Introduction aux Transmissions et aux Communications GCR 1	TP1 : Numérisation d'un signal analogique	2023/2024

I. Echantillonnage

Soit le signal analogique :

$$x(t) = sin(2\pi f_0 t + \varphi)$$

On se propose d'échantillonner $x(t)$ à la fréquence d'échantillonnage f_e pour obtenir le signal à temps discret suivant :

$$x(t) = sin\left(2\pi \frac{f_0}{f_e} n + \varphi\right)$$

1. Déterminer la fréquence d'échantillonnage f_e qui vérifie le théorème de Shannon.

2. A quoi sert le théorème de Shannon.

3. Expliquer le phénomène de recouvrement ''aliasing''

4. Créer une série contenant des valeurs temporelles sur une durée de 10 ms à une fréquence d'échantillonnage de 10 kHz.

5. Créer une série contenant les valeurs du signal $x(t)$ (fréquence de 440 Hz) correspondant aux valeurs temporelles de la question I.1.4.

6. Le signal $x(t)$ est généré par le code suivant: $t = [0 : 0.01 : 10]$; x=sin(2* pi*5*t) .

a. Quelle est la fréquence maximale du signal analogique ?

b. Quelle est la fréquence d'échantillonnage ?

c. Le théorème d'échantillonnage est-il respecté ? Quelle conclusion en tirez-vous ?

d. Même question que la précédente si on augmente la fréquence du signal analogique à 30 Hz à 80 Hz.

Travail à réaliser :

Partie 1 :

On souhaite générer et afficher le signal sinusoïdal (1) avec f_0 =800 Hz , $f_e = 8\,KHz$ et φ arbitraire.

1. Générer un "vecteur de temps" v_t, contenant les instants des échantillons de ce signal avec une période d'échantillonnage (T_e= 1/ f_e), v_t=(0: $N-1$) $*$ T_e; on prendra N = 11.
2. Générer le signal à l'aide de la commande suivante : $s = A * sin(2 * p_i * f * v_t)$;
3. Tracer la sinusoïde obtenue en utilisant la fonction MATLAB stem.
4. Tracer la sinusoïde obtenue en utilisant la fonction MATLAB plot.
5. Quel est le rôle de la fonction plot par rapport à la fonction stem ?

for i = 1 : Ne
ff(i) = n. fe / Ne

Partie 2 :

Générer 0.8 secondes du signal sinusoïdal (1) de fréquence f_0 =100 Hz échantillonné à f_e=1000 Hz.

1. Combien y a t'il d'échantillons (N_e) ? $N_e = \frac{0,8}{Te} = 0,8 \times Te = 800$
2. Générer le vecteur des fréquences freq = n. f_e / N_e, avec n variant de 0 à N_e-1.
3. Représenter le module de la TFD en fonction du vecteur des fréquences *freq*.
4. Vérifier qu'il y a bien une raie à la fréquence du signal.

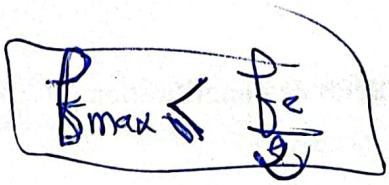
Partie 3:

A partir d'un signal sinusoïdal analogique (1), de fréquence $f_0 = 1000$ Hz, créer trois signaux sinusoïdaux échantillonnés respectivement

a) à 20 000 Hz ,

b) à 5 000 Hz

c) à 1 500 Hz.



Représenter les différents signaux sur le même graphe. Comparer les représentations fréquentielles de ces trois signaux. Interprétez les résultats obtenus.

II. Quantification

La quantification est l'opération qui affecte une valeur numérique codée en binaire à un échantillon donné. La valeur numérique qui est affectée à l'échantillon est choisie parmi un ensemble de valeur possibles, dont le nombre dépend de la résolution (ou nombre de bits) du convertisseur.

Résolution du convertisseur : N bits => 2^N valeurs possibles.

Les valeurs possibles pour les échantillons sont les *niveaux de quantification*. L'intervalle entre deux niveaux de décision est le *pas de quantification*.

Un codage sur N bits permet d'utiliser 2 N niveaux de décision, il y a donc 2 N-1 intervalles

Partie 1 : Quantification d'un signal sinusoidal

1. Créer un vecteur de 1000 éléments, représentant le temps. Nous choisissons une période unité: t = linspace(0,1,1000) ;
2. Créer ensuite un vecteur représentant un signal sinusoïdal, d'amplitude et de fréquence unité : s = sin(2*pi*t) ;
3. Créer maintenant une fonction réalisant la quantification :

```
function y = numerise(s,N)
q = 2/(2^N);
y = q * floor((s*(1-q)/q) + 0.5) ;
```

4. Cette fonction renvoie dans le vecteur y les échantillons du vecteur s quantifiés sur N bits. Observez le signal quantifié y pour N valant successivement 16, 4, et 2 bits. Que remarquez-vous ?

Partie 2 : Bruit de quantification

Pour caractériser l'erreur introduite par la quantification, nous allons utiliser un modèle de bruit additif. Avec ce modèle, on considère que le signal quantifié est le signal original auquel le processus de quantification a rajouté un bruit.

Observez le bruit de quantification pour N valant successivement 16, 8, 4 et 2 bits. Que remarquez-vous ?