

TP 1 : échantillonnage et quantification de signaux analogiques

1- Introduction

L'objectif de ce TP est de vous familiariser avec les opérations de base sur les signaux et d'étudier les opérations de quantification et d'échantillonnage d'un signal analogique. L'étude de ces opérations est essentielle pour la compréhension de la conversion analogique-numérique utilisée pour la numérisation des signaux. L'échantillonnage et la quantification sont souvent utilisés de manière conjointe pour numériser un signal. Cependant, pour une meilleure compréhension des concepts, dans un premier temps, nous les étudierons de manière séparée avant de les utiliser simultanément dans la dernière partie.

Lors de ce TP, nous utiliserons le langage Python, que vous avez étudié lors du premier semestre.

2- Génération et représentation de signaux

Dans le but de réaliser des opérations sur des signaux, nous allons tout d'abord, générer des signaux types. Il existe plusieurs manières de représenter un signal. Pour ce TP, nous représenterons un signal sous la forme de deux listes : Une liste de nombres réels T qui représentera le temps et une liste de nombre réels X qui représentera l'amplitude du signal.

Le code Python suivant génère un signal sinusoïdal.

```
import numpy as np
```

```
F=2
```

```
T=np.linspace(0,1,1001)
```

```
X1=np.sin(F*np.pi*2*T)
```

1)- Expliquer ce que font les lignes de code de ce programme.

Dans le but de visualiser graphiquement ce signal, nous utiliserons la bibliothèque Matplotlib

2)- Représenter graphiquement ce signal et vérifier qu'il est conforme aux attentes.

3)- Générer un second signal X_2 de fréquence 5 Hz et d'amplitude 2 et représentez les deux signaux X_1 et X_2 sur une même figure avec des couleurs différentes.

4)- Produire un signal X_3 représentant la somme des signaux X_1 et X_2 et afficher son allure.

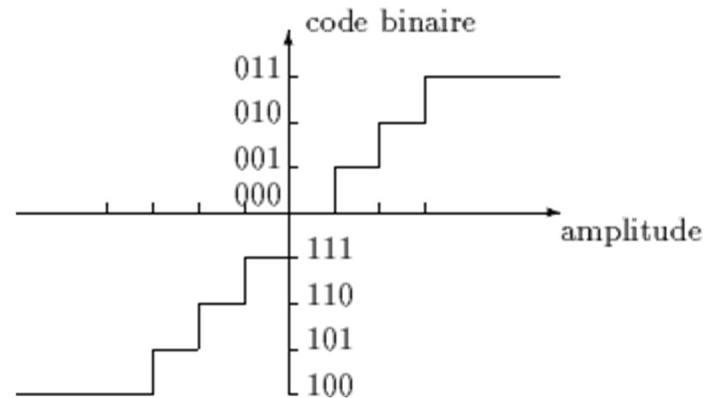
5)- afficher simultanément les trois signaux X_1 , X_2 et X_3 et vérifier sur quelques points de ces courbes que les amplitudes des signaux sont conformes aux attentes.

6)- Les signaux réels sont souvent bruités. Reprendre le signal X_1 et ajouter différents niveaux de bruit puis observer le signal obtenu. Utiliser pour cela la bibliothèque random.

7)- Tester d'autres types de signaux en utilisant la librairie mathématique (cos, abs, etc.).

3- Quantification des signaux

Afin d'enregistrer des signaux de manière numérique, il est nécessaire qu'ils soient représentés par un nombre fini de données binaires. Un convertisseur analogique numérique possède une résolution de conversion finie. La figure ci-dessous illustre la caractéristique de quantification d'un convertisseur à 3 bits.



8)- Réutiliser le signal X1 (d'amplitude 1 et de fréquence 2 Hz) et obtenir un signal quantifié X1q en utilisant le code suivant :

```
q=0.5 # pas de quantification
```

```
X1q = q*np.round(X1/q) # signal quantifié
```

Représenter graphiquement les signaux X1 et X1q sur la même figure. Commenter.

9)- La quantification d'un signal s'accompagne nécessairement d'une perte d'information. En effet, le signal quantifié est différent du signal d'origine. Représenter la différence entre le signal X1 et le signal X1q que l'on appellera l'erreur de quantification $E=X1-X1q$ ainsi que les deux signaux X1 et X1q. Commenter.

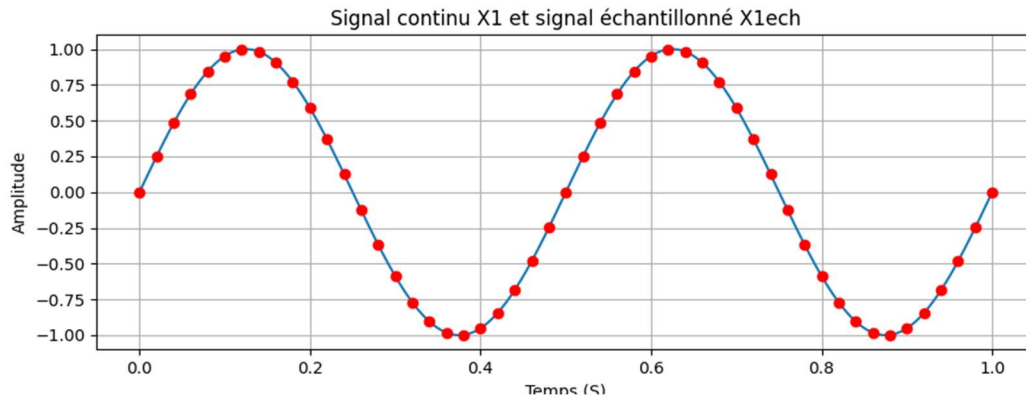
10)- Tester différentes valeurs de pas de quantification allant de $q=0.1$ à $q=1$. Que constatez-vous concernant l'erreur de quantification ? Commenter.

11)- Appliquer l'opération de quantification sur divers autres signaux (formes et fréquences différentes, signaux bruités, etc.). Commenter.

4- Echantillonnage des signaux

L'échantillonnage consiste à relever l'intensité du signal à des instants réguliers séparés par une durée appelée « période d'échantillonnage » (voir TD).

12)- écrire un programme en python permettant de produire un signal échantillonné X1ech à partir du signal X1. On choisira une période d'échantillonnage $T_e=20$ ms. Représenter le signal X1 et le signal X1ech sur un même graphique. Vérifier que les résultats obtenus sont similaires à ceux représentés sur la figure ci-dessous.



13)- Afficher uniquement le signal échantillonné pour différentes périodes d'échantillonnage (10 ms, 20 ms, 50 ms et 100 ms). Que constatez-vous ? Est-il possible de deviner mentalement le signal d'origine pour une période d'échantillonnage de 100 ms ?

14)- D'après le théorème d'échantillonnage de Shannon, quelle est la limite maximale de la période d'échantillonnage nécessaire à la reconstruction du signal pour notre signal sinusoïdal X1 de fréquence 2 Hz ?

15)- Représenter le signal échantillonné pour cette valeur maximale. Commenter.

16)- Utiliser à présent le signal X3. Ecrire un programme en python permettant de produire un signal échantillonné X3ech à partir du signal X3. On choisira une période d'échantillonnage $T_e=20$ ms. Représenter le signal X3 et le signal X3ech sur un même graphique.

17)- D'après le théorème d'échantillonnage de Shannon, quelle est la limite maximale de la période d'échantillonnage nécessaire à la reconstruction du signal pour notre signal sinusoïdal X3 ?

18)- Représenter le signal échantillonné pour cette valeur maximale. Commenter.

5- Echantillonnage et quantification des signaux

La numérisation d'un signal analogique à l'aide d'un convertisseur analogique-numérique utilise les deux opérations vues précédemment : l'échantillonnage pour prélever un nombre fini d'échantillons et la quantification pour représenter le signal sous forme numérique sur un nombre limité de bits dans la mémoire d'un ordinateur.

19)- A partir du signal X1, effectuer une opération d'échantillonnage avec une période $T_e=20$ ms suivie d'une opération de quantification avec un pas de quantification de 0.2. Tracer sur une figure unique, les signaux X1, X1ech et X1echq.

20)- Représenter la différence entre le signal X1ech et le signal X1echq que l'on appellera l'erreur de quantification $E=X1ech-X1echq$ ainsi que les deux signaux X1ech et X1echq. Commenter.

21)- Etudier l'effet du choix de la période d'échantillonnage et du pas de quantification sur l'erreur (pour cela, modifier T_e et étudier son influence, puis modifier q et étudier son influence).

22)- En partant du signal échantillonné et quantifié obtenu à la question 19), écrire un programme qui donne les valeurs numériques enregistrées en mémoire de l'ordinateur.