TEK-UP
Ecole Supérieure Privée
Technologie & Ingénierie

Travaux Pratiques

« Traitement Numérique du Signal avec Python »

Niveau : 2e année Ingénieur Data Science

TP2 - Génération de signaux numériques

Année Universitaire : 2021/2022

Génération de signaux numériques

Un signal numérique est une suite de valeurs numériques :{x(k)}

- Réelles ou complexes
- Certaines ou aléatoires
- Finie ou infinie.

Notation : x(k) $k \in Z$, $\Delta t=1$ ou $x(k\Delta t)$.

La première notation est utilisée pour représenter un signal qui n'est pas forcément une fonction du temps. La deuxième notation implique une répartition périodique des valeurs selon la variable temps. Le terme Δt est appelée période d'échantillonnage ($Fe=1/\Delta t$).

Rq. : on peut passer de la deuxième notation à la première notation en posant $\Delta t=1$.

L'objectif de ce TP est de générer, de visualiser les signaux numériques élémentaires, faire des quelques opérations mathématiques sur ces signaux, de déterminer leur énergie et leur puissance en utilisant l'environnement de traitement numérique de signal sous Python.

Manipulation

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

1. Génération des signaux numériques élémentaires :

1.1.Soit le signal numérique impulsion unité défini par :

$$d(k) = \begin{cases} 1 & si \quad k = 0 \\ 0 & si \quad k \neq 0 \end{cases}$$

Le programme P1 peut être utilisé pour générer et tracer le signal d(k) dans le domaine temporel de k compris entre -5 et 15.

- Dans le domaine temporel de k compris entre -5 et 15, générer et tracer le signal impulsion d(k). (Utiliser la fonction 'np.zeros' de Python).

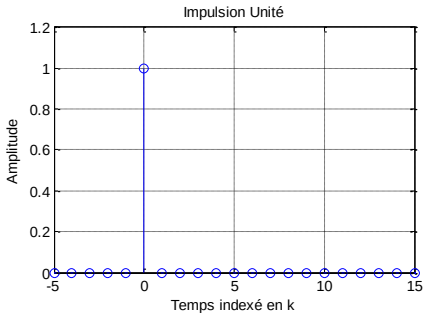

Figure 1 : représentation temporelle de l'Impulsion unité

- Application : Modifier le programme précédent pour générer et tracer les signaux suivants : d(k-5), d(k+3).

1.2.Soit le signal numérique Échelon unité, défini par :

$$u(k) = \begin{cases} 1 & si \quad k \geq 0 \\ 0 & si \quad k < 0 \end{cases}$$

- Dans le domaine temporel de k compris entre -5 et 15, générer et tracer le signal échlon u(k). (Utiliser la fonction 'np.ones' de Python).

- Application : Modifier le programme précédent pour générer et tracer les signaux suivants : u(k-5) et 2u(k+5)

1.3.Soit le signal numérique rectangulaire de largeur N, défini par :

$$rect_N(k) = \begin{cases} 1 & si \quad 0 \leq k \leq N-1 \\ 0 & si \quad ailleurs \end{cases}$$

- Dans le domaine temporel de k compris entre -5 et 15 générer et tracer le signal numérique $rect_{10}(k)$, donné par :

$$rect_{10}(k) = \begin{cases} 0 & si \quad -5 \leq n \leq -1 \\ 1 & si \quad 0 \leq n \leq 9 \\ 0 & si \quad 10 \leq n \leq 15 \end{cases}$$

1.4.Soit le signal numérique exponentiel à des valeurs réelles, défini par :

$$s_1(k) = \lambda\, a^k$$

- Générer et tracer le signal exponentiel $s_1(k)$ dans le domaine temporel de k compris entre 0 et 20; on prend : $\lambda = 1$ et $a = 0.5$
- Calculer l'énergie de ce signal

1.5.Le signal numérique sinus, défini par :

$$s_1(k) = A\sin\left(\frac{2\pi}{N}k - phase\right), \quad \text{N =20 étant la période du signal}$$

- Générer et tracer le signal exponentiel s(k) dans le domaine temporel de k compris entre 0 et 40; on prend : *amplitude* "$A=$ 1" et "$phase = 0$"

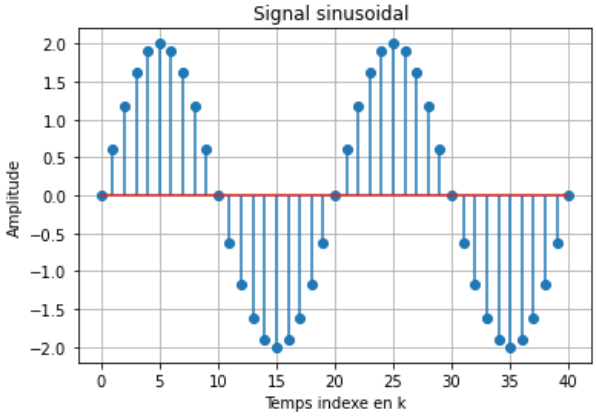

Figure 2: représentation temporelle de signal sinus

- Application :
  - Modifier le programme précédant pour générer et tracer un signal sinusoïdal $s_2(k)$ d'amplitude 2.5, de fréquence 0.05, et de phase décalée de 90degrées.
  - Tracer le signal $s_2(k)$ sur 5 périodes.

2. Génération des signaux numériques complexes :

Un signal aléatoire de longueur N avec une distribution uniforme dans l'intervalle (0,1) peut être généré par la commande de Python suivante :

x = np.random.rand (1,N);

Application :

- Générer et tracer un signal aléatoire de longueur 50 tels que ces éléments sont uniformément distribués dans l'intervalle [-3,3].

3. Génération des signaux numériques aléatoires

Des signaux complexes peuvent être générés en utilisant les opérations faites pour les signaux élémentaires.

Par exemple, un signal de modulation d'amplitude peut être généré par la modulation d'un signal sinusoïdal de haute fréquence $x_h(k) = \cos(\omega_h k)$ avec un signal sinusoïdal de basse fréquence $x_b(k) = \cos(\omega_b k)$. Le signal résultant $y(k)$ à la forme suivante : $y(k) = A(1 + m.x_b(k)).x_h(k) = A(1 + m.\cos(\omega_b k)).\cos(\omega_h k)$, avec m, appelé facteur de modulation, est choisi pour s'assurer que l'expression $1 + m.\cos(\omega_b k)$ est positive pour tous les valeurs de k.

- Le programme « P » est utilisé pour générer un signal modulé en amplitude.

```
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
# Signal de modulation d'amplitude

k=np.arange(150);
m=0.8; fh=0.1; fl=0.01;
xh=np.sin(2*np.pi*fh*k);
xl=np.sin(2*np.pi*fl*k);
y=(1+m*xl)*xh;
plt.figure()
plt.plot(k,y)
plt.grid; plt.ylabel('Amplitude');
plt.xlabel('temps index en k')
plt.title('Signal modulé en amplitude')
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
```