
TD1 : Signal numérique

Le but de ce projet consiste à programmer les différentes étapes du processus de numérisation d'un signal, puis d'observer et de mesurer les erreurs engendrées par ce processus.

Ainsi, la première partie s'intéresse à l'étape d'**échantillonnage**, alors que la deuxième partie traite du processus de **quantification**.

1 Echantillonnage

Dans cette première partie, vous allez vous familiariser avec le logiciel matlab (ou scilab, ou octave), tout en vous intéressant à l'échantillonnage d'un signal.

Exercice 1 : Synthèse de signaux numériques

Ecrire deux fonctions de synthèse numérique pour les signaux sinusoïdaux et pour les signaux créneaux. La fréquence d'échantillonnage doit être un des paramètres de chaque fonction.

Exercice 2 : Echantillonnage

1. Générer un signal de 0.5 s composé de la somme de deux sinusoïdes d'amplitude 1 V, échantillonnées à 256 Hz, et qui battent respectivement aux fréquences de 100 et 156 Hz. Afficher le signal obtenu. Expliquer le phénomène observé en affichant les deux signaux sinusoïdaux séparément. Proposer si besoin une solution.
2. Générer et représenter un signal sinusoïdal d'amplitude 1 V qui bat à la fréquence de 356Hz, et échantillonnée à 256 Hz. Comparer ce signal au signal sinusoïdal de fréquence 100 Hz, échantillonné à la fréquence 256 Hz. Expliquer.

Exercice 3 : Modification de la fréquence d'échantillonnage

Pour écouter un signal, vous pouvez utiliser la commande `playsnd`. Pour charger un signal sonore au format wav, utiliser la commande `wavread` :

```
[s,Fe,bits]=wavread('file.wav');  
// jouer le son s1  
soundsc(s,Fe,bits)
```

Nous souhaitons modifier la fréquence d'échantillonnage d'un signal sonore (pour faire simple, la diviser par 2). Nous proposons deux approches : soit la fréquence d'échantillonnage à la lecture est modifiée (lors de l'appel à `soundsc`), soit les valeurs des échantillons sont modifiées (modification de `s` avant l'appel à `soundsc`).

1. Tester les deux approches.
2. Comparer et expliquer les deux résultats obtenus.
3. Je souhaite modifier la fréquence d'échantillonnage d'un signal numérique pour passer de 44100 Hz à 8000 Hz. Cette modification est-elle triviale ? Proposer **rapidement** une ou plusieurs solutions, sans trop détailler et sans les implémenter. Tester la fonction `resample`.

2 Quantification

Dans cette deuxième partie, nous nous intéressons à la quantification. Vous allez observer et mesurer les erreurs induites par cette étape. Nous supposons l'amplitude du signal comprise entre -1 et 1 .

Exercice 4 : Quantification d'un signal - mesure du SNR

Nous considérons ici un signal sinusoïdal, échantillonné à une fréquence F_e Hz. Les échantillons obtenus sont quantifiés linéairement sur N bits.

1. Représenter l'erreur obtenue par cette opération de quantification.
2. Etablir expérimentalement le lien entre le rapport signal sur bruit (SNR) et le nombre de bits N . Comparer avec la théorie vue en TD.

Exercice 5 : Quantification non linéaire

La voix humaine possède une dynamique très large mais l'information compréhensible se trouve dans les sons de bas niveau. Il est donc inapproprié de quantifier un tel signal selon une loi linéaire, puisque les échantillons de forts niveaux comporteront moins d'information que ceux de bas niveau. C'est la raison qui justifie la quantification à pas logarithmique d'un signal vocal.

Une quantification logarithmique peut s'interpréter par trois étapes successives : une étape de compression (passage de l'échelle linéaire à l'échelle logarithmique), une étape de quantification uniforme, et une étape d'expansion (inverse de l'étape de compression).

La fonction de compression pour une quantification logarithmique de type μ -law est donnée par :

$$C(x) = \text{sign}(x) \frac{\log(1 + \mu|x|)}{\log(1 + \mu)} \quad (1)$$

où μ prend des valeurs supérieures à 0. La valeur typique est $\mu = 255$.

Vous allez charger en mémoire un fichier son `Toms_diner.wav`.

Ensuite, vous devez créer et comparer deux signaux numériques obtenus par :

1. quantification linéaire des échantillons sur 8 bits
2. quantification logarithmique des échantillons sur 8 bits (μ -law)

Vous tenterez de répondre notamment aux questions suivantes :

- Lorsque μ vaut 0, que vaut $C(x)$?
- Lorsque μ est très grand, quelle est la forme d'un signal sinusoïdal compressé ?
- Quelle est la fonction $E(x)$ réciproque de la fonction $C(x)$?

Pour sauvegarder un signal, vous pouvez utiliser la commande `auwrite` :

```
auwrite(s1, Fe, N, 'linear', 'toms_lin.au');
auwrite(s2, Fe, N, 'mu', 'toms_mu.au');
%% charger
[x1,Fe,N]=auread('toms_lin.au');
[x2,Fe,N]=auread('toms_mu.au');
%% écouter les sons x1 et x2
soundsc(x1,Fe,N)
soundsc(x2,Fe,N)
```