

TP1 : Numérisation des données multimédia

Compte-rendu demandé :

Le compte-rendu du TP est constitué des réponses aux questions et du programme Python.

Ce TP a pour but de numériser des signaux analogiques et de manipuler des objets multimédias afin de faire le lien avec les notions que vous avez abordées en cours.

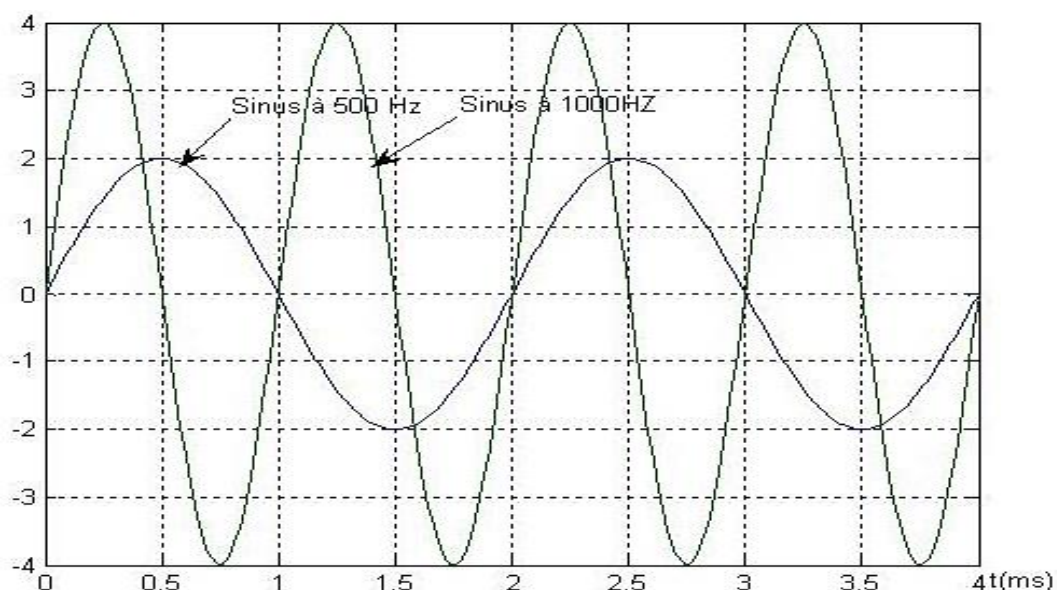
Exercice 1 :

Soit une fonction à une variable réelle $u(t)$. La variable t représente le temps (Ex : dans le cas d'audio) ou tout autre variable, par exemple une variable d'espace (Ex : dans le cas d'une image).

$$u_k = u(kT_e)$$

Où l'indice k est entier. La fréquence d'échantillonnage est $f_e = 1/T_e$.
Considérons la fonction $u(t) = 2\sin(2\pi ft)$ de fréquence $f = 0,2$ Hz.

1. Échantillonnez la fonction sur l'intervalle $[0, 0.1]$
2. Présentez graphiquement la fonction originale et la fonction échantillonnée.
3. Ecrivez le code python qui permet d'échantillonner les deux signaux ci-dessous :



Exercice 2 :

Nous allons utiliser la bibliothèque Pygame et ses différents modules afin de pouvoir jouer les sons dans le programme. Puisqu'on s'intéresse à des fichiers sonores, on va redéfinir les signaux Y1, et Y2 à l'aide d'attributs de formats sons typiques pour pouvoir les lire à l'aide de n'importe quel lecteur (on utilise ici le format "wave").

On utilise désormais une fréquence d'échantillonnage de 44 100 Hz pour définir Y1 et Y2. On choisit une durée des sons de 4 secondes.

```
# Définition des signaux Y1, Y2 et Y3
# Les sons sont échantillonnés à la fréquence 44100Hz

f = 200 # fréquence des signaux (200Hz)

temps_total = 4 # en seconde
fech = 44100 # nombres d'échantillons par seconde
dt = 1/fech # pas de temps en seconde

nb_points = int(temps_total/dt)
temps = np.linspace(0,nb_points,nb_points)*dt

phi = np.pi # valeur du déphasage

Y1 = np.sin(2*np.pi*f*temps)
Y2 = np.sin(2*np.pi*f*temps+phi)

Y3 = Y1+Y2
```

1. Vérifiez que le nombre de points dans le vecteur Y1 correspond bien au nombre d'échantillons présents dans un signal de 4 secondes à 44 110 Hz ,(La fonction *len* renseigne sur la taille (lenght) d'un vecteur).

On veut maintenant créer les fichiers sons à l'aide des vecteurs Y1, Y2 et Y3,

2. Ecrivez le code python qui nous permet d'obtenir les trois fichier sonores 'Son1', 'Son2' et 'Son3' sous format wave ('wav') à l'aide de la fonction `wave.open(...,'wb')` où 'wb' correspond à une écriture sous forme binaire (write binary) ; chaque signal doit être coder sur signal sur 8 bits = 1 octet.
3. Indiquez ensuite la fréquence d'échantillonnage (*frameRate*) et le nombre total d'échantillons (*nFrames*) dans chaque fichier son.
4. Ecrivez le code python qui permet la lecture des sons en utilisant l'outil "mixer" de pygame.