

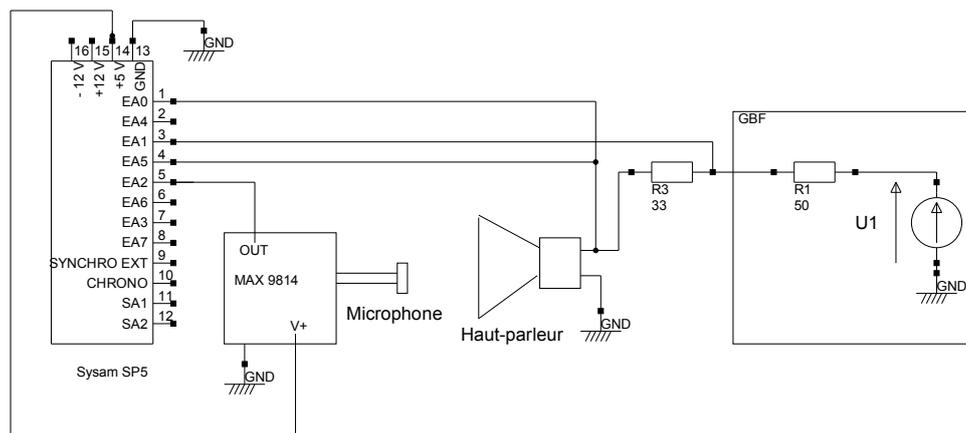
Étude électroacoustique d'un haut-parleur

1. Introduction

Ce document décrit une expérience de mesure de l'impédance d'un haut-parleur, associée à une mesure du signal délivré par un microphone électret.

La mesure du courant dans le haut-parleur se fait avec une résistance placée en série. La carte d'acquisition SysamSP5 permet de mesurer la tension aux bornes de cette résistance en utilisant le mode différentiel. La présence d'amplificateurs différentiels sur la carte SysamSP5 est un avantage important sur les oscilloscopes, car cette fonction est absente sur les oscilloscopes d'entrée de gamme utilisés couramment.

2. Dispositif de mesure



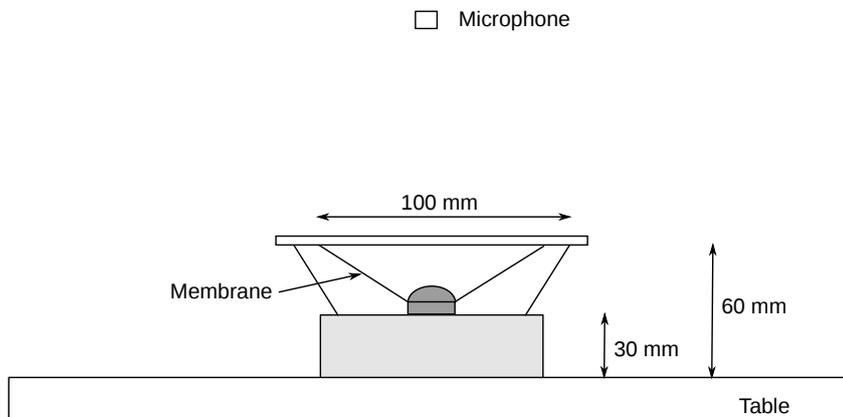
Le signal sinusoïdal est délivré par un générateur de fonctions (GBF). Le circuit de mesure est constitué du haut-parleur associé en série à une résistance $R_3 = 33\Omega$. Cette résistance doit être du même ordre de grandeur que l'impédance du haut-parleur, qui est de 8Ω (pour une fréquence audio moyenne).

Le microphone électret est placé à environ 10 cm du haut-parleur. Il est associé à un amplificateur MAX 9814. La platine comportant le microphone et l'amplificateur est vendue par [Adafruit](#). Elle est alimentée par la tension 5 V de la carte d'acquisition.

Les tensions acquises par la carte SysamSP5 sont

- ▷ La tension U_{hp} aux bornes du haut-parleur, sur la voie EA0.
- ▷ La tension U_r aux bornes de la résistance R_3 , en mode différentiel sur les voies EA1 et EA5.
- ▷ Le signal fourni par l'amplificateur du microphone, sur la voie EA2.

Le haut-parleur est posé sur la table et émet verticalement. Le microphone est fixé à environ 10 cm au dessus.



3. Programme d'acquisition

Le script python effectue l'acquisition des trois signaux. On définit en première ligne la fréquence approximative, ce qui permet de calculer la fréquence d'échantillonnage et le nom des fichiers de sauvegarde. Le module de l'impédance est le rapport des valeurs efficaces de la tension et du courant. Pour le calcul du déphasage entre la tension et le courant, on procède comme expliqué dans [Mesure d'impédance](#). Il faut tout d'abord déterminer précisément la fréquence, ce qui se fait en recherchant le maximum dans le spectre. On calcule ensuite le cosinus et le sinus du déphasage, d'où l'on déduit le déphasage.

Le trois signaux sont enregistrés dans un fichier. Un second fichier enregistre la fréquence, l'impédance, le déphasage, et les valeurs efficaces des signaux.

[acquisitionHP.py](#)

```
# -*- coding: utf-8 -*-
import pycanum.main as pycan
from matplotlib.pyplot import *
import numpy
import numpy.fft

f = 1300.0
fe = f*100
nom="%d"%int(f)

sys=pycan.Sysam("SP5")
sys.config_entrees([0,1,2],[5,5,5],diff=[1])
te=1.0/fe
ne=80000
sys.config_echantillon(te*1e6,ne)
sys.acquerir()
t=sys.temps()[0]
te=t[1]-t[0]
fe=1.0/te
```

```
ne=len(t)
u=sys.entrees()
son=u[2]
Uhp=u[0]
R=33.0
Ihp=u[1]/R
sys.fermer()

figure()
plot(t,Uhp,label="U")
plot(t,u[1],label="RI")
plot(t,u[2],label="son")
xlabel("t (s)")
ylabel("u (V)")
axis([0,te*ne,-5,5])
grid()
legend(loc='upper right')

Ueff = Uhp.std()
Ieff = Ihp.std()
Z=Ueff/Ieff
Pson = son.std()
spectre = numpy.absolute(numpy.fft.fft(Uhp))
f = numpy.arange(ne)*1.0/(te*ne)
freq = numpy.argmax(spectre[0:int(ne/2)])*1.0/(te*ne)
periode = 1.0/freq
print("Frequence = %f"%freq)
p = int(periode/te/4) # decalage quadrature
Vhp = numpy.roll(Uhp,p,0)
cos = numpy.mean(Uhp*Ihp)
sin = numpy.mean(Vhp*Ihp)
norme = numpy.sqrt(cos**2+sin**2)
cos = cos/norme
sin = sin/norme
phi = numpy.angle(cos+sin*1j)

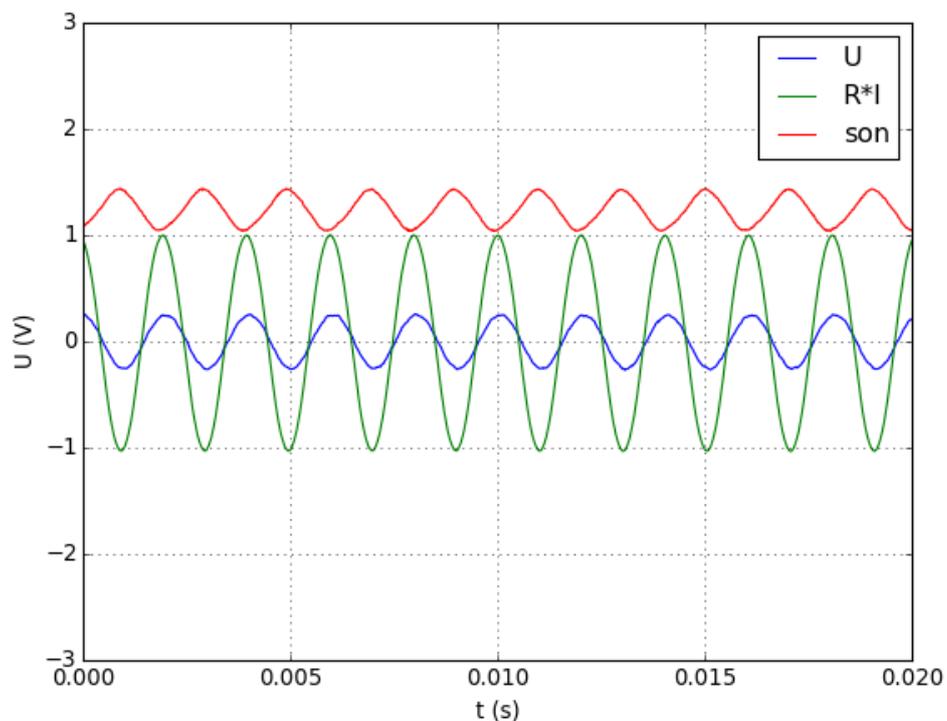
print("Z = %f"%Z)
print("phi = %f"%phi)
print("Pson = %f"%Pson)
print("Ieff = %f"%Ieff)
print("Ueff = %f"%Ueff)

numpy.savetxt("courbes-%s.txt"%nom,[t,Uhp,Ihp,son])
numpy.savetxt("impedance-%s.txt"%nom,[freq,Z,phi,Ueff,Ieff,Pson])
show()
```

4. Résultats

Voici un exemple de signaux, pour une fréquence d'environ 500 Hz :

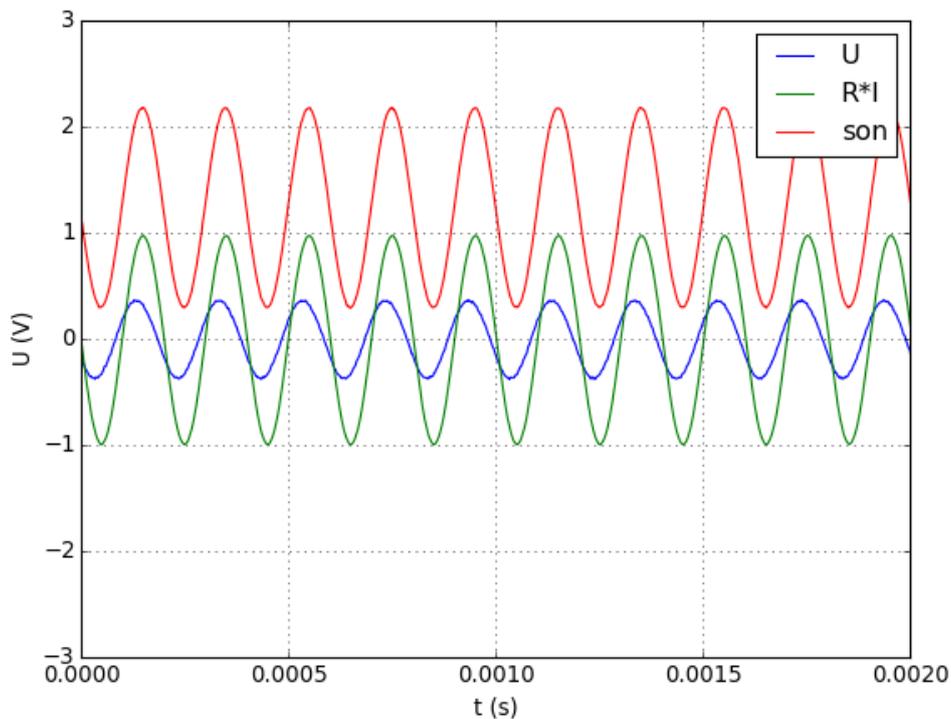
```
import numpy
from matplotlib.pyplot import *
[t,Uhp,Ihp,son] = numpy.loadtxt("courbes-500.txt")
figure()
R=33.0
plot(t,Uhp,label="U")
plot(t,Ihp*R,label="R*I")
plot(t,son,label="son")
axis([0,0.02,-3,3])
xlabel("t (s)")
ylabel("U (V)")
grid()
legend(loc='upper right')
```



Pour cette fréquence, le courant et la tension sont presque en phase car le haut-parleur est principalement résistif. Son inductance se manifeste à haute fréquence :

```
import numpy
from matplotlib.pyplot import *
```

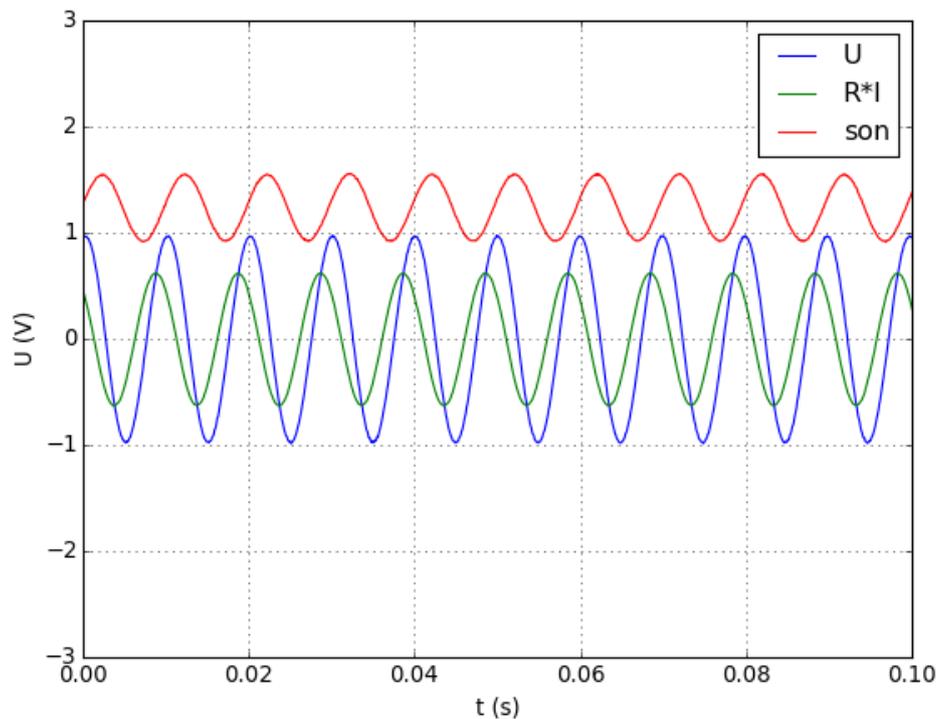
```
[t,Uhp,Ihp,son] = numpy.loadtxt("courbes-5000.txt")
figure()
R=33.0
plot(t,Uhp,label="U")
plot(t,Ihp*R,label="R*I")
plot(t,son,label="son")
axis([0,0.002,-3,3])
xlabel("t (s)")
ylabel("U (V)")
grid()
legend(loc='upper right')
```



À basse fréquence, il y a un phénomène de résonance mécanique qui introduit aussi un déphasage :

```
import numpy
from matplotlib.pyplot import *
[t,Uhp,Ihp,son] = numpy.loadtxt("courbes-100.txt")
figure()
R=33.0
plot(t,Uhp,label="U")
plot(t,Ihp*R,label="R*I")
plot(t,son,label="son")
axis([0,0.1,-3,3])
xlabel("t (s)")
ylabel("U (V)")
```

```
grid()
legend(loc='upper right')
```

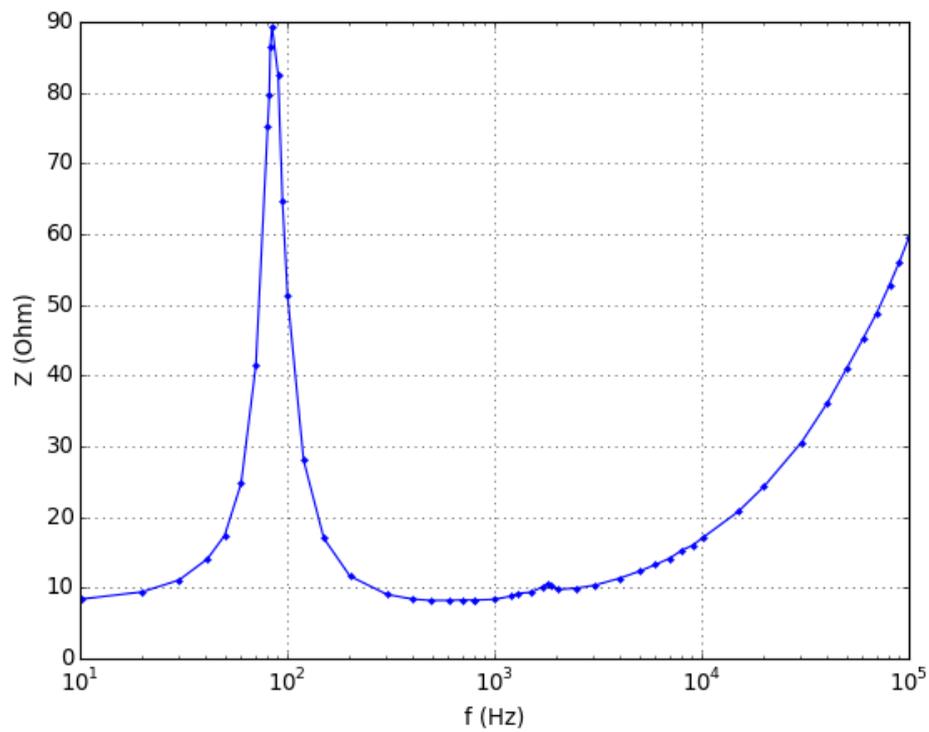


Voici les courbes d'impédance (module et déphasage) en fonction de la fréquence :

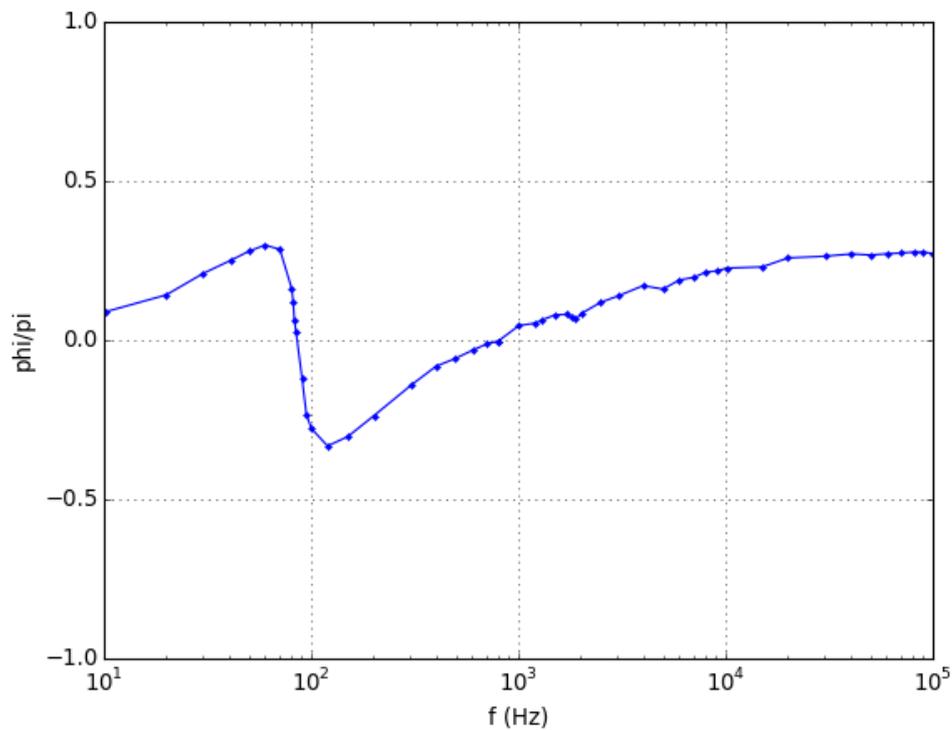
```
noms = ["10", "20", "30", "40", "50", "60", "70", "80", "82", "83", "85", "90", "95", "100", "120",
```

```
liste_f = []
liste_Z = []
liste_phi = []
liste_son = []
liste_P = []
for nom in noms:
    [freq,Z,phi,Ueff,Ieff,Pson] = numpy.loadtxt("impedance-%s.txt"%nom)
    liste_f.append(freq)
    liste_Z.append(Z)
    liste_phi.append(phi/numpy.pi)
    liste_P.append(Ueff*Ieff)
    liste_son.append(Pson/Ueff)
```

```
figure()
plot(liste_f,liste_Z,".-")
xlabel("f (Hz)")
ylabel("Z (Ohm)")
xscale("log")
grid()
```



```
figure()
plot(liste_f,liste_phi,".-")
xlabel("f (Hz)")
ylabel("phi/pi")
xscale("log")
axis([10,1e5,-1,1])
grid()
```

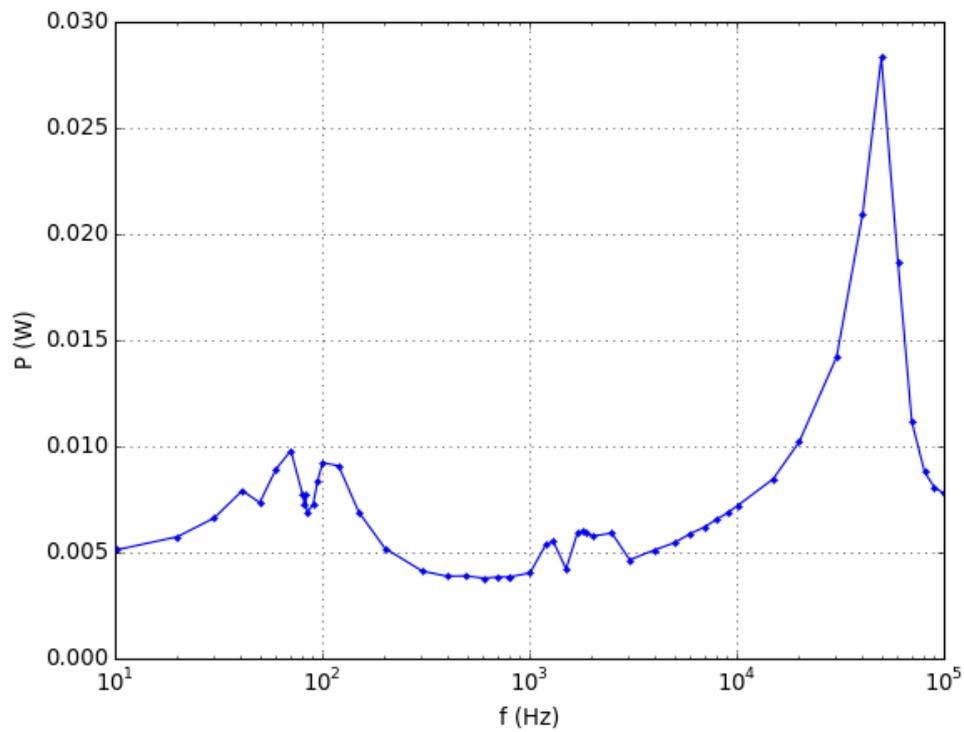


On voit la résonance mécanique à 83 Hz suivie d'une phase de faible impédance (environ 8Ω) où la résistance du bobinage domine. À plus haute fréquence, l'impédance augmente.

On voit sur la courbe de phase que le haut-parleur n'est pas du tout assimilable à une résistance, même pour les fréquences moyennes.

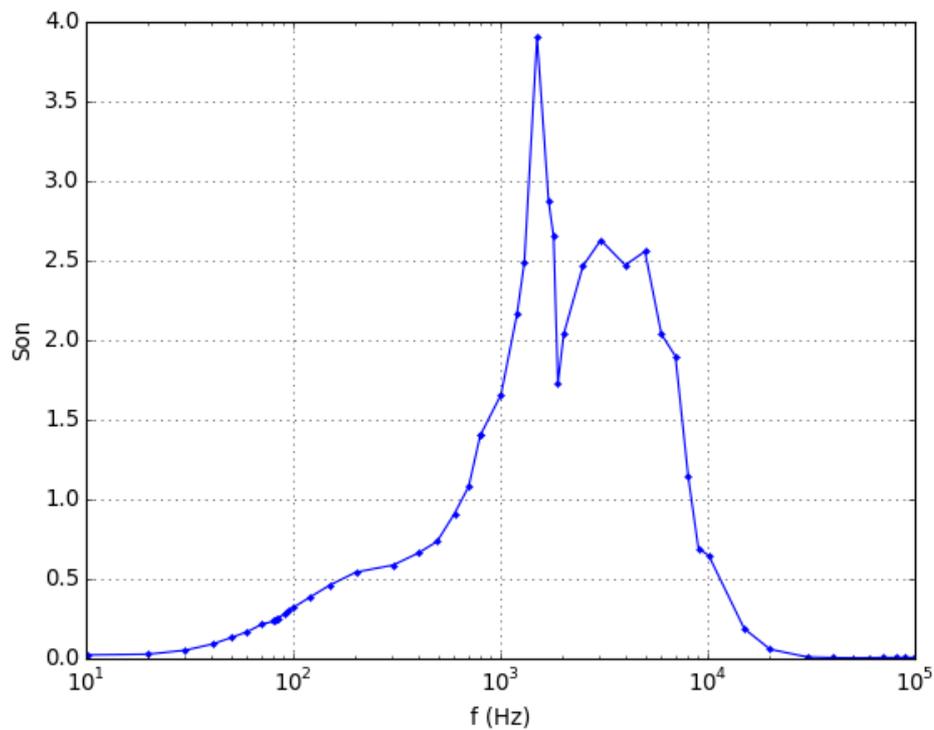
On trace aussi la puissance reçue par le haut-parleur :

```
figure()
plot(liste_f,liste_P,".-")
xlabel("f (Hz)")
ylabel("P (W)")
xscale("log")
grid()
```



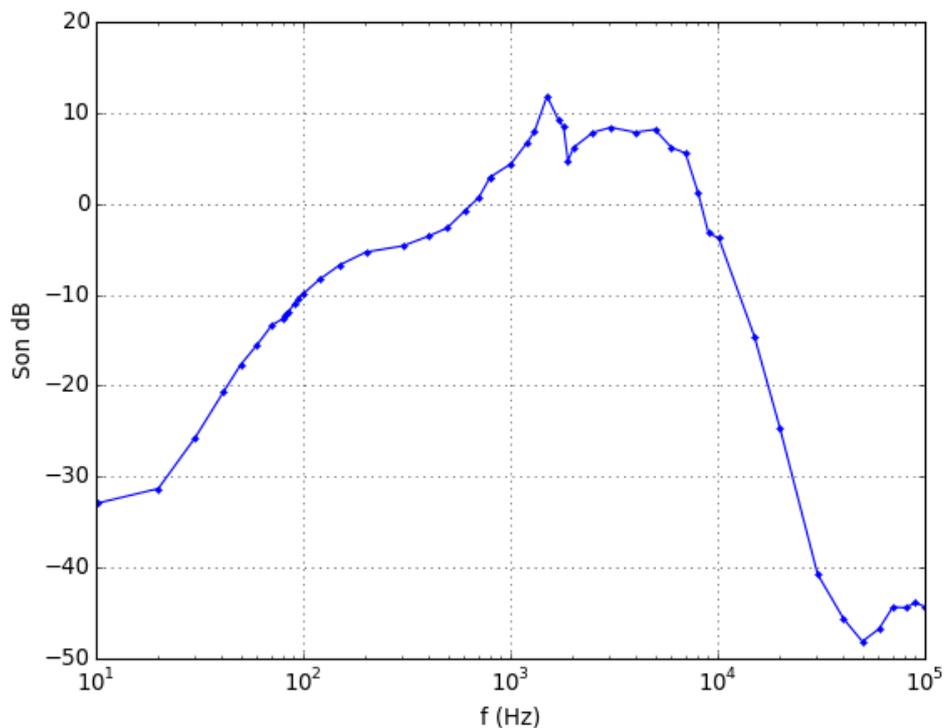
Voici la valeur efficace du signal transmis par l'amplificateur du microphone, divisée par la tension efficace :

```
figure()
plot(liste_f,liste_son,".-")
xlabel("f (Hz)")
ylabel("Son")
xscale("log")
grid()
```



Cette courbe est souvent donnée en échelle décibel (mais nous n'avons que des décibels relatifs), qui représente mieux l'intensité que nous percevons.

```
son = 20*numpy.log10(numpy.array(liste_son))
figure()
plot(liste_f,son,".-")
xlabel("f (Hz)")
ylabel("Son dB")
xscale("log")
grid()
```



La fiche constructeur du microphone indique une bande passante parfaitement plate de 50 à 3000 Hz. Sur ce domaine de fréquence (qui est le plus important pour les sons audibles), les variations observées sont donc dues au haut-parleur.

L'intensité sonore a été divisée par la tension efficace, car en pratique les haut-parleurs sont commandés en tension. L'intensité sonore augmente à haute fréquence. Entre 500 et 5000 Hz, l'augmentation est d'environ 10 dB. Ces résultats sont étonnants, car il s'agit d'un haut-parleur large bande (de diamètre 10 cm), prévu pour restituer correctement tout le spectre audible. Il faut cependant remarquer que le haut-parleur est posé horizontalement sur la table, sans caisson d'enceinte, ce qui ne correspond pas du tout aux conditions recommandées d'utilisation. Lorsque le haut-parleur est fixé dans une enceinte de type *bass-reflex*, les graves sont renforcés par résonance avec la cavité formée par l'enceinte.