

Signaux, Sons et Images pour l'Informaticien: Introduction

Diane Lingrand

Polytech SI3

2018 - 2019

- Sons

- Introduction au son (aujourd'hui)
- Transformée de Fourier - Reconstruction - Shannon
- Filtres et bancs de filtres
- Compression du son - MP3
- Description d'un son - Vers la reconnaissance
- TP de reconnaissance de sons

- Images

- Introduction aux images
- Echantillonnage et transformations 2D
- Détection de contours et points d'intérêt
- Bruit et restauration d'images
- Compression d'images
- TP de reconnaissance d'images

- 1 Physique du son
- 2 Représentation du son
- 3 Echantillonnage et Quantification
- 4 Formats de fichiers audio
- 5 Travaux Pratiques
- 6 Reconnaître un chronogramme

Outline

1 Physique du son

2 Représentation du son

3 Echantillonnage et Quantification

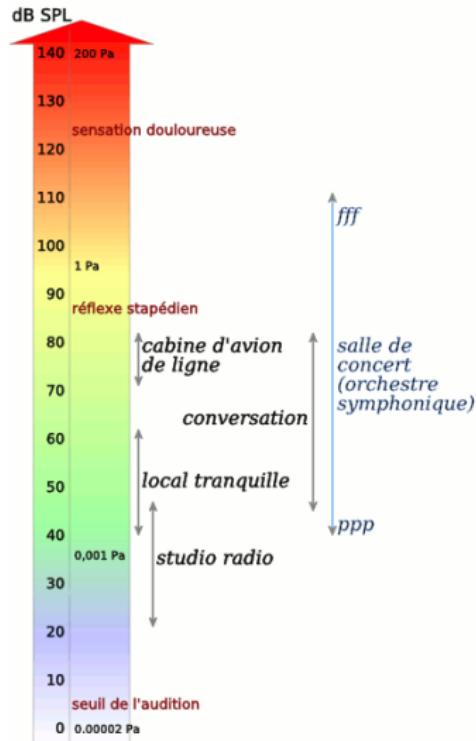
4 Formats de fichiers audio

5 Travaux Pratiques

6 Reconnaître un chronogramme

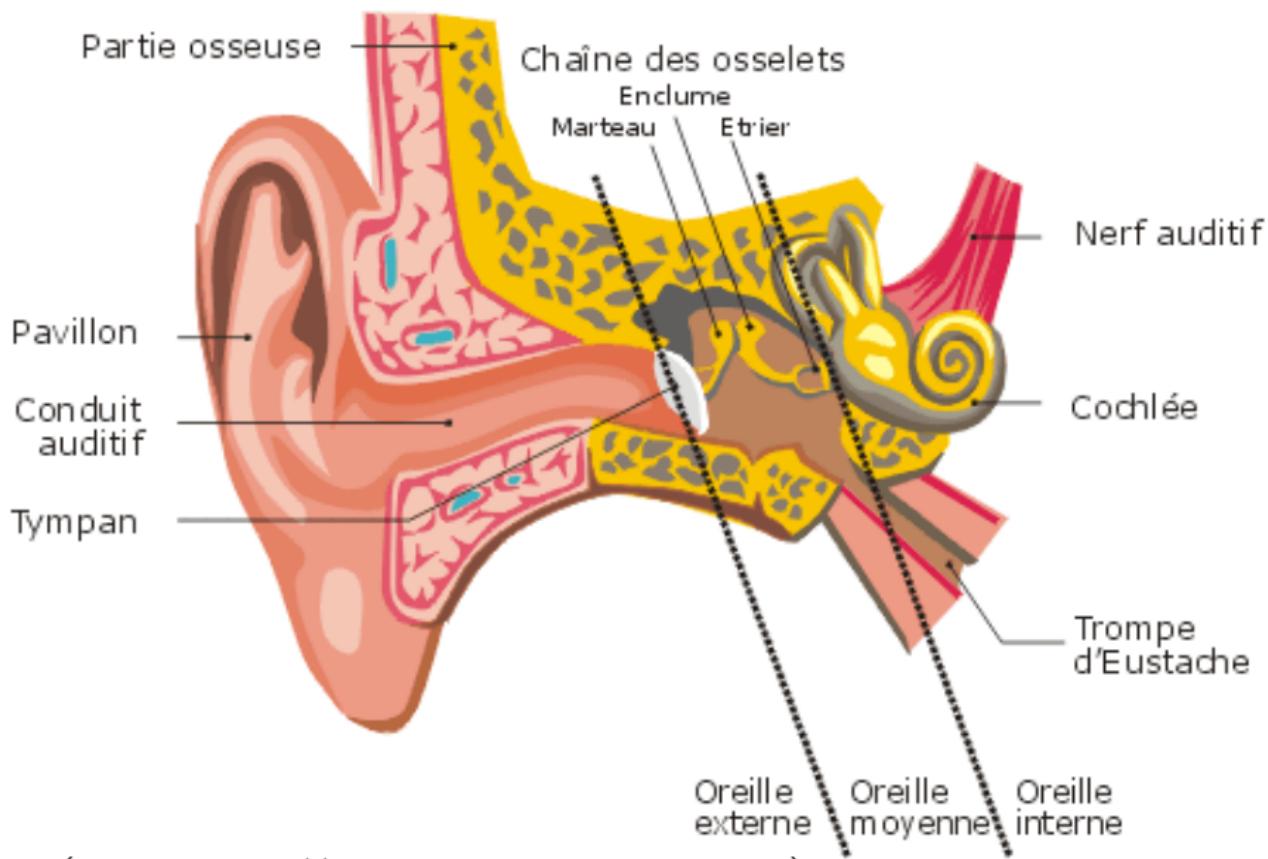
Le son

- variations de pression de l'air
- vitesse de propagation dans l'air : 340 m/s (1 224 km/h)
- seuil de perception auditive : $10^{-12} W/m^2$
- intensité : $10 \log \frac{P}{P_{th}}$ en dB : de 0 à 120 dB
- périodique, quasi-périodique ou non
- longueur d'onde : distance parcourue pendant un cycle
- fréquences audibles par un humain : de 20 Hz à 20 kHz



(source : wikipedia.org)

Oreille



(source : <http://claude.benoitgonin.free.fr>)

Oreille-cerveau auditif : R. Pujol, S. Blatrix



Outline

1 Physique du son

2 Représentation du son

3 Echantillonnage et Quantification

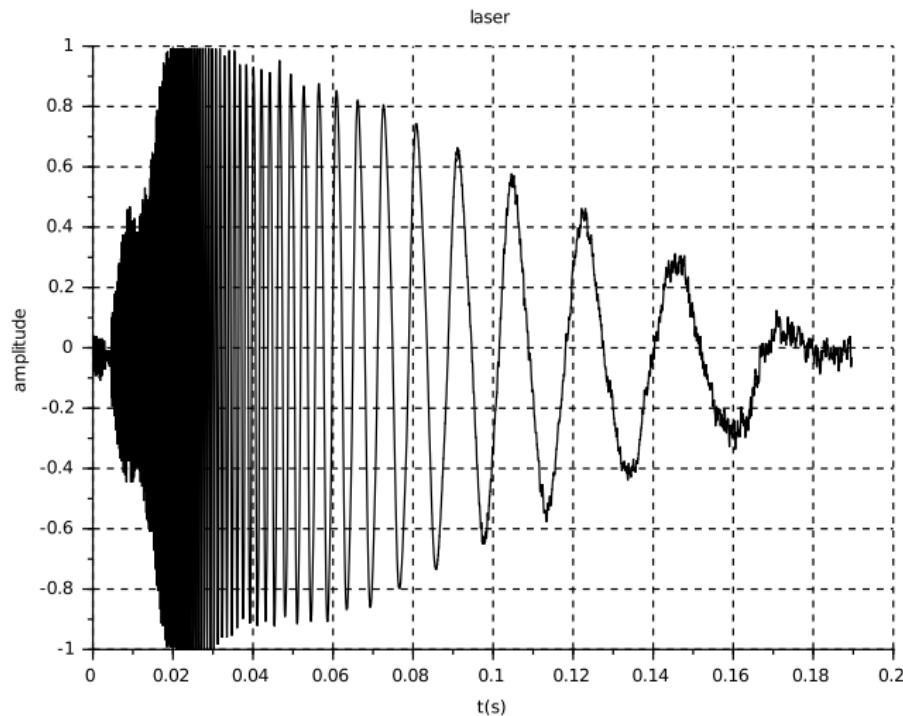
4 Formats de fichiers audio

5 Travaux Pratiques

6 Reconnaître un chronogramme

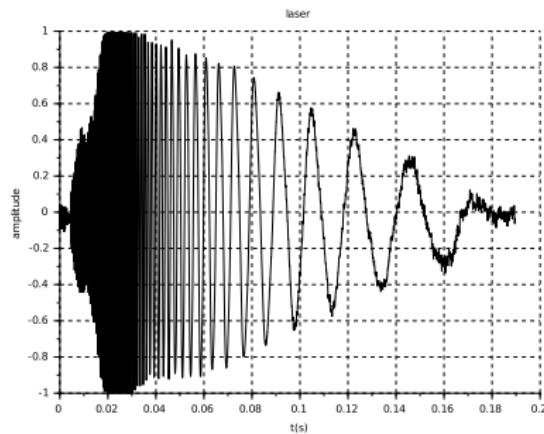
Chronogramme

Représentation temporelle



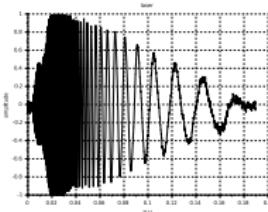
[ext]

Chronogramme (scilab)



```
[s,fe,b]=wavread('laser.wav');
t=[0:length(s)-1]/fe
plot2d(t,s)
xgrid
xtitle('laser','t(s)', 'amplitude')
```

Chronogramme (python)



```
import matplotlib.pyplot as plt
import numpy as np
import librosa.display
import librosa
x, fe = librosa.load('laser.wav')
plt.figure(figsize=(14, 5))
librosa.display.waveplot(x, sr=fe)
% default is mono. In case of stereo:
% librosa.load('fichierStereo.wab', mono=False)
plt.title('laser')
plt.show()
% n=len(x)
% t = np.linspace(0, n/fe, n, endpoint=False) %endpoint not :
% plot(t,x)
```

Harmoniques, spectre

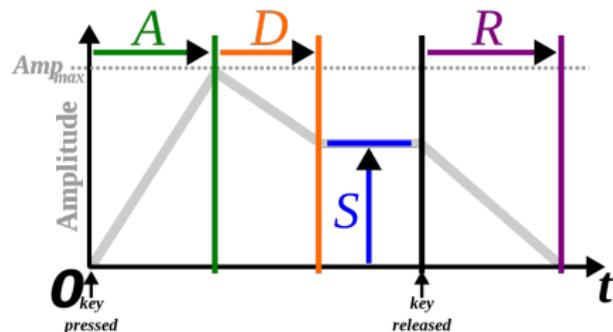
fondamentale : fréquence du signal périodique, note

harmonique : multiple entier de la fondamentale

formant : maxima des composantes fréquentielles du signal

spectre : ensemble des composantes fréquentielles du signal

enveloppe : évolution de l'intensité en fonction du temps (Attack Decay Sustain Release)



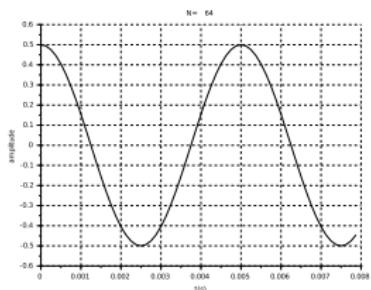
(source : wikipedia.org)

En combinant la composition fréquentielle et l'enveloppe, on peut synthétiser des sons.

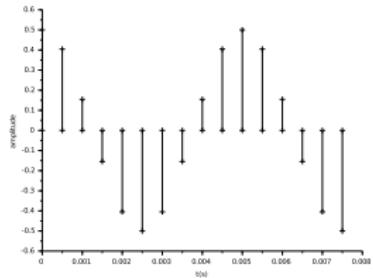
Outline

- 1 Physique du son
- 2 Représentation du son
- 3 Echantillonnage et Quantification
- 4 Formats de fichiers audio
- 5 Travaux Pratiques
- 6 Reconnaître un chronogramme

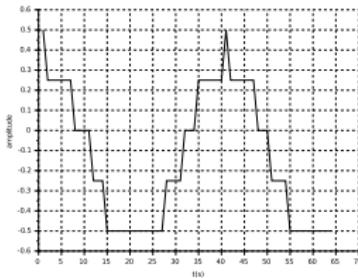
Echantillonnage et Quantification



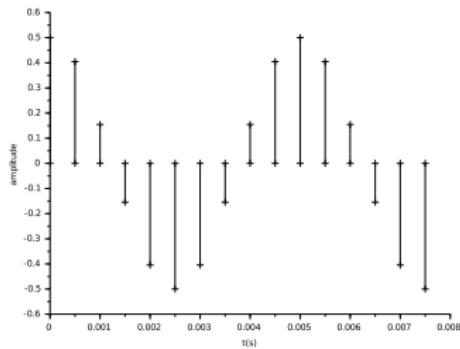
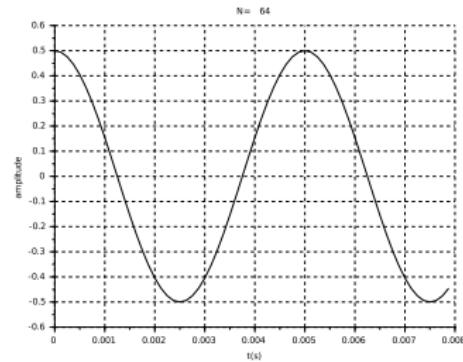
Echantillonnage : discréétisation du temps. Echantillons temporels.



Quantification : discréétisation des valeurs du signal.



Echantillonnage



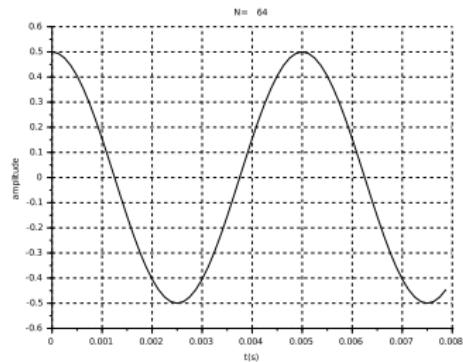
période d'échantillonnage : T_e

fréquence d'échantillonnage : $f_e = \frac{1}{T_e}$

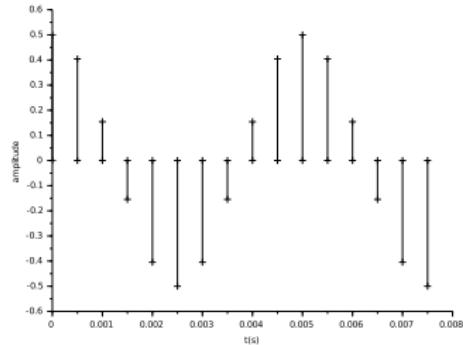
signal échantillonné : $s_e = \{s(nT_e) = s(\frac{n}{f_e}) = s_n, n = 0 \dots N - 1\}$

contrainte de Nyquist Shannon : $f_e \geq 2f_{max}$ où f_{max} est la fréquence maximale du signal

Echantillonnage (scilab)

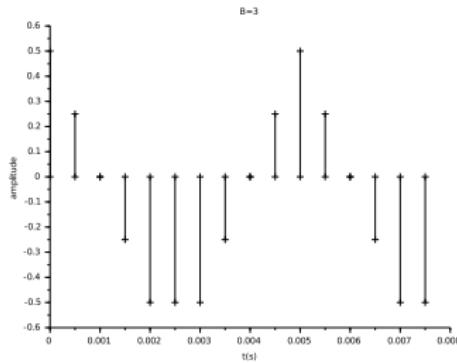
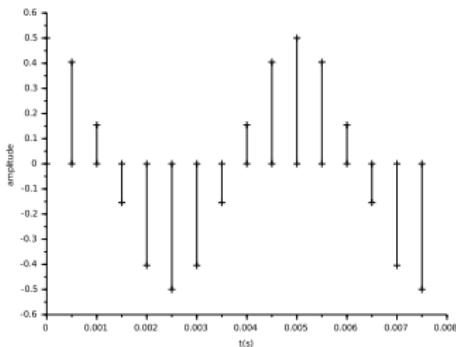


```
n=64  
te=1/8000.0  
t = np.linspace(0, n*te, n,\  
endpoint=False)  
s=0.5*np.cos(2*np.pi*200*t)  
plt.plot(t,s)
```



```
t2=t[::4]  
se=s[::4]  
plt.plot(t2,se)
```

Quantification



coder sur B bits : découper l'intervalle $[-1 : +1]$ en 2^B intervalles

pas de quantification : $Q = \frac{2}{2^B} = 2^{1-B}$

signal codé : $sb = \{sb(nT_e) = sb_n = E(\frac{s_n}{Q}), n = 0 \dots N - 1\}$

signal quantifié : $sq = \{sq = Q * E(\frac{s_n}{Q}), n = 0 \dots N - 1\}$

erreur ou bruit de quantification : $b = \{b(nT_e) = s_n - sq_n, n = 0 \dots N - 1\}$

rapport signal sur bruit (en dB) : $SNR = \frac{\text{écart type}(se)}{\text{écart type}(b)}$

Sinusoïde de fréquence 440 Hz (la)

B=1 ; 2 intervalles[ext]

B=2 ; 4 intervalles[ext]

B=3 ; 8 intervalles[ext]

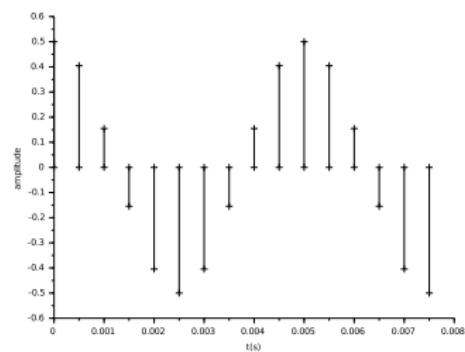
B=4 ; 16 intervalles[ext]

B=5 ; 32 intervalles[ext]

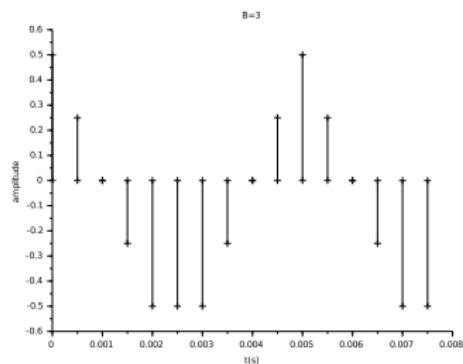
B=8 ; 256 intervalles[ext]

Echantillonnage - Quantification (scilab)

Echantillonnage



Quantification



```
t2=t[::4]  
se=s[::4]  
plt.plot(t2,se)
```

```
seq=0.25*np.floor(se*4)  
plt.plot(t2,seq)
```

Exemples de sous-échantillonnage et quantification

Son de référence : cello_G2_15_fortissimo_arco-normal.mp3[ext] du site http://www.philharmonia.co.uk/explore/sound_samples/ (dont on a extrait un seul canal), initiallement quantifié sur 16 bits et de fréquence d'échantillonnage 44100 Hz.

- Sous-échantillonnage

- 1 échantillon sur 2[ext]
- 1 échantillon sur 4[ext]
- 1 échantillon sur 8[ext]
- 1 échantillon sur 16[ext]
- 1 échantillon sur 32[ext]

- Quantification

- B=8[ext]
- B=7[ext]
- B=6[ext]
- B=5[ext]
- B=4[ext]
- B=3[ext]
- B=2[ext]
- B=1[ext]

- N nombre d'échantillons
- D durée du signal
- f_e fréquence d'échantillonnage
- B nb. bits pour coder un échantillon
- taille : $D * f_e * B * \text{nb. voies}$
- débit binaire (*bit rate*) : $f_e * B * \text{nb. voies}$
- pour compresser : réduire D , B , f_e ou le nombre de voies
- taux de compression : rapport nouvelle taille sur ancienne taille

Outline

- 1 Physique du son
- 2 Représentation du son
- 3 Echantillonnage et Quantification
- 4 Formats de fichiers audio
- 5 Travaux Pratiques
- 6 Reconnaître un chronogramme

Formats de fichiers audio

- sans compression
 - PCM (linear Pulse Code Modulation) - WAV AIFF AU
 - PDM (Pulse Density Modulation) - SA-CD
- avec compression
 - sans pertes : ALAC, FLAC,
 - avec pertes : Ogg Vorbis, MP3, WMA, AAC

Format Waveform Audio File Format (WAVE)

little endian

[Bloc de declaration d'un fichier au format WAVE]

FileTypeBlocID (4 octets) : Constante RIFF (0x52,0x49,0x46,0x46)
FileSize (4 octets) : Taille du fichier (sans les 8 premiers octets)
FileFormatID (4 octets) : Format = WAVE (0x57,0x41,0x56,0x45)

[Bloc decrivant le format audio]

FormatBlocID (4 octets) : Identifiant fmt (0x66,0x6D, 0x74,0x20)
BlocSize (4 octets) : Nombre d'octets du bloc - 16 (0x10)

AudioFormat (2 octets) : Format du stockage (1: PCM, ...)
NbrCanaux (2 octets) : Nb. de canaux (de 1 a 6)
Frequence (4 octets) : Freq. d'echant. (Hz) [11025, 22050, 44100, ...]
BytePerSec (4 octets) : Nb. octets a lire par s. (Freq. * BytePerBloc).
BytePerBloc (2 octets) : Nb. octets par bloc d'echant. (NbCanaux * BitsPerSample/8).
BitsPerSample (2 octets) : Nb. de bits pour chaque echant. (8, 16, 24)

[Bloc des donnees]

DataBlocID (4 octets) : Constante data (0x64,0x61,0x74,0x61)
DataSize (4 octets) : Nb. octets des donnees (tailleFichier - tailleEntete)
DATAS[] : [Sample 1 du Canal 1] [Sample 1 du Canal 2] [Sample 2 du Canal 1] ...

```
> hd laser.wav | head -n 5
00000000  52 49 46 46 fc 20 00 00  57 41 56 45 66 6d 74 20
00000010  10 00 00 00 01 00 01 00  22 56 00 00 44 ac 00 00
00000020  02 00 10 00 64 61 74 61  ac 20 00 00 00 06 00 f4
00000030  00 00 00 01 00 f6 00 01  00 fd 00 fb 00 05 00 f9
00000040  00 f8 00 03 00 f9 00 f7  00 ff 00 f8 00 fc 00 01
```

Taille du fichier laser.wav en octets : $8452 = \underbrace{32 * 256 + 15 * 16 + 2 + 8}_{fc\ 20\ 00\ 00}$

Outline

1 Physique du son

2 Représentation du son

3 Echantillonnage et Quantification

4 Formats de fichiers audio

5 Travaux Pratiques

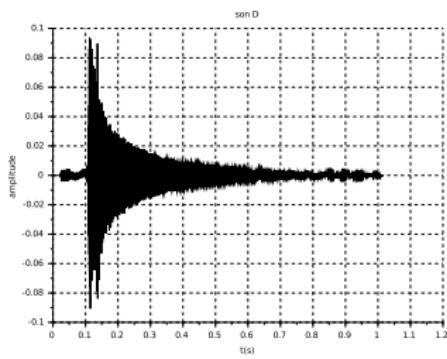
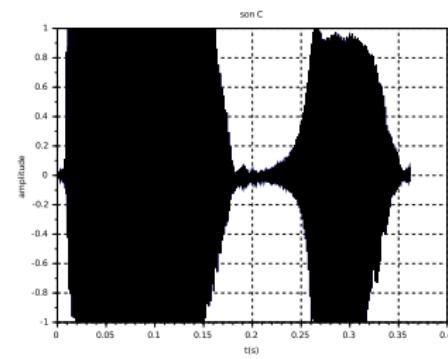
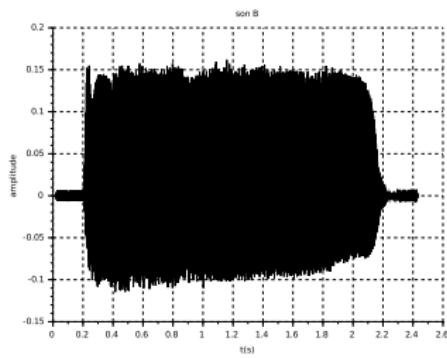
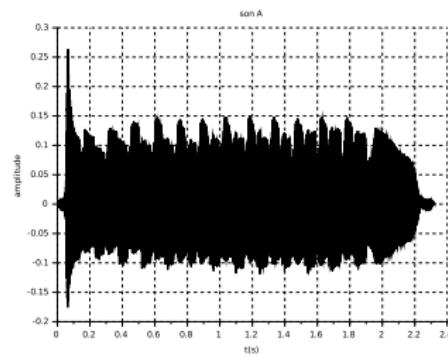
6 Reconnaître un chronogramme

- Outils pour les travaux pratiques :
 - Python 3 avec les librairies :
 - numpy (`pip3 install numpy`)
 - librosa (`pip3 install numpy`)
 - IPython (`pip3 install ipython`), aussi inclu dans Jupyter (`pip3 install Jupyter`)
 - Anciennement logiciel Scilab (Python 3 avec les librairies). A installer sur système natif (Linux, Windows ou Mac)
- Notation :
 - 1/3 en contrôle continu : compte-rendu de TPs, tests en TD ou amphi, participation
 - 1/3 DS à mi-semestre
 - 1/3 DS final

Outline

- 1 Physique du son
- 2 Représentation du son
- 3 Echantillonnage et Quantification
- 4 Formats de fichiers audio
- 5 Travaux Pratiques
- 6 Reconnaître un chronogramme

Ecoutez les quatre extraits et retrouvez le chronogramme



- 1  [ext]
- 2  [ext]
- 3  [ext]
- 4  [ext]