Échantillonnage – Notes de cours

1. Théorie de l'échantillonnage

1.1. Définition

Le **théorème de Nyquist-Shannon** est à la base de la conversion d'un signal d'une forme continue à une forme discrète. Il stipule que pour représenter correctement un signal numérisé, la fréquence d'échantillonnage d'un signal doit être égale ou supérieure au double de la fréquence maximale contenue dans ce signal.

L'échantillonnage d'un signal permet ainsi de transformer des grandeurs continues provenant de dispositifs analogiques (capteurs) en une séquence de valeurs discrètes. Lorsque l'on s'intéresse à des séries temporelles (telles que le son), on peut avoir des valeurs discrètes réparties uniformément dans le temps. Pour certaines séquences, les valeurs peuvent être séparées par des durées variables (par exemple les valeurs boursières au cours du temps).

1.2. Exemples

Certains exemples de signaux échantillonnés on été programmés en TP : impulsion, marche unitaire, rampe, exponentielle réelle et complexe, sinusoïde.

Exemple de la sinusoïde :

Considérons la sinusoïde suivante :
$s(t) = a.\sin(\omega_0 t) = a.\sin(2\pi f_0 t)$
où a est l'amplitude de la sinusoïde, ω_0 la pulsation, f_0 la fréquence du signal et t la variable temporelle.

Programme générant une sinusoïde :

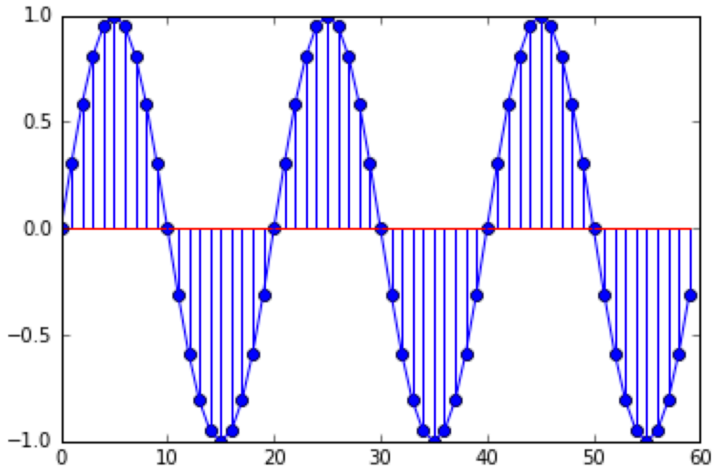
```
import numpy as np
def genSin (N,f,fs) :
        n = np.arange(N)                        # génère un vecteur de 0 à N-1
        t=np.linspace(0,(N-1)/float(fs),N)      # t = n / float(fs)  génère une échelle de
                                                # temps discrète
        pi = np.pi
        x=np.sin(2*pi*f*t)
        return(x)
```

Dans la fonction genSin, le vecteur t est le vecteur discret temporel
$[0, T_s, 2*T_s, \ldots (N-1) * T_s]$

Le vecteur x contient donc une séquence de N échantillons suivant la focntion sinusoïdale à la fréquence f.

On peut appeler la fonction genSin de la façon suivante :
s = genSin(60, 1, 20) pour générer une sinusoïde sur 3 périodes (60/20) à une fréquence de 1 Hz, avec une fréquence d'échantillonnage de 20 Hz.

La figure suivante montre le signal généré.



2. Opérateur dérivée sur des échantillons discrets

On considère la série temporelle x(t) échantillonnée de 0 à T s. Numériquement, nous enregistrons une séquence temporelle de N échantillons : x[n], avec n = 1 .. N
x[n] peut représenter une séquence temporelle de positions capturées.

Calcul numérique de la dérivée première :
On part de la formule de Taylor :
$x(t+h) = x(t) + h\, dx/dt + h^2/2\, d^2x/dt^2 + o(h^2)$
$x(t-h) = x(t) - h\, dx/dt + h^2/2\, d^2x/dt^2 + o(h^2)$

À l'ordre 1, on néglige le terme de la dérivée seconde :
En continu, on a : $v(t) = dx/dt = (x(t) – x(t-h))/h$
Numériquement, si on prend h = 1, on obtient :
$v(n) = x(n) – x(n-1)$ 1er ordre

À l'ordre 2, on garde le terme de la dérivée seconde :
$x(t+h) – x(t-h) = 2h\, dx/dt$
En continu, on a : $v(t) = dx/dt = (x(t+h) – x(t-h) /2h$
Numériquement, si on prend h = 1, on obtient :
$v(n) = x(n+1) – x(n-1) / 2$ 2nd ordre

Aux ordres supérieurs, on peut utiliser l'algorithme de Runge Kuta (cours en M2)

Exercice :
Utiliser le même principe pour déterminer l'algorithme de calcul de l'accélération (dérivée de la vitesse, ou dérivée seconde de la position).