

Traitement du Signal

TP 1

Dănuț Ovidiu POP

Objectifs :

Représentation et applications des signaux de base et applications de la transformée de Fourier sous Python

Ex.

Créer un tracé simple.

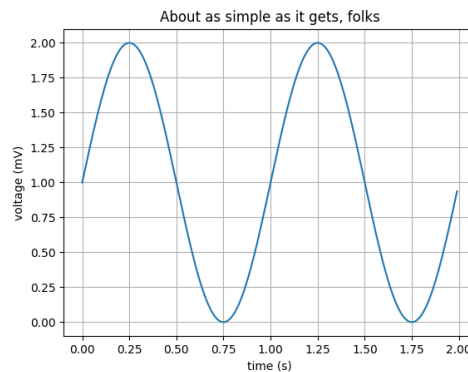
```
import matplotlib
import matplotlib.pyplot as plt
import numpy as np

# Data for plotting
t = np.arange(0.0, 2.0, 0.01)
s = 1 + np.sin(2 * np.pi * t)

fig, ax = plt.subplots()
ax.plot(t, s)

ax.set(xlabel='time (s)', ylabel='voltage (mV)',
       title='About as simple as it gets, folks')
ax.grid()

fig.savefig("test.png")
plt.show()
```



Créer et afficher les signaux suivants

1. Un signaux non stationnaires
2. Signal porte ou rectangle
3. Exponentielle décroissante
4. Un signal sur bruit
5. Sin, cos, sin+cos (utiliser les subplots)
6. le signal suivant $x(t) = e^{j100\pi t}$
7. Tracer le signal $x(t)$ entre -5 et 5 pour $a = 1$, avec un pas de temps $T_e = 0.01$ s.
$$x(t) = e^{-a|t|}$$
8. Calculer, de manière formelle, sa transformée de Fourier $X(f)$ et tracez la sur une autre figure entre -5 Hz et 5 Hz avec un pas de fréquence $F_e = 0.01$ Hz.
9. Tracer le module et la phase de la transformée de Fourier (fonctions abs et angle).
10. Pour approximer la Transformée de Fourier continue d'un signal $x(t)$, représenté suivant un pas T_e , on utilise la commande `fftshift`
11. La transformée de Fourier Inverse s'obtient par la commande `ifft`
12. Illustrer la propriété de décalage fréquentiel de la TF en représentant le module de la TF de $x(t) \cdot e^{j2\pi f_0 t}$ avec $f_0 = 5$ Hz
Représenter le module, la partie réelle et la partie imaginaire du signal temporel.

Commandes susceptibles de vous être utiles

`plot` permet de tracer une fonction
`xlabel` rajoute une légende à l'axe des abscisses
`ylabel` rajoute une légende à l'axe des ordonnées
`title` rajoute un titre à une figure
`axis` permet de modifier la valeur des axes
`fft` calcule une transformée de Fourier Rapide
`ifft` calcule une transformée de Fourier inverse
`linspace(a,b,n)` génère un vecteur de n valeurs équidistantes entre a et b
`abs` calcule une valeur absolue ou un module dans le cas complexe
`real` extrait la partie réelle d'un nombre complexe
`imag` extrait la partie imaginaire d'un nombre complexe

On va utiliser les modules Python: Numpy, SciPy, Matplotlib