Numérisation

1. Fréquence d'échantillonnage

Trop d'information ➔ problème de stockage
Pas assez d'information ➔ perte d'information

Trouver un compromis : **Théorème de Shannon**

$$F_e \geq 2 \times F_{max}$$

Exemple :

Signal audio pleine bande à un spectre s'étalant de 20 Hz à **20 kHz**

$F_e \geq$ **40kHz** (= 2 x 20 kHz)

Réduire le coût du traitement ou des données

➔ limitation du spectre par **filtrage** (fréquence de coupure F_c)

Numérisation

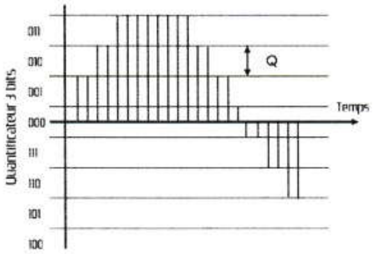
1. Fréquence d'échantillonnage

Qualité	Spectre du signal F_{max}	$F_e \geq 2 \times F_{max}$	Applications
Téléphonique	300-3400Hz	8kHz	Téléphonie fixe
« Bande élargie »	50-7000Hz	16 à 22kHz	PC, audio-conférence (compression ADPCM)
Radiodiffusion	50-15000Hz	32kHz	DAB, NICAM (transmission son TV)
« Hi-Fi »	20-20000Hz	44,1 à 48kHz	CD Audio, Studio numérique, DAT

Numérisation

2. Quantification

Réduire la taille de l'ensemble des valeurs possibles des échantillons (qui peut être très grande)



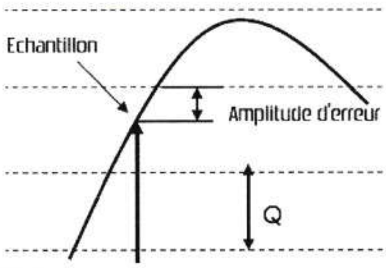
Les valeurs de quantification sont espacées d'un **pas** (Q) :

- de taille fixe pour une quantification uniforme
- ou variable pour une quantification non-uniforme

Numérisation

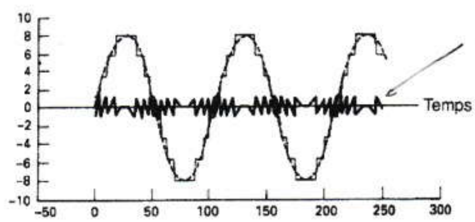
2. Quantification

Ceci produit une **erreur de quantification** entre le signal source et le signal quantifié



Pour une quantification uniforme, on a une erreur de **Q** / 2 *(max)*

Sur tout le signal on calcule l'erreur quadratique moyenne



erreur ~ bruit de quantification (parasite)

Numérisation

4. Formats

Il existe de nombreux formats de fichiers son :

- `.raw` : fichier binaire nu (brut)
- `.wav` : introduit par Microsoft
- `.aiff` : utilisé sur Silicon Graphics et Macintosh
- `.au` : format SUN
- `.mp3` : LE format d'échange (compressé)
- **etc.**

➔ encapsulent des informations complémentaires

Poids du fichier (bits/seconde)

Numérisation

✧ Exemples à écouter !

- De la parole et/ou de la musique
- Fichier original de parole : 16 bits, 16 kHz
- Variation de la fréquence d'échantillonnage

  on arrive plus à ~~saturer~~ comprendre ce qui est dit

  8 kHz
  4 kHz
  2 kHz

- Variation du nombre de bits de quantification

  do + en + saturé

  8 bits
  4 bits
  2 bits

Numérisation

2. Quantification

Le pas de quantification Q dépend du **rapport signal à bruit**

(SNR = Signal to Noise Ratio)

Capacité à faire ressortir une information par rapport au bruit

←→ qualité d'une transmission d'information (valeur élevée = bonne qualité)

$SNR_{dB} = 6{,}02b + 1{,}76\,dB$

	Nombre de Bits	**SNR_{dB}**
Qualité « Hi-Fi »	16 - 18 bit	~ 95
Codage la parole , NICAM	14 bit	~ 80
Codage son PC	8 bits	~ 40

Numérisation

3. Codage

Après une quantification (= limitation du **nombre de bits** nécessaires pour stocker l'information)

Codage de l'information sur ***b*** **bits** :

→ 2^b valeurs possibles : **de 0 à 2^b-1**

1 bit : 2 valeurs 0, 1 *2 bits : 4 valeurs 0,1,2,3*

8 bits : 256 valeurs de 0 à 255 ...

Approximation des valeurs du signal continu en les ramenant à un ensemble réduit de valeurs

Exemple : codage sur 2 bits, des valeurs entières de 0 à 11

$\underbrace{0\,1\,2}_{0}\ \underbrace{3\,4\,5}_{1}\ \underbrace{6\,7\,8}_{2}\ \underbrace{9\,10\,11}_{3}$ —

Exercices

1- Calculer la taille mémoire d'1h d'un enregistrement en qualité téléphonique (8 bits, 8 kHz, mono).

Signal échantillonné à 8 kHz ⇒ 8 000 val/s et 1h = 3600s

Par 1h d'enregistrement ⇒ 8000 x 3600 = 28 800 000 ~~val~~ échantillons

Codé sur 1 octet (8 bit) ⇒ 28 800 000 octets = 28,8 Mo

Exercices

2- Même question pour un enregistrement en qualité CD (16 bits, 44.1 kHz, stéréo).

Signal échantillonné à 44,1 kHz ⇒ 44 100 val/s et 1h = 3600s

Par 1h d'enregistrement ⇒ 44100 x 3600 = 158760000 octets

Sur 16 bit (x2)
Stéréo (x2) } 158760000 x 2 x 2 = 635 040 000 octets soit 635,04 Mo

Exercices

3- Quel est le débit (en bits/s) d'un signal en qualité studio (16 bits, 16 kHz) ?

Fréquence x Nb bit x Nb piste

16000 x 16 x 1 = 256 000 bit/s = 256 kb/s = 32 ko/s

/8

Exercices

4- Si un ordinateur possède une connexion internet qui peut fonctionner à 100 ko/s, combien de temps sera nécessaire pour télécharger un fichier d'une heure qualité CD ?

↳ 16 bit, 44,1 kHz, stéréo

44100 échantillons/s x 3600 pour 1h → 158 760 000 valeurs

2 pistes (2x) → 317 520 000 valeurs codées sur 16 bits (2 octets)

Exercices

5- Si le signal suivant a été quantifié sur 5 bits, quelles sont les valeurs du signal numérisé ? 010111101000100.

$\underbrace{01011}_{1}\ \underbrace{11010}_{2}\ \underbrace{00100}_{3}$

$2^4\ 2^3\ 2^2\ 2^1\ 2^0$
16 8 4 2 1

0 0 0 0 0 → 0
1 1 1 1 1 → 31

(1) : 11
(2) : 26
(3) : 4

Exercices

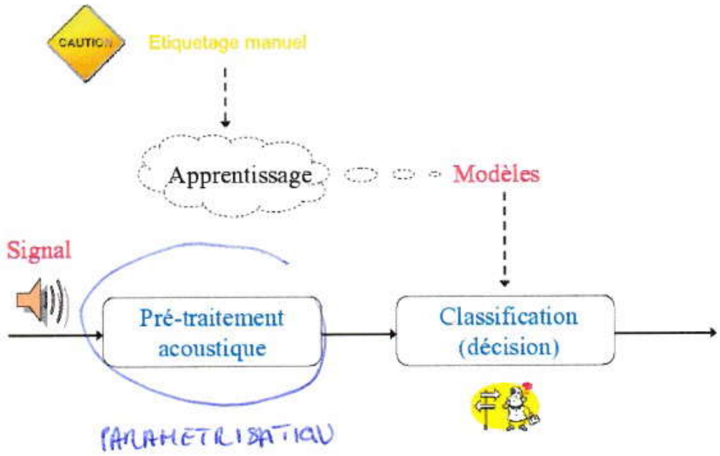
6- Quel est le rapport signal sur bruit en dB d'un signal DVD audio (24 bits 196 kHz) ?

$SNR_{dB} = 6{,}02b + 1{,}76\,dB$

$b = 24 \longrightarrow SNR_{dB} = 146{,}24$

Mais, que faire ?

- ✧ Système classique de reconnaissance des formes



Mais, que faire ?

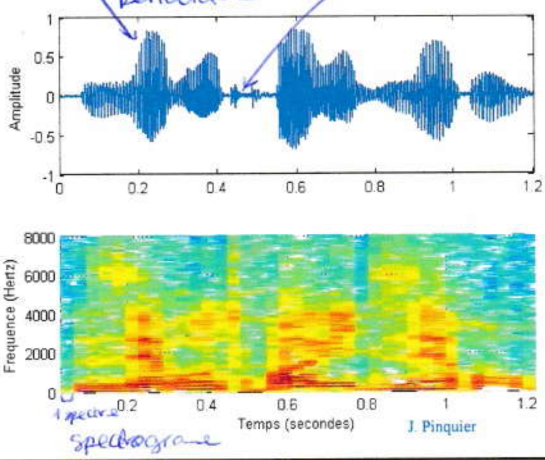
- ✧ Exemple : signal de parole

Partie voisée zone periodique

Partie non voisée zone bruitée

Representation temporelle

Representation temps/fréquence



Mais, que faire ?

- Pourquoi effectuer une paramétrisation ?
  - CAR l'évolution temporelle du signal ne fournit pas directement les informations acoustiques
  - zone voisée / non voisée

    energie (= variance du signal stationnaire)
    taux de passage à zéro (ZCR)
  - fréquence fondamentale (F_0)

    calcul d'autocorrélation
  - spectre (pour passer par les fréquences)

    décomposition du signal en une somme de sinusoïde, caractérisées chacunes ~~des~~ par son amplitude et sa fréquence

Mais, que faire ?

- Comment effectuer la paramétrisation ?
  - CAR le signal de parole n'est pas stationnaire
  - travailler sur des portions de signal et pas sur l'ensemble
  - statistiques à court terme / statistiques à long terme
  - fenêtres temporelles de 10 à 30 ms / plusieurs secondes

    généralement — par du traitement de détection de la langue
  - Spectre

    Transformée de Fourrier à court terme
    TF Discrète (algorithme TF Rapide)

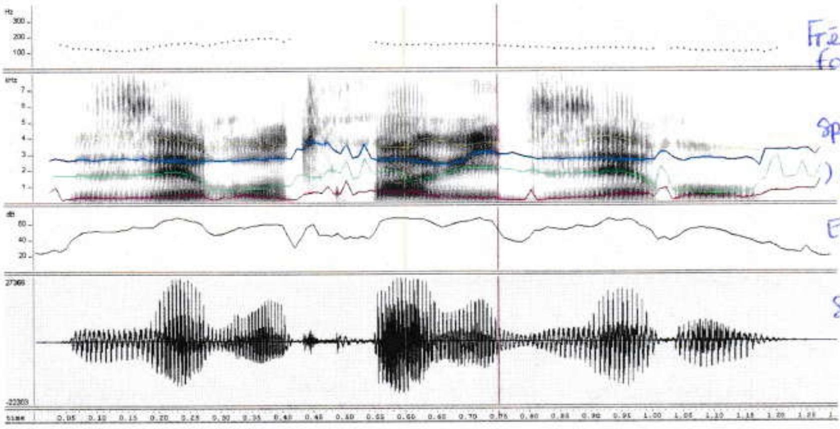
$$S(k) = \sum_{n=0}^{N-1} s(n) \cdot e^{-2i\pi k \frac{n}{N}}$$

  - Spectrogramme

    évolution à court terme du spectre de fréquence

Mais, que faire ?

✧ Avec un logiciel classique « wavesurfer »



Fréquence fondamentale en Hz (pitch)

Spectrogramme (formants)

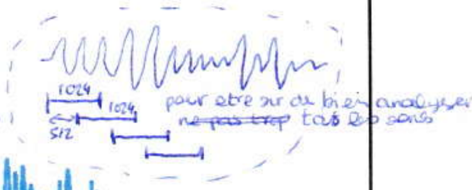
) on peut potentiellement détecter les voyelles avec les 2 premiers freq

Energie / Intensité (dB)

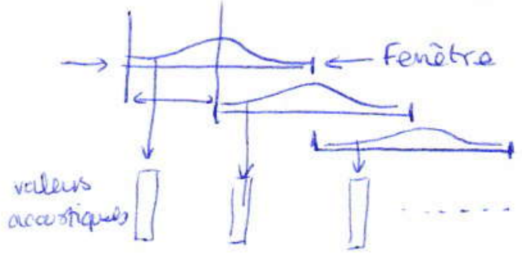
Signal

Mais, que faire ?

✧ Analyse sur une fenêtre glissante
- Ordre de 20 ms
- Recouvrement (moitié en général)



pour être sûr de bien analyser ~~ne pas trop~~ tous les sons



Mais, que faire ?

✧ Quelques paramètres

- Temporels : Energie, ZCR
- Fréquentiels : issus de la DSP, coefficients spectraux
  densité spectrale de puissance
- Autre : Mel Frequency Cepstrum Coefficients (MFCC)
  Traitements complémentaire aux paramètres fréquentiels pour aller plus loins
- Etc.

Mais, que faire ?

✧ **Exercice** : soit un enregistrement sonore mono (1 piste), échantillonné à une fréquence de 16 kHz. L'analyse se fait sur des fenêtres de 256 points (avec recouvrement sur la moitié). ⇒ décalage de 128 points

1. Quel est le nombre de points capturés par intervalle de temps ?
2. Quelle est la durée (en ms) d'une fenêtre d'analyse ?
3. Quel est le nombre de fenêtres (vecteurs) à traiter pour un fichier de 10 minutes ?

Mais, que faire ?

✧ **Exercice** (solution) :

1. Quel est le nombre de points capturés par intervalle de temps ?

   16kHz ⇔ 16000 point/s

2. Quelle est la durée (en ms) d'une fenêtre d'analyse ?

   ~~16000~~/256 points ⇔ 256/16000 = 0,016 = 16ms

3. Quel est le nombre de fenêtres (vecteurs) à traiter pour un fichier de 10 minutes ?

Durée : 10min ⇔ 600s

nb fenêtre (par s) : 16000/256 × 2 = 125 (recouvrement)

Total : 125 × 600 = 75000 fenetres

⇒ en réalité 75000 − 1 = 74999 fenetres

sinon ⇒ segmentation fault

Plan

✧ Introduction

✧ Paramétrisation générale

- Analyse temporelle
- Analyse fréquentielle
- Le cepstre
- Autres paramètres

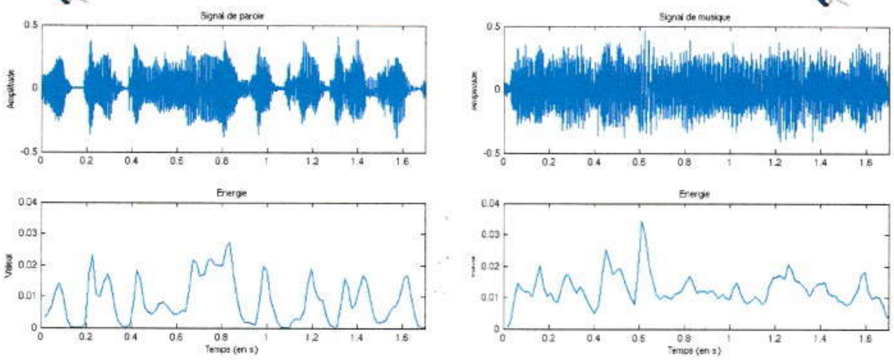
✧ Paramétrisation spécifique

Analyse temporelle

pour normaliser (faire la moyenne)

nombre de point

- Energie : $E(t) = \frac{1}{N} \sum_{n=1}^{N} x_n^2(t)$ = somme des amplitudes au carré
  - Discrimination des sons voisés / non-voisés
  - Détection de silence
  - Aide à la décomposition en parole / musique



Analyse temporelle

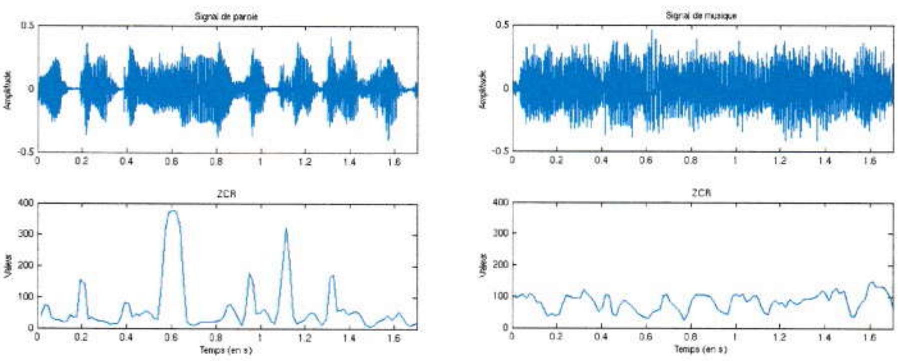
renvoie 1 si > 0

renvoie -1 si < 0

taux de passage par zéro

- ZCR : $ZCR(t) = \frac{1}{2N} \sum_{n=1}^{N} |sign(x_n) - sign(x_{n-1})|$
  - Plus discriminant que l'énergie

↳ taille de la fenêtre



signal -0,2 0,1 0,3 -0,4 -0,6 0,9

$\underbrace{-0{,}2 \;\; 0{,}1 \;\; 0{,}3 \;\; -0{,}4 \;\; -0{,}6}_{x_{n-1}}$

$\underbrace{0{,}1 \;\; 0{,}3 \;\; -0{,}4 \;\; -0{,}6 \;\; 0{,}9}_{x_n}$

$a = sign(x_{n-1})$	-1	1	1	-1	-1
$b = sign(x_n)$	1	1	-1	-1	1
$\lvert b-a \rvert$	2	0	2	0	2

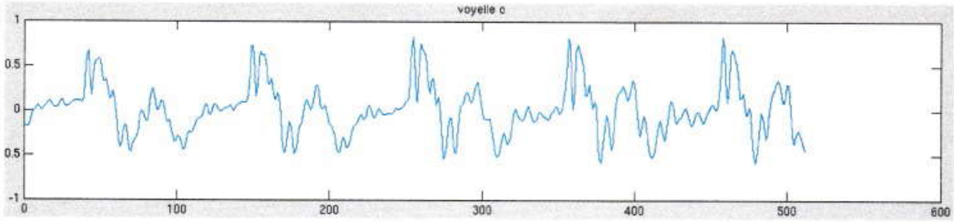
$\boxed{\frac{1}{2}\sum \lvert b-a \rvert = 3}$ ⇒ $\frac{2}{3}$

Analyse fréquentielle

✧ Spectre

$$x_n \rightarrow \boxed{\text{Accentuation}} \rightarrow \boxed{\text{Fenêtrage}} \rightarrow \boxed{|\text{FFT}|} \rightarrow \tilde{x}_n$$

- Signal x_n



Analyse fréquentielle

✧ Spectre

$$x_n \rightarrow \boxed{\text{Accentuation}} \rightarrow \boxed{\text{Fenêtrage}} \rightarrow \boxed{|\text{FFT}|} \rightarrow \tilde{x}_n$$

- Signal x_n
- Préaccentuation des aigus (←→ filtre passe-haut)

$H(z) = 1 - az^{-1}$ avec $a = 0,98$ en général

Analyse fréquentielle

✧ Spectre

$x_n \rightarrow$ Accentuation $\rightarrow$ Fenêtrage $\rightarrow$ | FFT | $\rightarrow \tilde{X}_n$

- Signal x_n
- Préaccentuation des aigus
- Fenêtrage

Analyse fréquentielle

✧ Fenêtrage : pourquoi ?

- Signal quasi-stationnaire sur fenêtres de 10 à 30 ms
- Pour limiter les effets de bord (réduire les discontinuités)

→ Pondération ~~des discontinuité~~ par fenêtre temporelle aplatie aux extrémités

✧ Analyse sur une fenêtre glissante

→ Recouvrement (moitié en général)

✧ Exemples de fenêtres

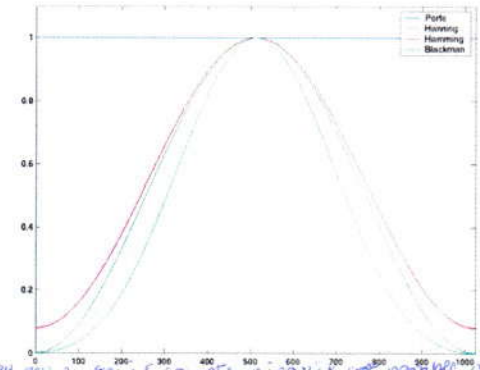
- Hamming → généralement utilisée en audio
- Hanning → généralement utilisée en image
- Triangulaire
- Rectangulaire

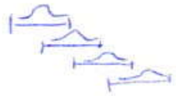
Analyse fréquentielle

✧ Fenêtre de Hamming

$$w(n) = 0{,}54 - 0{,}46\cos\left(\frac{2\pi\, n}{N-1}\right),\ 0 \le n \le N-1$$

avec N : taille de la fenêtre en échantillons avec N= 2^p





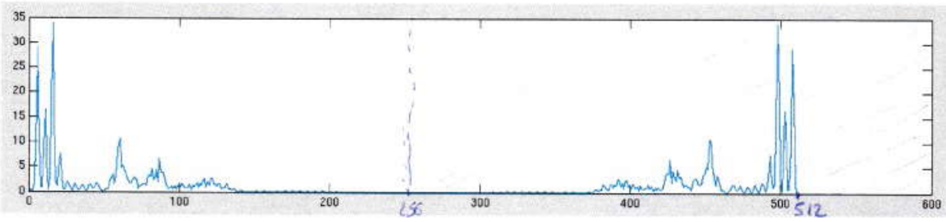
peu peu en tenir compte puisqu'il est possible que l'on coupe en plein milieu d'un mot ou d'un son

Analyse fréquentielle

✧ Spectre

$$x_n \longrightarrow \boxed{\text{Accentuation}} \longrightarrow \boxed{\text{Fenêtrage}} \longrightarrow \boxed{|\text{FFT}|} \longrightarrow \tilde{x}_n$$

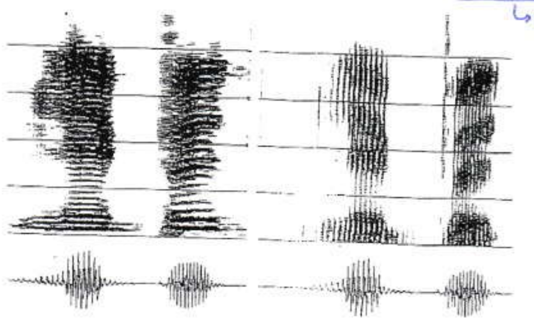
- Signal x_n
- Préaccentuation des aigus
- Fenêtrage de Hamming
- Transformée de Fourier Rapide



Analyse fréquentielle

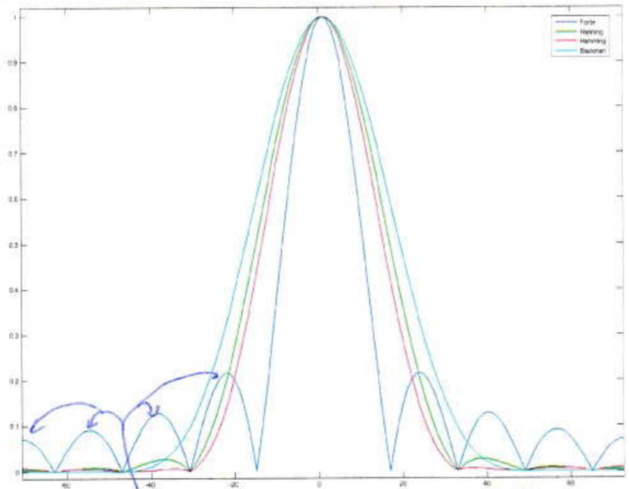
✧ Impact de la longueur de la fenêtre d'analyse

- Bande étroite : fenêtre longue ($>F_0$) → structure harmonique
- Bande large : fenêtre courte ($<F_0$) → structure formantique
  ↳ celui qu'on privilégie



Analyse fréquentielle

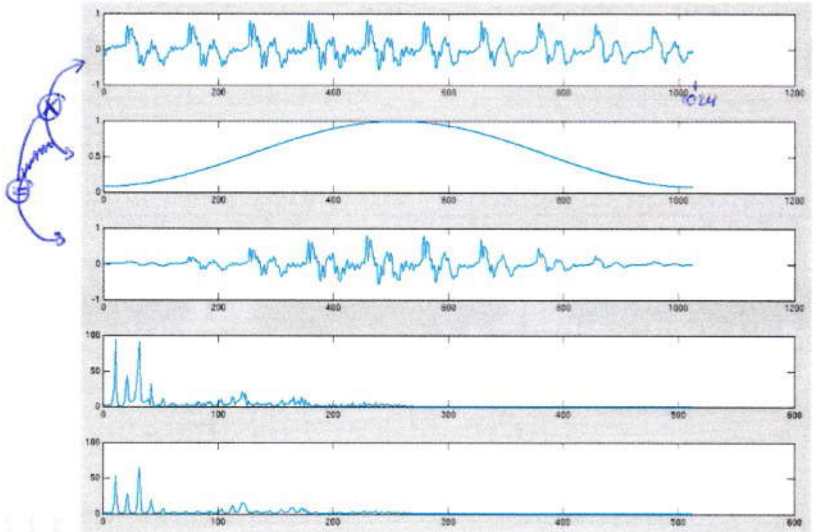
✧ Impact du type de fenêtrage



atténuer les bord de la fenêtre pour éviter les rebonds

Analyse fréquentielle

✧ Impact de la fenêtre de Hamming sur un spectre



57

Analyse fréquentielle

✧ Utilisation de la DSP (Densité Spectrale de Puissance)

- Transformée de Fourier de la fonction d'autocorrélation
- Paramètres de base

Centroïde spectral (centre de gravité)

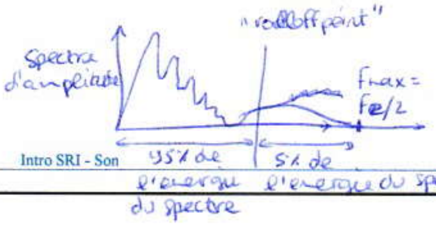
$$C(t) = \frac{\sum_{n=1}^{N} \omega_n S_t(\omega_n)}{\sum_{n=1}^{N} S_t(\omega_n)}$$

pour déterminer si c'est un signal basse freq, haute freq ...

Flux spectral (variation du spectre)

$$FS(t) = \sum_{n=1}^{N} \left(\frac{S_t(\omega_n)}{\|S_t\|} - \frac{S_{t-1}(\omega_n)}{\|S_{t-1}\|} \right)$$

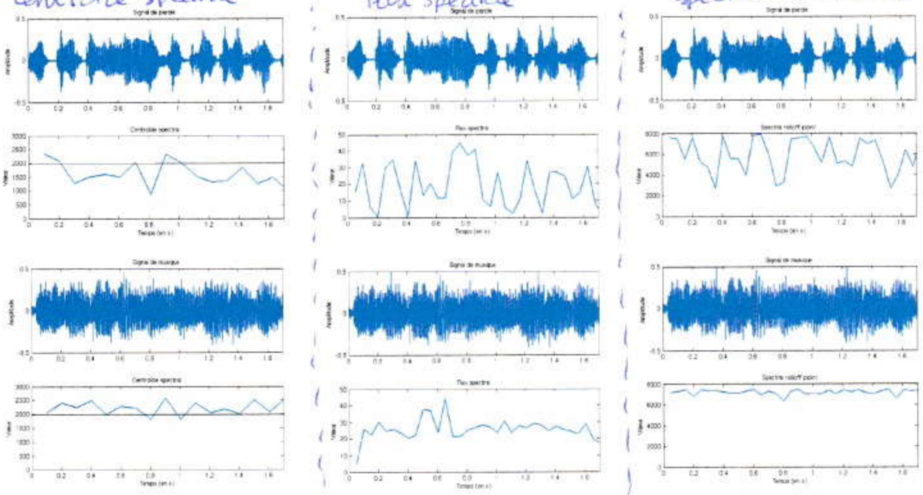
Spectral rolloff point (fréquence de coupure à 95%)



Analyse fréquentielle

✧ Application : décomposition parole / musique

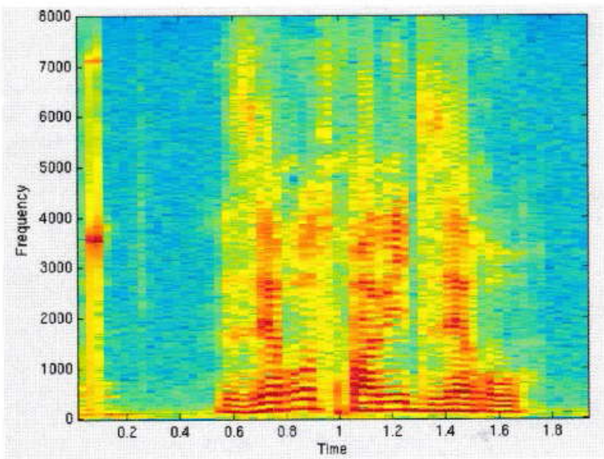
centroïde spectral — flux spectral — spectral rolloff point

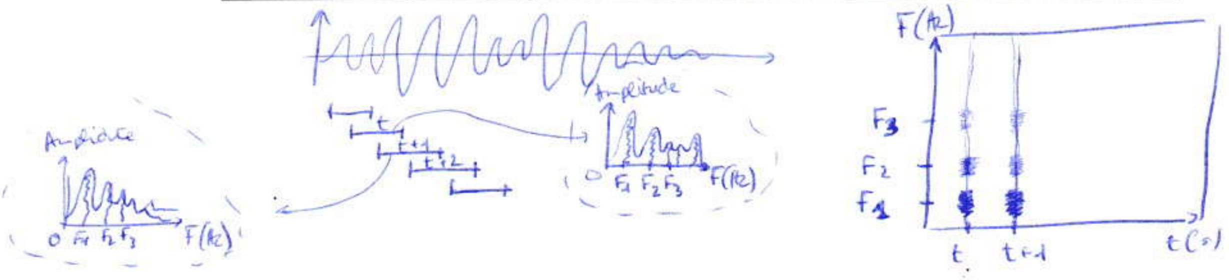


Analyse fréquentielle

✧ Spectrogramme

évolution à court terme du spectre de fréquence





Plan

- ✧ Introduction
- ✧ Paramétrisation générale
  - Analyse temporelle
  - Analyse fréquentielle
  - Le cepstre
  - Exercices
- ✧ Paramétrisation spécifique

Le cepstre

✧ MFCC

x_n → Accentuation → Fenêtrage → |FFT| → Filtrage (échelle MEL) → Log → FFT^{-1} → $\tilde{x}_n$

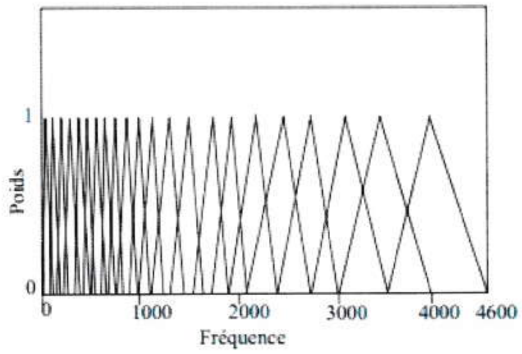
⚠ Ce n'est pas une transformée inverse

Juste pour montrer que l'on revient dans le domaine temporel

- Préaccentuation des aigus
- Fenêtrage de Hamming
- Transformée de Fourier Rapide
- Utilisation de filtres triangulaires et de l'échelle MEL
- Log
- Transformée de Fourier Rapide discrète

Le cepstre

- Echelle perceptive
  - Point de départ : la perception des sons de la parole des hautes fréquences est plus faible que celle des basses fréquences



Le cepstre

- Echelles perceptives
  - Echelle Mel : linéaire en basse fréquence, logarithmique en haute fréquence

$$M = \frac{1000}{\log(2)} \log(1 + \frac{F}{1000})$$

  - Echelle Bark : $B = 6.\text{Arcsinh}\frac{F}{600}$

24 bandes critiques (Zwicker)

Le cepstre

✧ MFCC

- Déconvolution source/conduit = transformation homomorphique

  séparer l'excitation du conduit vocal

  supprimer la fréquence fondamentale (F_0)

$$S_n = e_n \otimes c_n$$

$$\downarrow \text{FFT}$$

$$\hat{S}_n = \hat{e}_n \times \hat{c}_n \longleftarrow \text{spectre}$$

$$\downarrow \log$$

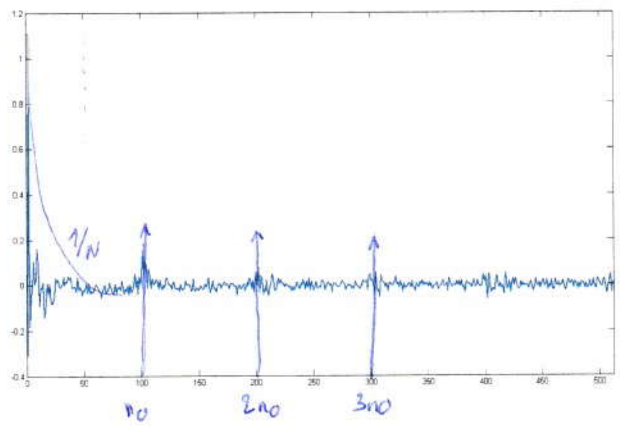
$$\log \hat{S}_n = \log \hat{e}_n + \log \hat{c}_n$$

$$\downarrow \text{FFT}^{-1}$$

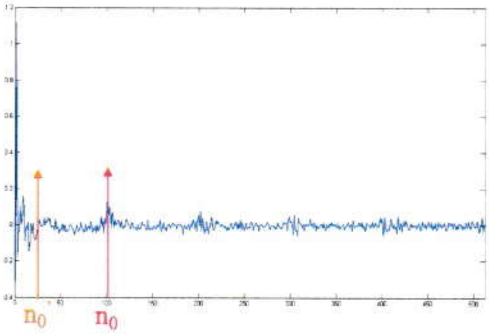
$$\hat{\hat{S}}_n = \hat{\hat{e}}_n + \hat{\hat{c}}_n \longleftarrow \text{cepstre}$$

Le cepstre

✧ Représentation du cepstre



Le cepstre



- $n_{0'} < 30 \rightarrow$ cepstre [1: $n_{0'}$]
- 1er coefficient, assimilé à l'énergie, souvent pas pris en compte

Exemple : 12 MFCC + energie Δ, $\Delta\Delta$
→ 1 vecteur de 39 coefficients par fenetre d'analyse

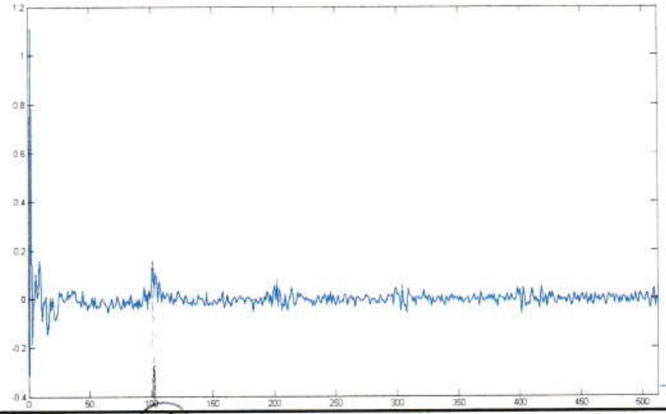
Plan

- ✧ Introduction
- ✧ Paramétrisation générale
  - Analyse temporelle
  - Analyse fréquentielle
  - Le cepstre
  - Exercices
- ✧ Paramétrisation spécifique

Exercices

✧ Exercice 1

- Calculer la F_0 du cepstre ci-dessous sachant que F_e=16 kHz et que la FFT a été réalisée sur 1024 points (fenêtre d'analyse de 1024 points)
- Homme ou femme ?
- Que vaudrait n_0 pour un enfant (F_0=450 Hz) ?



105

Exercices

✧ Exercice 1 : solution

- Calculer la F_0 de l'extrait ci-dessous sachant que F_e=16 kHz et que la FFT a été réalisée sur 1024 points (fenêtre d'analyse de 1024 points)

$n_0 = 105$ $\quad$ $F_e = 16$ kHz $= 16\,000$ points/s

$T_0 = 105/16000$

$F_0 = 1/T_0 = 16000/105 = 152$ Hz

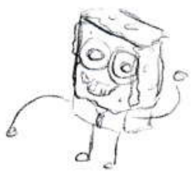
- Homme ou femme ?

Difficile à dire. Plutôt homme (cf cours précédent)

- Que vaudrait n_0 pour un enfant (F_0=450 Hz) ?

($n_0 < 30 \rightarrow$ cepstre)

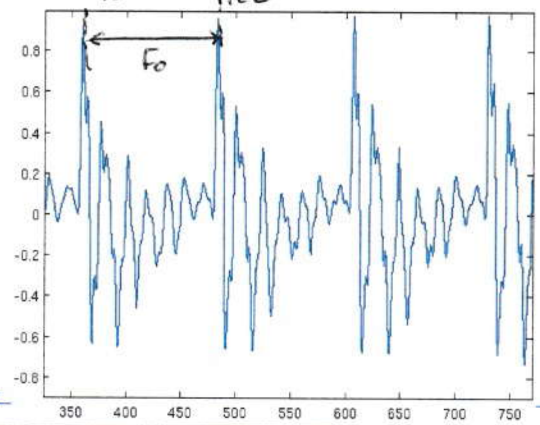
$F_0 = \frac{n_0}{F_e} \Leftrightarrow n_0 = \frac{F_e}{F_0} = \frac{16000\text{Hz}}{450\text{Hz}} = 30$



Exercices

✧ Exercice 2

- Calculer approximativement la valeur de la fréquence fondamentale en Hertz, sachant que la fréquence d'échantillonnage est de 16 kHz et que l'abscisse représente le numéro des échantillons
- Homme ou femme ? Pic 1 Pic 2



Exercices

✧ Exercice 2 : solution

- Déduire la valeur de la fréquence fondamentale en Hertz, sachant que la fréquence d'échantillonnage est de 16 kHz et que l'abscisse représente le numéro des échantillons

Par exemple Pic 1 = 360   Pic 2 = 480

Periode en points (échantillons) : 480 - 360 = 120

Periode en temps 120/16000 = 0,0075 s

Frequence : 16000/120 = 133 Hz

- Homme ou femme ?

A priori, un homme ...

Exercices

✧ Exercice 3

- Écrire une fonction (en algorithme ou en Python) qui permette de calculer l'énergie à court terme d'un signal en fonction de la taille des fenêtres d'analyse, avec un recouvrement de moitié.

```
def energie(signal, taille_fenetre):
#initialisation du vecteur résultat
    recouvrement = floor(taille_fenetre/2)
    nb_fen = floor((np.size(signal) - taille_fenetre / recouvrement) + 1
    nrj_res = np.zero(nb_fen)
#calcul de l'energie
    for fen in range(nb_fen):        navigue de (0) à (nbfen-1) par pas de 1
        p = fen * recouvrement       somme de..                    au carré
        nrj_res[fen] = np.sum((signal[p : p + taille_fenetre])**2)/taille_fenetre
    return(nrj_res)
```


Exercices

✧ Exercice 4

- En s'inspirant de l'exercice 3, écrire une fonction qui permette de calculer le taux de passage par zéro (ZCR) en fonction de la taille des fenêtres d'analyse, avec un recouvrement de moitié.

```
def ZCR(signal, taille_fenetre):
#initialisation du vecteur résultat
    recouvrement = floor(taille_fenetre/2)
    nb_fen = floor((np.size(signal) - taille_fenetre/recouvrement) + 1
    zcr_res = np.zero(nb_fen)
# calcul de la ZCR
    for fen in range(nb_fen):
        p = fen * recouvrement
        zcr_res[fen] = np.sum(np.abs(np.copysign(-1, signal[p+1 : p+taille_fenetre]
                    - n.copysign(1, signal[p+taille_fenetre -1])))/(2*taille_fenetre))
    return(zcr_res)
```

Introduction

✧ Spécificités de la musique : rappels

- Gamme 7 notes → DO RE MI FA SOL LA SI (DO)
- Octave 8 notes → ex : DO2 - DO3 et rapport de freq : 2
- Quinte 5 notes → ex : DO - SOL et rapport de freq : 3/2
- Quarte 4 notes → ex : DO - FA et rapport de freq : 4/3
- L'étalon ➔ le « LA3 » : 440 Hz

Introduction

✧ Spécificités de la musique : rappels

- Oreille humaine capte 10 octaves
- Gamme tempérée octave divisée en 12 intervalles (demi-tons) demi-ton espacé de $2^{1/12} = 1,0595$
- Rapports de fréquences des notes

do	do#	ré	ré#	mi	fa	fa#	sol	sol#	la	la#	si	do
1	$2^{1/12}$	$2^{1/6}$	$2^{1/4}$	$2^{1/3}$	$2^{5/12}$	$\sqrt{2}$	$2^{7/12}$	$2^{2/3}$	$2^{3/4}$	$2^{5/6}$	$2^{11/12}$	2

Gamme tempérée

					Rapport d'une note à l'autre	R
				Équivalent du physicien	P	
Note			Fréquence			
do⁷			4186,03		2^{12}	
do⁶			2093,02		2^{11}	
do⁵			1046,51		2^{10}	
	ré⁴			587,33		
do⁴			522,25		2^{9}	1,0595
	si³			493,88		1,0595
		si b³		466,16		1,0595
	la³			440,00		1,0594
		la b³		415,31		1,0595
	sol³			392,00		1,0595
		fa #³		369,99		1,0594
	fa³			349,23		1,0595
	mi³			329,63		1,0595
		mi b³		311,13		1,0595
	ré³			293,66		1,0595
		do #³		277,18		1,0594
do³			261,63		2^{8}	
do²			130,81		2^{7}	
do¹			65,41		2^{6}	
do⁰			32,70		2^{5}	
do⁻¹			16,35		2^{4}	
do⁻²			8,15		2^{3}	

Exercices

✧ Exercice 3

- Écrire une fonction (en algorithme ou en Python) qui permette de calculer l'énergie à court terme d'un signal en fonction de la taille des fenêtres d'analyse, avec un recouvrement de moitié.

```
def energie(signal, taille_fenetre):
 # initialisation du vecteur résultat
    recouvrement = floor(taille_fenetre/2)
    nb_fen = floor((np.size(signal) - taille_fenetre/recouvrement) + 1
    nrj_res = np.zero(nb_fen)      navigue de (0) à (nb fen -1) par pas de 1
 # calcul de l'énergie
    for fen in range(nb_fen):
       p = fen * recouvrement      somme de..                        au carré
       nrj_res[fen] = np.sum((signal[p : p + taille_fenetre])**2)/taille_fenetre
 return(nrj_res)
```


Exercices

✧ Exercice 4

- En s'inspirant de l'exercice 3, écrire une fonction qui permette de calculer le taux de passage par zéro (ZCR) en fonction de la taille des fenêtres d'analyse, avec un recouvrement de moitié.

```
def ZCR(signal, taille_fenetre):
 # initialisation du vecteur résultat
    recouvrement = floor(taille_fenetre/2)
    nb_fen = floor((np.size(signal) - taille_fenetre/recouvrement) + 1
    zcr_res = np.zero(nb_fen)
 # calcul de la ZCR
    for fen in range(nb_fen):
       p = fen * recouvrement
       zcr_res[fen] = np.sum(np.abs(np.copysign(-1, signal[p+1 : p+taille_fenetre]
                - n.copysign(1, signal[p+taille_fenetre - 1])))/(2*taille_fenetre)
 return(zcr_res)
```

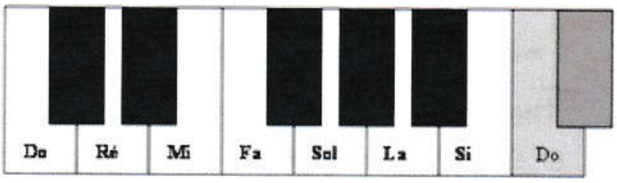
Introduction

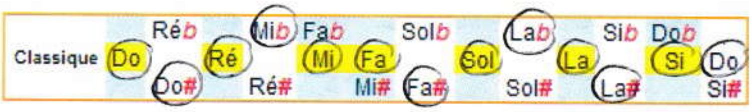
✧ Spécificités de la musique : rappels

- Nomenclature

A = LA B = SI C = DO D = RE E = MI F = FA et G = SOL

- Exemple : piano





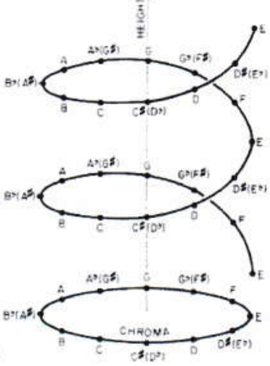
Introduction

✧ Spécificités de la musique

- En musique, deux dimensions : note et octave

Exemple : MIDI A3 désigne un LA (A), 3e octave. Une sinusoïde de fréquence 440Hz donne une sensation de hauteur équivalente

- Représentation sous forme d'hélice Shepard [She64]



Introduction

✧ Spécificités de la musique : lien entre la fréquence F et la note de musique H

- Fréquence F exprimée en Hertz (Hz) -- échelle linéaire
- Hauteur perçue H -- échelle logarithmique

$$F = F_0 2^{\frac{H-H_0}{O}} \qquad H = H_0 + O\log_2\left(\frac{F}{F_0}\right)$$

avec H_0 et F_0 les hauteurs et les fréquences de référence

$F_{ref} = F_0 = 440$ Hz et $H_{ref} = H_0 = 57$

$O = 12$: nombre de demi-tons par octave

Exemple : $F = 880$ Hz ⇒ $H = 69$

$F \times 2$ ⇒ octave supérieure

$F/2$ ⇒ octave inférieure

Introduction

✧ Spécificités de la musique : variations de fréquence

- à l'échelle macroscopique : mélodie
- à l'échelle microscopique :

variations linéaires : glissando ou portamento

variations périodiques (< 20 Hz) : vibrato

variations périodiques plus rapides : rajout de composants spectraux

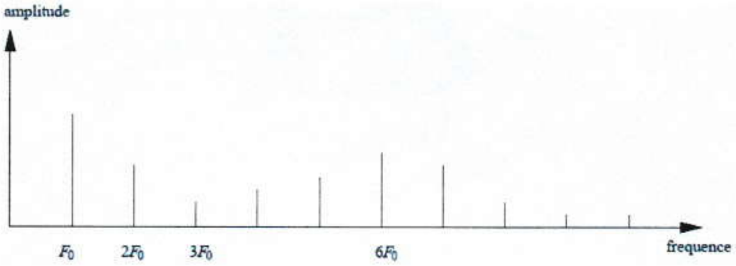
→ synthèse par modulation de fréquence (FM)

Introduction

- Spécificités de la musique : spectre
  - fréquences composant le son **régulièrement** espacées

→ son harmonique de fondamentale F_0

F_0 : fréquence **fondamentale** (ou première harmonique)
$k.F_0$: $k^{ème}$ harmonique



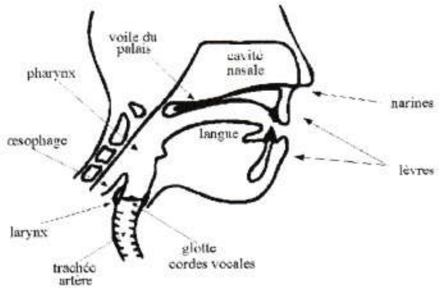
Plan

- Introduction
- Paramétrisation générale
- Paramétrisation spécifique
  - Introduction
  - Fréquence fondamentale
  - Constant-Q Transform
  - Chromas

Fréquence fondamentale

✧ Définition en parole

- F_0 : fréquence fondamentale de vibration des cordes vocales



Fréquence fondamentale

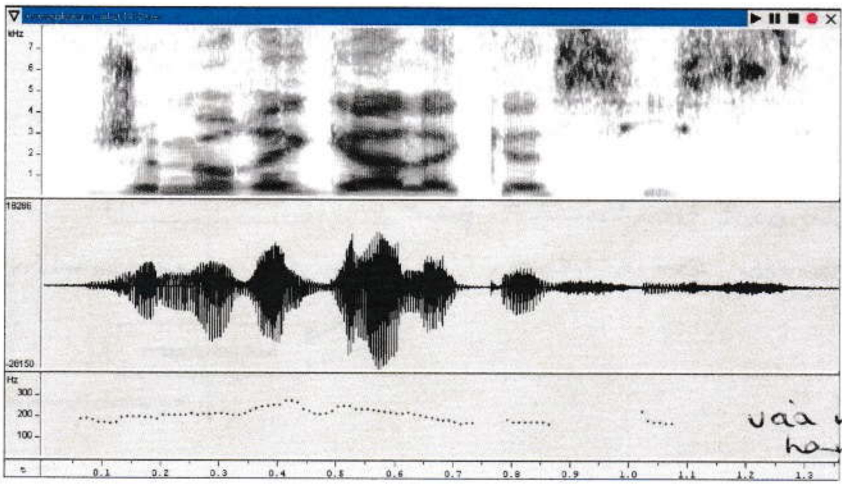
✧ Définition

- Correspond à la hauteur de la voix (mélodie)
- Intonation = variation de temps de ~~la~~ Fo
- Plage de valeur (en Hz) :
  80-250 : homme
  200-400 : femme
  > 400 : enfants

Liée à la taille des cordes vocales

Fréquence fondamentale

✧ Définition



va à monotone
homme ? / femme ?
=> entre les deux...

Fréquence fondamentale

✧ Intérêts et fonctions

- Liée à l'accentuation : ~~Détection~~ Distinction entre homonymes :
"segment" (nom) en EN → "segment" (verbe)

- De structuration
phrase, discours : "La petite brise la glace"

- Modale
Déclaration, interrogation, injonction, exclamation : "tu vas bien"

- Expressive
Manifestation d'un sentiment
Attitude

Fréquence fondamentale

✧ Extraction : principe général

Un prétraitement acoustique et un changement de représentation

L'extraction "brute" du fondamental

Un post traitement pour corriger les erreurs

Signal

PRETRAITEMENT

Representation du signal

EXTRACTION DU F0

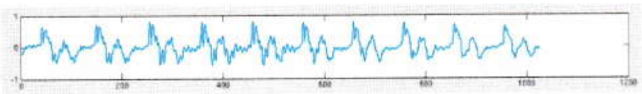
Estimation brute de f0

POST-TRAITEMENT

Estimation finale de f0

Fréquence fondamentale

✧ Extraction



- Autocorrélation [Hess83]

Algorithme de type « corrélation »

Hypothèse : le signal est stationnaire

$$FP_{autoc} = \frac{1}{N} \sum_{n=0}^{N-1-\tau} s(n)\, s(n+\tau)$$

avec τ délai ou retard

$\max \tau = T_0$

Fréquence fondamentale

✧ Extraction

- Fonction de distance (AMDF) [Miller et Weibel 56]

  critère de variation d'amplitude à court terme (Average Magnitude Difference Function)

$$FP_{AMDF}(\tau) = \sum_{n=0}^{N-1-\tau} |s(n) - s(n+\tau)|$$

Ambiguïté entre pics T0, 2*T0… atténuée par la non stationnarité du signal

Résiste aux erreurs grossières

Rapidité de calcul

Fréquence fondamentale

✧ Extraction

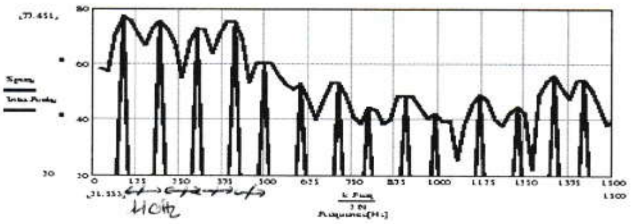
car dent espacé d'une distance quasi-constante

- Peigne spectral [Martin 81]

intercorrélation entre le spectre S du signal et un « peigne »
max $FP \rightarrow F_0$

$$FP_{peigne}(f) = \sum_{i=1}^{n(f)} \alpha_i |S(i * f)|$$

Exemple
f = 110 Hz



Fréquence fondamentale

✧ Liens vers programmes classiques

- MES SignAix
  http://www.lpl.univ-aix.fr/ext/projects/mes_signaix.htm
- Snack/Wavesurfer
  http://www.speech.kth.se/wavesurfer/
- Praat
  http://www.fon.hum.uva.nl/praat/
- Modélisation automatique de la fréquence fondamentale F0 (MOMEL)
  http://www.icp.inpg.fr/~rolland/mv_work/momel_french.html
- Snorri/WinSnorri
  http://www.loria.fr/~laprie/

YIN https://audition.ens.fr/adc/ (marche mieux)

Fréquence fondamentale

✧ Limite : la polyphonie (plusieurs notes jouées en même temps)

- Les méthodes précédentes sont dédiées à la monophonie
- Problème lié à la séparation de sources
- Problème très difficile
  Sources sonores variées
  Intervalle de notes possibles important
  Musique : présence de batterie, de bruit
- Prise en compte du contexte tonal local

→ chromas ou Pitch Class Profile

Plan

✧ Introduction

✧ Paramétrisation générale

✧ Paramétrisation spécifique

- Introduction
- Fréquence fondamentale
- Constant-Q Transform
- Chromas

Constant-Q Transform

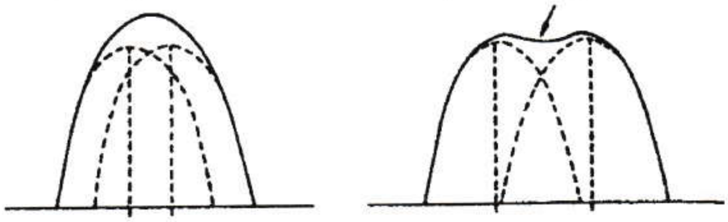
✧ Signal de musique : choix de la fenêtre d'analyse est prépondérant !

- Fenêtre courte (bonne résolution temporelle)

→ Détecter les changements rapides du signal (attaques de note)

- Fenêtre grande

Distinguer les sinusoïdes proches (hauteurs voisine)



Constant-Q Transform

✧ Illustration

- Soit un signal de musique avec 2 sinusoïdes de fréquences f_1 et f_2 qui correspondent à 2 notes adjacentes
- Notons $\Delta f = f_2 - f_1$
- Pour la musique occidentale (échelle logarithmique), les fréquences correspondant à 2 notes adjacentes sont d'autant plus proches que l'on se rapproche des basses fréquences
- 2 notes adjacentes (correspondant à un demi-ton) sont séparées par 6 % de la fréquence de la note la plus basse

Exemple : $f_{DO1} = 32{,}7\,Hz$ et $f_{DO\#1} = 32{,}7\,Hz + \overbrace{(0{,}06 * 32{,}7)}^{6\%} = 34{,}6\,Hz$

Constant-Q Transform

✧ Solution : analyse fréquentielle multi-résolution

- Résoudre le dilemme résolution temps/fréquence
- FFT (transformée en résolution fixe)

Fréquences espacées de manière égale (linéaire)
⇔ résolution fréquentielle constante

- Analyse fréquentielle multi-résolution

⇒ Spectre de fréquences divisée en sous-bandes et chacune est traitée de manière indépendante
↳ fenêtre courtes pour les freq élevée
↳ fenêtre longues pour les freq ~~faible~~ basse

Constant-Q Transform

✧ Transformée en constante Q (Constant-Q Transform – CQT)

- Proposée par Brown en 1991

- Canaux fréquentiels pas linéairement espacés (comme FFT)

MAIS géométriquement espacés : $Q = \frac{f}{\Delta f}$ est constant

avec f la fréquence centrale et Δf la résolution

Freq centrales ⟶ notes de musique occidentales

- Résolution temporelle augmente avec la fréquence

  Basses fréquences : grande taille de fenêtre

  Quand Fréquence augmente ➔ taille de fenêtre diminue

Constant-Q Transform

✧ Transformée en constante Q (Constant-Q Transform – CQT)

- Formule :

$$X(k) = \sum_{n=0}^{N(k)-1} w(n,k)\, x(n)\, e^{-j2\pi f_k n}$$

avec :
- $X(k)$ la k[ième] composante de la CQT,
- $x(n)$ le signal,
- $w(n,k)$ la fenêtre d'analyse

La longueur de $w(n,k)$ à la fréquence f_k est : $N(k) = \frac{Q \cdot f_e}{f_k}$

avec $Q = \frac{f_k}{\Delta f_k}$ et f_e la fréquence d'échantillonnage

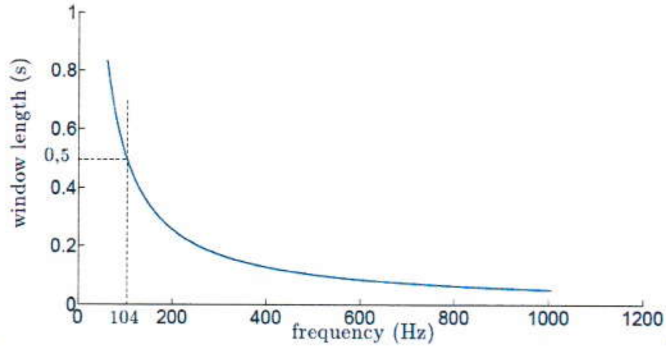
Constant-Q Transform

✧ Transformée en constante Q (Constant-Q Transform – CQT)

- Illustration

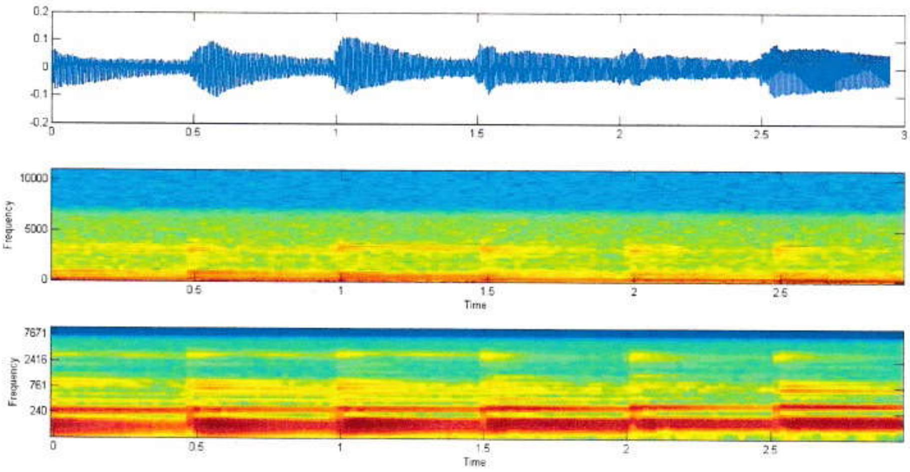
Taille de la fenêtre N (en secondes) en fonction de la fréquence (en Hertz) pour un espacement d'un demi-ton $(Q = 2^{12}-1)$

Exemple : 104Hz ⟷ 0,5s



Constant-Q Transform

✧ Transformée en constante Q (Constant-Q Transform – CQT)



Chromas

✧ Méthode

- Le spectre est divisé en 12 classes (12 demi-tons de la gamme)

Le i^e coef du chroma vector correspond à l'énergie dans les bandes de freq correspondant à la i^e note de la gamme, à toutes les octaves possibles

- Intensité des 12 demi-tons

$$M(f_k) = \text{round}\left(12 \log\left(\frac{f_k}{f_{ref}} \bmod 12\right)\right)$$

- Vecteur de chromas (12 dimensions)

$$PCP(n) = \sum_{k\,:\,M(f_k)=n} |X_k|^2$$ avec ou sans le carré

- Normalisation possible

Chromas

✧ Chromagram

- Affichage compact de la représentation spectrale (FFT ou CQT)
- Exemple sur la gamme