Créer un moment cinétique avec du son

grâce aux résonateurs de Helmholtz

AMRANI Taha — 35198

2024-2025

Plan

Introduction et étude théorique

Observation initiale

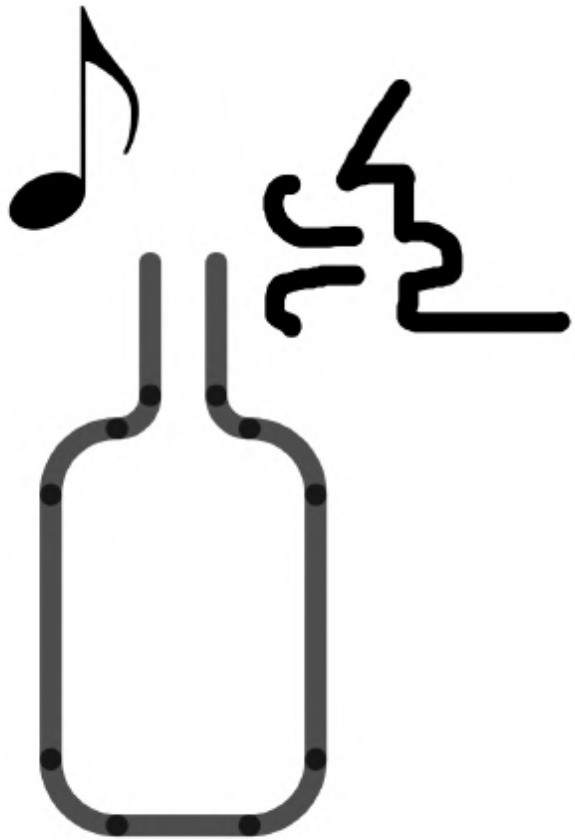


Figure 1 – Souffler sur une bouteille laisse entendre un son

Problématique

Problématique :

Comment peut-on exploiter les propriétés mécaniques des résonateurs de Helmholtz pour produire un mouvement exploitable à l'échelle macroscopique à partir d'une onde sonore ?

Le modèle de Helmholtz - Hypothèses

Hypothèses du modèle :

1. $L << \lambda$, où λ est la longueur de l'onde sonore
2. L'air $\approx$ **gaz parfait**
3. Transformations rapides $\rightarrow$ Parois **calorifugées**

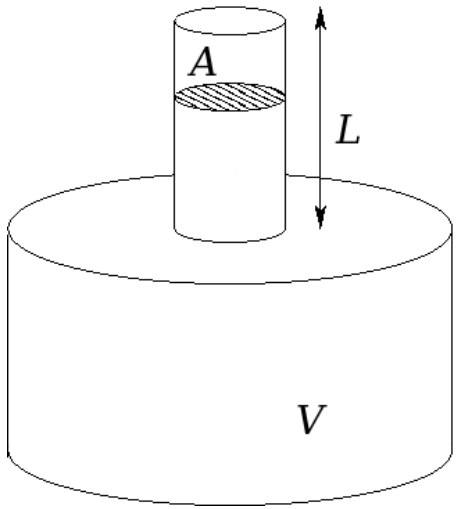


Figure 2 – Modèle classique du résonateur de Helmholtz

Le modèle en action

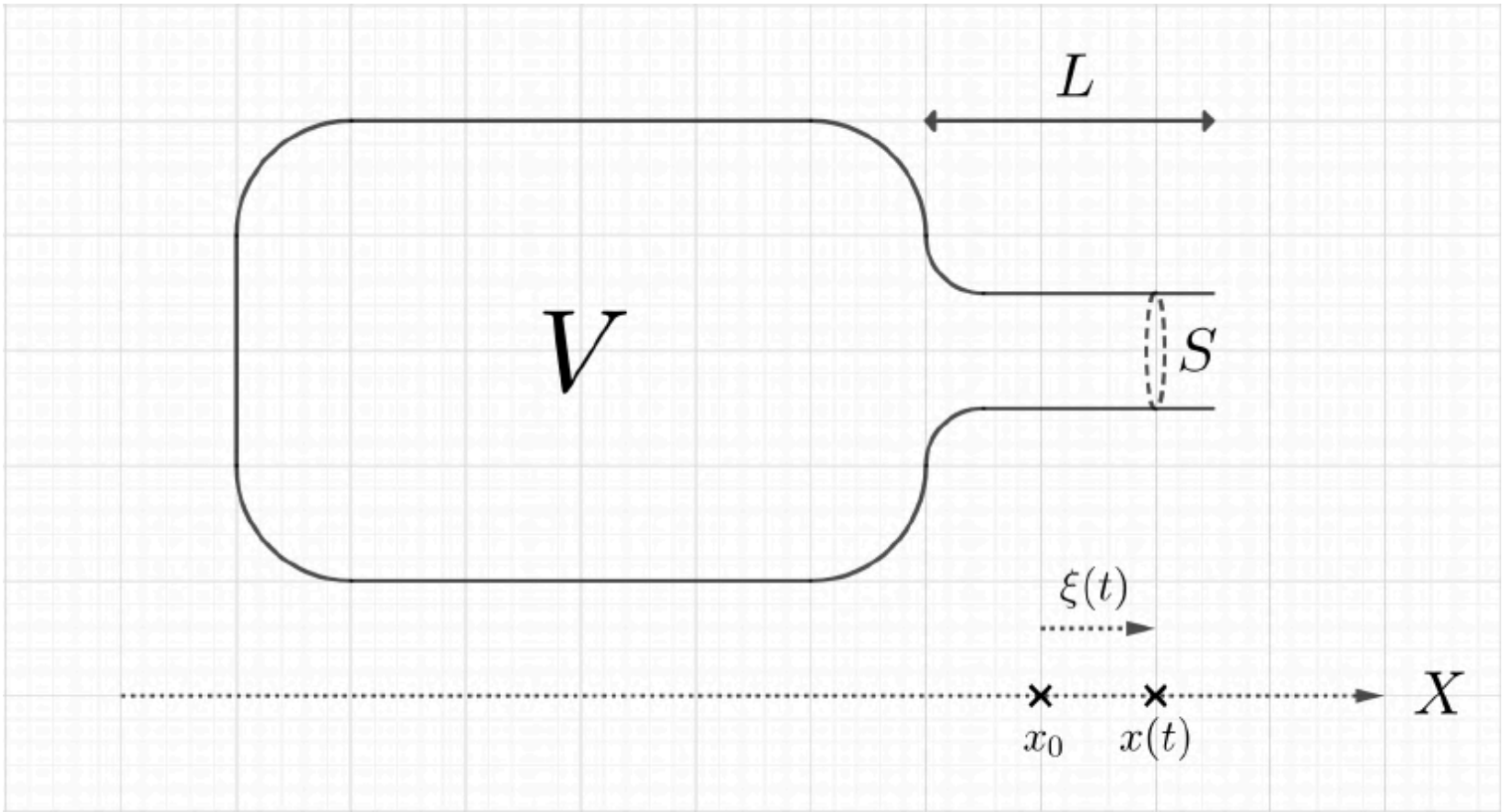


Figure 3 – Introduction des notations

Le modèle en action

Théorème I

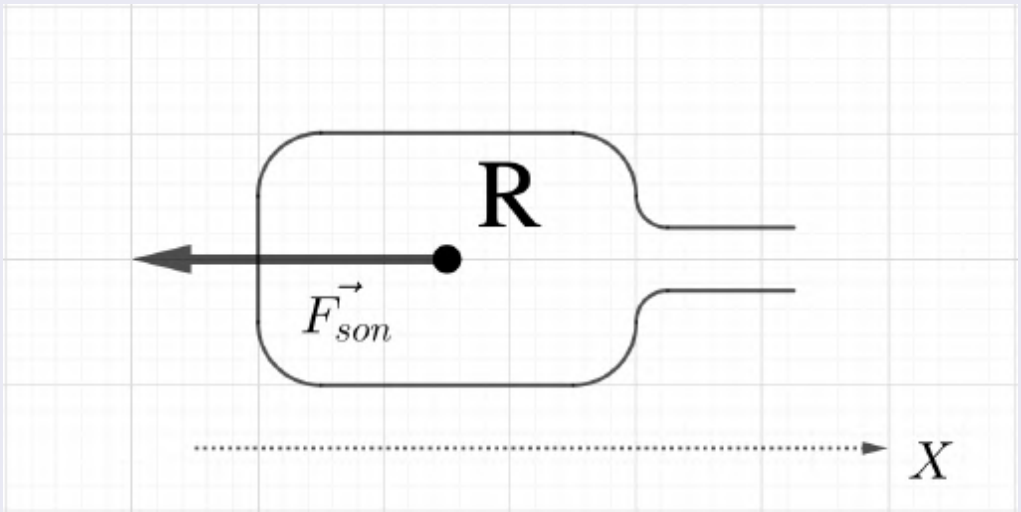
Sous l'effet d'une force extérieure F, S se met à osciller dans l'axe (O_x) du tube selon l'équation

$$\frac{d^2\xi}{dt^2} + \omega_0^2 \xi = \frac{F}{\rho S L} \qquad \text{où } \omega_0 = \sqrt{\frac{\gamma p A}{\rho V L}}$$

Le modèle en action

Théorème II

Modélisation de l'action sur le résonateur R d'une onde sonore à la fréquence ω_0 : force de norme constante, dirigée selon l'axe du goulot.



Problème

Comment évaluer la norme de ladite force ?

"Moulin acoustique"

L'expérience de Vinko Dvorak et Alfred Mayer permet :

- d'obtenir la norme de la force
- d'en exploiter la puissance mécanique

Figure 4 – Dispositif utilisé par Mayer en 1878
Source de l'image : Scientific American, Supplément n°139, 31 août 1878

"Moulin acoustique"

(a) Vue globale

(b) Vue du dessus

Figure 5 – Dispositif utilisé dans ce TIPE

"Moulin acoustique"

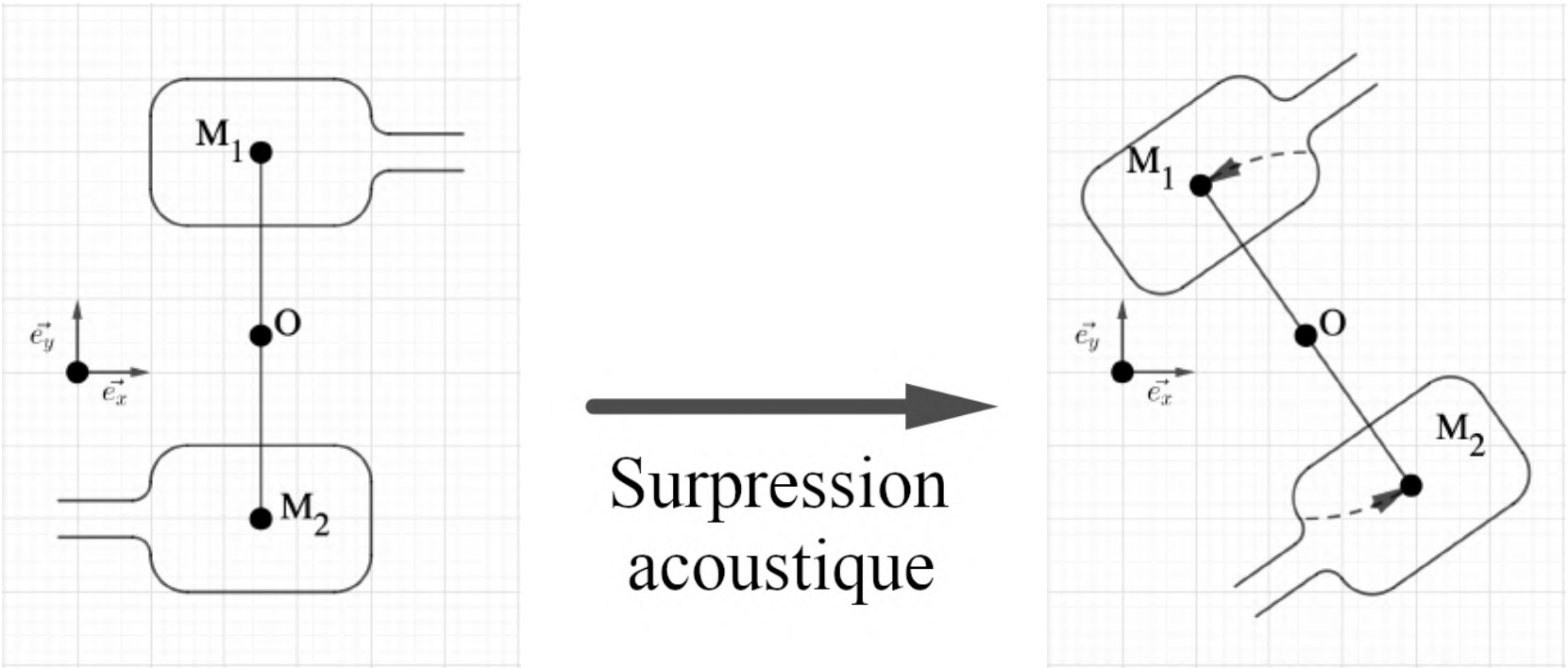
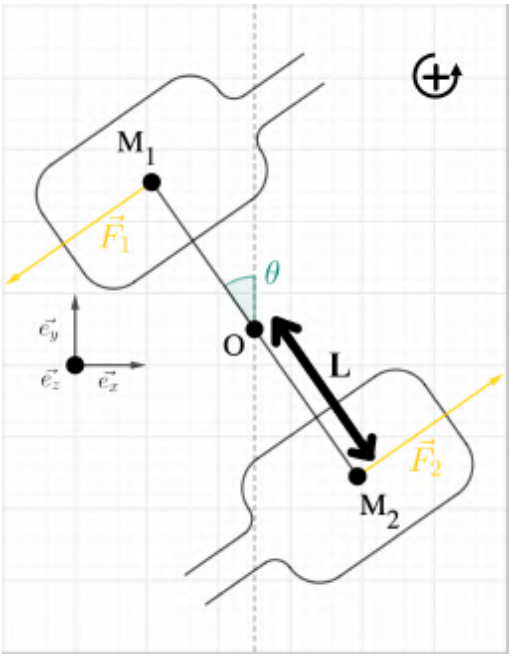


Figure 6 – Description de l'expérience de Dvorak

Étude théorique de l'expérience de Dvorak



Bilan des actions exercées sur le mécanisme :

- Les forces $\vec{F_1}$ et $\vec{F_2}$ → action des surpressions sonores.
  - On note M_F leur moment selon $\vec{e_z}$ en O.
- Un couple $-h\dot{\theta}\vec{e_z}$ → forces de frottements $\vec{f_1}$ et $\vec{f_2}$
- Un couple $-C\theta\vec{e_z}$ → torsion de la corde

On a alors :

$$\boxed{\ddot{\theta} + \frac{\omega_1}{Q}\dot{\theta} + \omega_1^2\theta = \omega_1^2\theta_\infty}$$

où $\theta_\infty = \frac{M_F}{C}, \omega_1 = \sqrt{\frac{C}{J}}, Q = \frac{\sqrt{JC}}{h}$, J le moment d'inertie du système {Bouteilles+Règle} par rapport à l'axe de la corde

Étude théorique de l'expérience de Dvorak

Le comportement du mécanisme dépend alors du signe de

$$\Delta = \frac{h^2}{J^2} - 4\frac{C}{J}.$$

Expérience

Parenthèse pratique : mesure de la fréquence propre d'une cavité

Figure 7 – Les grandeurs L, V et S ne sont pas toujours bien définies

Mesure de la fréquence propre d'une cavité

Quand on propage une onde sonore de **pulsation** ω sur un résonateur de **pulsation propre** ω_0, celui-ci produit un son.

Lorsqu'on mesure l'intensité sonore résultante, l'ordinateur affiche une valeur proportionnelle à :

$$\boxed{G_{dB}(\omega) := -log(|\omega_0^2 - \omega^2|)}$$

Mesure de la fréquence propre d'une cavité

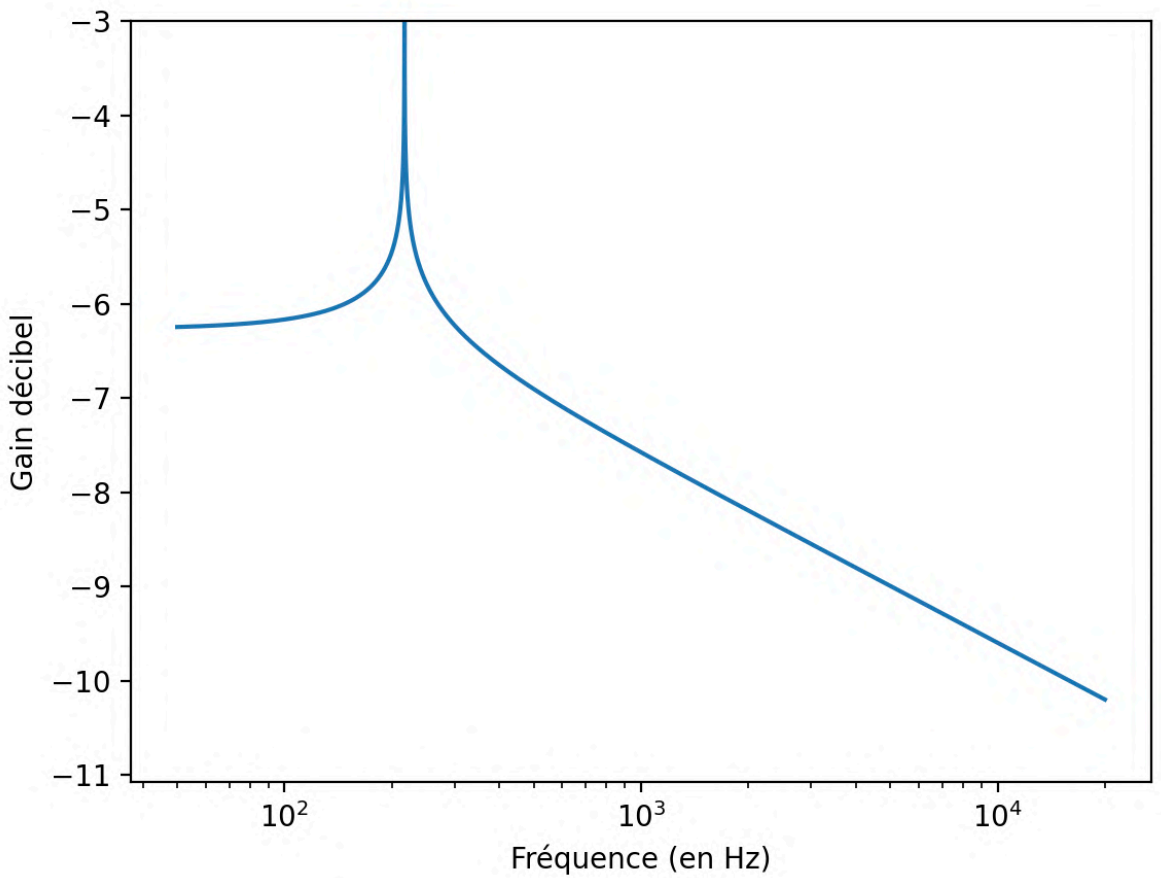


Figure 8 – Exemple : Diagramme de Bode résultant pour $f_0 = 217$ Hz

Mesure de la fréquence propre d'une cavité

En conclusion, pour mesurer la fréquence propre d'une cavité :

1. Souffler dessus (bruit blanc $\rightarrow$ perturbations mécaniques de pulsations variées)
2. Mesurer la fréquence associée au maximum du spectre sonore du son résultant

Mesure de la fréquence propre d'une cavité

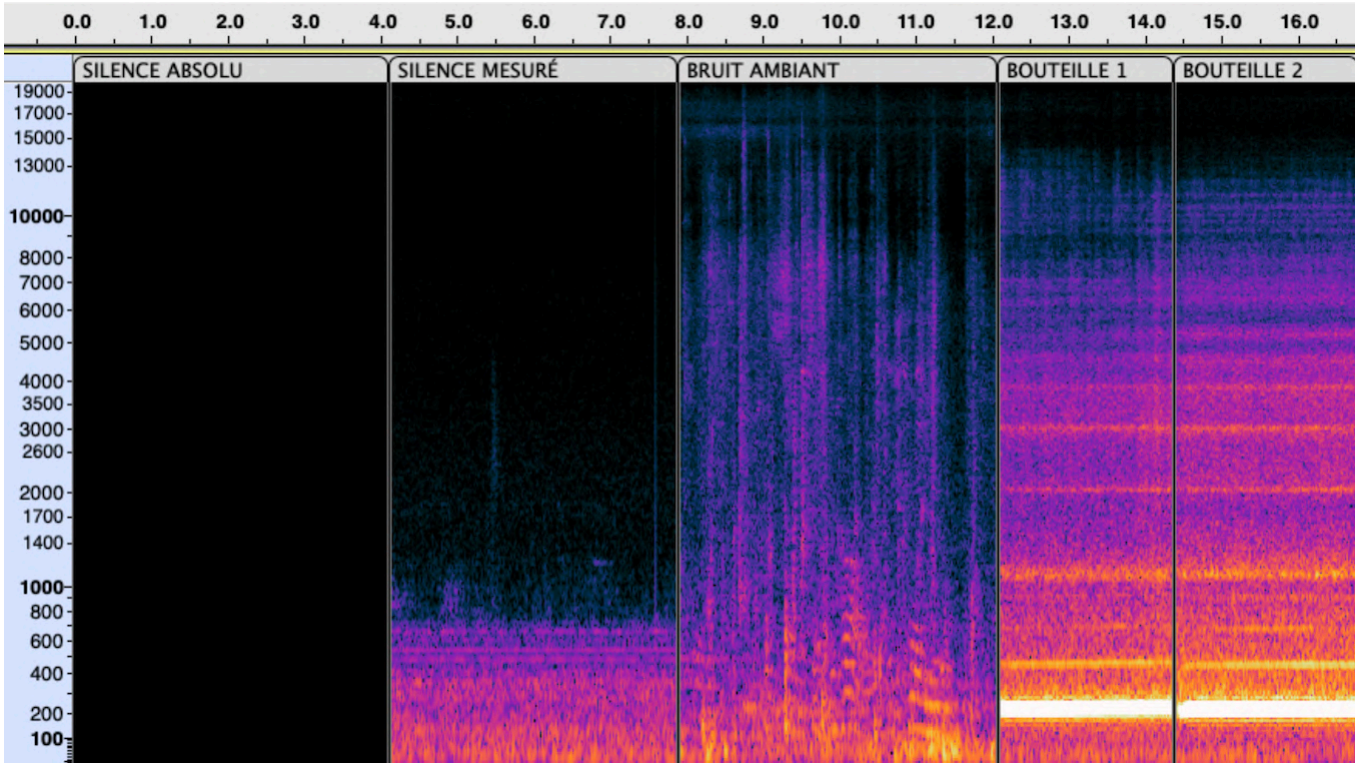
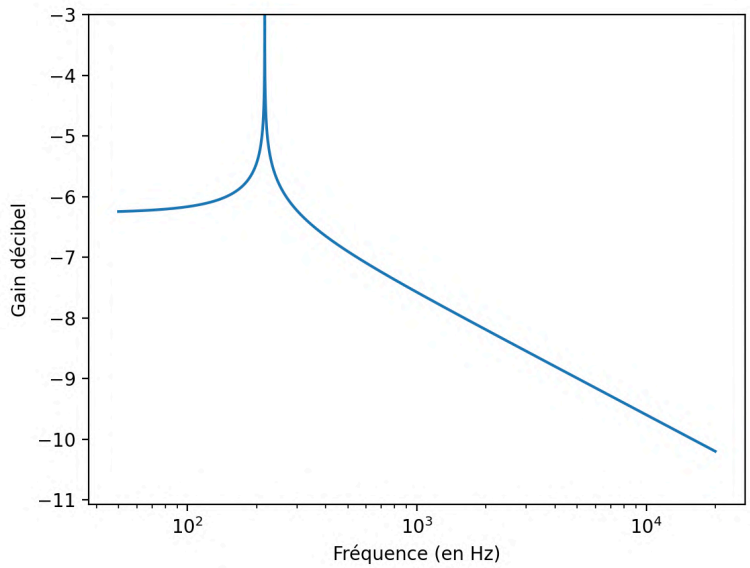


Figure 9 – Visualisation du spectre obtenu dans différentes situations avec le logiciel Audacity

Mesure de la fréquence propre d'une cavité



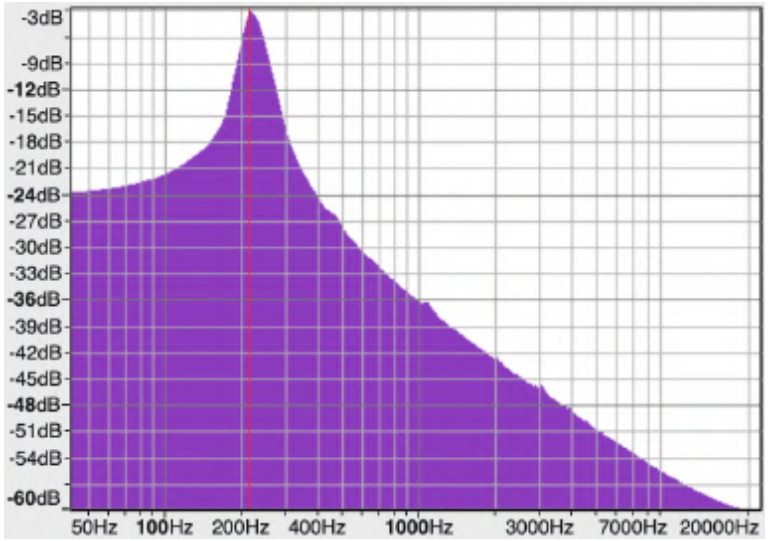
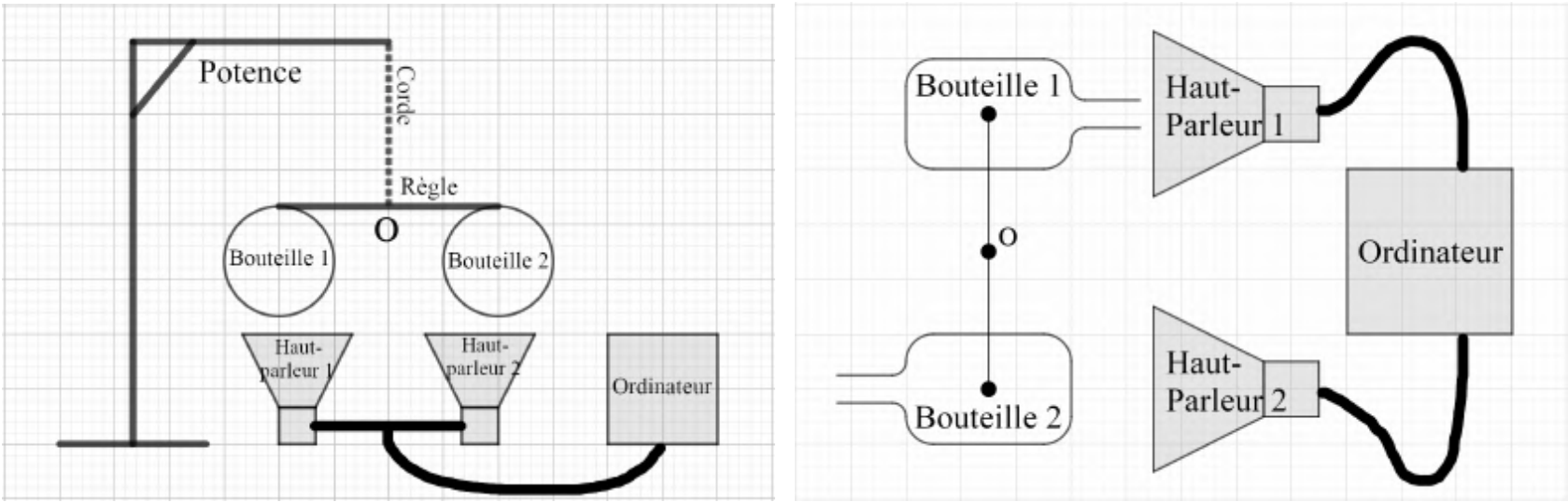


Figure 10 – Comparaison entre l'allure théorique et pratique du diagramme de Bode. On mesure dans cet exemple $f_0 = 217 \pm 1$ Hz.

Adaptation personnelle de l'expérience de Dvorak



(a) Vue latérale (b) Vue du dessus

Figure 11 – Schéma du montage expérimental

Adaptation personnelle de l'expérience de Dvorak

Figure 12 – Photo du montage

Adaptation personnelle de l'expérience de Dvorak

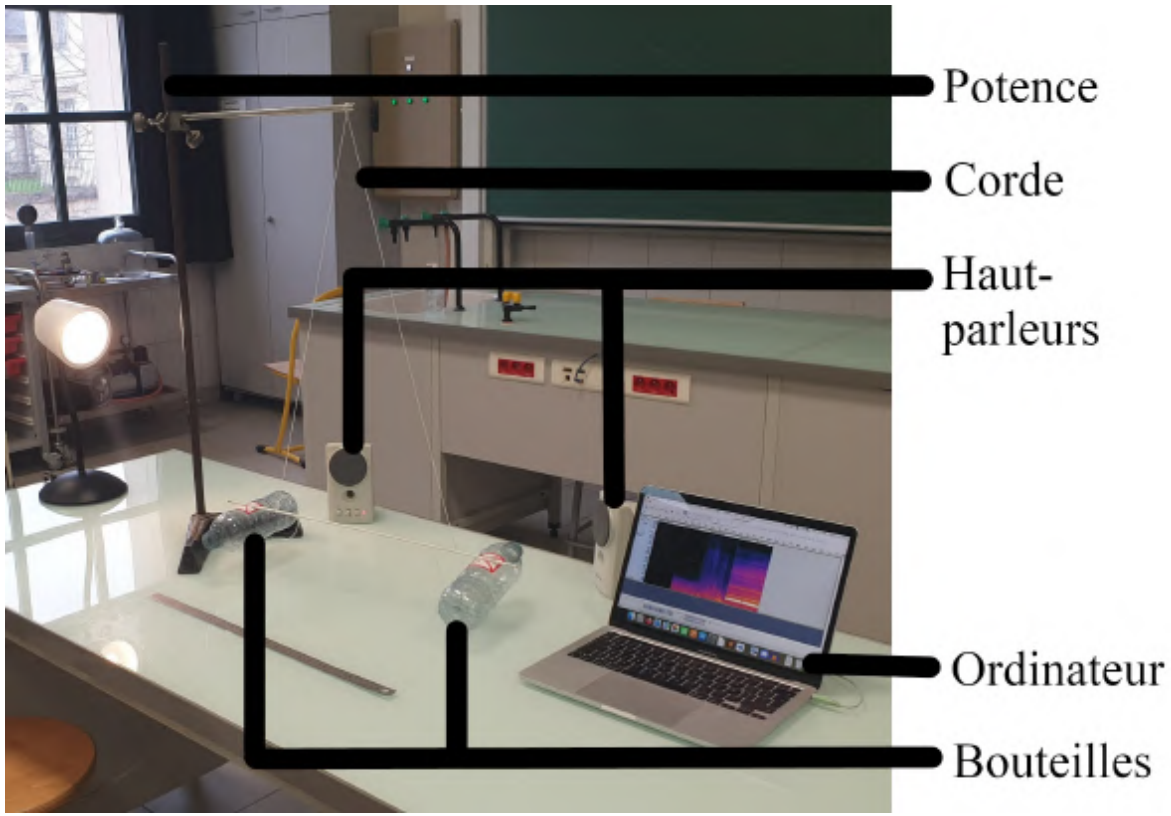


Figure 13 – Photo légendée du montage

Méthodes d'exploitation - Protocole

1. Lancer une capture en vue du dessus du montage
2. Produire une onde sonore monochromatique intense à la fréquence propre
3. Utiliser le pointeur CSRT du module opencv de Python pour effectuer le pointage
4. Calculer numériquement l'angle de la règle par rapport à son orientation de départ, en fonction du temps

Méthodes d'exploitation - Pointeur CSRT

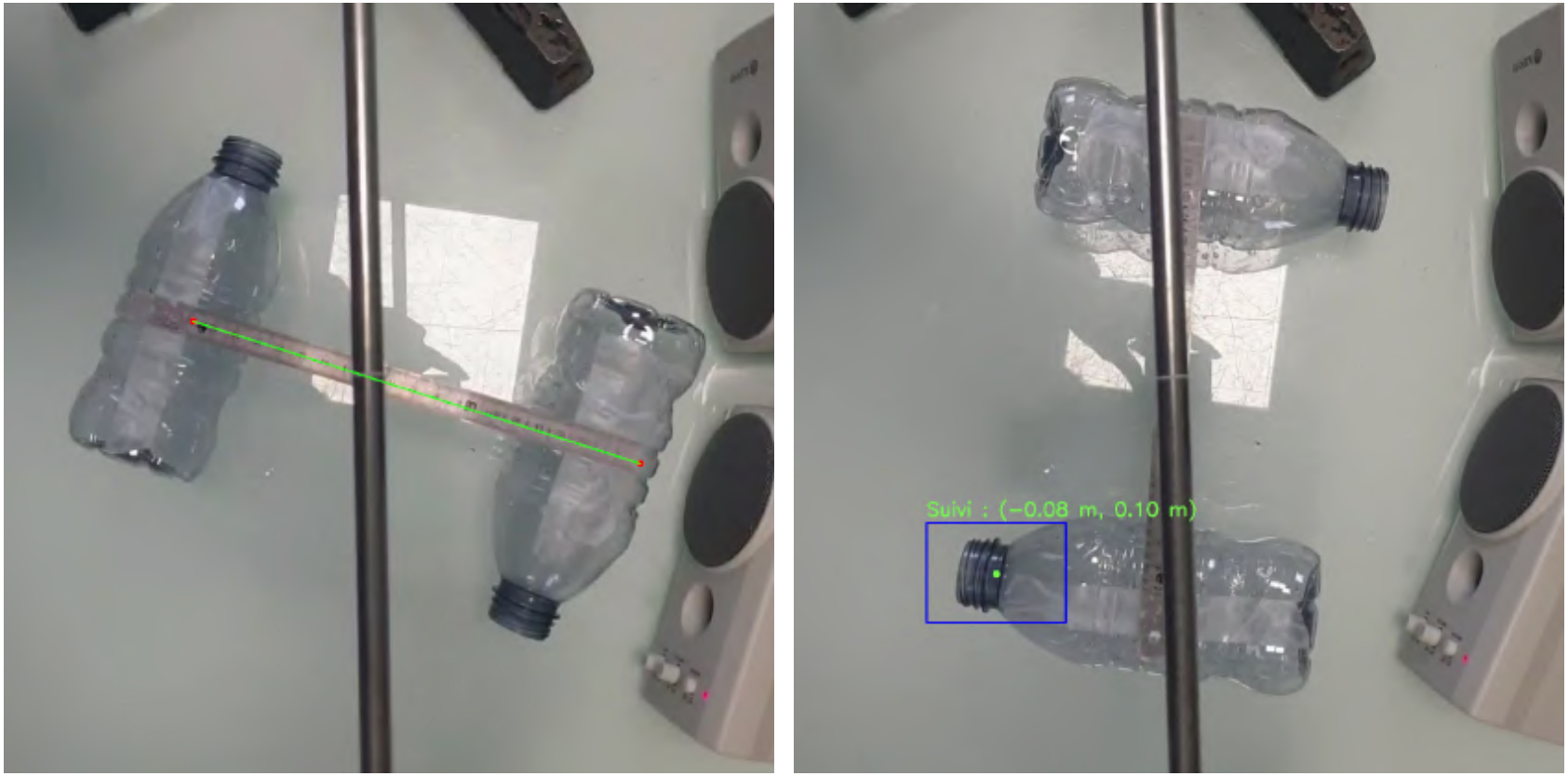


Figure 14 – Fonctionnement du module

Méthodes d'exploitation - Calcul avec tableur

	A	B	C	D	E
1	Frame	Temps (en s)	X (en mètres)	Y (en mètres)	Angle (en rad)
2	1	0.02	-0.053034500969214	-0.0820423219413256	0
3	2	0.03	-0.053034500969214	-0.0820423219413256	0
4	3	0.05	-0.053034500969214	-0.0821009235998551	-0.00032549
5	4	0.07	-0.0530931026277435	-0.0820892032681492	0.0002430737
6	5	0.08	-0.0529641789789786	-0.082138428661314	-0.001138105
7	6	0.10	-0.0529196417184962	-0.0822364106343753	-0.002063833
8	7	0.12	-0.0529426135686397	-0.0823147962128244	-0.00229986
9	8	0.13	-0.0529609910487546	-0.0823775046755837	-0.00248841
10	9	0.15	-0.0529756930328465	-0.0824276714457911	-0.002639077
11	10	0.17	-0.0530460562786495	-0.0824092032034275	-0.001933227
12	11	0.18	-0.0532195501923509	-0.0823944286095366	-0.000364415
13	12	0.20	-0.0532411420062531	-0.0823240072758944	0.0002103089
14	13	0.22	-0.0532584154573748	-0.0822090685504511	0.000995609
15	14	0.23	-0.0532722342182721	-0.082058515911567	0.001908503
16	15	0.25	-0.05328328922699	-0.0818208704834006	0.0033710945
17	16	0.27	-0.0532921332339643	-0.081572152482338	0.0048401187
18	17	0.28	-0.0534164117566028	-0.0813731780814879	0.0070270017
19	18	0.30	-0.0535158345747136	-0.0810967952437488	0.0094434051
20	19	0.32	-0.0535953728292022	-0.0807584856564985	0.0120508145
21	20	0.33	-0.0536590034327931	-0.0804878379866982	0.0141459776
22	21	0.35	-0.0537685095741954	-0.080271319850658	0.0163321495
23	22	0.37	-0.0539733178043762	-0.0802153086592449	0.0184145524
24	23	0.38	-0.0541371643885209	-0.0801704997059544	0.020078372
25	24	0.40	-0.0543268433143661	-0.080134652543322	0.0219092654
26	25	0.42	-0.0544199847965128	-0.0801059748132161	0.0228715188
27	26	0.43	-0.0545530996407597	-0.0802002359461904	0.0234606202
28	27	0.45	-0.0546009898576276	-0.0802170431940403	0.0237712762
29	28	0.47	-0.0547565053481811	-0.0803476923093793	0.0243374038
30	29	0.48	-0.0548809177406238	-0.0804522116016505	0.024788669
31	30	0.50	-0.054980447654578	-0.0805358270354674	0.0251486386

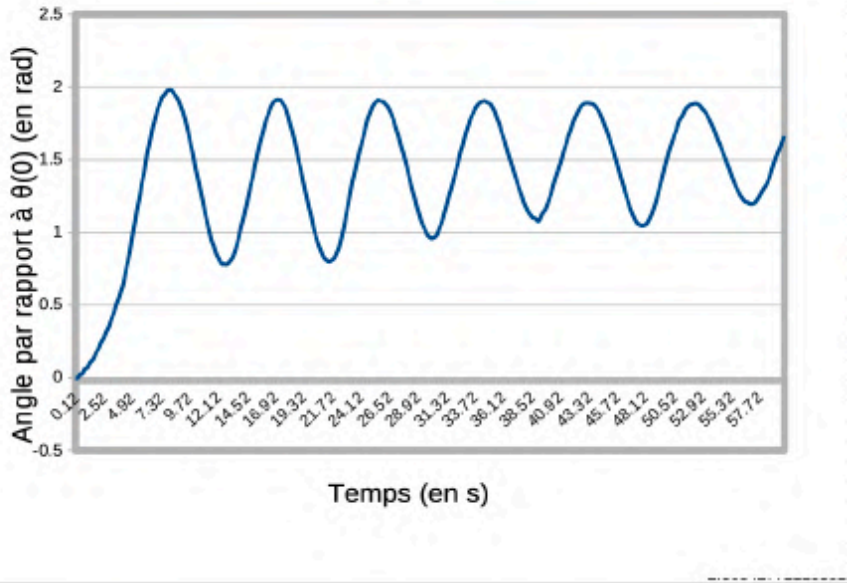


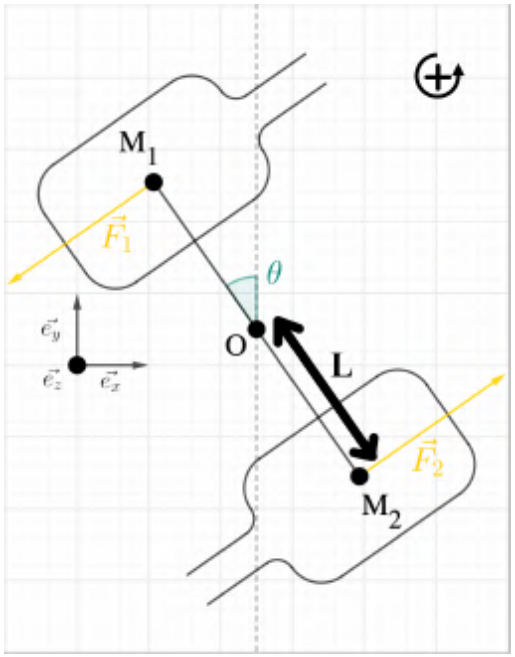
Figure 15 – Exploitation numérique des données

Difficultés rencontrées

1. Il a été difficile de choisir le bon algorithme de pointage
2. Il a fallu utiliser des haut-parleurs très puissants
3. Il a fallu faire attention aux effets de saturation
4. Maintenir le dispositif en équilibre était crucial

Exploitation

Rappel avant l'exploitation : équation du mouvement



Bilan des actions exercées sur le mécanisme :

- Les forces $\vec{F_1}$ et $\vec{F_2}$ → action des surpressions sonores.
  - On note M_F leur moment selon $\vec{e_z}$ en O.
- Un couple $-h\dot{\theta}\vec{e_z}$ → forces de frottements $\vec{f_1}$ et $\vec{f_2}$
- Un couple $-C\theta\vec{e_z}$ → torsion de la corde

On a alors :

$$\boxed{\ddot{\theta} + \frac{\omega_1}{Q}\dot{\theta} + \omega_1^2\theta = \omega_1^2\theta_\infty}$$

où $\theta_\infty = \frac{M_F}{C}, \omega_1 = \sqrt{\frac{C}{J}}, Q = \frac{\sqrt{JC}}{h}$, J le moment d'inertie du système {Bouteilles+Règle} par rapport à l'axe de la corde

Exploitation : cas $\Delta < 0$

Pour obtenir $\Delta < 0$: maximiser la torsion de la corde $\rightarrow$ diminuer sa longueur

Si $\Delta < 0$, la solution générale de l'équation est de la forme

$$\boxed{\theta(t) = \theta_\infty + \mathbf{A} e^{-\frac{\omega_1 t}{2Q}} cos(\omega_1 t + \phi)}$$

Exploitation : cas $\Delta < 0$

On obtient les résultats suivants :

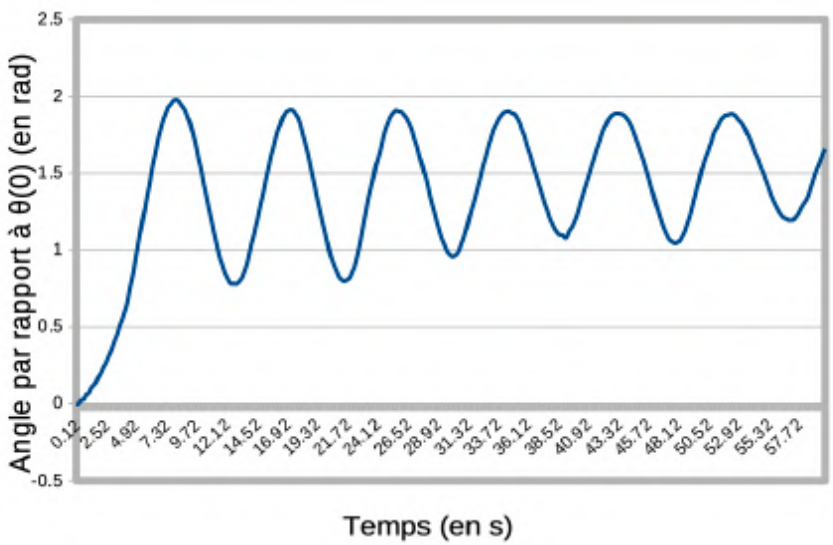


Figure 16 – $\theta_\infty = 1,45$ radians,
$\omega_1 = 0,7$ rad/s, $Q = 119$

Exploitation : cas $\Delta < 0$

On a :

$$\boxed{M_F = J\theta_\infty \omega_1^2}$$

Avec, $J \approx 6 \cdot 10^{-4}$ kg·m^2 :

$$\boxed{M_F \approx 4,3 \cdot 10^{-4} \text{ N} \cdot \text{m}}$$

Donc, en notant L la demi-longueur de la règle :

$$\boxed{||\vec{F_1}|| = ||\vec{F_2}|| = \frac{M_F}{2L} \approx 2.2 \cdot 10^{-3} \text{ N}}$$

Exploitation : cas $\Delta > 0$

Il est plus difficile d'établir M_F dans le cas $\Delta > 0$.

Dans ces conditions, la solution générale de l'équation est de la forme

$$\boxed{\theta(t) = \theta_\infty + \mathbf{A}e^{r_+ t} + \mathbf{B}e^{r_- t}}$$

où

$$r_-^+ = -\frac{\omega_1}{Q} \pm \omega_1 \sqrt{\frac{1}{4Q^2} - 1}$$

Exploitation : cas $\Delta > 0$

On obtient les résultats suivants :

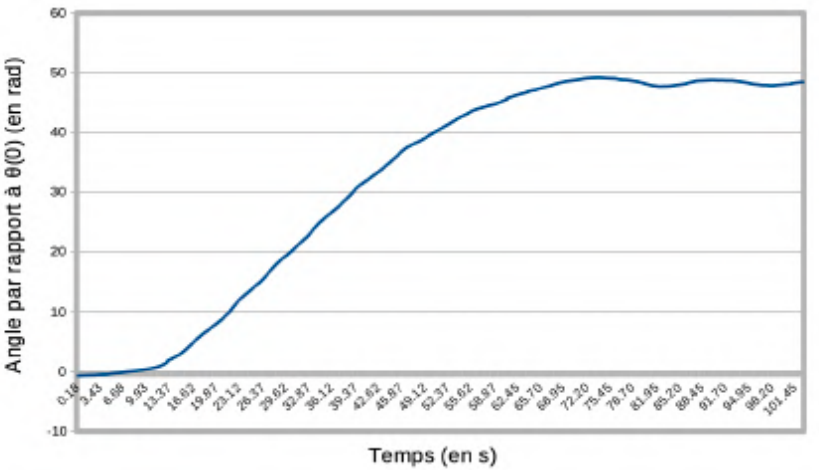


Figure 17 – Pour obtenir les valeurs de θ_∞, ω_1, et Q, il est plus difficile d'exploiter directement le graphe de la fonction.

Expérience : cas $\Delta > 0$

On utilise le module scipy de Python pour la comparaison, avec la **méthode des moindres carrés**.

- Choisir une forme générale pour la fonction $\theta_{\text{théorique}}(t)$ recherchée :

$$\boxed{\theta(t) = \theta_\infty + \mathbf{A}e^{r_+ t} + \mathbf{B}e^{r_- t}}$$

- Initialiser les constantes θ_∞,r_+,r_-,$\mathbf{A}$,$\mathbf{B}$ à des valeurs aléatoires
- Minimiser l'écart quadratique entre $\theta_{\text{théorique}}(t)$ et $\theta_{\text{expérimental}}(t)$ par petites variations des constantes
- Reproduire les deux étapes précédentes et conserver la meilleure fonction obtenue

Expérience : cas $\Delta > 0$

On obtient :

$$\boxed{\theta_\infty = 48,35 \text{ radians}, \omega_1 = 0,08 \text{ rad/s}, Q \lessapprox 1/\sqrt{2}}$$

On constate que Q est très proche de la valeur limite $1/\sqrt{2}$...

Expérience : cas $\Delta > 0$

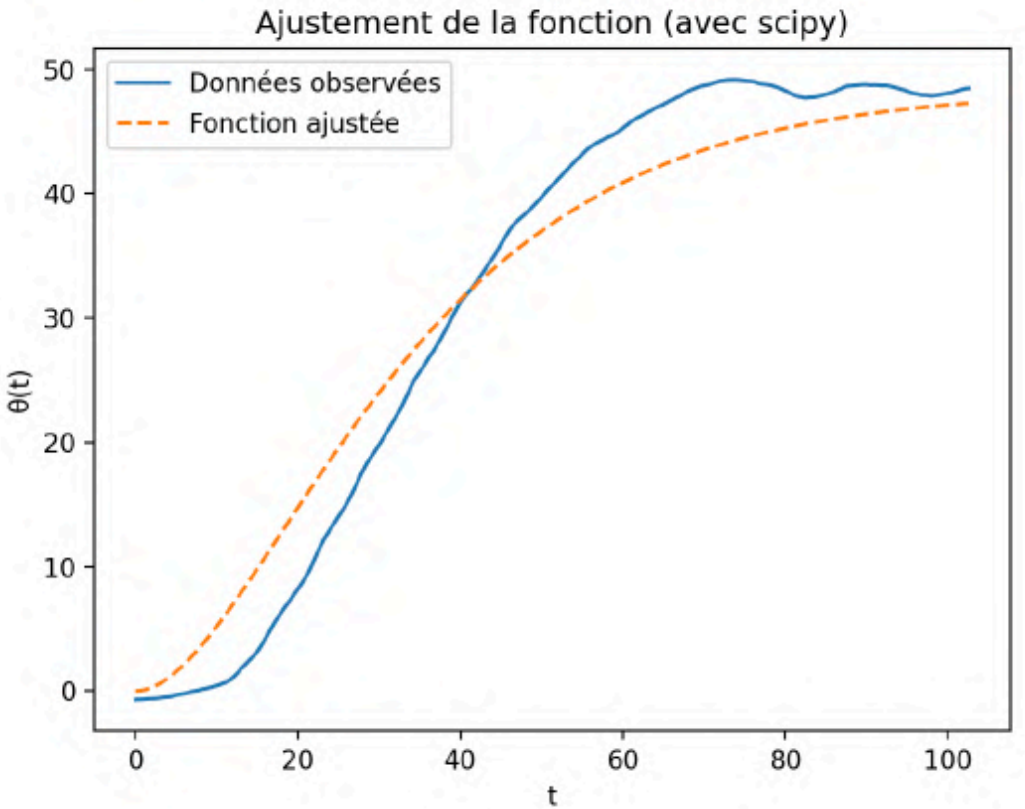


Figure 18 – Comparaison avec les résultats théoriques pour de telles valeurs de $\theta_\infty, \omega_1, Q$

Expérience : cas $\Delta > 0$

On établit de même :

$$\boxed{||\vec{F_1}|| = ||\vec{F_2}|| = \frac{J\theta_\infty \omega_1^2}{2L} \approx 1.0 \cdot 10^{-3} \text{ N}}$$

Conclusion

Conclusion

Résonateurs de Helmholtz :

- Conversion efficace de l'énergie d'une onde sonore en énergie cinétique macroscopique.
- Applications en bio-ingénierie, micro-ingénierie...

Applications

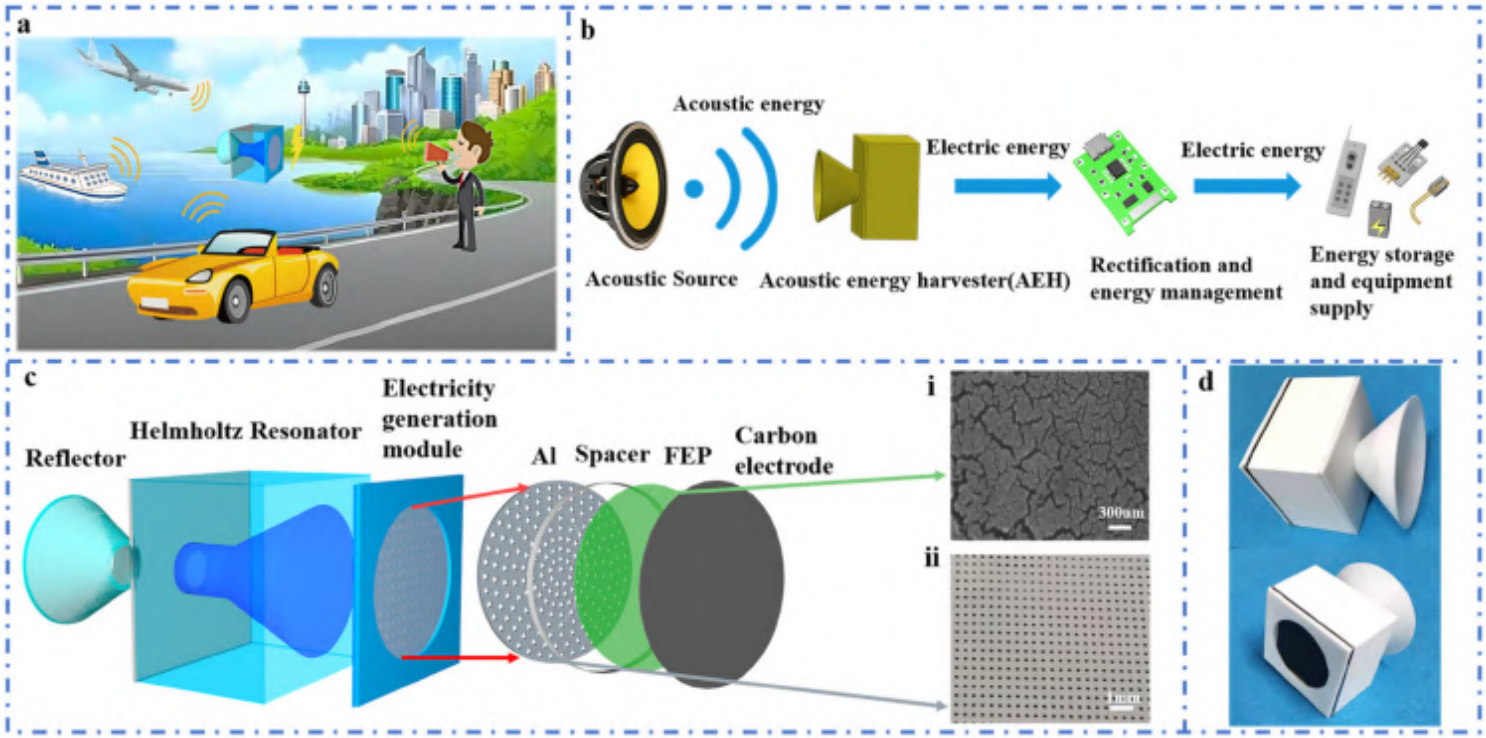


Figure 19 – Le nanogénérateur chinois H-Teng fonctionne grâce à la résonance de Helmholtz

Source de l'image : présentation de l'ingénieur Ling Lui lors de la *16th International Conference on Nano/Micro Engineered and Molecular Systems* de l'IEEE, en 2021

Merci pour votre attention.

Code Python

```
import matplotlib.pyplot as plt
import numpy as np

t = np.arange(0.,1000.,0.1)
plt.plot(t, 20*np.log10(1/abs(500**2-t**2)))
plt.xlabel('Pulsation (rad/s)')
plt.ylabel('Gain décibel')
plt.show()
```

Figure 20 – Code qui a permis la visualisation du diagramme de Bode en figure 8

Code Python

```python
import cv2
import math
import numpy as np
import csv

# Fonction pour lisser la position à l'aide de la moyenne mobile exponentielle
def smooth_position(prev_pos, new_pos, alpha=0.1):
    """Lisse la nouvelle position avec une moyenne mobile exponentielle"""
    if prev_pos is None:
        return new_pos
    else:
        smooth_x = alpha * new_pos[0] + (1 - alpha) * prev_pos[0]
        smooth_y = alpha * new_pos[1] + (1 - alpha) * prev_pos[1]
        return (smooth_x, smooth_y)

# Initialiser la capture vidéo
cap = cv2.VideoCapture('Video.mp4')
# Lire la première image
ret, frame = cap.read()
# Laisser l'utilisateur sélectionner manuellement la ROI (Région d'Intérêt) (dessiner un rectangle autour de l'objet)
roi = cv2.selectROI("Sélectionnez l'objet à suivre", frame, fromCenter=False, showCrosshair=True)
# Définir automatiquement l'origine au centre de l'image
frame_height, frame_width = frame.shape[:2]
origin_x, origin_y = frame_width // 2, frame_height // 2
cv2.destroyAllWindows()
# Demander à l'utilisateur de sélectionner deux points pour mesurer la distance de référence (échelle)
print("Sélectionnez deux points sur l'objet de référence pour définir l'échelle.")
# Créer une liste pour stocker les points sélectionnés
points = []
# Définir l'échelle globalement
scale = None
```

Figure 21 – Code qui a permis l'exploitation de la vidéo de l'expérience, partie I

Code Python

```python
def click_event(event, x, y, flags, param):
    """ Fonction de rappel pour stocker les points cliqués et calculer l'échelle """
    global points, scale, frame
    if event == cv2.EVENT_LBUTTONDOWN:
        points.append((x, y))
        cv2.circle(frame, (x, y), 5, (0, 0, 255), -1)
        cv2.imshow("Définir l'échelle", frame)
        # Une fois deux points sélectionnés, calculer l'échelle
        if len(points) == 2:
            # Obtenir les coordonnées des deux points
            point1, point2 = points
            # Dessiner la ligne reliant les points
            cv2.line(frame, point1, point2, (0, 255, 0), 2)
            cv2.imshow("Définir l'échelle", frame)
            cv2.waitKey(0)
            cv2.destroyAllWindows()
            # Demander à l'utilisateur la distance réelle entre les deux points
            distance = float(input("Entrez la distance réelle entre les deux points sélectionnés (en mètres) : "))
            pixel_distance = math.dist(point1, point2)  # Calculer la distance en pixels
            scale = distance / pixel_distance  # Calculer l'échelle en mètres par pixel
            print(f"Échelle : {scale} mètres par pixel.")

cv2.imshow("Définir l'échelle", frame)
cv2.setMouseCallback("Définir l'échelle", click_event)
# Attendre que deux points soient sélectionnés et que l'échelle soit calculée
cv2.waitKey(0)
cv2.destroyAllWindows()
# S'assurer que l'échelle est définie avant de continuer
if scale is None:
    print("Échelle non définie, arrêt du programme.")
    cap.release()
    cv2.destroyAllWindows()
    exit()
# Initialiser le tracker
tracker = cv2.TrackerCSRT_create()
# Initialiser le tracker avec la première image et la ROI sélectionnée
tracker.init(frame, roi)
# Variable pour le lissage
prev_position = None
# Ouvrir le fichier CSV pour écrire les coordonnées
csv_file = open('tracked_coordinates_bottlecap.csv', mode='w', newline='')
csv_writer = csv.writer(csv_file)
# Écrire l'en-tête dans le fichier CSV
csv_writer.writerow(['Frame', 'X (mètres)', 'Y (mètres)'])
frame_count = 0
```

```python
while True:
    ret, frame = cap.read()
    if not ret:
        break
    frame_count += 1
    # Mettre à jour le tracker avec la nouvelle image
    success, roi = tracker.update(frame)

    if success:
        # Si le suivi est réussi, dessiner le rectangle autour de l'objet
        x, y, w, h = [int(v) for v in roi]
        # Calculer le centre de l'objet (rectangle de suivi)
        centerX = int(x + w / 2)
        centerY = int(y + h / 2)
        # Lisser la position du centre
        smooth_center = smooth_position(prev_position, (centerX, centerY), alpha=0.2)
        prev_position = smooth_center  # Mettre à jour la position précédente avec la position lissée
        # Convertir les coordonnées en pixels en mètres par rapport à l'origine
        real_world_X = (smooth_center[0] - origin_x) * scale
        real_world_Y = (smooth_center[1] - origin_y) * scale
        # Écrire les coordonnées dans le fichier CSV
        csv_writer.writerow([frame_count, real_world_X, real_world_Y])
        # Dessiner le rectangle et le point central
        cv2.rectangle(frame, (x, y), (x + w, y + h), (255, 0, 0), 2)
        # Dessiner un petit cercle au centre
        cv2.circle(frame, (int(smooth_center[0]), int(smooth_center[1])), 5, (0, 255, 0), -1)
        cv2.putText(frame, f"Suivi : ({real_world_X:.2f} m, {real_world_Y:.2f} m)",
                    (x, y - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.9, (0, 255, 0), 2)
    else:
        # Si le suivi échoue, tenter de réinitialiser le tracker
        cv2.putText(frame, "Suivi perdu. Réinitialisation...", (20, 30),
                    cv2.FONT_HERSHEY_SIMPLEX, 0.9, (0, 0, 255), 2)
        # Permettre à l'utilisateur de sélectionner à nouveau l'objet
        roi = cv2.selectROI("Re-sélectionnez l'objet à suivre", frame, fromCenter=False, showCrosshair=True)
        tracker.init(frame, roi)  # Réinitialiser le tracker avec la nouvelle ROI

    # Afficher le résultat
    cv2.imshow("Suivi d'objet", frame)

    # Quitter en appuyant sur la touche 'ÉCHAP'
    if cv2.waitKey(1) & 0xFF == 27:
        break

# Libérer la capture vidéo et fermer les fenêtres OpenCV
cap.release()
# Fermer le fichier CSV après avoir écrit toutes les coordonnées
csv_file.close()
cv2.destroyAllWindows()
```

Figure 22 – Code qui a permis l'exploitation de la vidéo de l'expérience, partie II

Code Python

```python
import numpy as np
import pandas as pd
from scipy.optimize import least_squares
import matplotlib.pyplot as plt

# Définir la fonction à ajuster
def model(t, a, b):
    theta = 48.3453662431804  # constante fixée
    return theta * (1 - (b * np.exp(a * t) - a * np.exp(b * t)) / (b - a))
# Fonction de résidus pour least_squares
def residuals(params, t, y):
    a, b = params
    return model(t, a, b) - y

data = pd.read_csv('AAAAAAA.csv')
# Nettoyer les données
data = data[['t', 'f']].replace([np.inf, -np.inf], np.nan).dropna()
# Extraire les données
t_data = data['t'].values
f_data = data['f'].values

# Estimation initiale
initial_guess = [-0.0541294687745729, -0.0239127232659019]
```

Figure 23 – Code qui a permis l'application de la méthode des moindres carrés, partie I

Code Python

```python
# Estimation initiale
initial_guess = [-0.0541294687745729, -0.0239127232659019]

# Ajustement par moindres carrés
result = least_squares(
    residuals,
    x0=initial_guess,
    args=(t_data, f_data),
    loss='linear',     # robuste aux outliers
    max_nfev=10000000
)

# Extraire les paramètres estimés
a_est, b_est = result.x
print(f"Paramètres estimés avec least_squares : a = {a_est:.6f}, b = {b_est:.6f}")

# Tracer les données et la courbe ajustée
plt.plot(t_data, f_data, label="Données observées")
plt.plot(t_data, model(t_data, a_est, b_est), label="Fonction ajustée", linestyle='--')
plt.legend()
plt.xlabel("t")
plt.ylabel("θ(t)")
plt.title("Ajustement de la fonction (avec scipy)")
plt.show()
```

Figure 24 – Code qui a permis l'application de la méthode des moindres carrés, partie II