

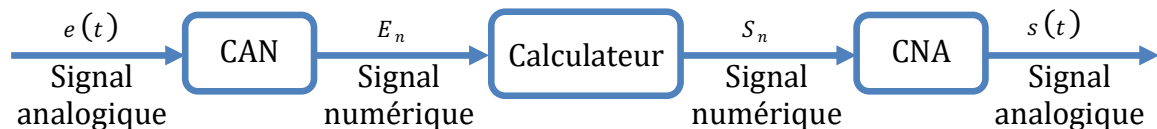
FILTRAGE NUMERIQUE

Extraits du programme

Filtrage numérique.	Mettre en œuvre une chaîne d'acquisition et de conversion. <u>Capacité numérique</u> : réaliser, à l'aide d'un langage de programmation, un filtrage numérique d'un signal issu d'une acquisition, et mettre en évidence la limitation introduite par l'échantillonnage.
---------------------	---

Un filtre numérique opère sur des signaux numériques en leur appliquant un algorithme de calcul qui dépend du traitement recherché.

Un convertisseur analogique-numérique (CAN) doit d'abord convertir un signal analogique en signal numérique, qui est traité numériquement, puis converti en signal analogique par un convertisseur numérique-analogique (CNA).



Le signal analogique sera chanté ou joué, enregistré et numérisé par le logiciel Audacity, filtré grâce à Python, et éventuellement reconverti en signal analogique pour être écouté sur les HP de votre ordinateur.

L'objectif de ce travail est de filtrer un signal numérique et de mettre en évidence la limitation introduite par l'échantillonnage.

1. Préparation

On réalise le filtrage d'un signal $e(t)$ analogique à l'aide d'un filtre passe bas du premier ordre de fonction de transfert :

$$\underline{H}(j\omega) = \frac{1}{1 + j\omega/\omega_c}$$

- Que représente la pulsation ω_c ?
- Déterminer l'équation différentielle qui relie la tension de sortie du filtre $s(t)$ à la tension d'entrée $e(t)$.

Pour réaliser des signaux numériques, $e(t)$ et $s(t)$ sont échantillonnés pendant une durée Δt à une fréquence d'échantillonnage $f_e = 1/T_e$ pour fournir des tableaux à N éléments $e[k]$ et $s[k]$ avec $0 \leq k \leq N-1$.

- En utilisant une méthode d'Euler explicite déterminer $s[k + 1]$ en fonction de $s[k]$, $e[k]$, T_e et f_c .

2. Matériel

Un ordinateur avec Python et un son disponible sur le répertoire de classe.

3. Manipulations

3.1. Fichier son

Charger un fichier son et repérer dans les caractéristiques du son la fréquence d'échantillonnage.
Exporter le son dans un répertoire personnel.

3.2. Extraction du signal échantillonné

Ouvrir Pyzo (ou Spyder) et charger le fichier TPFiltrage_Eleves.py.

Sauvegarder ce fichier dans le même répertoire que le fichier wav précédent.

La commande

```
fe,signal_entree = wav.read("nom_du_fichier.wav")
```

extrait la fréquence d'échantillonnage f_e et un tableau des valeurs du signal échantillonné.

Compléter les trois lignes permettant de générer un tableau des instants d'échantillonnage, nommé temps.

Ce tableau permet de tracer les signaux d'entrée et de sortie en fonction du temps.

Le spectre du signal d'entrée est calculé comme dans le TP Echantillonnage.

Exécuter le script et noter les principales fréquences du signal.

3.3. Filtrage

La partie filtrage est à écrire.

Définir une fréquence de coupure f_c pour le filtre passe-bas, à peu près au milieu du spectre.

Définir un tableau de Ne zéros appelé signal_sortie.

Ecrire la boucle permettant le calcul du signal de sortie, après passage dans le filtre, en utilisant la méthode d'Euler vue en préparation.

Ecrire les lignes permettant le calcul du spectre du signal de sortie.

Tracer le signal de sortie et son spectre.

Sauvegarder le signal de sortie sous format .wav afin de pouvoir l'écouter sur Audacity.

Observer les spectres des signaux et interpréter.

On peut éventuellement modifier la fréquence de coupure.

4. Influence de la fréquence d'échantillonnage

On souhaite diviser la fréquence d'échantillonnage par quatre : cela revient à prendre dans la liste Signal_entrée un échantillon sur quatre, grâce à l'instruction :

```
Nouvelle_liste=Ancienne_liste[::4]
```

A partir du tableau signal_entree, générer un second signal nommé par exemple signal_entree2 qui ne contient plus qu'un échantillon sur quatre.

Modifier en conséquence le tableau temps.

Filtrer le signal, et écouter le signal obtenu. Commenter.

Reprendre en divisant la fréquence d'échantillonnage par 8, soit $f_e = 5512,5$ Hz. Que constatez-vous ?

Expliquer la dégradation du signal.

5. Etude de filtres d'ordre deux

Si le temps le permet, programmer un filtrage numérique passe-bas du second ordre avec :

- Un facteur de qualité Q à définir ;
- Une fréquence propre f_0 à définir.
- Un gain maximal H_0 à définir.