# Séance 2 : Implémentation d'un FIR en virgule fixe

## Partie 1 – Implémentation d'un FIR en virgule flottante avec Matlab

Le but de cette partie est d'implémenter l'équation aux différences d'un filtre FIR en virgule flottante. En effet, dans le dernier TP, vous avez conçu un filtre à l'aide de **filterDesigner** (ou **fdatool**) et vous avez réalisé le filtrage du signal PDM traité à l'aide de la fonction **filter**, i.e sans utiliser l'équation aux différences du FIR conçu.

### Exercice 1 – Filtrage à l'aide de la fonction filter

Réaliser le filtrage du signal PDM traité à l'aide de la fonction **filter**.

### Exercice 2 – Filtrage à l'aide de l'équation aux différences

Réaliser le filtrage du signal PDM traité à l'aide de l'équation aux différences. Pour cela, vous pourrez créer une fonction **FIRFiltering** qui prend en argument le signal à filtrer et les coefficients du FIR.

### Exercice 3 – Comparaison des signaux filtrés

Comparer la représentation temporelle et fréquentielle du résultat des deux filtrages : signal PDM filtré à l'aide de la fonction **filter** et signal PDM filtré à l'aide de l'équation aux différences. Vous pourrez vous écouter le résultat à l'aide de la fonction **soundsc**.

## Partie 2 – Implémentation d'un FIR en virgule fixe avec Matlab

Le but de cette partie est d'implémenter l'équation aux différences d'un filtre FIR en virgule fixe. Ainsi, il sera nécessaire de travailler avec des entiers 32 bits, à l'aide de la fonction **int32**. De plus, le signal filtré en virgule fixe devra être compris en 0 et 4095, i.e correspondant aux 12 bits du DAC.

### Exercice 1 – Traitement PDM en entier 32 bits

Réaliser le traitement du signal PDM, pour modéliser le fonctionnement de la fonction `__builtin_popcountll`.

### Exercice 2 – Filtrage du signal PDM traité en virgule fixe

Réaliser le filtrage en virgule du signal PDM traité. Pour cela, vous pourrez créer une fonction **FIRFiltering_q** qui prend en argument le signal à filtrer, les coefficients du FIR et le nombre de bits de partie décimal de la virgule fixe. De plus, vous devrez réfléchir aux nombres de bits de la partie entier et de la partie décimale.

### Exercice 3 – Mise à l'échelle

Créer une fonction scaling qui prend un signal codé en entier 32 bits dont les valeurs sont comprises entre 0 et 4095 et retourne un signal codé en virgule flottante comprise entre -1 et 1.

### Exercice 4 – Comparaison

Comparer la représentation temporelle et fréquentielle entre le signal filtré en virgule flottante et le signal filtré en virgule fixe. Vous pourrez écouter le résultat à l'aide de la fonction **soundsc**.

## Partie 3 – Implémentation d'un FIR en virgule fixe en C

Maintenant que vous avez conçu le filtre FIR et valider son implémentation en virgule fixe. Vous pouvez maintenant implémenter votre filtrage en virgule fixe en C sur votre microcontrôleur. Pour vous aider vous pouvez faire une implémentation C et comparer le résultat avec celui de Matlab.

## Pour aller plus loin

Vous pourrez évaluer l'impact de l'amplitude du signal audio sur le signal PDM traité ainsi que le signal PDM filtré.