

Isolation acoustique en ville à l'aide de métamatériaux



Présentation

- Problématique étudiée :
- L'isolation d'une rue longeant une voie ferrée est-elle possible a l'aide d'un réseau de résonateurs ?
- Plan :
- **0- Une étude théorique autour du train**
- **I- Autour des résonateurs de Helmholtz**
 - a) Le modèle de Helmholtz
 - b) Une première application du modèle
 - c) Une seconde application
- **II- Mise en place d'un réseau de résonateurs**
 - a) Le modèle du réseau de Bragg
 - b) Création d'un réseau de résonateurs
- **III- Fabrication d'un gradient de résonateurs**
- **IV- Conclusion et interprétation des résultats**

Introduction

- On appelle « méta-matériau » un matériau artificiel structuré qui présente certaines propriétés vis-à-vis de la propagation des ondes électromagnétiques ou des propriétés mécaniques non intuitives ainsi que des propriétés acoustiques auxquelles nous nous intéresserons particulièrement.

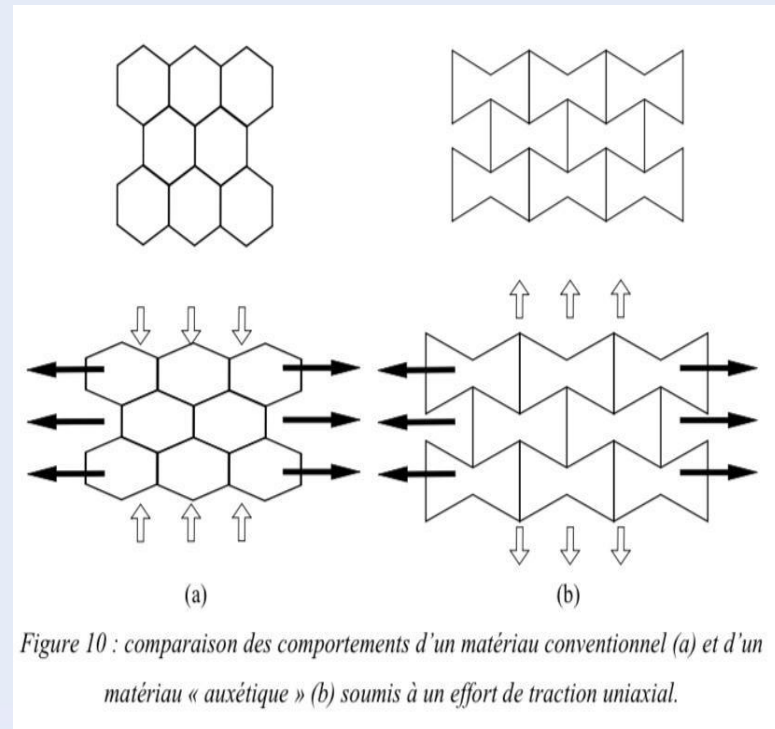
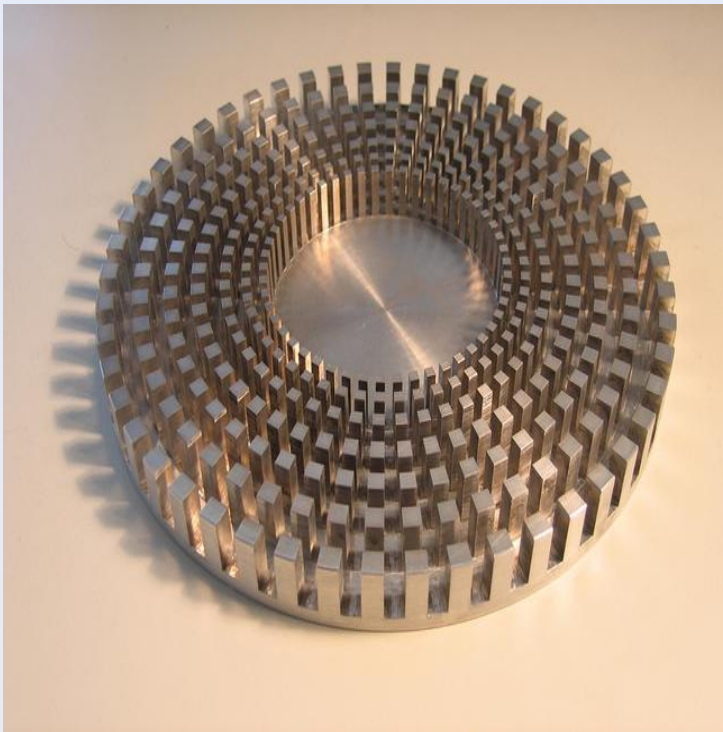
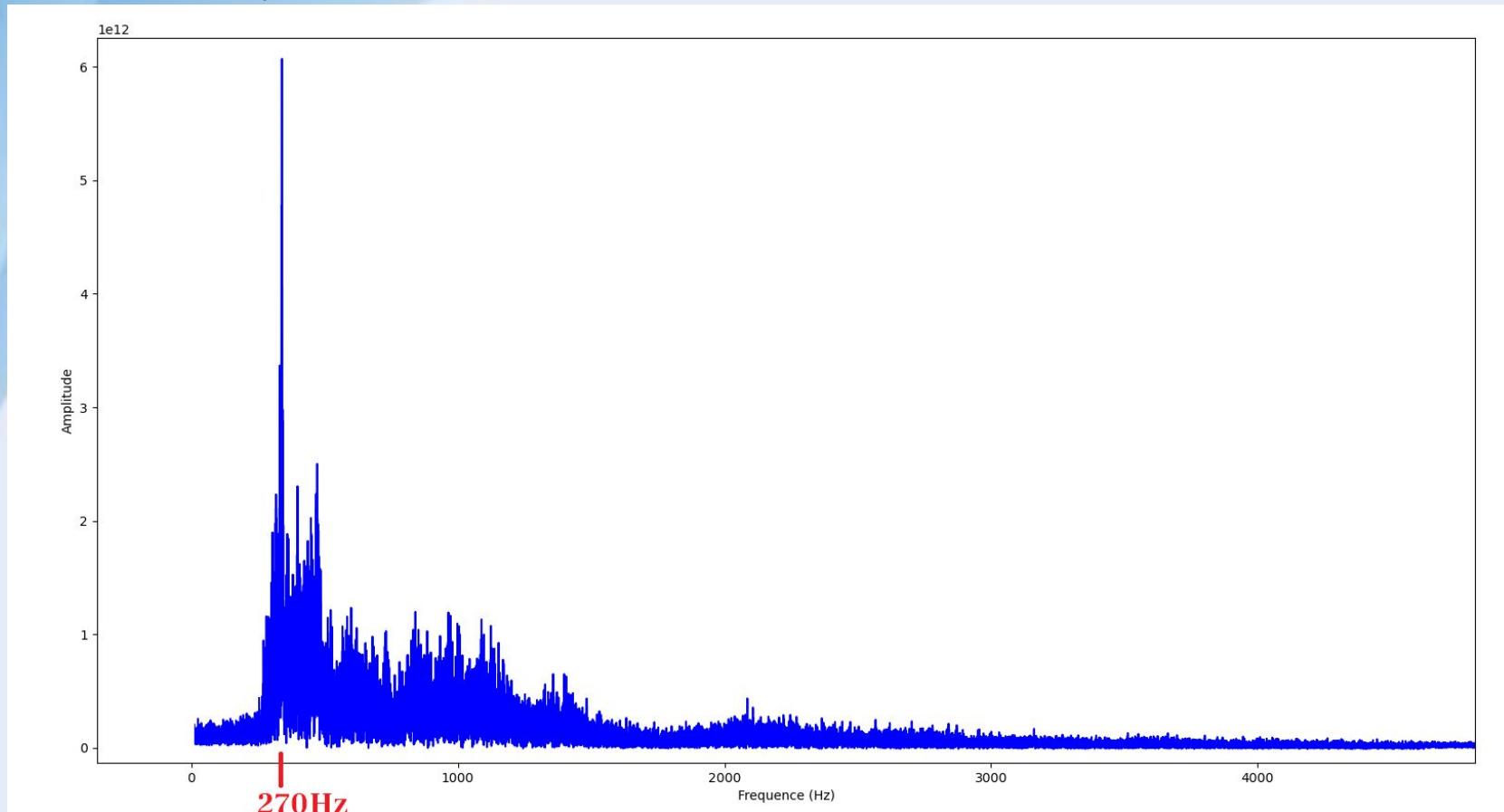


Figure 10 : comparaison des comportements d'un matériau conventionnel (a) et d'un matériau « auxétique » (b) soumis à un effort de traction uniaxial.

Structure en réseau périodique caractéristique des métamatériaux

0- Une étude théorique autour du train

Spectre représentant l'amplitude sonore en fonction des fréquences émises (en moyenne) par le train :



Les fréquences que nous cherchons à atténuer sont principalement entre 200 Hz et 2000 Hz : ce sont celles qui possèdent la plus forte amplitude et qui sont donc les plus nocives pour les passants.

I- Autour des résonateurs de Helmholtz

a) Le modèle de Helmholtz

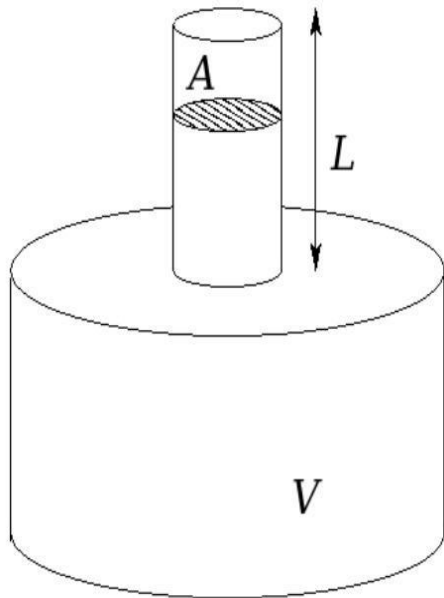


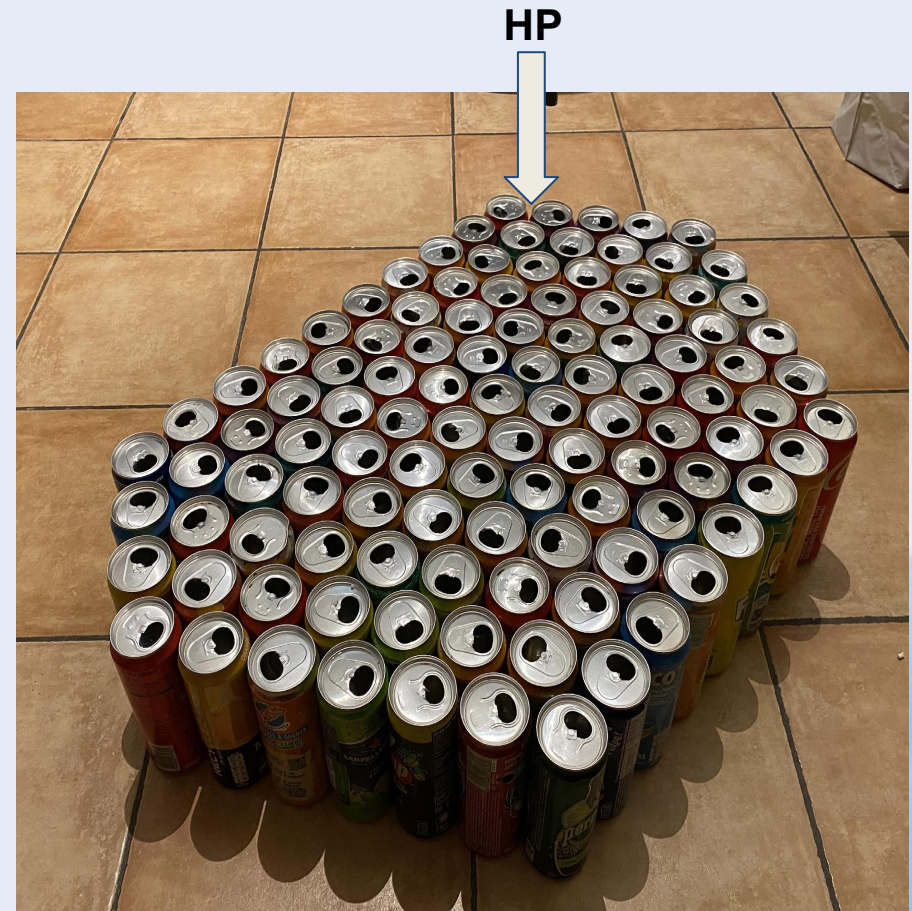
Schéma d'un résonateur de Helmholtz.

$$f_0 = \frac{c}{2\pi} \sqrt{\frac{A}{VL}}.$$

Calcul d'une fréquence de résonance selon le modèle de Helmholtz



I-b) Une première application du modèle



Utilisation de canettes vides en tant que résonateurs de Helmholtz

Le calcul théorique de la fréquence de résonance de ces canettes selon le modèle de Helmholtz nous donne une valeur de $418,6 \pm 0,5$ Hz en prenant :

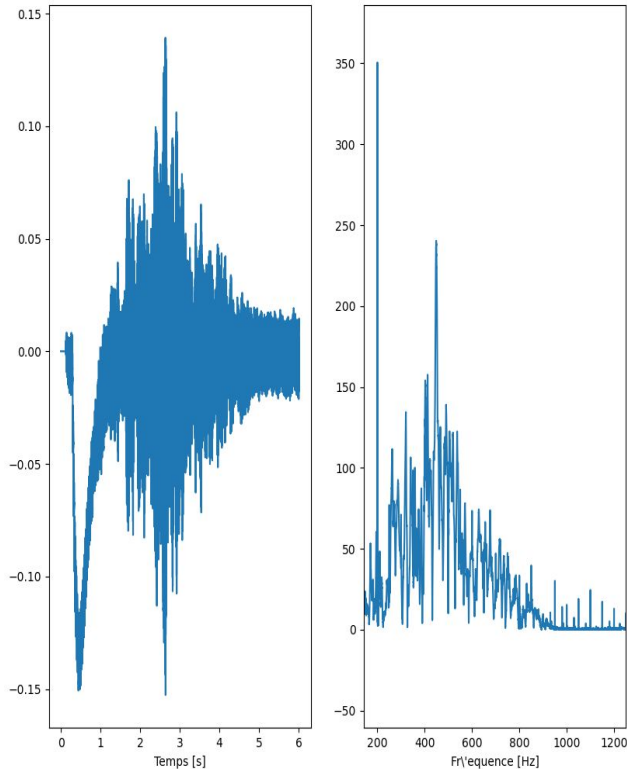
$$A = \pi \text{ cm}^2$$

$$v = 350 \cdot 10^{-6} \text{ m}^3$$

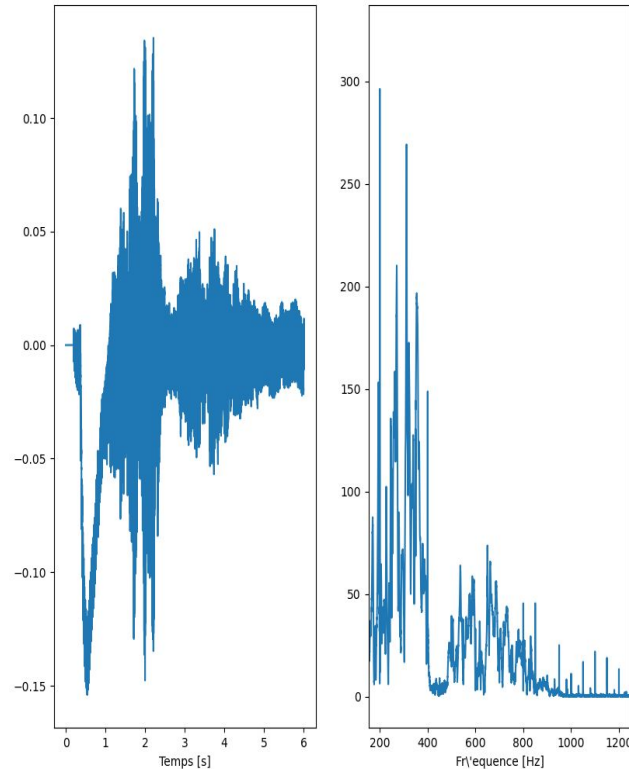
$$L = 1,5 \text{ cm}$$

I-b) Mise en évidence des résultats

Spectre obtenu en plaçant un haut parleur au dessus du sol :



Spectre obtenu en plaçant le haut parleur au dessus des résonateurs



Mise en évidence d'une atténuation autour de 420 Hz (prévu par la théorie).

I- c) Une seconde application

Seconde expérience autour des résonateurs de Helmholtz (plus adaptée pour l'application à une rue longeant une voie ferrée) :



Utilisation de plaques de polystyrène pour mieux guider le son

Selon le modèle de Helmholtz, pour cette organisation des résonateurs, la fréquence de résonance se trouve théoriquement à $269,0 \pm 0,9$ Hz. Où :

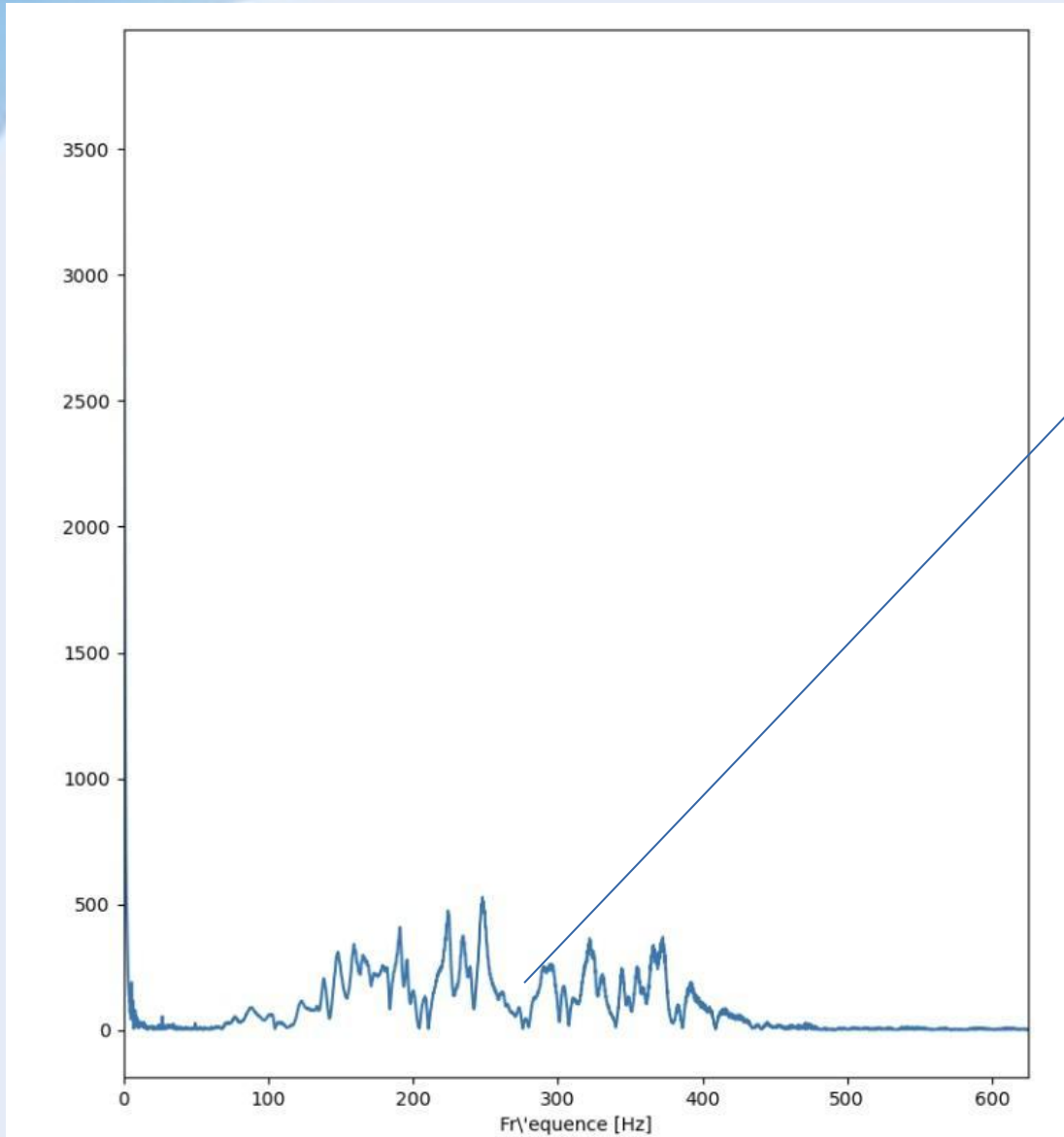
$$A = 1,6 \cdot 10^{-4} \text{ m}^2$$

$$V = 350 \cdot 10^{-6} \text{ m}^3$$

$$L_{\text{corrigé}} = 0,1 \cdot 10^{-3} + 1,8 \cdot 10^{-2} \text{ m}$$

I- c) Mise en évidence des résultats

Spectre obtenu en plaçant le micro derrière le mur de canettes :



Atténuation aux
alentours de 270
Hz

II- Mise en place d'un réseau de résonateurs

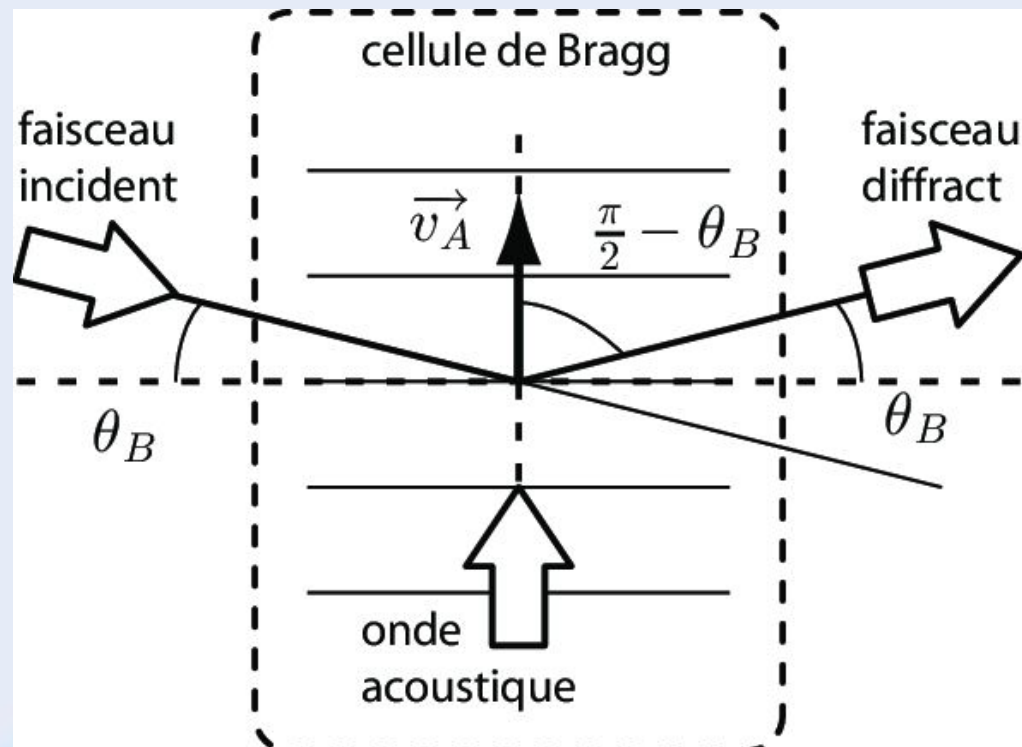
a) modèle du réseau de Bragg

- Formule de Bragg :
- $\lambda = 2 \cdot n \cdot \Lambda$, où : n est l'indice moyen effectif de réfraction
 λ correspond à la longueur d'onde
 Λ est la période du réseau de Bragg, c'est-à-dire ici l'écart se trouvant entre chaque centre de chaque canettes.

- Condition de Bragg :

Ici : $n=1$ donc : $\Lambda = \lambda/2$

or $\lambda = c/f \Rightarrow f = c/\lambda = c/2 \cdot \Lambda$



II- b) Création d'un réseau de résonateurs

Mise en place du dispositif dans l'ouverture d'une porte en suivant le modèle du réseau de Bragg :

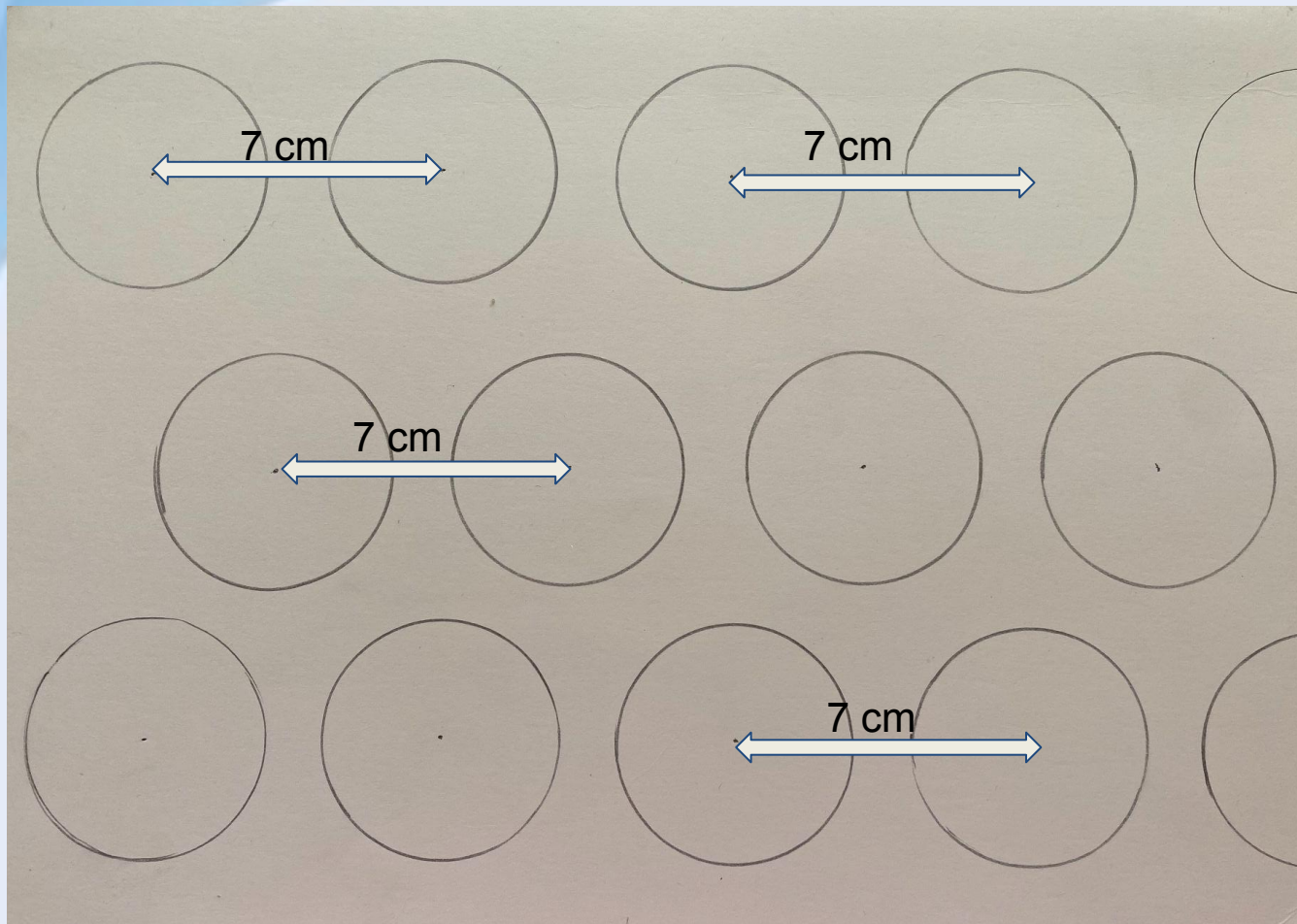


Isolation du reste de la porte à l'aide de mousse et de polystyrène

Réseau de canettes (résonateurs) à trois rangées et trois étages.
D'après la formule de Bragg, les centres des canettes sont tous espacés de 7 cm entre eux.

II- b) Création d'un réseau de résonateurs

Vue du dessus de la mise en place du réseau :



Les écarts entre chaque lignes sont les mêmes afin d'avoir une certaine régularité dans le réseau.

Le réseau est organisé de telle sorte que les canettes sont placées en quinconce.

II- b) Création d'un réseau de résonateurs

- Utilisation d'un code python :
- Ce code génère des fréquences sonores allant de 0 à 40 000 Hz et trace ensuite le graphique représentant le signal reçu par le micro.
- Les graphiques finalement obtenus correspondent à une moyenne des différentes mesures, prenant alors en compte une incertitude de type A.

```
# -*- coding: utf-8 -*-
from matplotlib.pyplot import *
import numpy as np
from scipy.signal import chirp
import pyaudio

# global variables
fe=44100

# some functions
def ChirpSignal(duration,fstart,fstop):
    t=np.arange(0,duration,1./fe)
    s=chirp(t,fstart,tend,fstop)*np.hanning(len(t))
    return t,s

def CreateSpectrum(s):
    Nfft=len(s)
    S=abs(np.fft.fft(s,Nfft))
    freq=(1.*np.arange(0,Nfft))/Nfft*fe
    return freq,S

def ShowSignalAndSpectrum(t,s):
    [freq,S]=CreateSpectrum(s)
    figure(),clf()
    subplot(121)
    plot(t,s)
    xlabel("Temps [s]")
    subplot(122)
    plot(freq,np.abs(S))
    xlim(0,1.25*fstop)
    xlabel(r"Fréquence [Hz]")
```

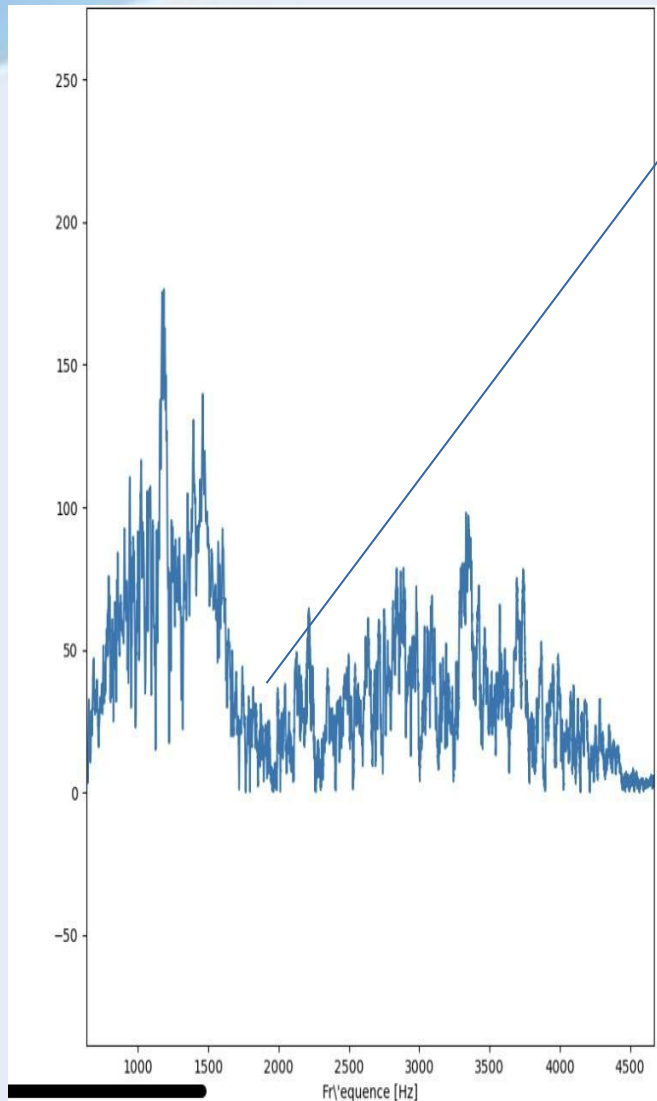
```
def PlayAndRecord(s,RecordTime):
    FORMAT = pyaudio.paFloat32
    FRAMESIZE = 1024
    NOFFRAMES = int(np.ceil(fe*RecordTime/FRAMESIZE))
    if NOFFRAMES*FRAMESIZE>len(s):
        s=np.append(s,np.zeros(NOFFRAMES*FRAMESIZE-len(s)))
    p = pyaudio.PyAudio()
    stream = p.open(format=FORMAT,
                    channels=1,
                    rate=fe,
                    input=True,
                    output=True,
                    frames_per_buffer=FRAMESIZE)
    ss=np.zeros(NOFFRAMES*FRAMESIZE)
    for i in range(0,NOFFRAMES):
        stream.write((s[i*FRAMESIZE:(i+1)*FRAMESIZE]).astype(np.float32).tostring())
        tmp=stream.read(FRAMESIZE)
        tmp = np.fromstring(tmp,dtype=np.float32)
        ss[i*FRAMESIZE:(i+1)*FRAMESIZE]=tmp
    stream.stop_stream()
    stream.close()
    p.terminate()
    return ss
```

```
# fabrication du signal:
fstart=50
fstop=1000
tend=5
[t,s]=ChirpSignal(tend,fstart,fstop)
ShowSignalAndSpectrum(t,s)

# reception du signal
record_time=6
ss=PlayAndRecord(s,record_time)
t=np.arange(0,len(ss))*1./fe
ShowSignalAndSpectrum(t,ss)
```

II- b) Mise en évidence des résultats

Spectre représentant le signal sonore reçu par le micro derrière le réseau de résonateurs :

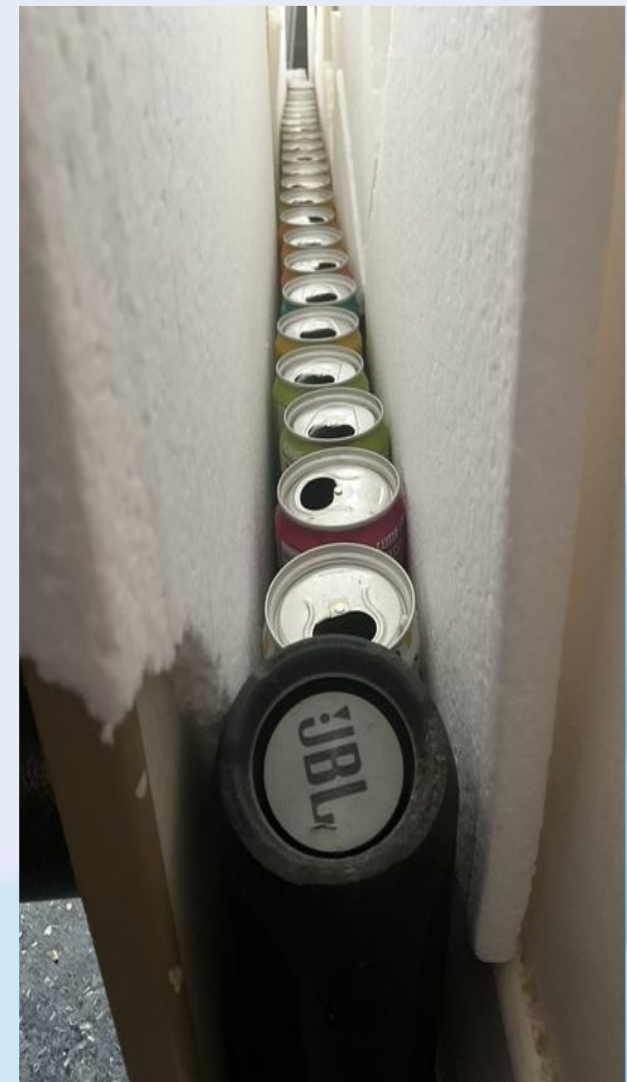
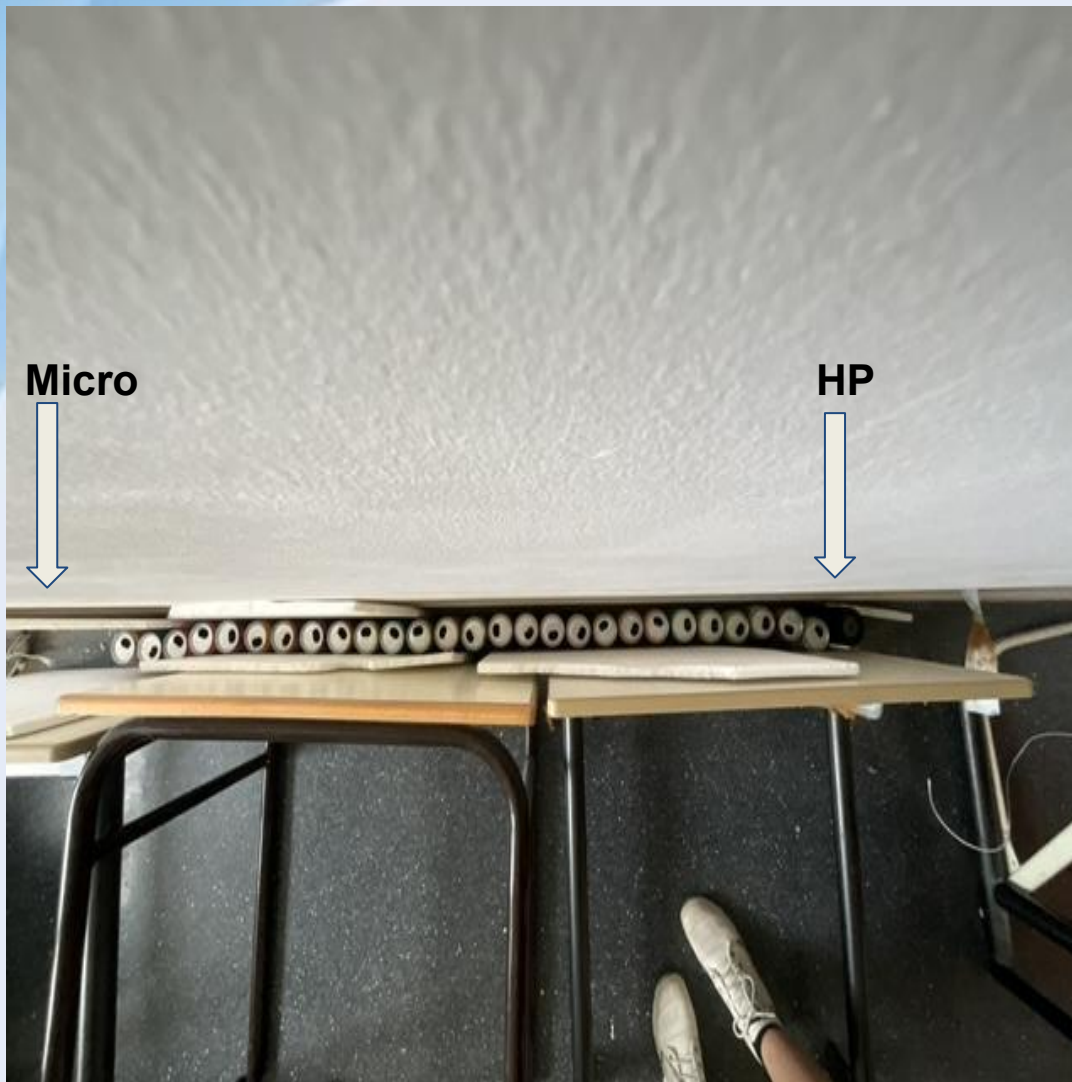


Chute de l'intensité sonore
entre 1500 Hz et 2000 Hz

Ce graphe prend en compte une
incertitude de type A sur les
mesures.

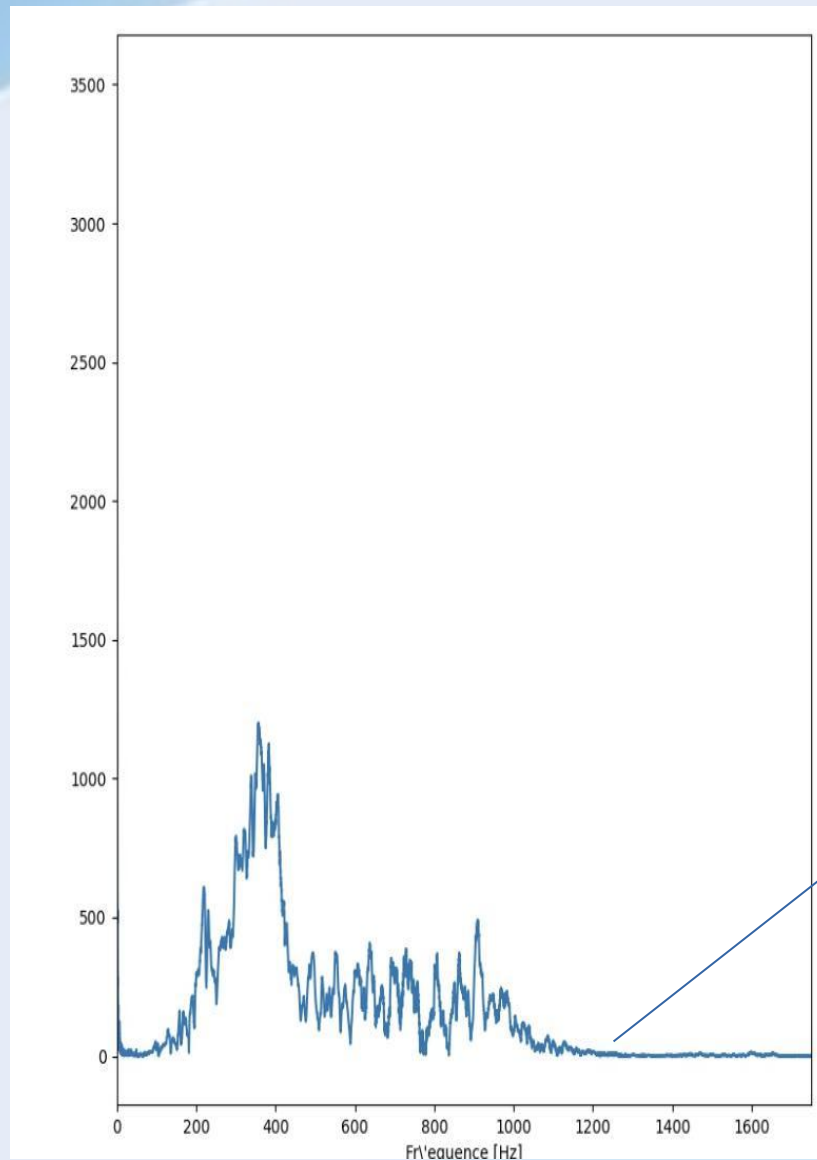
III- Fabrication d'un gradient de résonateurs

- Le dispositif : les résonateurs (canettes) sont remplies d'eau de façon croissante et placées dans un guide d'onde afin d'obtenir des résultats plus précis.



III- Mise en évidence des résultats

Spectre obtenu en sortie du gradient :



Obtention d'une plage
d'atténuation à partir de
1100 Hz

Conclusion et interprétation des résultats

Modèle du résonateur de Helmholtz : efficace pour l'atténuation de certaines fréquences précises.

Réseau de résonateurs : également efficace pour une plage de fréquences relativement faible.

Gradient de résonateurs : atténue une large plage de fréquences de façon efficace.

Domaines d'application :
aux abords :

- d'une voie ferrée
- d'une route (exemple : autoroute proche d'habitations)
- d'un lieu accueillant des événements (salle de spectacle, de concert)



Annexes

Code python utilisé pour générer les fréquences et tracer les différents spectres :

```
# -*- coding: utf-8 -*-
from matplotlib.pyplot import *
import numpy as np
from scipy.signal import chirp
import pyaudio

# global variables
fe=44100

# some functions
def ChirpSignal(duration,fstart,fstop):
    t=np.arange(0,duration,1./fe)
    s=chirp(t,fstart,tend,fstop)*np.hanning(len(t))
    return t,s

def CreateSpectrum(s):
    Nfft=len(s)
    S=abs(np.fft.fft(s,Nfft))
    freq=(1.*np.arange(0,Nfft))/Nfft*fe
    return freq,S

def ShowSignalAndSpectrum(t,s):
    [freq,S]=CreateSpectrum(s)
    figure(),clf()
    subplot(121)
    plot(t,s)
    xlabel("Temps [s]")
    subplot(122)
    plot(freq,np.abs(S))
    xlim(0,1.25*fstop)
    xlabel(r"Fr\ 'equence [Hz]")
    return
```

Annexes

```
def PlayAndRecord(s, RecordTime):
    FORMAT = pyaudio.paFloat32
    FRAMESIZE = 1024
    NOFFRAMES = int(np.ceil(fe*RecordTime/FRAMESIZE))
    if NOFFRAMES*FRAMESIZE >= len(s):
        s = np.append(s, np.zeros(NOFFRAMES*FRAMESIZE - len(s)))
    p = pyaudio.PyAudio()
    stream = p.open(format=FORMAT,
                    channels=1,
                    rate=fe,
                    input=True,
                    output=True,
                    frames_per_buffer=FRAMESIZE)
    ss = np.zeros(NOFFRAMES*FRAMESIZE)
    for i in range(0, NOFFRAMES):
        stream.write((s[i*FRAMESIZE:(i+1)*FRAMESIZE]).astype(np.float32).tostring())
        tmp = stream.read(FRAMESIZE)
        tmp = np.fromstring(tmp, dtype=np.float32)
        ss[i*FRAMESIZE:(i+1)*FRAMESIZE] = tmp
    stream.stop_stream()
    stream.close()
    p.terminate()
    return ss
```

Annexes

```
# fabrication du signal:
fstart=50
fstop=1000
tend=5
[t,s]=ChirpSignal(tend,fstart,fstop)
ShowSignalAndSpectrum(t,s)

# reception du signal
record_time=6
ss=PlayAndRecord(s,record_time)
t=np.arange(0,len(ss))*1./fe
ShowSignalAndSpectrum(t,ss)
```


Annexes

Code python réalisant une FFT, utilisé pour tracer le spectre de l'amplitude en fonction des fréquences émises par le train :

```
import scipy.io.wavfile as wavfile
import scipy
import scipy.fftpack as fftpk
import numpy as np
from matplotlib import pyplot as plt

s_rate, signal = wavfile.read("Recording 20.wav")

FFT = abs(scipy.fft.fft(signal))
freqs = fftpk.fftfreq(len(FFT), (1.0/s_rate))

plt.plot(freqs[range(len(FFT)//2)], FFT[range(len(FFT)//2)])
plt.xlabel('Frequency (Hz)')
plt.ylabel('Amplitude')
plt.show()
```