Travaux Pratiques

« Traitement Numérique du Signal avec Python »

Niveau : 2e année Ingénieur Data Science

TP3 - Transformée de Fourier Discrète (TFD – FFT)

Youssef ZOUHIR 2021/2022

Transformée de Fourier Discrète

Le présent TP est une initiation à la manipulation, sous Matlab, des représentations spectrales. Ces représentations basées sur le spectre permettent de mettre en évidence des propriétés du signal dans le domaine fréquentiel. Dans ces représentations, plusieurs facteurs sont à prendre en considération : l'échantillonnage temporel et fréquentiel, la troncature, la résolution spectrale.

1. Rappel théorique de la TFD

Un calculateur numérique nécessite la transformée de Fourier discrète pour calculer la Transformée de Fourier d'un signal. D'un point de vue mathématique, la TFD transforme une séquence de données, du domaine de définition original (le domaine temporel) au domaine transformé (le domaine fréquentiel).

La transformée de Fourier d'un signal d'énergie finie x(t) est définie par :

$$X(\gamma) = \int_{-\infty}^{+\infty} x(t) e^{-j2\pi\gamma t} dt$$

Afin d'être adaptée pour un calculateur numérique, le signal x(t) doit être remplacé par des échantillons x(kTe) calculés sur un ensemble limité de valeurs N. L'expression devient alors :

$$X(\gamma) = \sum_{k=0}^{N-1} x(kTe) e^{-j2\pi\gamma kTe}$$

Les valeurs de la fréquence γ doivent également constituer un nombre limité, multiple d'un certain pas de fréquence $n\Delta\gamma$, l'expression devient :

$$X(n\Delta\gamma) = \sum_{k=0}^{N-1} x(kTe) e^{-jn\Delta\gamma kTe}$$

Par un choix simplificateur de $\Delta\gamma = \frac{1}{NTe}$, l'expression prend la forme suivante :

$$X(n\Delta\gamma) = X\left(\frac{2\pi n}{NTe}\right) = \sum_{k=0}^{N-1} x(kTe) e^{\frac{-j2\pi kn}{N}}$$

On pose $X\left(\frac{2\pi n}{NTe}\right) = X(n)$, la transformée de Fourier discrète devient :

$$X(n) = \sum_{k=0}^{N-1} x(kTe) e^{\frac{-j2\pi kn}{N}}$$

Comme l'application TFD est bijective, la transformée de Fourier inverse existe et elle prend la forme suivante :

$$x(k) = \frac{1}{N}\sum_{n=-N/2}^{N/2-1} X(n)\, e^{\frac{j2\pi nk}{N}}$$

Ces transformées sont calculées par des algorithmes efficaces de calculs appelées FFT (Fast Fourier Transforme). Cet algorithme requiert un nombre de points fréquentiel multiple de 2. Afin d'obtenir la meilleure efficacité en termes de temps de calcul, le spectre est calculé aux fréquences normalisées $\gamma = \frac{n}{N_f}$, $n = 0,1,\dots,N_f - 1$.

Manipulation

1. Analyse spectrale

Cette section aborde les méthodes de calcul du spectre d'amplitude d'un signal échantillonné et périodique par la Transformée de Fourier Discrète (TFD – FFT). Les résultats obtenus sont commentés lorsque la FFT d'un signal est calculé avec un nombre entier ou non-entier de périodes.

1.1. On considère un signal sinusoïdal x(t), de longueur 32 échantillons, de fréquence 10 Hz et échantillonné à 80 Hz, compléter la séquence Python suivante, pour générer et représenter le spectre d'amplitude du signal x(t), (le signal x(t) est observé sur un nombre entier de périodes, voir Figure 1).

```
%    La figure 1 donne le résultat que vous devez obtenir.
%    Le calcul de la FFT avec un nombre entier de périodes

import numpy as np
import matplotlib.pyplot as plt

f0 = 10;                          % fréquence du signal sinusoidal
Fe = 80;                          % fréquence d'échantillonnage
N  = 32;                          % longueur du signal:N-échantillons
T  =  …… ;                        % durée d'acquisition
T0 =  …… ;                        % pas d'échantillonnage

t  =  ……… ;                       % vecteur temps
x  = np.cos(…………) ;               % signal
```

```
% Représentation du signal
plt.subplot(211);
plt.stem(t,x);
plt.xlabel('temps(s)')
plt.ylabel('amplitude')
                                    % Nf est appelé l'ordre de la fft,
% Calcul de la TFD avec la fft      il détermine le nombre de points
N  = len(x);                        fréquentiels
Nf = nextpow2(N);
                                    % Spectre d'amplitude
                                    % Définition de l'axe fréquentiel

X  = np.fft.fft(x,Nf)/N;
Xp = ………;
fp=…………………………;
                                    % Représentation du module du
%-Représentation fréquentiel        spectre symetrisé avec fftshift

plt.subplot(212)
plt.stem(fp,fftshift(Xp))
plt.xlabel('fréquence (Hz)')
plt.ylabel('spectred''amplitu)
```

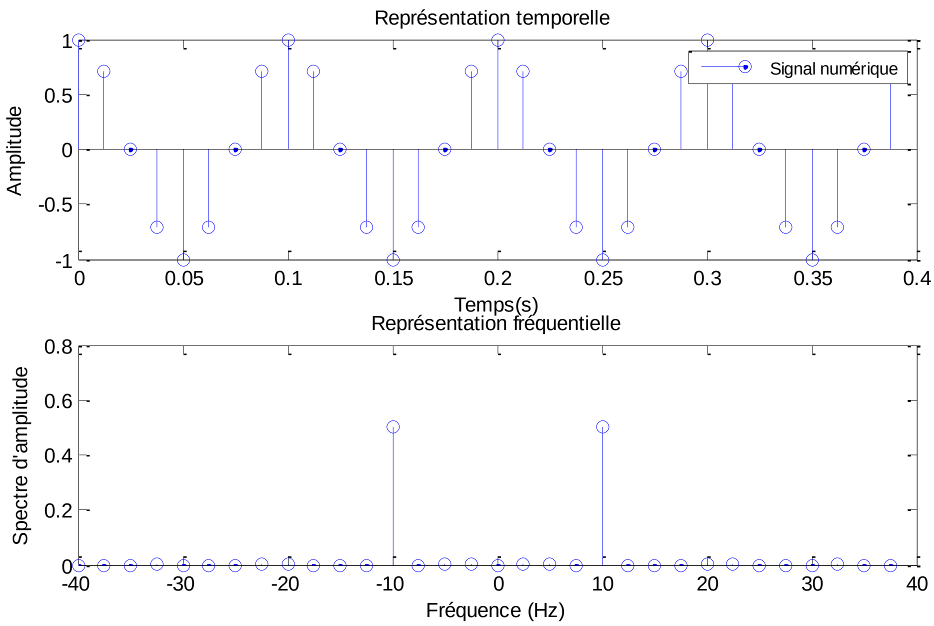

Figure 1 : Représentation temporelle et fréquentielle de signal

1.2. Calculer et représenter le spectre d'amplitude du signal x(t) sur 100 périodes. (redéfinir N=…).

1.3. Afin de chercher l'effet d'un nombre non entier de périodes, redéfinir l'axe temporel pour avoir uniquement 20 échantillons au lieu de 32 (donc 2.5 périodes). Recalculer et représenter le spectre d'amplitude du signal x(t).

1.4. Interpréter les résultats obtenus en 1.1), 1.2) et 1.3).

1.5. Refaire le même travail de la question 1.2, mais cette fois le signal est noyé dans un bruit aléatoire additif (bruit blanc de moyenne zéro, générer par la fonction rand de Python).

2. Les fenêtres de troncature

Remarques préliminaire : un signal ne peut être représenté complètement par N échantillons de sa TFD que si sa durée est limitée à N. on ne peut pas définir exactement la TFD d'un signal à durée illimitée. La TFD dans ce cas n'est définie qu'approximativement en limitant la durée du signal par un moyen approprié. On présente, dans cette partie, les principaux moyens de limiter la durée d'un signal et la qualité de l'approximation obtenue pour TFD.
Le but de cette partie est de comparer les performances de l'utilisation des fenêtres rectangulaires et de la fenêtre hamming.

2.1. Pour des longueurs de fenêtres de N_1=30 points et une séquence totale de N=512 points, générer et représenter sur un même graphe les fenêtres rectangulaire w_1 et hamming w_2.

- Générer et représenter le spectre d'amplitude (en échelle linéaire et en échelle logarithmique sur l'axe des ordonnées) de chacune de ces fenêtres. (La figure 2 donne le résultat que vous devez obtenir).

2.2. Soit un signal $x(t) = A1.\cos(2\pi.f_1 t) + A2.\cos(2\pi.f_2 t)$ avec (f_1= 13 Hz , f_2= 13.5 Hz et A1=A2=1) échantillonné à 256 Hz pendant 2 secondes. Les signaux $y_1(t)$ et $y_2(t)$ représentent respectivement la troncature (multiplication point par point) de signal x(t) par la fenêtre rectangulaire et la fenêtre hamming pour des longueurs de fenêtres sur 512 points.

- Générer et représenter sur un même graphe les signaux $y_1(t)$ et $y_2(t)$.
- Générer et représenter le spectre d'amplitude (en échelle logarithmique sur l'axe des ordonnées) des signaux $y_1(t)$ et $y_2(t)$. (La figure 3 donne le résultat que vous devez obtenir).
- Finaliser la comparaison en précisant la fenêtre la plus adaptée.

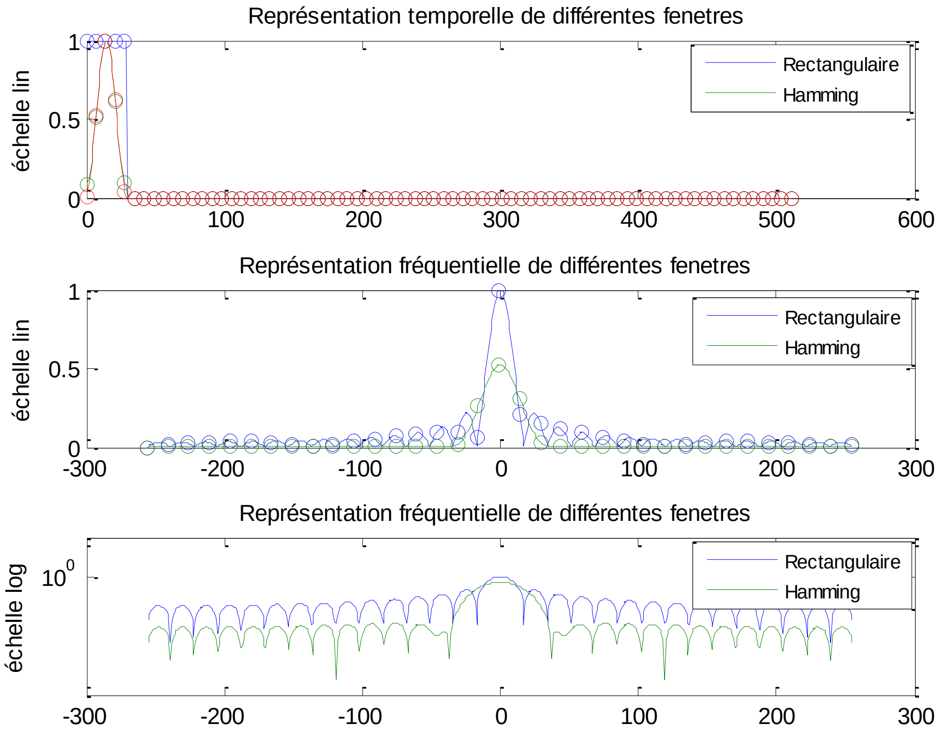

Figure 2 : Représentation temporelle et fréquentielle de fenêtres rectangulaire et hamming

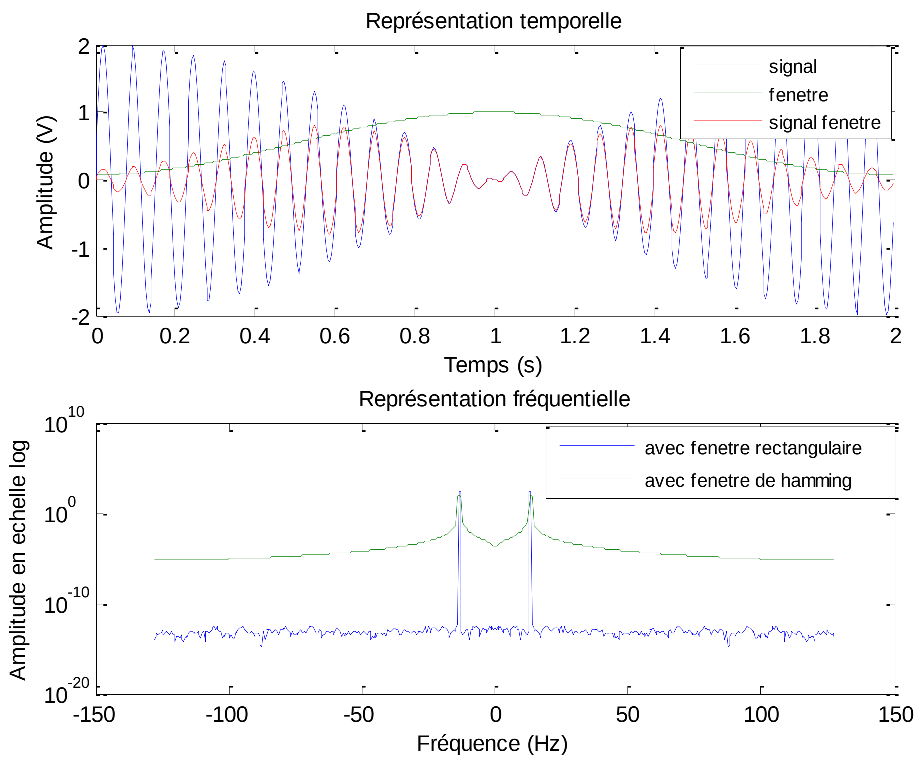

Figure 3 : Représentation temporelle et fréquentielle de la troncature du signal par la fenêtre rectangulaire et la fenêtre hamming