

COURS 3: INTRODUCTION AU TEMPS RÉEL

---

# TRAITEMENT DU SIGNAL

## 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 delà, 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 cf. MatLab

# TEMPS-RÉEL ET FILTRAGE IDEAL

- ▶ Implémenter un filtrage idéal par bloc
- ▶ Qu'observe-t-on ?
- ▶ Pourquoi entend-t-on ce phénomène ?

## TEMPS-RÉEL ET FILTRAGE FIR

- ▶ Quels échantillons sont nécessaires pour le filtre FIR ?
- ▶ Par quoi semble-t-il raisonnable de limiter l'ordre du filtre ?
- ▶ Proposer une implémentation d'un filtrage temps-réel, avec un filtre FIR
- ▶ A partir de quel ordre le signal correctement filtré ?

# 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  $h * s$  le signal filtré:

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

- ▶ 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 suffisamment d'échantillons 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 20 000 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:
- ▶ Division de la bande fréquentielle
  - ▶ 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 ?
- ▶ Implémenter trois fonctions qui calcule les coefficients FIR de ces trois filtres

## 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,fenetre,gainDB,gainMax)`

`[FIRcoeffAigu] = coeff_passe_haut_fir(fe,fc,ordre,fenetre,gainDB,gainMax)`

# VERS UN EQUALIZER NUMÉRIQUE TROIS BANDES: INTERFACE GRAPHIQUE

- ▶ Voir MatLab