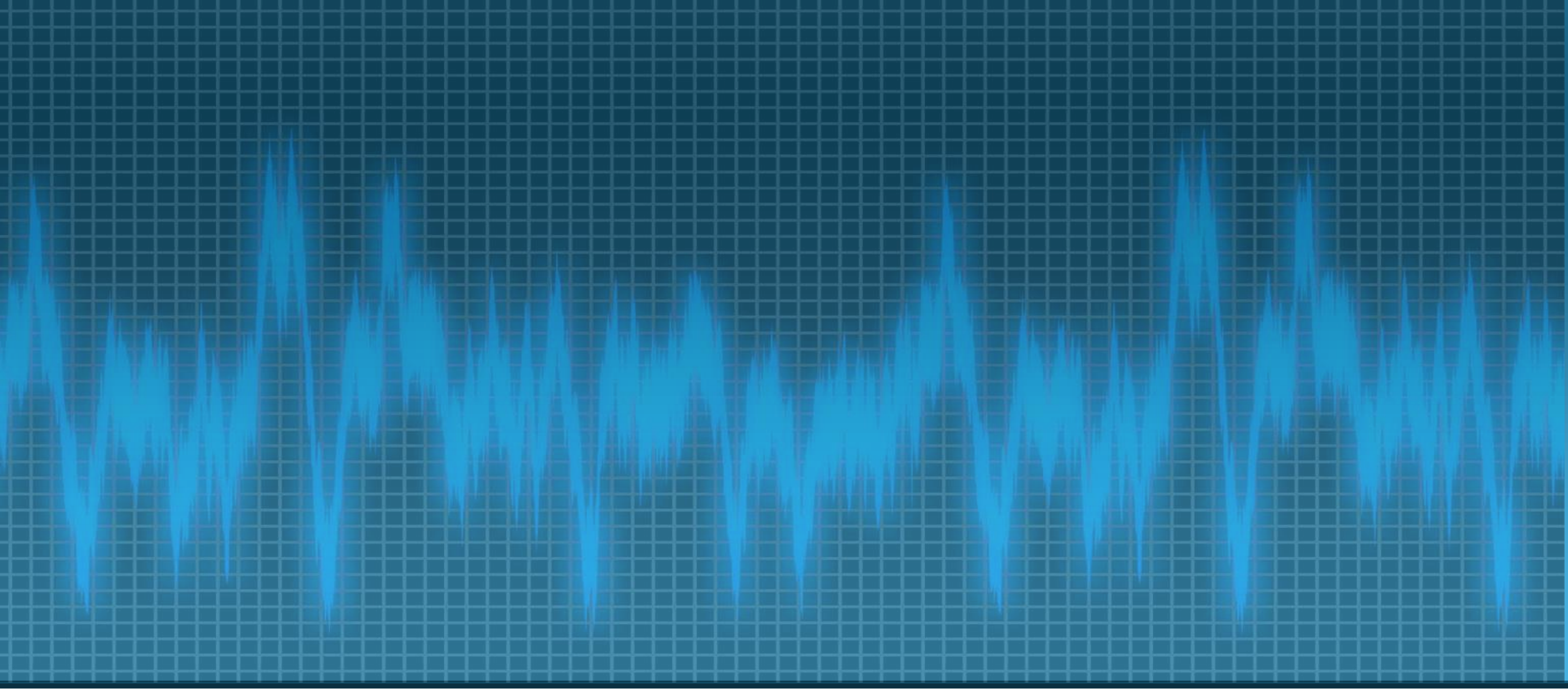


Compte Rendu TP1-part II

Filtrage numérique d'un signal d'entrée

Réalisé par : Hachem Squalli ElHoussaini N°29

Dirigé par : Pr. H. TOUZANI

Exercice :

I. Etude temporelle

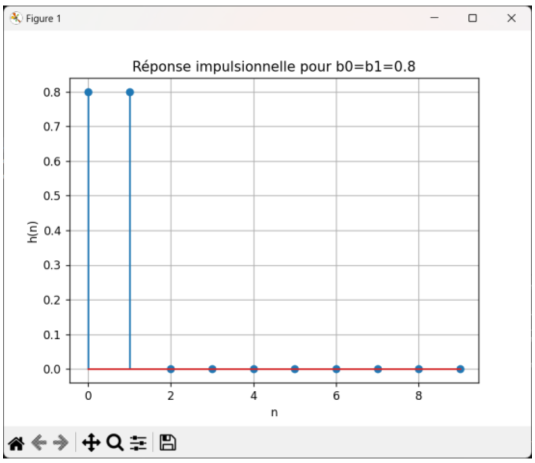
1. Calculez la réponse impulsionnelle (RI), sur le papier, en fonction de b0 et b1, en supposant le système causal, et les conditions initiales éventuelles nulles

La réponse impulsionnelle est donc :

$$h(n) = \begin{cases} b_0 & \text{pour } n = 0 \\ b_1 & \text{pour } n = 1 \\ 0 & \text{pour } n \geq 2 \end{cases}$$

2. En utilisant la fonction lfilter, calculer la Réponse Impulsionnelle du filtre, puis contrôlez graphiquement l'allure de la RI, avec b1= b2 = 0.8.

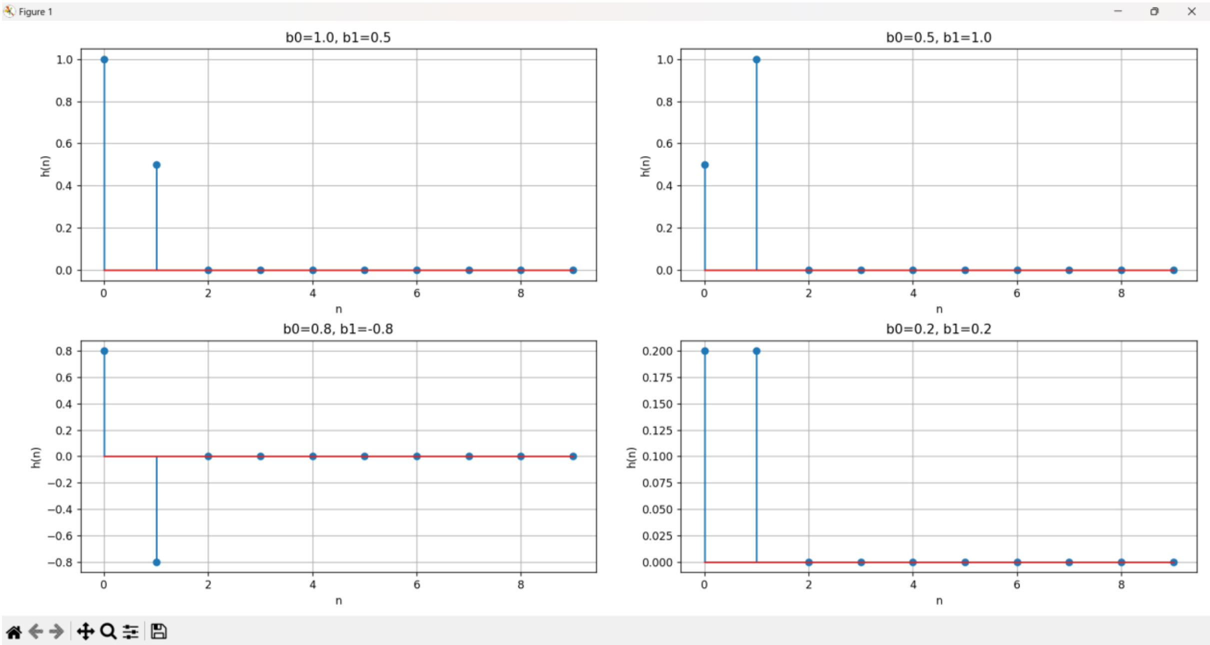
```
import numpy as np
import matplotlib.pyplot as plt
from scipy.signal import lfilter

b0, b1 = 0.8, 0.8
b = [b0, b1]
a = [1]

impulsion = np.zeros(10)
impulsion[0] = 1
h = lfilter(b, a, impulsion)

# Visualisation
plt.stem(h)
plt.title('Réponse impulsionnelle pour b0=b1=0.8')
plt.xlabel('n')
plt.ylabel('h(n)')
plt.grid()
plt.show()
```



3. Calculez et visualisez la réponse impulsionnelle pour différentes valeurs de b0 et b1. Conclusions.

```python
# Différentes combinaisons de coefficients
coeffs = [(1.0, 0.5), (0.5, 1.0), (0.8, -0.8), (0.2, 0.2)]

plt.figure(figsize=(12, 8))
for i, (b0, b1) in enumerate(coeffs, 1):
    b = [b0, b1]
    h = lfilter(b, a, impulsion)

    plt.subplot(2, 2, i)
    plt.stem(h)
    plt.title(f'b0={b0}, b1={b1}')
    plt.xlabel('n')
    plt.ylabel('h(n)')
    plt.grid()

plt.tight_layout()
plt.show()
```

Résultat



Conclusions :

- La réponse impulsionnelle est finie (FIR) et ne dure que 2 échantillons
- Les coefficients b0 et b1 déterminent directement l'amplitude des deux premiers échantillons
- Le signe de b1 influence la phase du filtre

I. Etude fréquentielle

1. Donnez l'expression de la fonction de transfert en z correspondant à cette équation aux différences :

$$H(z) = b_0 + b_1 z^{-1}$$

2. Donnez l'expression de la fonction de transfert H(f), puis de |H(f)| pour b1 et b2 quelconque .

- En substituant $z = e^{j2\pi fz} = e^{j2\pi f}$ (où f est la fréquence normalisée par Fe), on obtient :

$$H(f) = b_0 + b_1 e^{-j2\pi f}$$

- Le module de H(f) est :

$$|H(f)| = \sqrt{b_0^2 + b_1^2 + 2b_0 b_1 \cos(2\pi f)}$$

3. Préciser les amplitudes théoriques en f=0 et f=1/2

Pour f = 0 :

$$H(0) = b_0 + b_1$$
$$|H(0)| = |b_0 + b_1|$$

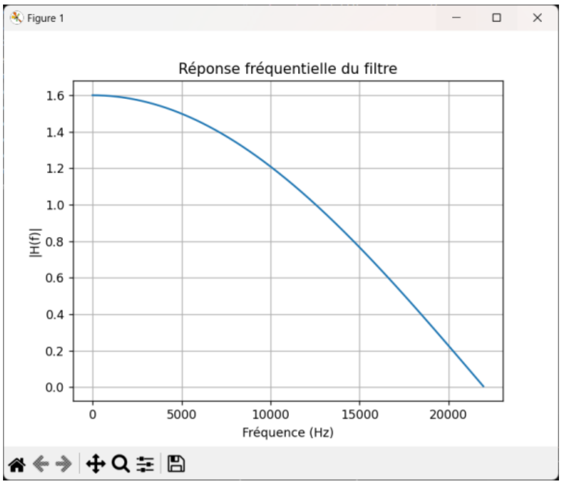
Pour f = 1/2 :

$$H\left(\frac{1}{2}\right) = b_0 + b_1 e^{-j\pi} = b_0 - b_1$$
$$\left|H\left(\frac{1}{2}\right)\right| = |b_0 - b_1|$$

4. Sous Python, calculez la TF du filtre en utilisant la TF (fonction fft) de la RI, visualisez les résultats. Conclusions.

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.fft import fft, fftfreq

b0, b1 = 0.8, 0.8
b = [b0, b1]
a = [1]


N_fft = 1024
h_padded = np.zeros(N_fft)
h_padded[:2] = [b0, b1]
H = fft(h_padded)
freqs = fftfreq(N_fft, d=1/44e3)


# Visualisation du module
plt.figure()
plt.plot(freqs[:N_fft//2], np.abs(H[:N_fft//2]))
plt.title('Réponse fréquentielle du filtre')
plt.xlabel('Fréquence (Hz)')
plt.ylabel('|H(f)|')
plt.grid()
plt.show()
```

Résultat :



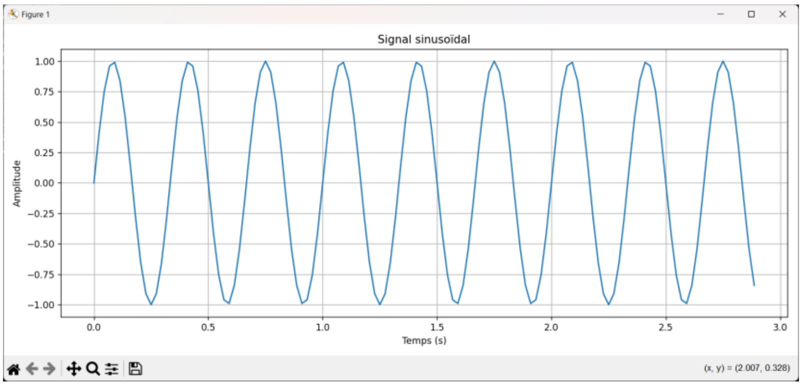
Conclusion :

- La réponse fréquentielle montre un comportement de filtre passe-bas ou passe-bande selon les valeurs de b0 et b1
- Les valeurs aux extrêmes (f=0 et f=Fe/2) correspondent aux calculs théoriques
-

II. Filtrage :

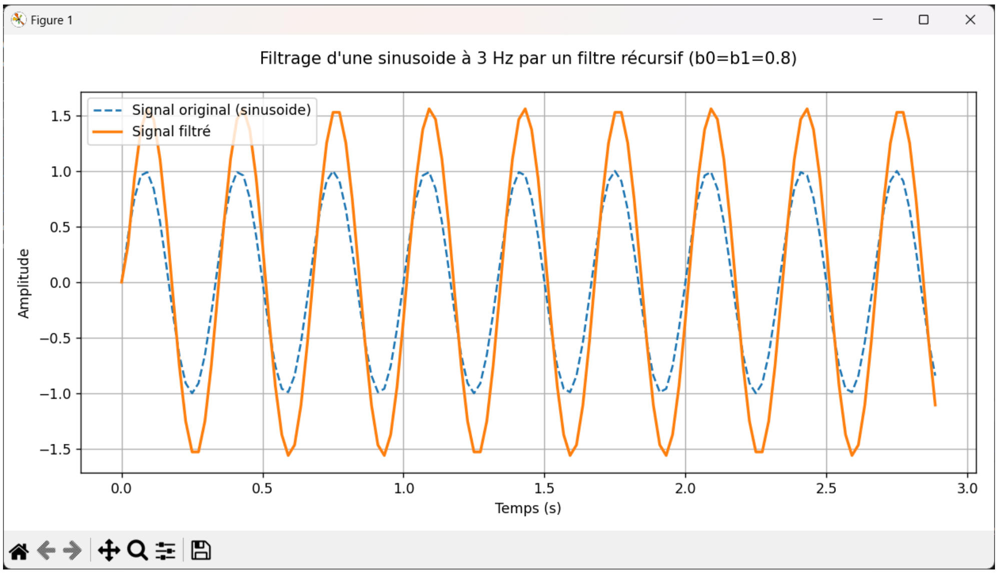
1. Créer une sinusoïde x, à la fréquence fo = 3, échantillonnée à Fe = 44, sur N = 128 points

```python
import numpy as np
import matplotlib.pyplot as plt

N = 128
Fe = 44
fo = 3

t = np.arange(N)/Fe
x = np.sin(2*np.pi*fo*t)

# Affichage
plt.figure(figsize=(12, 5))
plt.plot(t, x)
plt.xlabel('Temps (s)')
plt.ylabel('Amplitude')
plt.title('Signal sinusoïdal')
plt.grid(True)

plt.tight_layout()
plt.show()
```



2. Filtrez cette sinusoïde par le filtre précédent en utilisant la fonction lfilter, y1=lfilter(b,a,x)

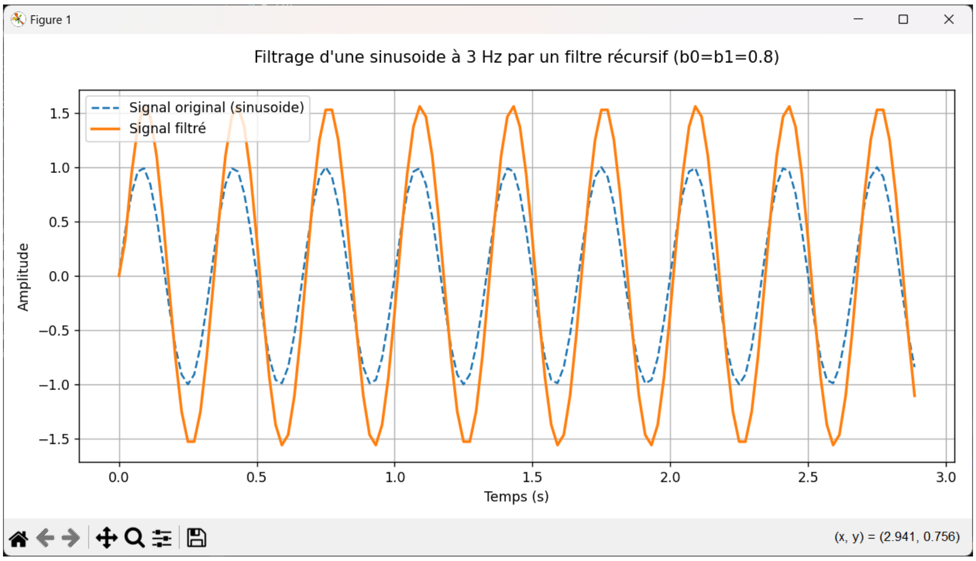
```python
import numpy as np
import matplotlib.pyplot as plt
import scipy

N = 128
Fe = 44

t = np.arange(N)/Fe
x = np.sin(2*np.pi*3*t)

#application de filtre récursif
b0, b1 = 0.8, 0.8
b = [b0, b1]
a = [1]  # Pas de terme récursif
y1 = scipy.signal.lfilter(b,a,x)

#affichage
plt.figure(figsize=(10,5))
plt.plot(t,x,label="Signal original (sinusoide)",linestyle="--")
plt.plot(t,y1,label="Signal filtré",linewidth=2)

plt.title(f"Filtrage d'une sinusoide à 3 Hz par un filtre récursif (b0=b1=0.8)", pad=20)
plt.xlabel('Temps (s)')
plt.ylabel('Amplitude')
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()
```



3. Filtrez cette sinusoïde par le filtre précédent :

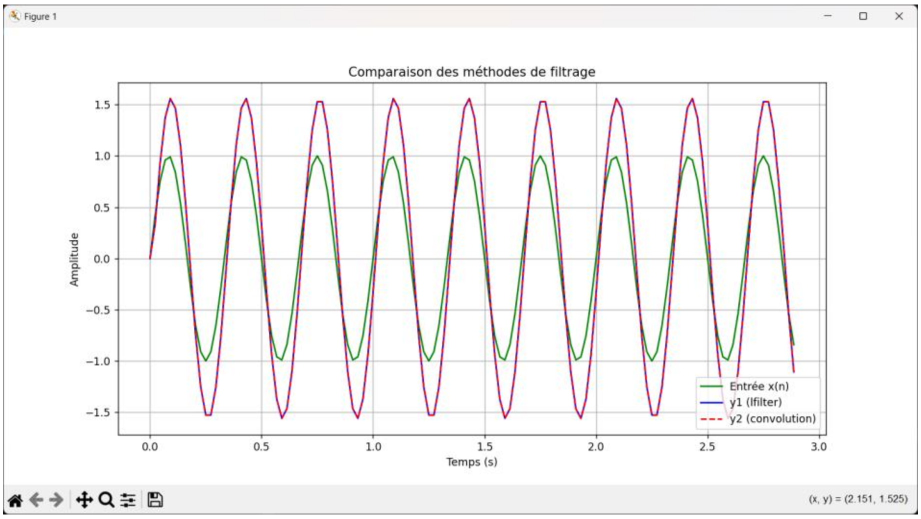
- en utilisant une convolution : y2=lfilter(h,[1],x)

```python
import numpy as np
import matplotlib.pyplot as plt
import scipy

N = 128
Fe = 44

b0, b1 = 0.8, 0.8
b = [b0, b1]
a = [1]
t = np.arange(N)/Fe
x = np.sin(2*np.pi*3*t)
h = np.array([b0, b1])

y2= scipy.signal.lfilter(h,[1],x)

#affichage
plt.figure(figsize=(10,5))
plt.plot(t,x,label="Signal original (sinusoide)",linestyle="--")
plt.plot(t,y2,label="Signal filtré",linewidth=2)

plt.title(f"Filtrage d'une sinusoide à 3 Hz par un filtre récursif (b0=b1=0.8)", pad=20)
plt.xlabel('Temps (s)')
plt.ylabel('Amplitude')
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()
```

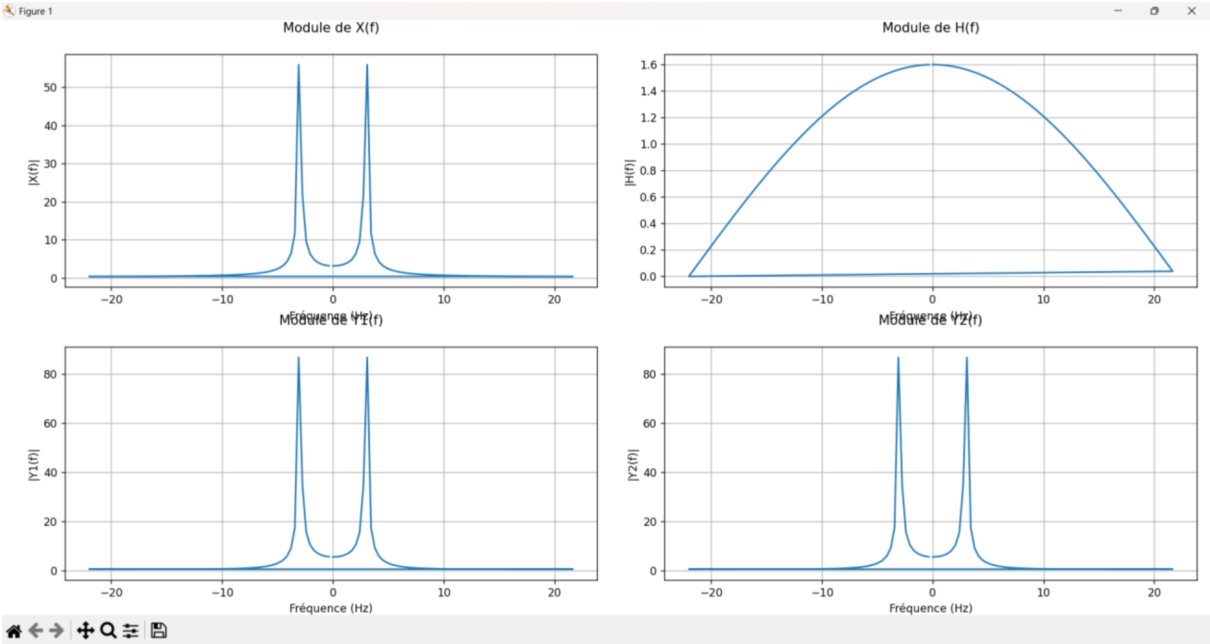


4. Comparez graphiquement ces deux résultats. Affichez les deux courbes, voire la différence des signaux entre y1 et y2

```python
#affichage
plt.figure(figsize=(12, 6))
plt.plot(t, x, 'g', label='Entrée x(n)')
plt.plot(t, y1, 'b', label='y1 (lfilter)')
plt.plot(t, y2, 'r--', label='y2 (convolution)')
plt.title('Comparaison des méthodes de filtrage')
plt.xlabel('Temps (s)')
plt.ylabel('Amplitude')
plt.legend()
plt.grid()
plt.show()
# Différence entre les deux méthodes
plt.figure()
plt.plot(t, y1-y2)
plt.title('Différence entre y1 et y2')
plt.xlabel('Temps (s)')
plt.ylabel('Amplitude')
plt.grid()
plt.show()
```



5. Calculez la TF : X(f) du signal x et la TF : H(f) de la réponse impulsionnelle h et des sorties y1 et y2. Visualisez ces deux résultats. Interpréter.



Interprétation :

- Les deux méthodes de filtrage (lfilter et convolution) donnent des résultats identiques (différence nulle)
- La TF du signal filtré montre l'effet du filtre sur la sinusoïde
- Le gain du filtre à la fréquence f0=3Hz peut être vérifié en comparant les amplitudes de X(f) et Y(f)