

TP 5- Filtrage linéaire LTI- Application au filtrage de sons

Objectifs

L'objectif de ce TP est de concevoir et synthétiser des filtres numériques linéaires à Réponse Impulsionnelle Finie (FIR) et à réponse impulsionnelle infinie (IIR), en s'appuyant sur les différentes représentations de ces filtres : réponse impulsionnelle, réponse fréquentielle. Ces filtres sont utilisés ici pour traiter des données audio (parole, musique).

Les programmes seront écrits en Python, en utilisant les libraires *numpy*, *signal* et *matplotlib*.

Exercice 1

On souhaite réaliser un système constitué d'un banc de filtres montés en parallèle, capables de séparer le signal en trois composantes, chacune étant définie sur une bande de fréquences (figure 2).

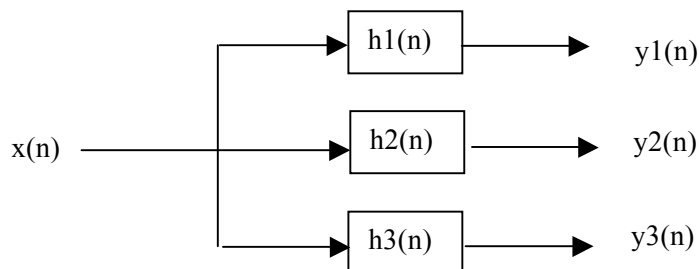


Figure 2. Structure de filtres en cascade
 h_1 , h_2 et h_3 sont les réponses impulsionnelles des 3 filtres

1. Synthétisez trois filtres FIR d'ordre 100. Ces filtres séparent l'axe des fréquences en trois zones : de 0 à 500 Hz, de 500 Hz à 2000 Hz et de 2000 Hz à 4000 Hz. La fréquence d'échantillonnage F_e est de 8000 Hz.

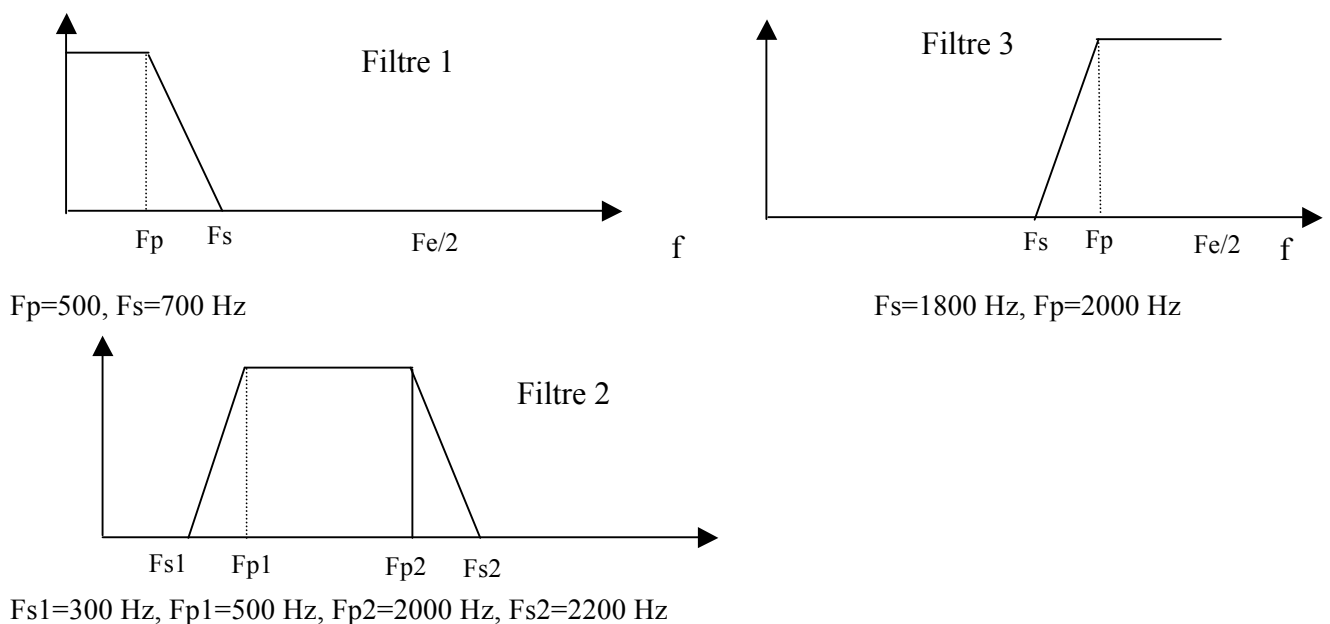


Fig. 3 Gabarit des 3 filtres : Réponses en fréquences $|H_1(f)|$, $|H_2(f)|$, $|H_3(f)|$

Calculez les réponses impulsionnelles et fréquentielles des 3 filtres.

2. Même question pour la conception de 3 filtres IIR d'ordre 10.
3. Pour chacune des méthodes de conception, visualisez la réponse fréquentielle de l'assemblage parallèle des 3 filtres.
4. Appliquez la transformation du signal au signal sonore choisi.
Vous pourrez écouter les signaux produits y1.wav, y2.wav et y3.wav.
Recomposez ensuite le signal $y = y1 + y2 + y3$ et réécoutez le.
Peut-on entendre la réponse en phase non linéaire ?
Vous pouvez jouer aussi sur la pondération des différents signaux y1, y2 et y3.