utilise la fonction fftshift() pour décaler circulairement le spectre d'amplitude de la TFD de x(t). Enfin, il crée un vecteur de fréquences qui correspond à la TFD et trace le spectre d'amplitude décalé en utilisant la fonction plot() de Matlab. La fonction abs() est utilisée pour obtenir l'amplitude de chaque composante de fréquence de la TFD décalée. ****

Un bruit correspond à tout phénomène perturbateur gênant la transmission ou l'interprétation d'un signal.
Dans les applications scientifiques, les signaux sont souvent corrompus par du bruit aléatoire, modifiant ainsi leurs composantes fréquentielles. La TFD peut traiter le bruit aléatoire et révéler les fréquences qui y correspond.
4- Créer un nouveau signal xnoise, en introduisant un bruit blanc gaussien dans le signal d'origine x(t), puis visualisez-le. Utiliser la commande randn pour générer ce bruit. Il est à noter qu'un bruit blanc est une réalisation d'un processus aléatoire dans lequel la densité spectrale de puissance est la même pour toutes les fréquences de la bande passante.

Ce bruit suit une loi normale de moyenne 0 et d'écart type 1.

```
%****************** 4 ****************
% Calcul du bruit blanc gaussien
noise = randn(1, N);

% Création du signal bruité xnoise
xnoise = x + noise;

% Tracé de xnoise
figure (4);
plot(t,noise)
xlabel('Temps (s)');
ylabel('Amplitude');
title('noise')
figure (5);
plot(t, xnoise);
xlabel('Temps (s)');
ylabel('Amplitude');
title('xnoise')
```





Commentaire :

****La commande randn génère un vecteur de nombres aléatoires suivant une loi normale de moyenne 0 et d'écart type 1, qui représente le bruit blanc gaussien. La commande plot permet de tracer le signal xnoise.**

Il est à noter que le bruit blanc gaussien est un type de bruit aléatoire qui a une densité spectrale de puissance constante pour toutes les fréquences de la bande passante. Cela signifie que le bruit est présent à toutes les fréquences de manière égale et que sa puissance est indépendante de la fréquence.**

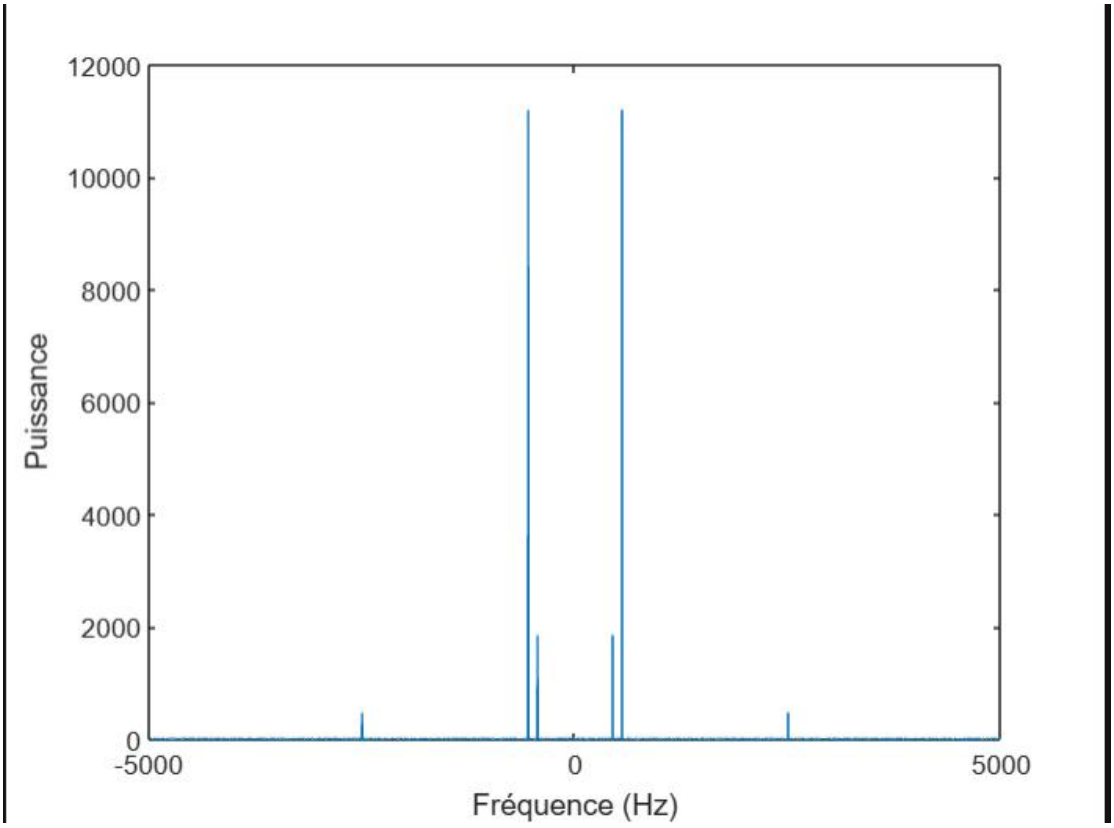
5)– Utiliser la commande sound pour écouter le signal et puis le signal bruité.

```
%****************** 5 ******************
% Écoute du signal x
sound(x, fe);

% Écoute du signal bruité xnoise
sound(xnoise, fe);
```

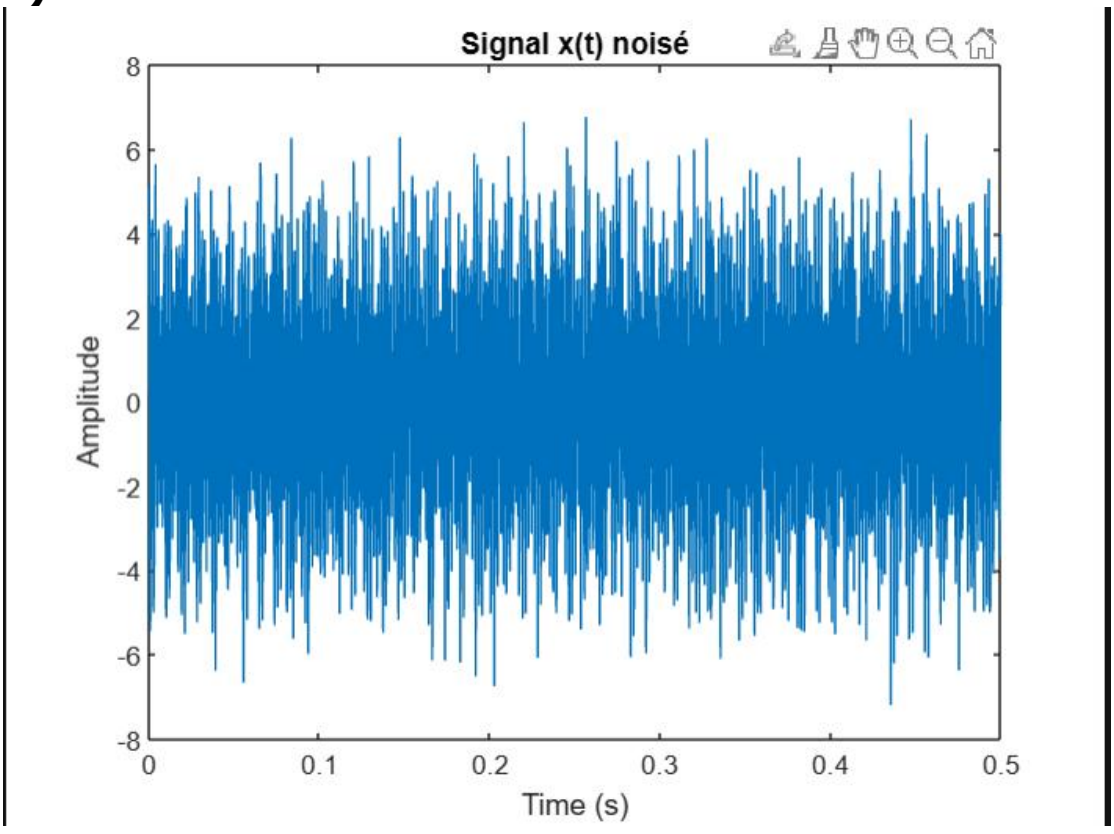
La puissance du signal en fonction de la fréquence (densité spectrale de puissance) est une métrique couramment utilisée en traitement du signal. Elle est définie comme étant le carré du module de la TFD, divisée par le nombre d'échantillons de fréquence.

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

```
%****************** 6 ******************
% Calcul de la TFD du signal bruité xnoise
Xnoise = fft(xnoise);

% Calcul du spectre de puissance du signal bruité
Pxnoise = (abs(Xnoise).^2)/N;

% Utilisation de la fonction fftshift pour centrer le spectre en fréquence
autour de 0 Hz
Pxnoise_shift = fftshift(Pxnoise);

% Tracé du spectre de puissance du signal bruité centré en fréquence autour
de 0 Hz
figure (6);
plot(f, Pxnoise_shift);
xlabel('Fréquence (Hz)');
ylabel('Puissance');
```


12000
10000
8000
6000
4000
2000
0
Puissance
-5000
0
5000
Fréquence (Hz)

7)


Signal x(t) noisé
8
6
4
2
0
-2
-4
-6
-8
Amplitude
0
0.1
0.2
0.3
0.4
0.5
Time (s)

;

Analyse fréquentielle du chant du rorqual bleu :

Il existe plusieurs signaux dont l'information est encodée dans des sinusoïdes. Les ondes sonores est un bon exemple. Considérons maintenant des données audios collectées à partir de microphones sous - marins au large de la Californie.
On cherche à détecter à travers une analyse de Fourier le contenu fréquentiel d'une onde sonore émise pas un 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 2.45e4 à 3.10e4.

```
clear all
close all
clc



[x,f] = audioread('phrase.wav');

chant=x(2.45e4:3.10e4);
```

Il semble que ce script Matlab utilise la fonction audioread pour lire un fichier audio phrase.wav, puis extrait un segment du signal audio (un "chant") en utilisant la syntaxe x(2.45e4 :3.10e4).

Le signal audio est stocké dans la variable chant.
La fonction audioread est une fonction de la bibliothèque d'échantillonnage de Matlab qui permet de lire des fichiers audio au format WAV ou FLAC. Elle prend en entrée le nom du fichier audio à lire, et renvoie un tableau de données audio et une fréquence d'échantillonnage (‘x’ et ’f’ respectivement).

2- Écoutez ce signal en utilisant la commande sound

```
clear all
close all
clc



[x,f] = audioread('phrase.wav');

chant=x(2.45e4:3.10e4);
soundsc(x,f);
```

3)

```
clear all
close all
clc

[x,f] = audioread('phrase.wav');

chant=x(2.45e4:3.10e4);
soundsc(x,f);
Nombrechant=length(chant);
t=[0:Nombrechant-1]*1/f;

%visualisez le signal
figure (1)
plot(t,chant);
title('Le signal de phrase fichier');
fshift=[-Nombrechant/2:(Nombrechant/2)-1]*(f/Nombrechant)/10;

%Densité spectrale de puissance de 2
densitespectralepuissance=abs(fft(chant).^2/Nombrechant);

%afficher la courbe
figure (2)
```

```
plot(fshift,fftshift(densitespectralepuissance));
title('la densité spectrale de puissance du signal');
```