TP 2 : traitement numérique du signal
Echantillonnage et TFD

1. Introduction

L'objectif de ce TP est de se familiariser avec les opérations d'échantillonnage des signaux dans le domaine temporel et le calcul des spectres de Fourier discrets en utilisant la transformée de Fourier discrète (TFD) ainsi que son algorithme rapide la FFT (Fast Fourier Transform).

2. Signaux numériques

Un signal à numériser doit d'abord être échantillonné. D'après Shannon la fréquence d'échantillonnage doit être égale au moins au double de la fréquence maximale du signal à échantillonner.

Commençons par créer un signal échantillonné. Il est composé de N échantillons et formé par la somme de trois sinusoïdes de fréquences égales respectivement à 1kHz, 2kHz et 4kHz. La fréquence d'échantillonnage est de 20kHz. Les amplitudes maximales de ces trois sinusoïdes sont respectivement 2.0, 1.2 et 0.75.

- Est-ce que la condition de Shannon est respectée ?

Supposons que le nombre d'échantillons à avoir dépend de la fréquence maximale du signal et aussi de la fréquence d'échantillonnage en utilisant l'expression suivante :

$$N = 2 \times \frac{f_e}{f_{\max}}$$

f_e fréquence d'échantillonnage

$f_{\max}$ fréquence maximale

- Calculer le nombre d'échantillons N
- Calculer la puissance moyenne de ce signal
- Ecrivez et exécutez ce script

```
f1 = 1000; %La fréquence de la première sinusoïde
f2 = 2000; %La fréquence de la deuxième sinusoïde
f3 = 4000; %La fréquence de la troisième sinusoïde
fe = 20000; %La fréquence d'échantillonnage
N = 2*fe/f3; % Dans cet exemple on va prendre le nombre d'échantillons qui
dépend de la fréquence maximale
t = [0:1:N-1]; %Nous générons un vecteur normé
T = t / fe; %Vecteur de temps [Te 2*Te 3*Te ... (N-1)*Te]
x = 2*cos(2*pi*f1*T)+ 1.2*cos(2*pi*f2*T)+ 0.75*cos(2*pi*f3*T); %x est donc un
vecteur de N points représentant un cosinus échantillonné
plot(T, x, '-+') %On trace la courbe
title('Echantillonnage d'un signal périodique')
xlabel('Temps (s)')
ylabel('Amplitude')
E = (x*x')/N %Calcul de l'énergie : Somme des carrés des termes divisée par le
nombre d'échantillons
```

- Quelles sont les résultats obtenus ?
- Que représente la valeur de E calculée dans ce script ?
- Quelle est la formule utilisée dans le programme pour calculer E ?
- Comparer avec le calcul théorique de l'énergie.
- Reprendre le même script en multipliant par 4 le nombre de point du vecteur t (le pas au, lieu de 1 doit être de 0.25). Y'a-t-il un changement ? Pourquoi ?

Travail à effectuer

- Refaire la même chose, en utilisant successivement les deux cas pour le vecteur t, mais avec un signal généré composé de quatre sinusoïdes. Les trois premières comme pour l'exemple précédent. La quatrième possède une fréquence égale à 12kHz et une amplitude égale à -0.5
- Répondez de nouveaux à toutes les questions précédentes
- Adapter le programme précédent à ce cas
- Refaire de nouveau la même chose en utilisant un fichier son (wave) de quelques dizaines de secondes.
- Commentez

2. Transformation de Fourier discrète

Nous rappelons que le TFD permet d'obtenir un spectre discret pour un signal discret. Si le signal discret est composé de N échantillons alors le spectre discret obtenu grâce à la TFD aura également N coefficients. La TFD périodise le signal et son spectre. En effet, l'échantillonnage du signal introduit une périodisation du spectre. Comme l'échantillonnage du spectre introduit une périodisation du signal. En utilisant une TFD sur un signal de N échantillons on aura un spectre de N échantillons par période. L'expression de la TFD est donc la suivante

$$X\left(\frac{n}{T_0}\right) = TFD(x(kT_e)) = \sum_{k=0}^{N-1} x(kT_e) \cdot e^{-2\pi \cdot \frac{kn}{N}}$$

Avec $n=0, 1, 2, \dots, N-1$

T_e : pas d'échantillonnage dans le domaine temporel

$\frac{1}{T_0}$: pas d'échantillonnage dans le domaine fréquentiel

$$T_0 = NT_e$$

Nous allons maintenant implémenter sous Matlab l'expression de la TFD. Nous allons utiliser le signal x du premier exemple formé par les trois sinusoïdes avec le même nombre de points déterminé précédemment. Evidemment, en prenant les deux cas successivement pour $f_e=10000$ et ensuite pour $f_e=20000$.

```
tic      %Pour lancer un chronomètre qui va nous permettre de calculer le temps
d'exécution de la TFD
X=1:N; %On crée un vecteur de N points
X=X*0; %On initialise tous les points à 0
W=exp(2*pi*j/N); %Il s'agit d'un terme constant que nous calculons à la
puissance kn
for n=1:N %On calcule le spectre X(n)du signal x(k)
    Wk=W^(n-1);
    X(n)=0 ;
    for i=1:N %On fait la somme de 0 à N-1
        X(n) =X(n) + x(i)*Wk^(-i+1);
    end
end
temps=toc      %mesure et affiche le temps d'exécution
figure
plot((0:N-1)*fe/2,abs(X)) % On trace le module du spectre
title('TFD calculée du signal périodique')
xlabel('Fréquence (Hz)')
ylabel('Amplitude')
```

- Existe-t-il des pics dans le spectre ? de combien de pics est formé le spectre tracé ?
- Pourquoi doit-on trouver beaucoup de pics ? Les pics sont ils lisibles ? Pourquoi ?
- Retrouvez la valeur E calculée précédemment à partir des amplitudes des pics. Y'a-t-il conservation d'énergie comme le stipule le théorème de Parseval ?
- Le temps d'exécution de la TFD calculé par le programme (instructions tic toc) est il important ? Dépend il du nombre d'échantillons N (c-à-d pour les deux cas du vecteur t)?

Remarque : La largeur des raies est importante à cause de f_e . En effet, nous n'avons pas assez de points surtout pour le premier cas (10 pour un intervalle de 20000Hz) et c'est pour cela que les raies sont larges de 2000Hz.

L'incrément fréquentiel est donc : $\Delta f = \frac{f_e}{N} = 2000\text{Hz}$

Travail à effectuer

- Refaire de nouveau la même chose en utilisant un fichier son (wave) de quelques dizaines de secondes.
- Commentez

3. Transformée de Fourier Rapide FFT

Comparer le résultat précédent avec celui obtenu à l'aide de l'algorithme de la Transformée de Fourier Rapide (fonction `fft()` de MATLAB).

```
tic
X2 = fft(x)
Temps2=toc
figure
hold on %On va afficher les deux courbes sur la même figure
plot((0:N-1)*fmax/2,abs(X2),'-+') %On trace la fft calculée par MATLAB en bleu
plot((0:N-1)*fmax/2,abs(X),'r') %On trace celle que nous avons calculé plus
haut en rouge
legend('Calcul avec fft()', 'TFD calculée')
title('Comparaison entre la TFD calculée et le calcul avec fft()')
xlabel('Fréquence (Hz)')
ylabel('Amplitude')
```

- Comparer le temps de calcul de la TFD et la FFT.
- Comparer les deux spectres par TFD et par FFT. Laquelle est plus précise TFD ou FFT ? pourquoi ?

4. Transformée de Fourier Inverse - Reconstruction

Sous matlab la TFD inverse rapide est calculée à l'aide de l'instruction `ifft`. Comparer le signal reconstruit à l'aide de la FFT inverse au signal original de la question 1. Pour cela nous utilisons ce script qui permet d'afficher les deux courbes, à savoir le signal `x` et le signal déterminé par `ifft`, afin de les comparer :

```
figure
hold on
plot(T, x, '-+') %On affiche notre signal échantillonné
plot(T, ifft(X), 'r') %On affiche la reconstruction
legend('Signal initial', 'Calcul avec ifft()')
title('Comparaison entre le signal initial et la reconstitution')
xlabel('Temps (s)')
ylabel('Amplitude')
```

- Que peut-on en déduire ?