
ENSEA

Beyond Engineering

# TP TRAITEMENTS NUMÉRIQUES AVANCÉS

## Ecouteurs à réduction de bruit

Nicolas Guérin et Emma Perret

Promo ESE 2022

# Introduction

Pour chaque bloc il faut réfléchir aux fréquences d'échantillonnage et de coupure mais également à la bande passante et au gain de cette bande. Il faut également penser à l'ordre du filtre. Pour chauqe signal, la fréquence d'échantillonnage et la longueur des mots sont déjà donnés.
Pour un signal PCM, nous pouvon facilement trouver la bande passante et le rapport signal à bruit :

1. La bande passante équivaut à la fréquence d'échantillonnage ($f_e \ ou \ f_s$) divisée par deux.
2. Le rapport signal à bruit ($RSB \ ou \ SNR$) se définit de la sorte : $RSB = 6.02 * N + 1.76$ avec $N$ : $la \ longueur \ des \ mots$. Il s'exprime en $dB$.
3. Pour un signal sur 16 bits, $RSB = 98dB$.
4. Nous acceptons un $RSB - 4dB$. Pour cela, nous sommes obligés de passer par un dithering. Sans le dithering, le $RSB$ est de $-8dB$ ce qui n'est pas acceptable.

ENSEA
Beyond Engineering

# Introduction

Nous avons donc listé les données pour tous les signaux PCM.
Pour le signal de sortie de l'égalisateur :

1. $f_e = 48kHz$
2. $Bande - passante = f_e/2 = 24kHz$
3. $N = 20bits$
4. $RSB = 122.16dB$
5. $RSB_{accepte} = RSB - 4 = 118.16dB$

Pour le signal de sortie du SR converter :

1. $f_e = 48kHz$
2. $Bande - passante = f_e/2 = 24kHz$
3. $N = 16bits$
4. $RSB = 98.08dB$
5. $RSB_{accepte} = RSB - 4 = 94.08dB$

ENSEA
Beyond Engineering

# Introduction

Pour le signal de sortie du player audio :

1. $f_e = 44.1kHz$
2. $Bande - passante = f_e/2 = 22.05kHz$
3. $N = 16bits$
4. $RSB = 98.08dB$
5. $RSB_{accepte} = RSB - 4 = 94.08dB$

Pour les signaux de sortie du micro et du PDM modulator :

1. $f_e = 6.144MHz$
2. $Bande - passante = f_e/2 = 3.72MHz$
3. $N = 1bits$
4. $RSB = 122.16dB$
5. $RSB_{accepte} = RSB - 4 = 118.16dB$

ENSEA
Beyond Engineering

# Partie 1 : Modulateur PDM

Le modulateur PDM permet de moduler un signal analogique en un signal binaire en fonction de la densité du signal. En effet, plus l'amplitude du signal analogique est élevée, plus le signal binaire sera constitué d'une suite longue de 1.

La génération du modulteur PDM a déjà été faite. Après avoir compris son fonctionnement, nous sommes donc directement passés au démodulateur PDM.

ENSEA
Beyond Engineering

# Partie 1 : Démodulateur PDM

## Théorie

La fréquence d'échantillonnage d'entrée vaut $f_e = 6.144MHz$ et celle de la sortie vaut $f_e = 48kHz$. Il y a un rapport de 128 entre les deux.
Il faut sous-échantillonner mais si nous échantillonnons par 128, cela est trop brutal et entraîne un ordre de filtre trop élevé et une pulsation de coupure très faible. Il faut donc faire par étape. Mais avant de sous-échantilloner il faut filtrer. Nous avons choisi de prendre un filtre FIR, cela nous permet aussi de conserver la forme du signal puisque c'est un filtre à phase linéaire. Cependant le FIR entraîne des problèmes de latence car nous avons un RSB élevé ($120dB$). Il faut donc faire attention au délai.

ENSEA
Beyond Engineering

# Partie 1 : Démodulateur PDM

## Signal d'entrée

Nous observons d'abord le signal d'entrée **??** :

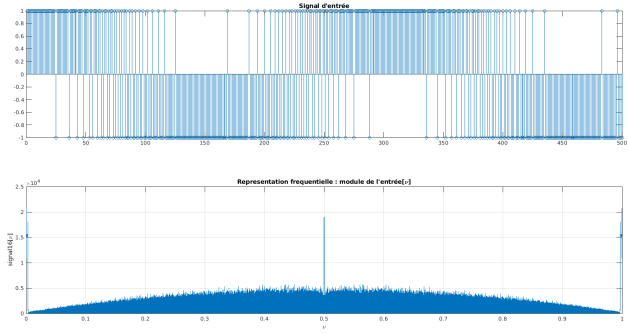



Figure: Signal d'entrée

# Partie 1 : Démodulateur PDM

## FIR : Mise en place

Dans un premier temps nous décidons de sous-échantilloneur par 16 puis par 8 (le produit des deux donne bien 128).
Nous utilisons Filter Designer pour designer nos filtres FIR.
Pour le filtre avant le sous-échantillonneur par 16, nous prenons :

1. $f_s = 6.144\ MHz$
2. $f_{stop} = f_e/16 = 384\ kHz$
3. $A_{stop} = 120\ dB$

Pour le filtre avant le sous-échantillonneur par 8, nous prenons :

1. $f_s = 384\ kHz$
2. $f_{stop} = f_e/16 = 24\ kHz$
3. $A_{stop} = 120\ dB$

ENSEA
Beyond Engineering

# Partie 1 : Démodulateur PDM

## FIR : Mise en place

Nous créons nos deux filtres :

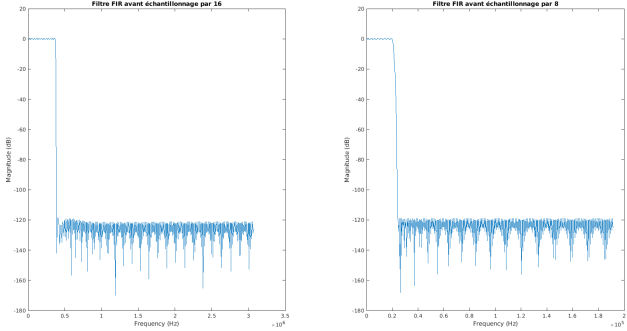



Figure: Filtres FIR

# Partie 1 : Démodulateur PDM

## FIR : Sous échantillonneur par 16

Nous observons les sorties de chaque sous-échantillonneur :

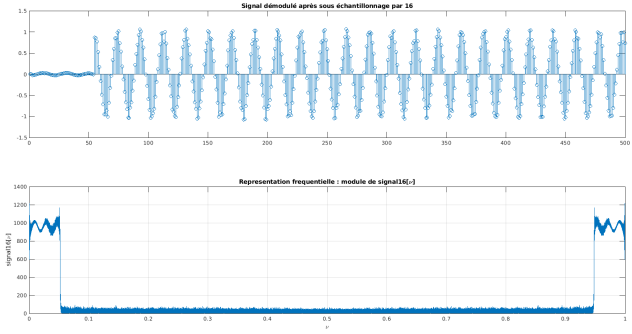



Figure: Sortie du sous-échantillonneur par 16

# Partie 1 : Démodulateur PDM

## FIR : Sous échantillonneur par 8

Voici alors notre sortie finale du démodulateur PDM :

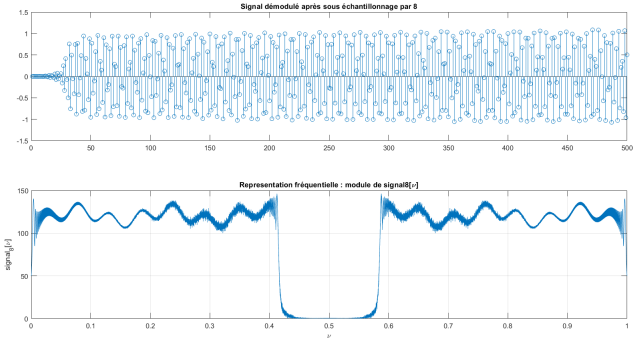



Figure: Sortie du sous-échantillonneur par 8

# Partie 1 : Démodulateur PDM

## Améliorations

Nous pouvons améliorer le démodulateur PDM en faisant plus d'étapes. En effet, les ordres de nos filtres sont encore assez élevés et créent un certain délai.
Nous pouvons faire 8 puis 4 puis 4 ou encore 4 puis 4 puis 4 puis 2.
Plus nous auront de blocs, plus la latence sera minimale.

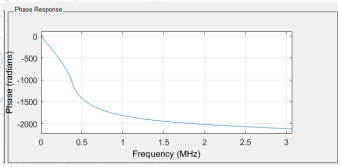
Par ailleurs, la latence est aussi dûe au choix du filtre. En effet, un filtre FIR crée un certain délai.
Au lieu de créer plus de bloc, nous décidons de changer de filtre et d'opter pour des filtres IIR.
Nous avons alors le choix entre un filtre Butterworth, Chebytchev type I, Chebytchev type II ou Elliptique.
Pour le choix du type de IIR, nous regarderons la phase et l'ordre.

ENSEA
Beyond Engineering

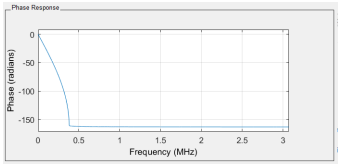
# Partie 1 : Démodulateur PDM

## IIR : Choix du type

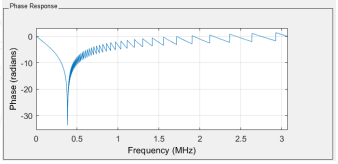
Voici les phases des 4 filtres designés pour être placé avant le sous-échantillonneur par 16 :



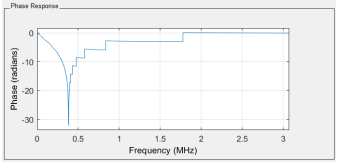

(a) Butterworth




(b) Chebytchev type I




(c) Chebytchev type I




(d) Elliptique

Figure: Phases des filtres IIR

# Partie 1 : Démodulateur PDM

## IIR : Choix du type

Les filtres Butterworth a un retournement de phase ce que nous souhaitons évité. Le filtre de Chebytchev type I a une phase qui atteint une valeure minimale importante en valeur absolue.Nouos éliminons ces deux là.
La différence entre Chebytchev type II et Elliptique se fait au niveau de l'ordre. En effet le filtre de Chebytchev type II proposé par FilterDesigner est un ordre 104 alors que l'Elliptique est un ordre 22. Comme nous cherchons à minimiser l'ordre, nous choisissons un filtre IIR elliptique.

ENSEA
Beyond Engineering

# Partie 1 : Démodulateur PDM

## IIR : Mise en place

Nous utilisons alors ces filtres :

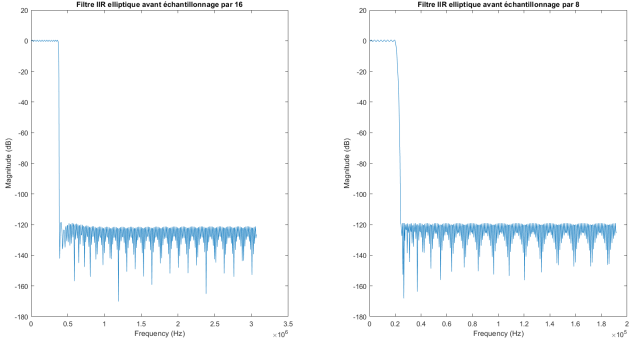



Figure: Filtres FIR

# Partie 1 : Démodulateur PDM

## IIR : Sous échantillonneur par 16

Nous observons les sorties de chaque sous-échantillonneur :

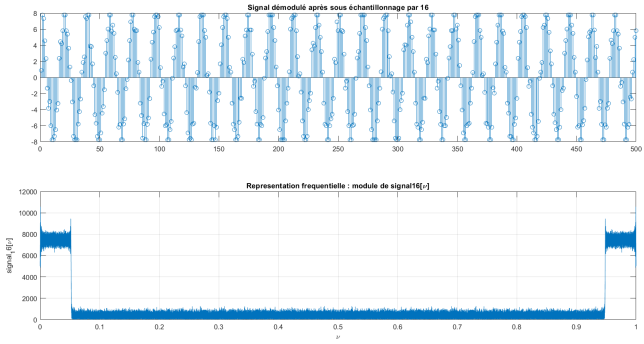



Figure: Sortie du sous-échantillonneur par 16

# Partie 1 : Démodulateur PDM

## IIR : Sous échantillonneur par 8

Voici alors notre sortie finale du démodulateur PDM :

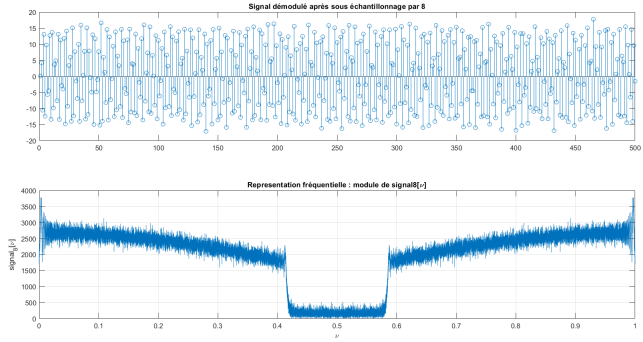


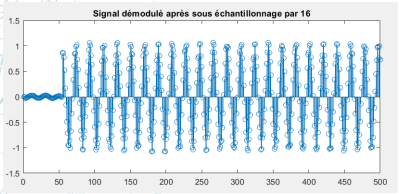

Figure: Sortie du sous-échantillonneur par 8

ENSEA
Beyond Engineering

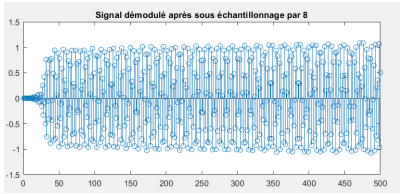
# Partie 1 : Démodulateur PDM

## Comparaison FIR et IIR

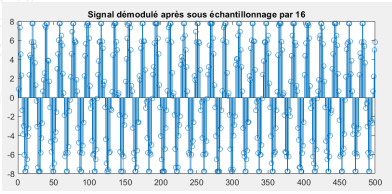
Nous constatons bien que les IIR suppriment la latence :



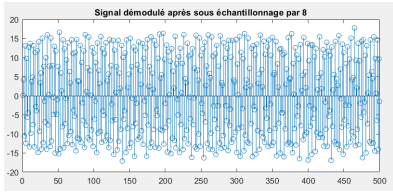

(a) Filtre FIR avant sous-échantillonneur 16




(b) Filtre FIR avant sous-échantillonneur 8




(c) Filtre IIR avant sous-échantillonneur 16




(d) Filtre IIR avant sous-échantillonneur 8

Figure: Comparaison des latences

ENSEA
Beyond Engineering

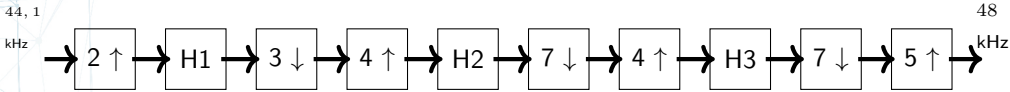
# Partie 2 : Sample-Rate Converter

Démarche:

Nous allons créer un étage Sample Rate Converter : $\boxed{F_{s_{in}} = 44,1kHz \quad F_{s_{out}} = 48kHz}$

- Rapport de réduction : $\frac{480}{441} = \frac{160}{147}$ **Problème:** Pour sur et sous échantillonner de cet ordre de grandeur, la puissance de calcul demandée est considérable nous choisirons plutôt de décomposer cette fraction en facteur premier pour mettre des filtres plus petit en cascade.
- Décomposons en facteurs premier : $\frac{160}{147} = \frac{2}{3} \times \frac{4}{7} \times \frac{4}{7} \times 5$
- Nous allons générer nos filtres avec l'application matlab **filter designer** onglet Polyphase Sample-Rate

Voici un schéma de la cascade que nous allons implémenter; le premier bloc est généré par Filter Designer et les deux suivants sont réaliser grâce a une implémentation polyphase pour réussir à obtenir un rapport $\frac{L}{M} = \frac{4}{7} < 1$ de ré-échantillonnage qui n'introduise pas de délai sur le filtrage.




ENSEA
Beyond Engineering

# Génération Premier étage de la cascade

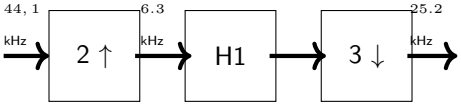



Figure: Représentation des blocs qui réalisent le premier étage

Nous avons choisi cet étage en première position car il permet de filtrer sans perdre d'informations sans pour autant trop augmenter la puissance de calcul due à l'upsampling.

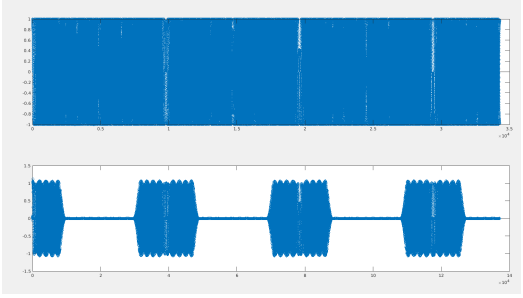


Figure: En haut: signal d'entrée sous-échantilloné, en bas: signal de sortie du premier etage de la

# Génération second étage de la cascade

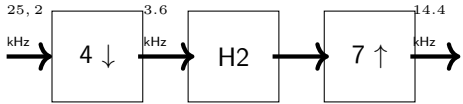



Figure: Représentation des blocs qui réalisent le premier étage

Le filtre polyphase est construit de la manière suivante : On réalise une répartition dans 4 branches des composants du vecteur colonne obtenu en sortie de l'étage précédent. On introduit 7 sous-branche déterminées à partir de la branche de plus haut niveau. sur chacune de ces sous branches on filtre par un second ordre. Ce second ordre est issue de la décomposition d'un filtre général en plusieurs niveaux par branches et sous-branches. Ainsi pour la branche 3, sous branche 6 on aura un filtre $H_{3,6}$ du second degré issue du filtre général $H$. On reproduit ce schéma pour le dernier étage également.

ENSEA
Beyond Engineering

# Génération dernier étage de la cascade

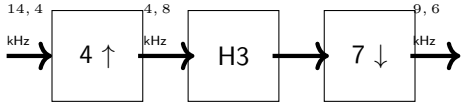



Figure: Représentation des blocs qui réalisent le premier étage

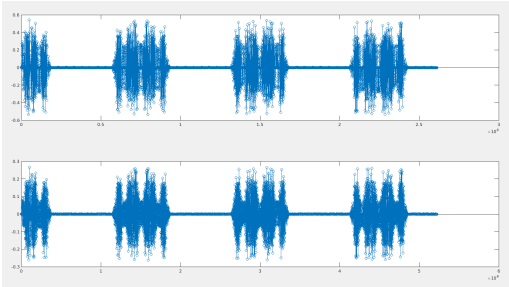


Figure: En haut: signal d'entrée sous-échantilloné, en bas: signal de sortie du premier étage de la cascade de filtre

# Upsampling de sortie

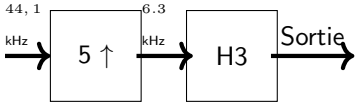
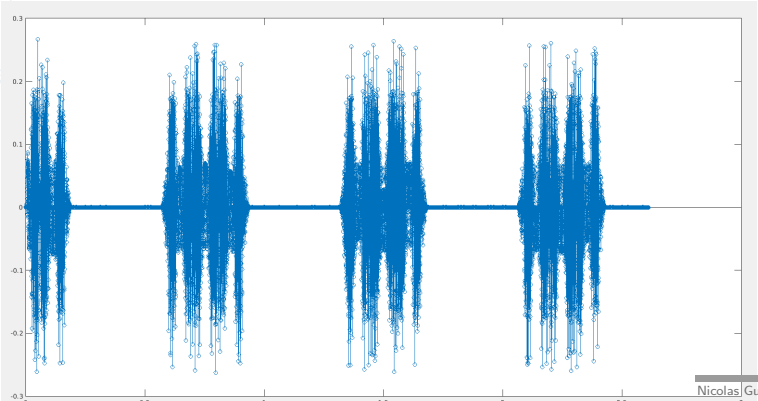



Figure: Représentation des blocs qui réalisent le premier étage



# Addition des signaux

Nous effectuons la somme des signaux et le rééquilibrage des tailles pour les vecteurs.

ENSEA
Beyond Engineering

# Méthode fréquentielle

On applique l'algorithme WOLA pour permettre la reconstruction parfaite du signal.

1. On débute par sélectionner une fenêtre (Hamming) et un taux de recouvrement (50%) et vérifier le critère COLA par la fonction `iscola`.
2. On génère deux vecteurs avec une répétition de ces fenêtres et un décalage de 50% entre les deux vecteurs.
3. On applique les vecteurs au signal et on

ENSEA
Beyond Engineering

# Méthode fréquentielle

On applique l'algorithme WOLA pour permettre la reconstruction parfaite du signal.

1. On débute par sélectionner une fenêtre (Hamming) et un taux de recouvrement (50%) et vérifier le critère COLA par la fonction `iscola`.
2. On génère deux vecteurs avec une répétition de ces fenêtre et un décalage de 50%entre les deux vecteurs.
3. On applique les vecteurs au signal et on

ENSEA
Beyond Engineering

# Partie 3 : Banc de filtres temporels

## Principe

Dans un premier temps nous allons utiliser une méthode temporelle et faire un banc de filtres octaves à 9 étages pour avoir 10 bandes comme souhaité.
Le but est d'ajouter un gain par bande entre l'analyse et la synthèse pour créer l'égaliseur.
Nous construisons 4 filtres complémentaires avec la fonction $firpr2chfb$ pour ce banc de filtres.
Le vu mètre correspond simplement à la mesure de l'énergie sur chaque bande et sera matérialisé par une matrice.

ENSEA
Beyond Engineering

# Partie 3 : Banc de filtres temporels

## Analyse

Pour l'analyse, nous utilisons les filtres $H_0$ et $H_1$ et nous sous-échantillonnons par 2 à chaque étage comme ci-dessous.

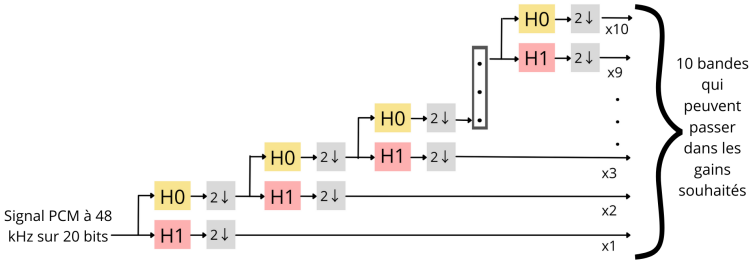



Figure: Analyse du banc de filtres

# Partie 3 : Banc de filtres temporels

## Traitement : gains

Pour la partie traitement entre l'analyse et la synthèse, il s'agit juste d'offrir la possbilité d'appliquer à chaque bande le gain souhaité pour modifier l'ambiance de l'audio. Il suffit donc de multiplier chacun des 10 signaux de sortie de l'analyse par un gain associé (exmple : nous multiplions $x_1$ par $G_1$ etc.).

ENSEA
Beyond Engineering

# Partie 3 : Banc de filtres temporels

## Synthèse

Pour l'analyse, nous utilisons les filtres $F_0$ et $F_1$ et nous sur-échantillonnons par 2 à chaque étage comme ci-dessous.

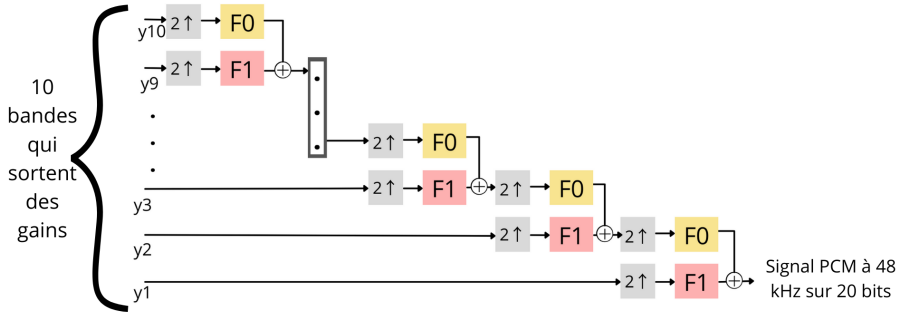



Figure: Synthèse du banc de filtres