

Compte rendu du TP4 - Fourier

Louis Allain

October 14, 2019

1 Transformée de Fourier 2

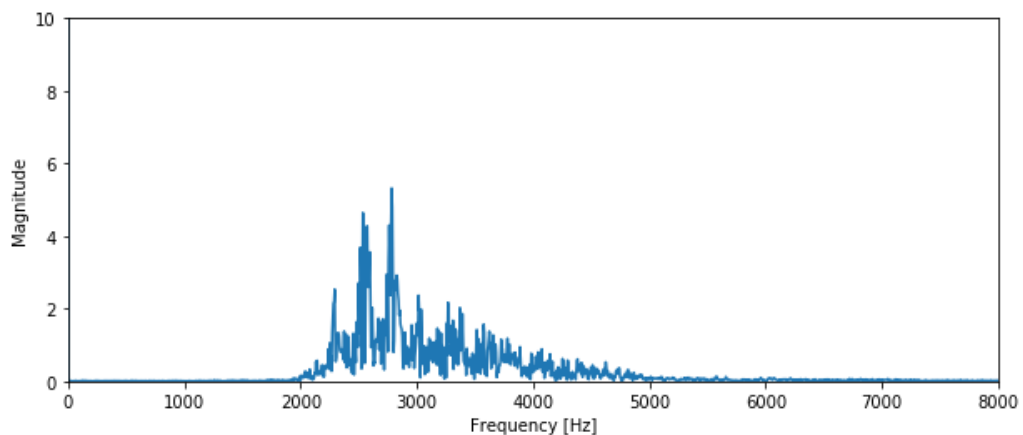
1.1 Transformée de Fourier appliquée à des signaux sonores réels

Réponses aux question :

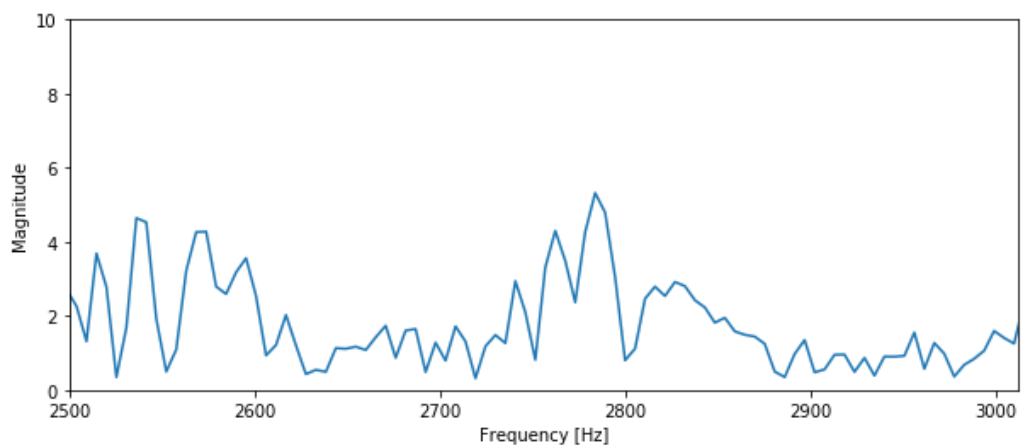
1. En ayant pris le son fourni de la corneille noire ”‘corneillenoire.wav’”, voici les résultats :

```
taille du fichier : 26067  
fréquence d'échantillonnage : 8000  
durée du signal : 3.258375 s
```

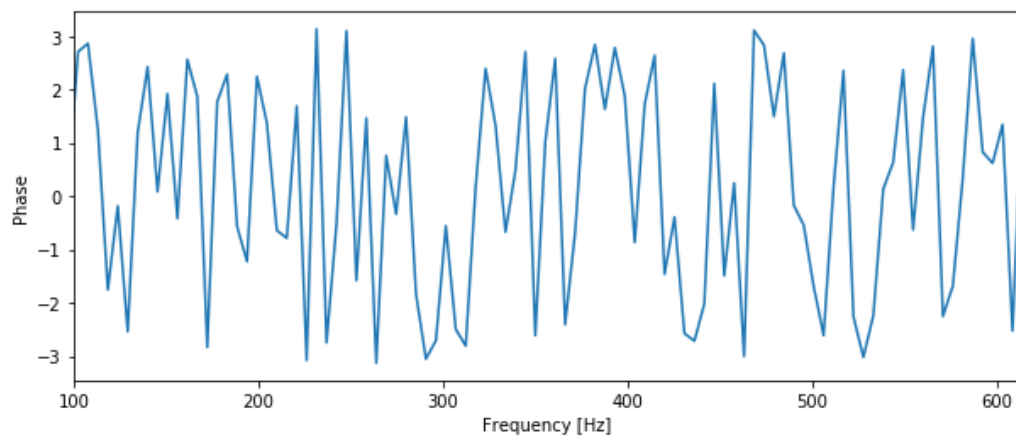
2. Voici un affichage en amplitude du spectre complet du signal :



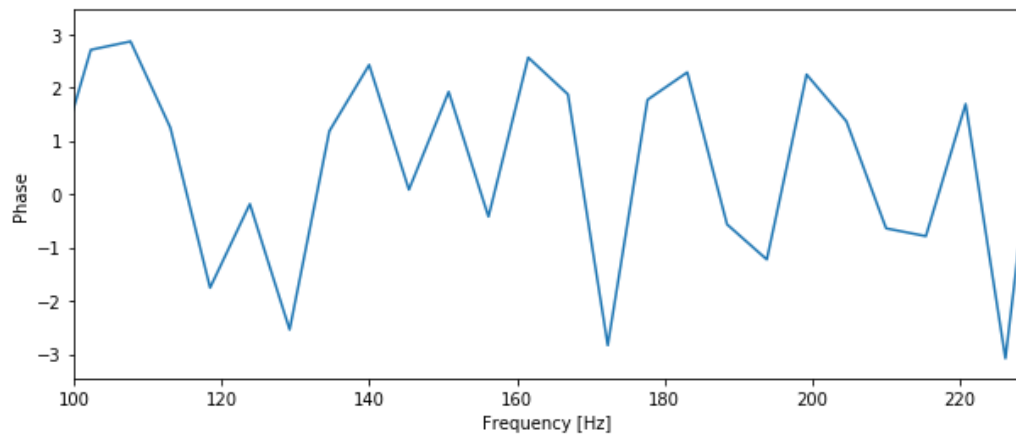
Voici un autre affichage en amplitude du spectre du signal avec un offset égal à 2500 et une fenêtre de 512 :



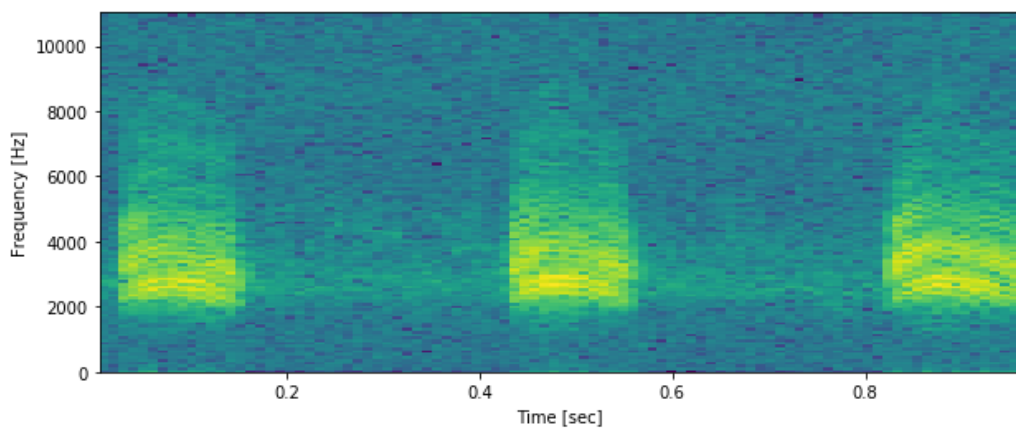
Voici un affichage de la phase du spectre avec un offset 100 et une fenêtre de 512 :



Voici un affichage de la phase du spectre avec un offset 100 et une fenêtre de 128 :



3. Traçage du spectrogramme du signal :



4. Voici la fonction `calculerSpectre` qui renvoie le vecteur fréquentiel et lecteur du spectre (en dB ou non) :

```
def calculerSpectre(echantillons, fs, dB=False):

    N = echantillons.size
    sf = np.zeros(N)
```

```

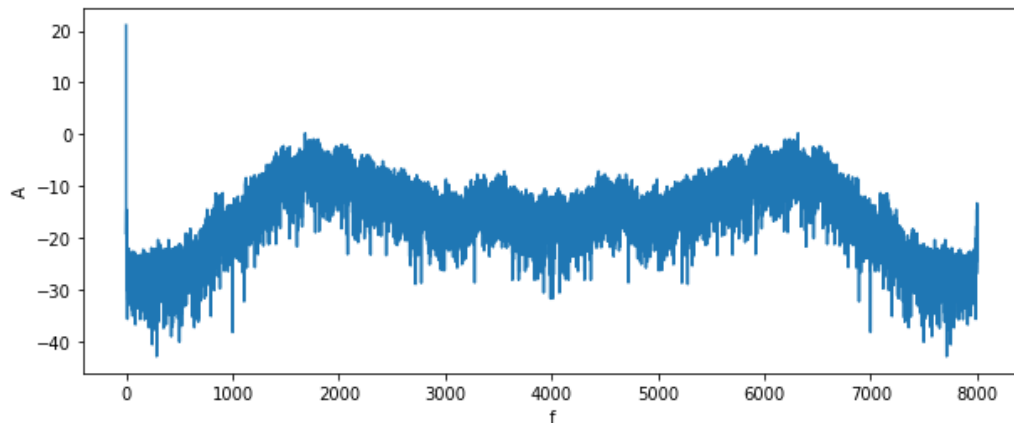
sf[:] = echantillons[0:N]
X = fft(sf)/N # tfd
F = np.linspace(0, fs, N)

if(dB):
X = 10 * np.log10(abs(X)) # dB

return (F, X)

```

De plus voici, par exemple le résultat de la fonction. On affiche ici le spectre en dB en fonction de la fréquence :



5. Voici la fonction movingFFT :

```

def movingFFT(filename, winsize, offset, wintype):
(fs, x) = read(filename)
(freq, spectre) = calculerSpectre(x, fs, False)
N = x.size
plt.figure(figsize=(10,4))
plt.xlim(offset, winsize+offset)
plt.ylim(-0.5, 0.5)
spectre = spectre*signal.get_window(wintype,N)
plt.plot(freq, spectre)

```

Et voici le résultat de cette fonction sur le fichier piebavarde.wav avec un offset de 2600 et une fenêtre de 256 :

