Filtrage temps-réel et MATLAB (introduction)

Matthieu Kowalski

Table des matières

1 Rappels des cours précédents : filtrage idéal et FIR

Filtrage

Définition
Un filtre est un système **linéaire** et **invariant dans le temps**. Il peut donc s'écrire comme une convolution.

Réponse impulsionnelle
Soit $\mathcal{S}$ un filtre. La réponse impulsionnelle h de $\mathcal{S}$ correspond à la sortie du système à l'impulsion unité (Dirac). Ainsi

$$h = \mathcal{S}(\delta)$$

et l'on a, pour tout signal x

$$y = \mathcal{S}(x) = h \star x = x \star h \quad y_n = \sum_{k=-\infty}^{+\infty} h_k x_{n-k}$$

Pour les signaux finis, la convolution suppose les signaux périodiques, de même période !

Filtres réalisables – 1

Filtre réalisable
Un filtre de réponse impulsionnelle h est réalisable ssi il est stable et causal.

Remarque 1

— Si un filtre est stable, alors il admet une transformée de Fourier

— Réciproquement, si un filtre admet une transformée de Fourier, alors il est stable.

Remarque 2

— Un filtre réalisable admet forcément une transformée de Fourier

— Si un filtre admet une transformée de Fourier il n'est pas forcément réalisable, car il peut ne pas être causal

Filtres réalisables – 2

Filtre stable
Un filtre de réponse impulsionnelle h est stable ssi

$$\sum_{k=-\infty}^{+\infty} |h_k| < +\infty$$

Filtre stable
Un filtre de réponse impulsionnelle h est causal ssi h est causal, ie

$$h_k = 0 \quad \forall k < 0$$

Filtre réalisable
Un filtre de réponse impulsionnelle h est réalisable ssi il est stable et causal.

Filtrage et transformée de Fourier

Réponse en fréquence ou Gain complexe
La réponse en fréquence, ou gain complexe, d'un filtre est sa transformée de Fourier (quand elle existe !).

Filtrage dans le domaine fréquentielle
Soit $\mathcal{S}$ un filtre de impulsionnelle h et x un signal. On a

$$y = \mathcal{S}(x) = h \star x$$

Si h et x admettent une transformée de Fourier, on a dans le domaine fréquentiel :

$$\hat{y} = \hat{h} \cdot \hat{x}$$

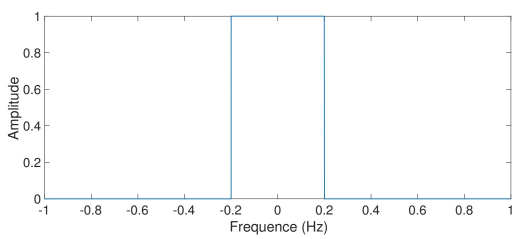
Filtrer un signal, c'est agir directement sur son spectre !

Filtre passe-bas idéal – 1

Définition
La réponse en fréquence d'un filtre passe-bas idéal de fréquence de coupure ν_0 est donnée par :

$$\hat{h}(\nu)^{\mathrm{pb}_{\nu_0}} = \begin{cases} 1 & \text{si } |\nu| < \nu_0 \\ 0 & \text{sinon} \end{cases}$$

Réponse en fréquence

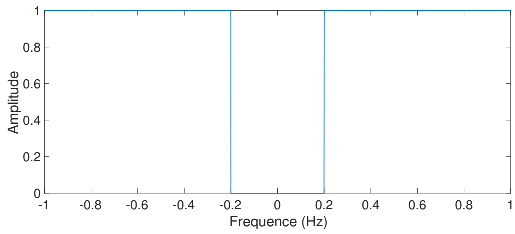


Filtre passe-haut idéal – 1

Définition
La réponse en fréquence d'un filtre passe-haut idéal de fréquence de coupure ν_0 est donnée par :

$$\begin{aligned}\hat{h}(\nu)^{\mathrm{ph}_{\nu_0}} &= \begin{cases} 0 & \text{si } |\nu| < \nu_0 \\ 1 & \text{sinon} \end{cases} \\ &= 1 - \hat{h}(\nu)^{\mathrm{pb}_{\nu_0}}\end{aligned}$$

Réponse en fréquence

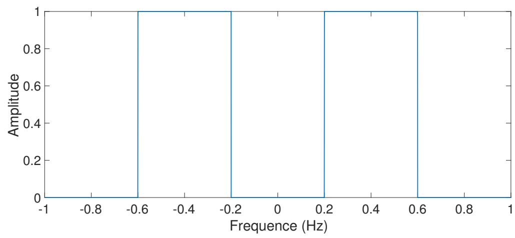


Filtre passe-bande – 1

Définition
La réponse en fréquence d'un filtre passe-bande idéal de fréquences de coupures $\nu_0 > 0$ et $\nu_1 > 0$ est donnée par :

$$\begin{aligned}\hat{h}(\nu)^{\mathrm{pbande}_{\nu_0;\nu_1}} &= \begin{cases} 1 & \text{si } \nu_0 < \nu < \nu_1 \\ 1 & \text{si } -\nu_0 < -\nu < -\nu_1 \\ 0 & \text{sinon} \end{cases} \\ &= \hat{h}(\nu)^{\mathrm{pb}_{\nu_1}} - \hat{h}(\nu)^{\mathrm{pb}_{\nu_0}}\end{aligned}$$

Réponse en fréquence



Filtres – équation aux différences
Soit un filtre de réponse impulsionnelle h. Alors le signal y, version filtrée du signal x par h, est donnée par :

$$y_n = (h \star x)_n = \sum_{k=-\infty}^{+\infty} h_k x_{n-k}$$

Comment réaliser un tel filtre en "temps réel" ?

Filtres FIR ou MA

Définition
Soit un filtre de réponse impulsionnelle h. Le filtre est dit "à réponse impulsionnelle finie" (FIR) ou "à moyenne mobile" (MA) si h est finie : $h = \{h_{-k_1}, \dots, h_0, \dots, h_{k_2}\}$

L'équation aux différences s'écrit alors :

$$y_n = \sum_{k=-k_1}^{k_2} h_k x_{n-k}$$

On appelle **ordre** du filtre, le nombre d'échantillons de sa réponse impulsionnelle.

Remarques

— Un filtre FIR est forcément **stable**

— Il n'est pas forcément **causal**

— Un filtre FIR est **réalisable** ssi il est causal

Synthèse de filtre RIF

But
Synthétiser un filtre RIF (ou MA) **causal**, qui s'approche le plus possible du filtre idéal recherché.

RI du filtre RIF recherché VS RI du filtre idéal

— Le filtre idéal a une RI h^{ideal} a support infini, non causal :

$$y_n = \sum_{k=-\infty}^{+\infty} h_k x_{n-k}$$

— Le filtre RIF que l'on cherche étant causal, sa réponse impulsionnelle h doit être causale :

$$y_n = \sum_{k=0}^{K} h_k x_{n-k}$$

Synthèse de filtre RIF

But
Synthétiser un filtre RIF (ou MA) **causal**, qui s'approche le plus possible du filtre idéal recherché.

Synthèse par fenêtrage

— **Calcul** de la RI h^{ideal} par TF inverse :

$$h_n^{\text{ideal}} = \int_{-1/2}^{1/2} H(\nu) e^{i2\pi n\nu} \, \mathrm{d}\nu$$

— **Fenêtrage** de la RI h^{ideal}

$$h^{\text{win}} = w \cdot \{h_{-N/2}^{\text{ideal}}, \dots, h_{N/2}^{\text{ideal}}\}$$

— Application d'un **retard** sur h^{tronc}, afin de décaler les indices pour rendre le filtre causal

$$h_n^{\text{RIF}} = h_{n-N/2}^{\text{win}}$$

Paramètres d'un filtre FIR

— Ordre (nombre de coefficients)

— Fenêtre (Rectangulaire, Hamming, Hann, Blackman ...)

2 Les bases du temps-réel audio sous matlab

Notion de temps réel

- La notion de temps-réel est liée à celle de latence, et dépend de l'application
- Latence = durée qui s'écoule entre l'action et la réaction.
- Par exemple, un décalage entre le son et l'image de 20 ms est acceptable. Au dela, on perçoit le décalage.
- Traiter un signal en temps-réel = traiter un signal en minimisant la latence

Structure général de traitement

Structure de traitement

- Lire un "bloc" de signal (fichier, micro. . .)
- Traiter ce bloc
- Écrire ce bloc (fichier, enceintes. . .)

Problèmes

- Le traitement ne doit pas introduire de latence
- En video : < 20 ms. En audio : < 5 ms !

Temps réel audio sous Matlab

Boucle temps-réel Matlab avec la toolbox dsp.systems

```
% Lecture d'un fichier audio:
AFR = dsp.AudioFileReader('fichier.wav','SamplesPerFrame',frameLength);

% Ecriture d'un fichier audio:
ADW = dsp.AudioPlayer(AFR.SampleRate);

while ~isDone(AFR)
    % Lecture d'un block de signal
    audio = step(AFR);

    % Traitement de ce block
    audio_processed = signal_processing(audio);

    % Ecriture du block de signal
    step(ADW, audio_processed)
end

% Fermeture des ressources
close(AFR);
close(ADW);
```

Temps-réel et filtrage ideal

- Implémenter un filtrage ideal par bloc
  voir cours-TD 1
- Qu'observe-t-on ?
  Écoute très éloignée du signal original
- Pourquoi entend-t-on ce phénomène ?
  Le filtre n'est pas compatible avec le temps-réel. Les artefacts viennent de la convolution circulaire !

Temps-réel et filtrage FIR

- Quels échantillons sont nécessaires pour le filtre FIR ?
  des K précédents échantillons, où K est l'ordre du filtre
- Par quoi semble-t-il raisonnable de limiter l'ordre du filtre ?
  Par la taille des blocs utilisée pour le temps-réel
- Proposer une implémentation d'un filtrage temps-réel, avec un filtre FIR
- A partir de quel ordre le signal correctement filtré ?
  Cela dépend de la fenêtre. Un ordre d'environ 100 semble nécessaire avec une fenêtre de Hann.

Gain en décibel

Pour agir sur les fréquences, on peut leur appliquer un gain en décibel. Soit s le signal original et $h \star s$ le signal filtré :

Gain en décibel

$$G = 10 \log_{10} \left(\frac{\|h \star s\|^2}{\|s\|^2} \right) = 10 \log_{10} \left(\frac{\|\hat{h} \cdot \hat{s}\|^2}{\|\hat{s}\|^2} \right)$$

Ainsi, si l'on veut appliquer un gain de 3 dB, il suffit d'amplifier les fréquences de :

$$\hat{h}_k = 10^{G/10}$$

On voit que pour $G = 3$, alors $h \simeq 2$. Un gain de 3 dB double l'énergie du signal !

Vers un Equalizer numérique

Reprendre le programme ci-dessus avec :

- Une fonction pour calculer les coefficients du filtre voulu, selon la fréquence de coupure, l'ordre et le gain (en DB) voulu :

  [FIRcoeff] = coeff_passe_bas_fir(fe,fc,ordre,fenetre,gainDB,gainMax)

- Une fonction pour filtrer en temps-réel

  [sig_filtered] = realTime_filtering(sig_pad,frameLength,FIRcoeff)

  où sig_pad est le signal contenant suffisament d'échatillons passés pour appliquer le filtre.

Vers un Equalizer numérique

- Le but est de créer un equalizer numérique.
- On considère que l'oreille humaine est sensible aux fréquences sur une échelle logarithmique, de 20 Hz à 20000 Hz.

Division de la bande fréquentielle

- sub-basses : 30 a 63 Hz
- basses : 63 a 250 Hz
- bas-mediums : 250 a 500 Hz
- mediums : 500 a 2000 Hz
- haut-mediums : 2 a 4 kHz
- aigus : 4 a 20 kHz

Vers un Equalizer numérique trois bandes

Ce qui donne, pour un equalizer trois bandes, la division suivante :

Equalizer trois bandes

— Basses : < 250 Hz

— Medium : 250 à 2000 Hz

— Aigus : > 2000 Hz

Vers un Equalizer numérique trois bandes

— Quels sont les types de filtres qui vont intervenir pour créer un equalizer trois bandes ?
  Un passe-bas, un passe-bande et un passe-haut

— Implémenter trois fonctions qui calcule les coefficients FIR de ces trois filtres
  [FIRcoeffBass] = coeff_passe_bas_fir(fe,fc,ordre,fenetre,gainDB,gainMax) [FIRcoeffMedium] = coeff_passe_bande_fir(fe,fc1,fc2,ordre,fenet
  [FIRcoeffAigu] = coeff_passe_haut_fir(fe,fc,ordre,fenetre,gainDB,gainMax)

Vers un Equalizer numérique trois bandes : interface graphique

Voir Matlab. . .