La récupération d'énergie vibratoire à l'aide d'un couplage électromagnétique et piézoélectrique

Khalil EL AZRI

SCEI: 15526

Plan

Contextualisation

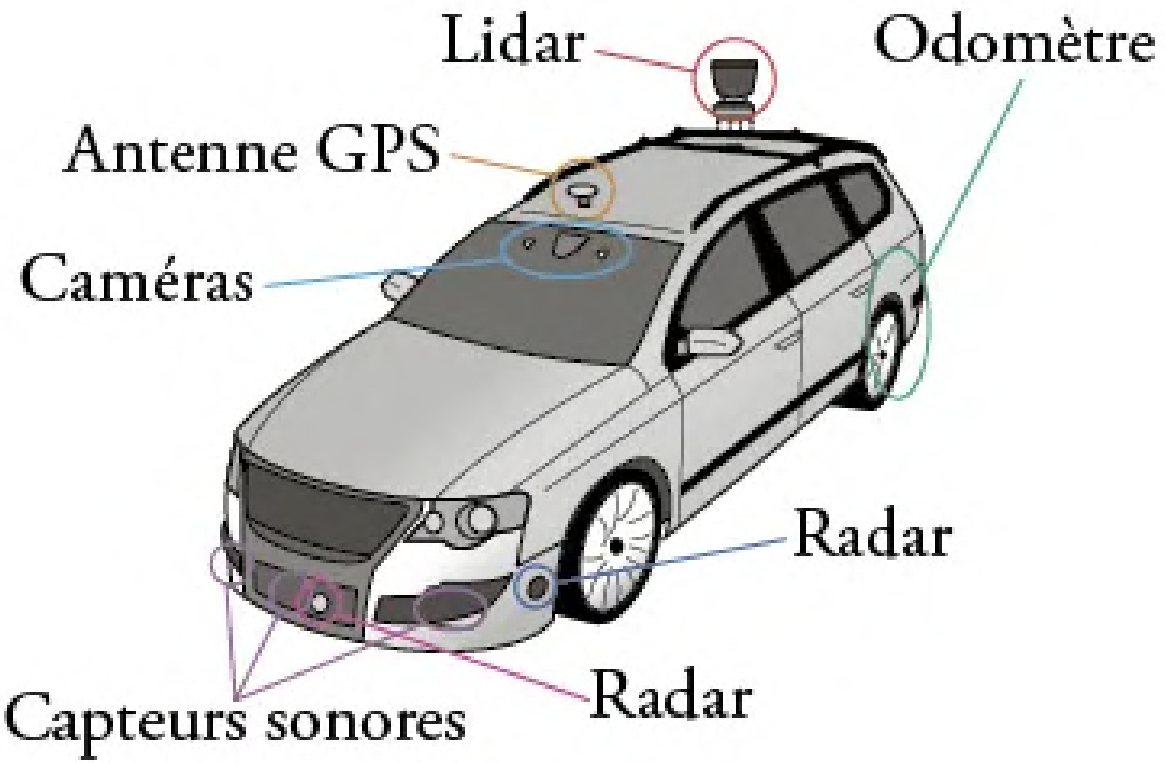


Figure: Placement des différents capteurs sur une voiture

Définition

Matériau Piézoélectrique :

Un matériau piézoélectrique (presser) est par définition capable de convertir :

- l'énergie mécanique en énergie électrique , c'est l'effet piézoélectrique direct

- l'énergie électrique en énergie mécanique , c'est effet piézoélectrique indirect

Problèmatique

Comment peut-on récupérer une telle énergie vibratoire ?

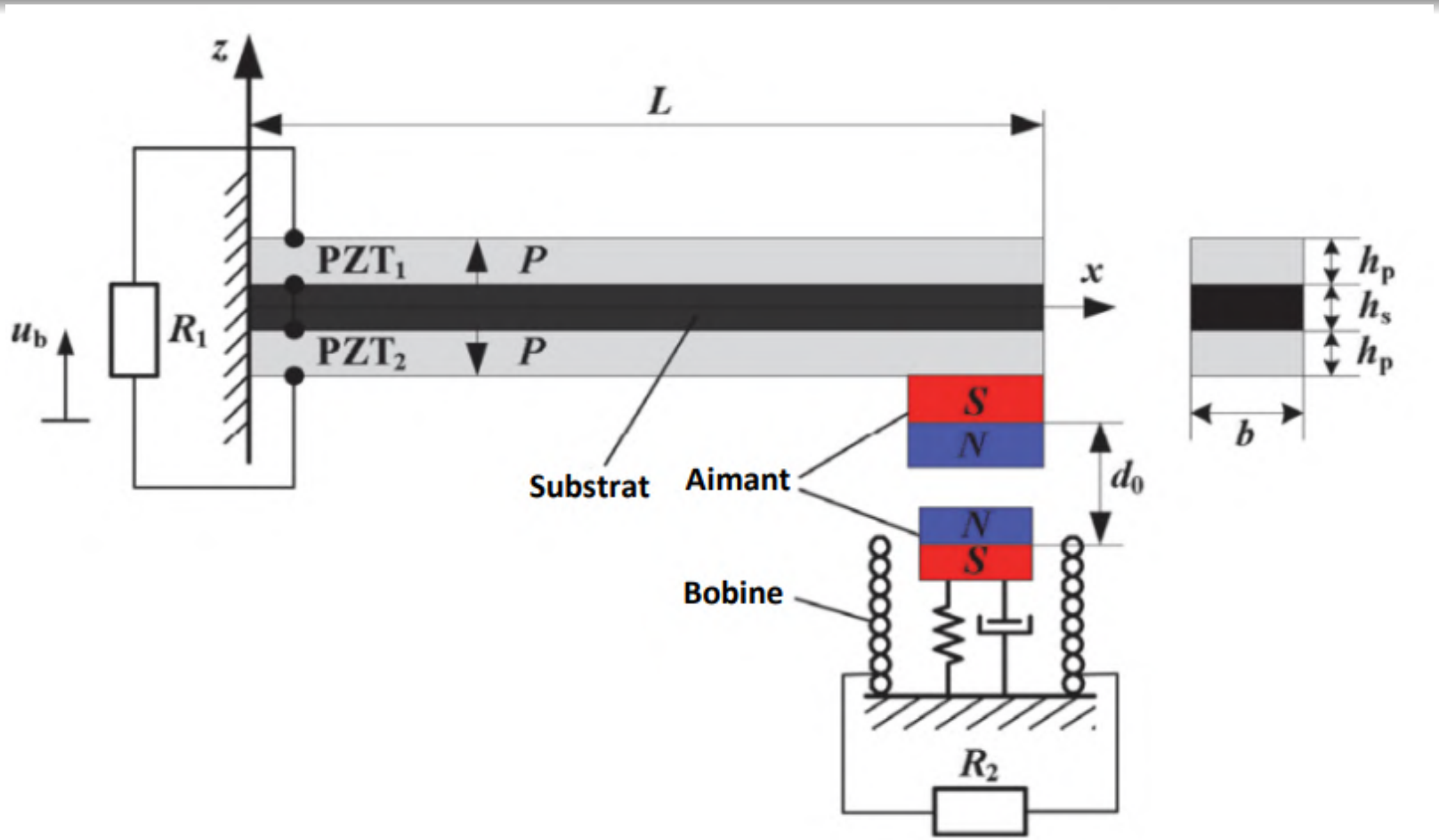


Figure: Schéma de principe du récupérateur d'énergie hybride proposé

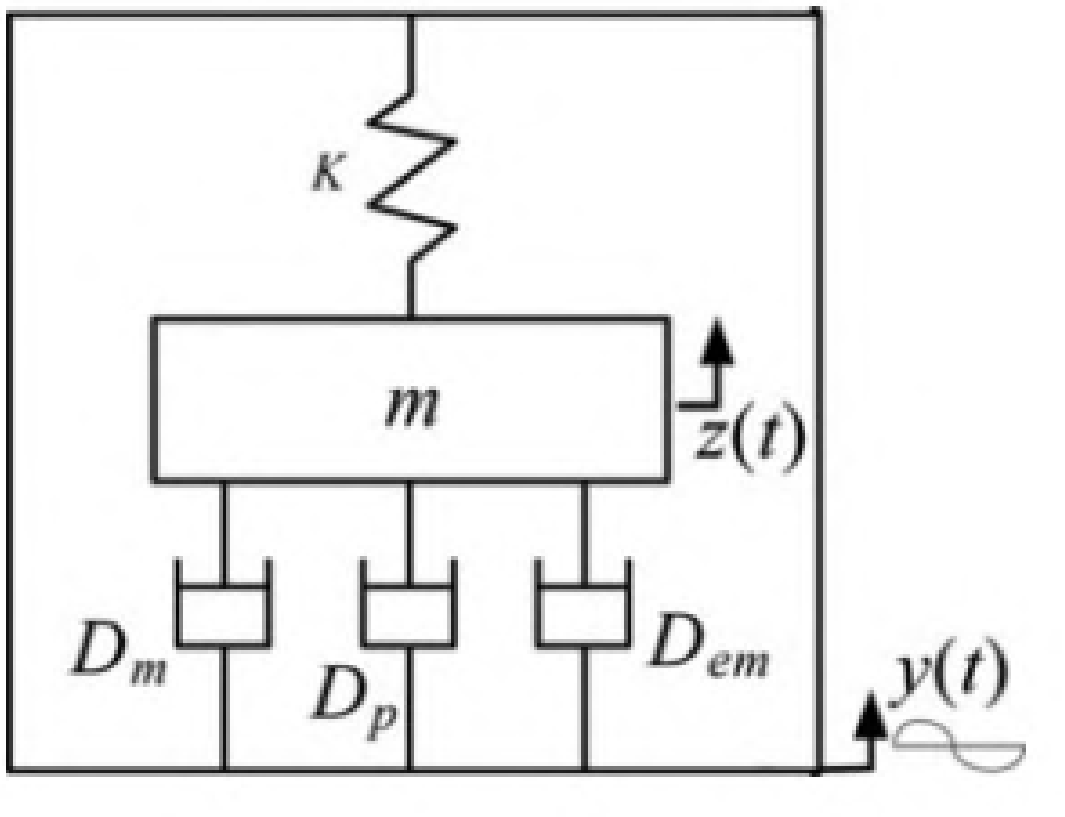


Figure: Modèle équivalent du récupérateur hybride proposé

L'équation différentielle qui décrit ce système est :

$$m\ddot{z} + D\dot{z} + Kz = -m\ddot{y}$$

- m : la masse équivalente du système (Kg)
- K : la rigidité équivalente du système (N/m)
- D : le coefficient d'amortissement équivalent

$$D = D_{em} + Dp + D_m$$

Pour une vibration d'excitation sinusoïdale:

$$y(t) = Y_0 \sin(\omega t)$$

$$z(t) = \frac{Y_0(\frac{\omega}{\omega_n})^2}{\sqrt{(1-(\frac{\omega}{\omega_n})^2)^2 + (2\xi(\frac{\omega}{\omega_n}))^2}} \sin(\omega t - \phi)$$

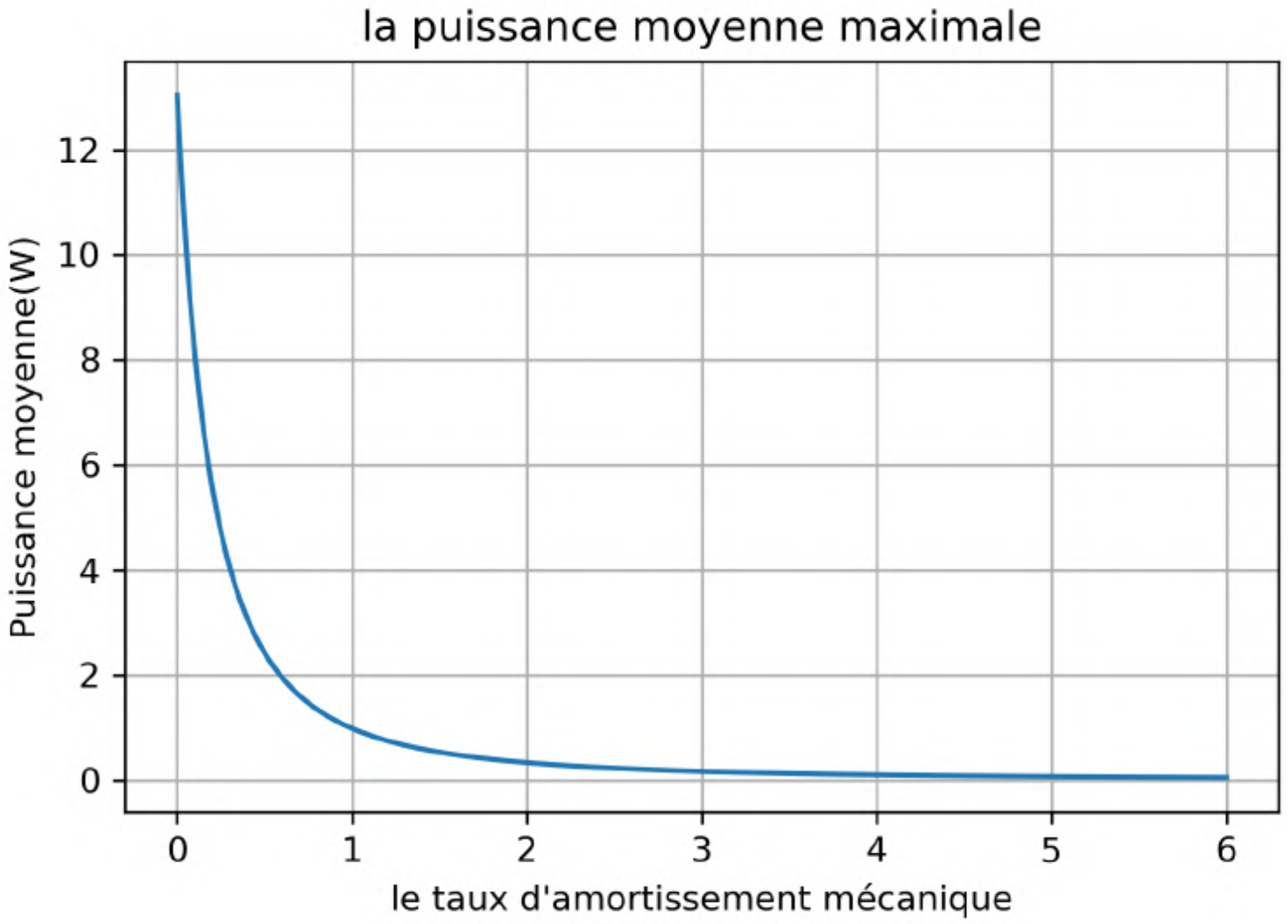
- $\omega_n = \sqrt{\frac{K}{m}}$: La pulsation propre du système (rad/s)
- $\xi = \frac{D}{2m\omega_n}$: Le taux d'amortissement total

La puissance moyenne consommée par la partie électromagnétique et piézoélectrique est :

$$P_{couple} = \frac{1}{T}\int_0^T (D_p + D_{em})\dot{z}dz = \frac{m(\xi_p + \xi_{em})\omega^3(\frac{\omega}{\omega_n})^3 Y_0}{(1-(\frac{\omega}{\omega_n})^2)^2 + (2\xi(\frac{\omega}{\omega_n}))^2} \quad (1)$$

Ainsi à la résonance :

$$P_{couple}^{RS} = \frac{m\omega_n^3(\xi_p + \xi_{em})Y_0^2}{4(\xi_p + \xi_{em} + \xi_m)^2}$$


la puissance moyenne maximale
Puissance moyenne(W)
0
2
4
6
8
10
12
0
1
2
3
4
5
6
le taux d'amortissement mécanique

modélisation

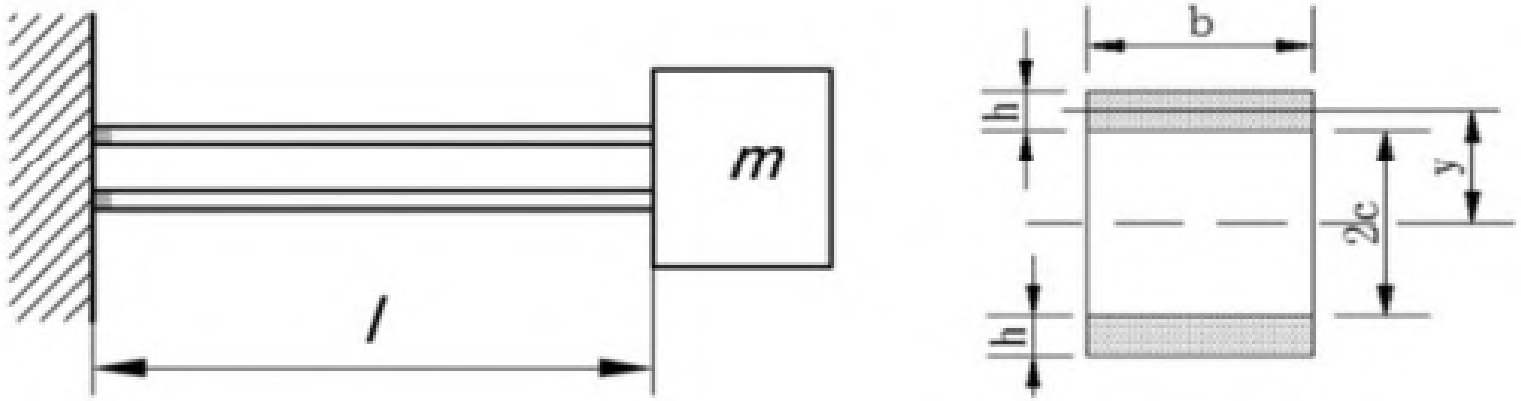


Figure: schéma de structure de la partie piézoélectrique

la tension en circuit ouvert du récupérateur d'énergie piézoélectrique peut être écrite comme :

$$u = 2\frac{hd\sigma}{\varepsilon}$$

- h : l'épaisseur du PZT (m)
- d : la constante de déformation piézoélectrique (C/m)
- σ : la contrainte mécanique (Pa)

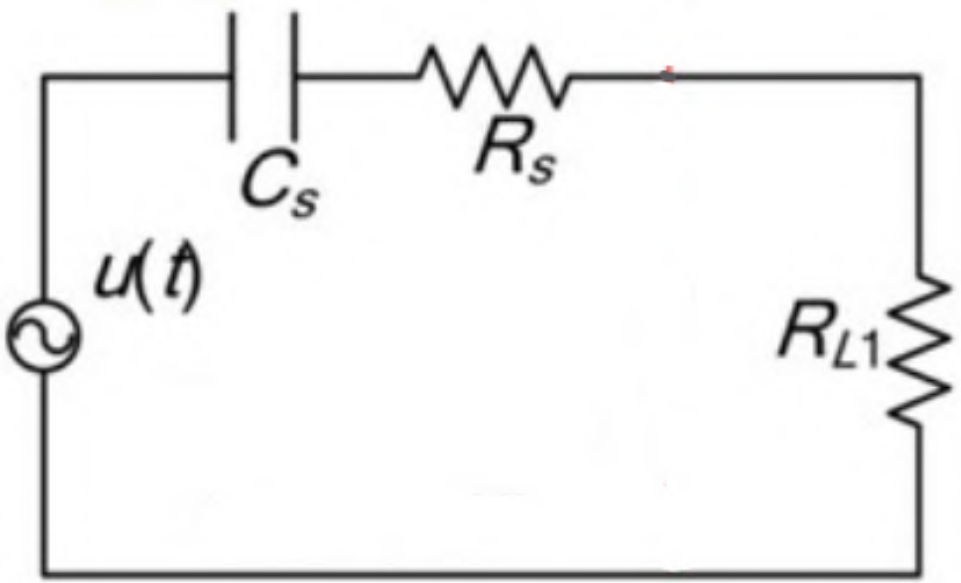


Figure: Modèle équivalent de circuit du récupérateur piézoélectrique

On peut exprimer la puissance du système piézoélectrique par :

$$P_p = \frac{1}{T}\int_0^T \frac{u_1^2}{R_1}dt$$

- on note :

$$u_1 = \frac{R_1}{R_1 + X_s}u$$

avec : $X_s = \frac{1}{2\pi f C_s}$ et $C_s = \frac{\varepsilon bl}{2h}$
en remplaçant dans l'équation (1), on trouve :

$$\xi_p = \frac{1}{8}\frac{mR_1}{(X_s + R_1)^2}(\frac{d(h+2c)hl}{I\varepsilon})^2\omega_n$$

D'après la loi de Faraday :

$$\epsilon_{em} = -\frac{d\phi}{dt} = -NBL\dot{z}$$

- N : le nombre de spires de la bobine
- B : le champs magnétique (T)
- L : la longueur effective d'une spire de la bobine (m)

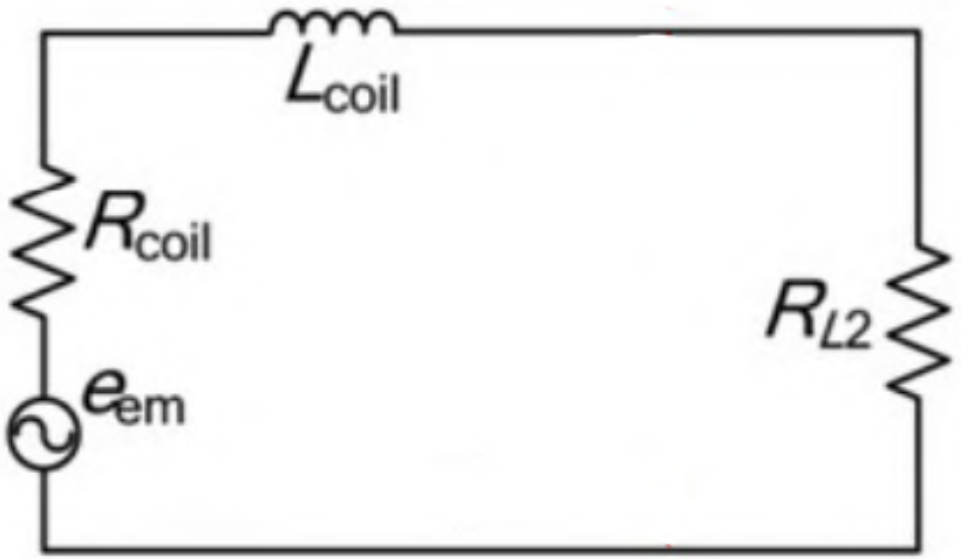


Figure: Modèle équivalent de circuit du récupérateur électromagnétisme

On obtient :

$$P_{em} = \frac{1}{T}\int_0^T \frac{e_{em}^2}{R_b + R_2}dt = \frac{1}{2}\frac{(NBL)^2}{R_b + R_2}\omega^2.Z_0^2$$

en remplaçant dans l'équation (1), on trouve :

$$\xi_{em} = \frac{(NBL)^2}{2m\omega_n(R_b + R_2)}$$

La puissance de la sortie du récupérateur

On conclut que :

$$P_{out} = P_p + P_{em}$$

Ainsi :

$$P_{out} = \frac{1}{2}\frac{R_1}{(X_s + R_1)^2}(\frac{m\omega_n^2 d(h+2c)hl}{2I\epsilon})^2.Z_0^2 + \frac{1}{2}\frac{R_2(NBL)^2}{(R_b+R_2)^2}\omega^2.Z_0^2$$

Avec :

$$Z_0^2 = \frac{Y_0^2(\frac{\omega}{\omega_n})^4}{(1-(\frac{\omega}{\omega_n})^2)^2 + (2\xi(\frac{\omega}{\omega_n}))^2}$$

Simulation

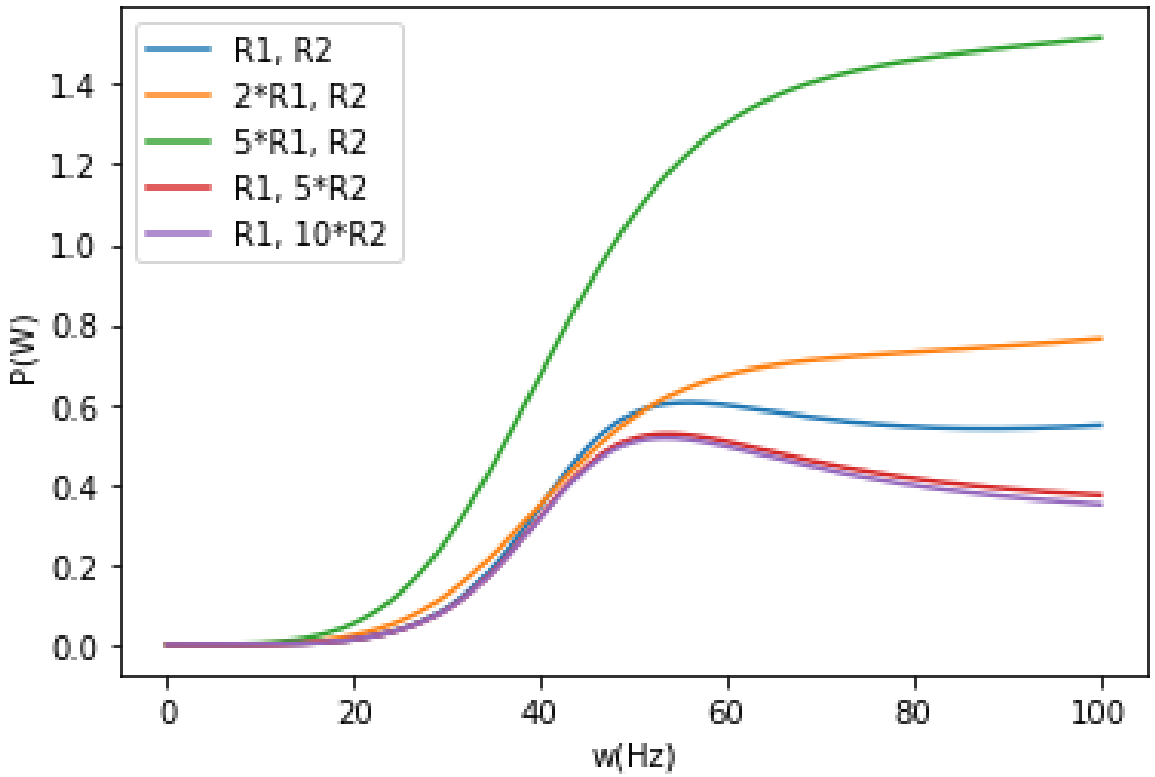


Figure: la variation de la puissance en fonction de la pulsation de l'exitatrice avec $R_1 = 82.5k\Omega$ et $R_2 = 70\Omega$

Expérience

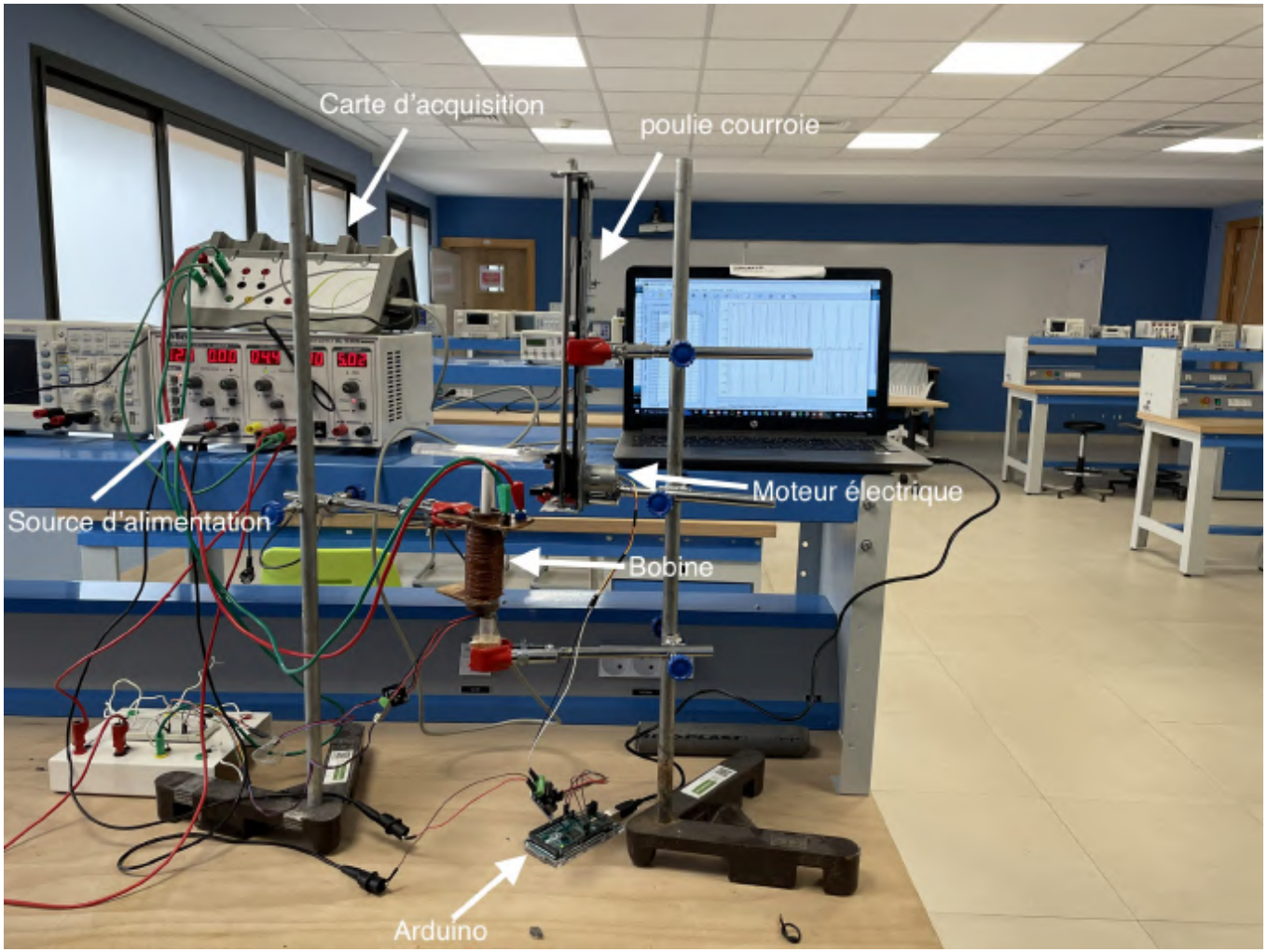
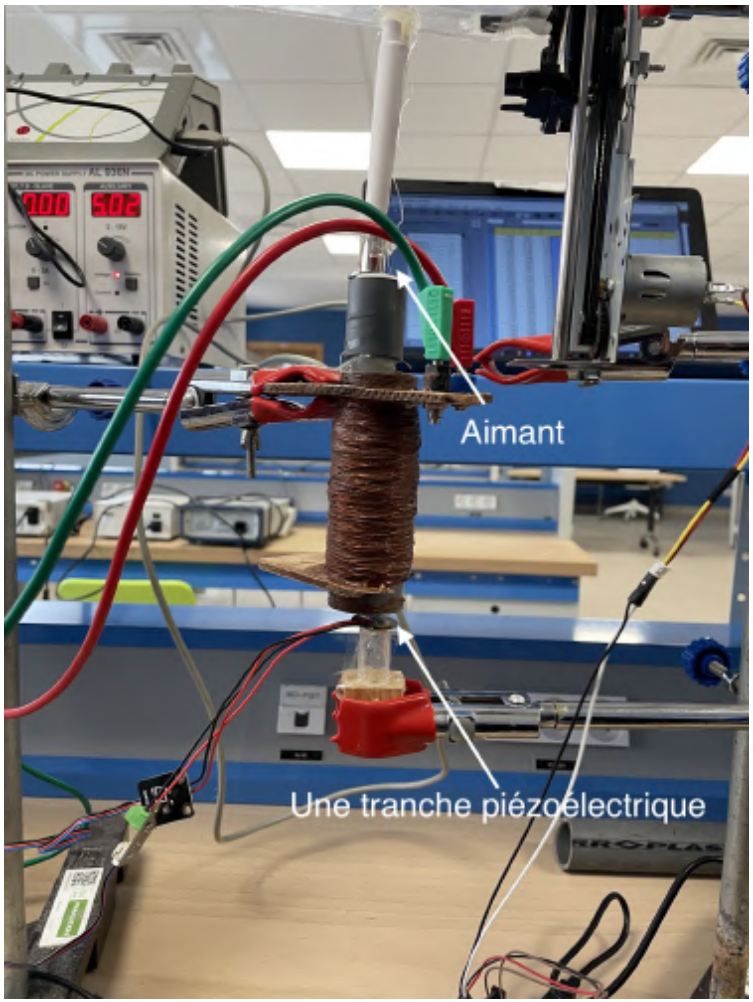


Figure: Montage expérimental



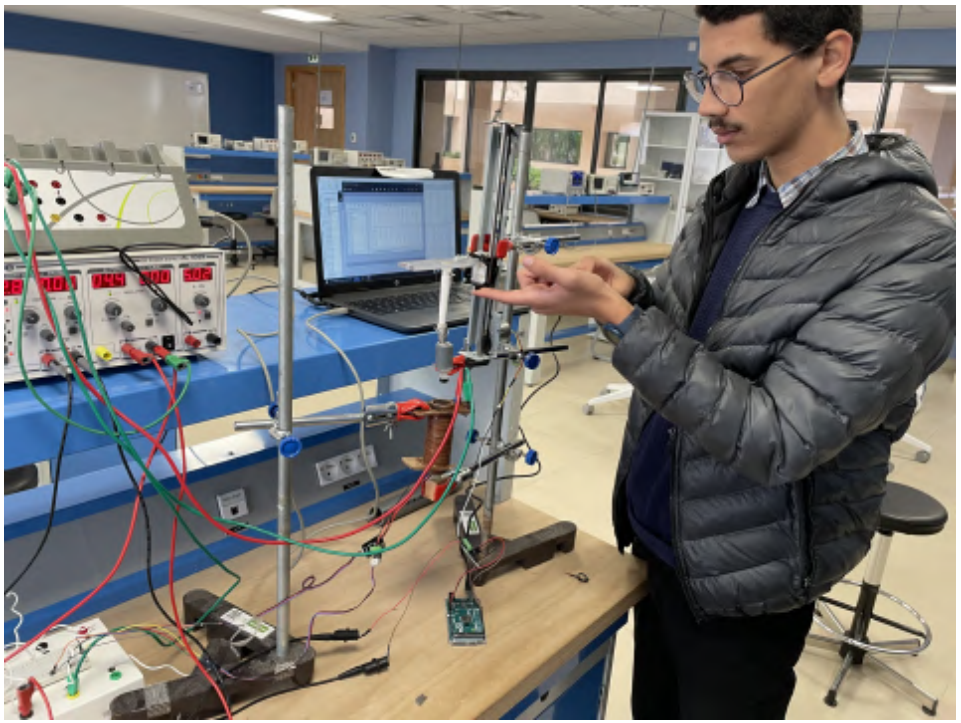


Figure: Montage expérimental

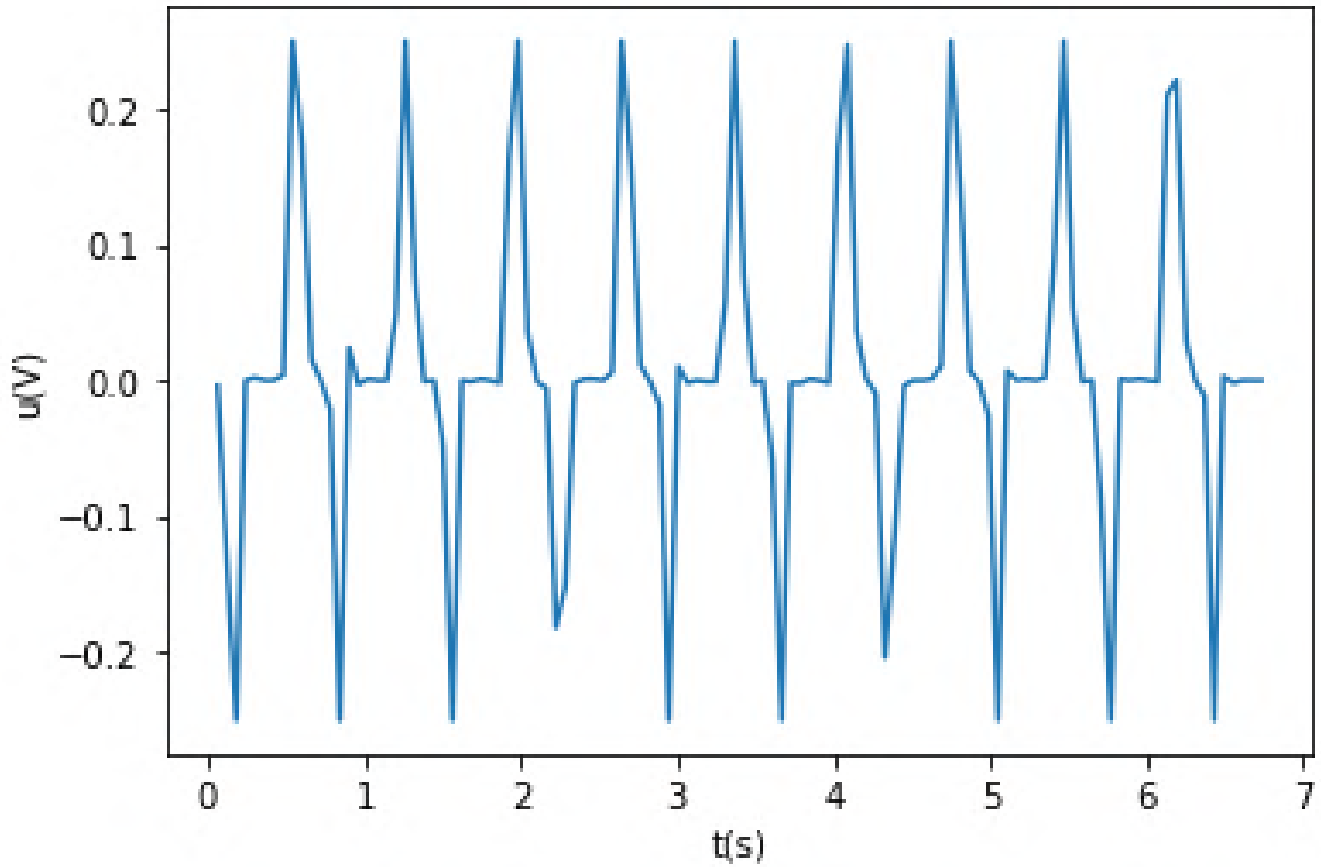


Figure: Résultats expérimentaux

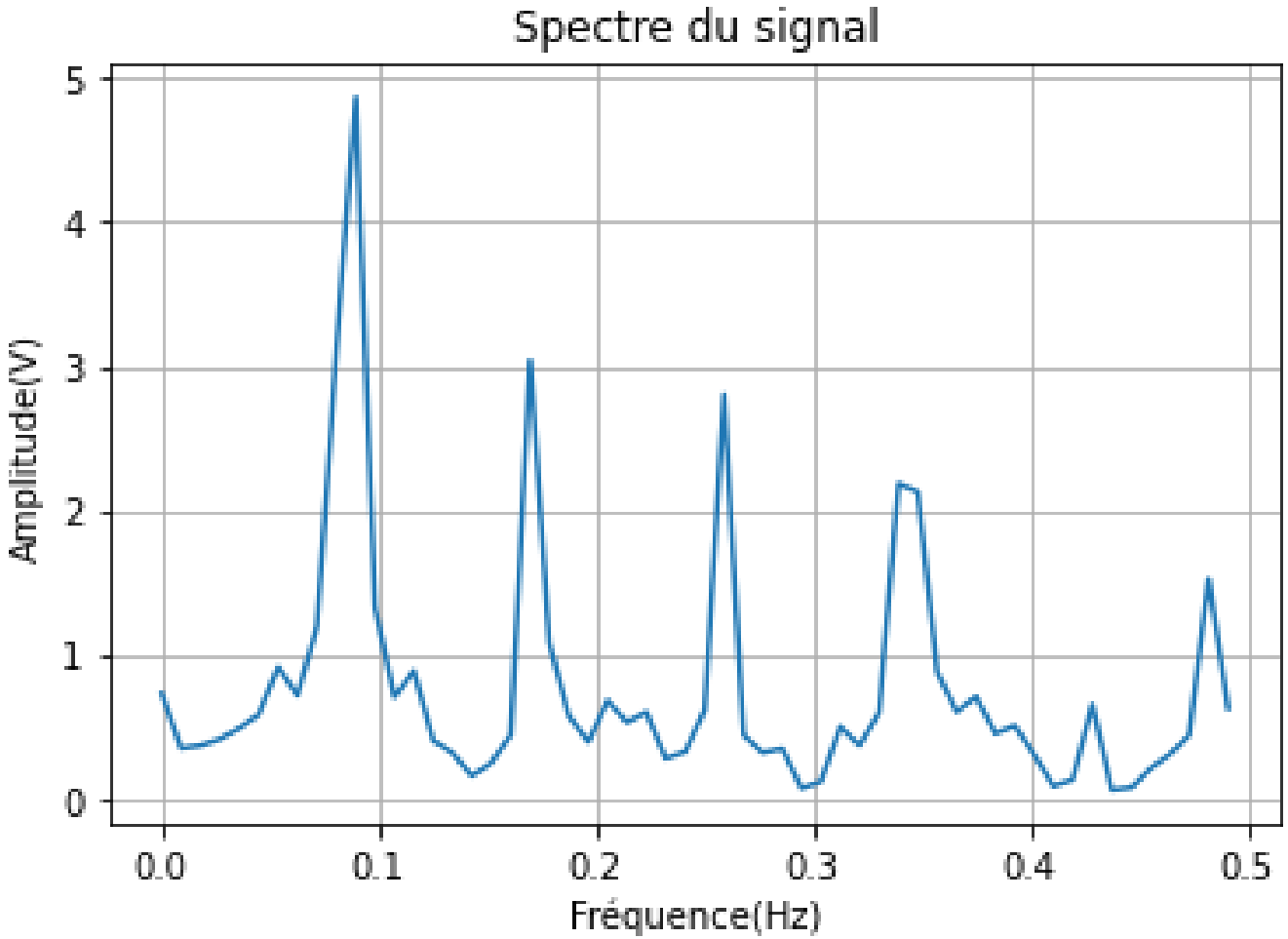


Figure: Résultats expérimentaux

Conclusion

- Résultats expérimentaux et simulation
- Avantages du modèle
- Inconvénients du modèle

Annexes

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

m = 0.0216424
Y0 = 0.1
wn = 45

def puissance(x, y, z):
    return (m*(wn**3)*(x+y)*Y0**2)/(4*(x+y+z)**2)


zp = 0.37664540536591834
zm = 0.001689800323940694
Z = np.linspace(0, 6, 1000)
P = puissance(zp, zm, Z)
plt.plot(Z, P)
plt.xlabel("le taux d'amortissement mécanique")
plt.ylabel('Puissance moyenne(W)')
plt.title('la puissance moyenne maximale')
plt.grid(True)
plt.savefig('figure.png', dpi=300, format='png')

plt.show()
```

Annexes

```python
import numpy as np
import matplotlib.pyplot as plt
from math import pi

B = 1.2
cs = 65*10**(-9)
N = 4000
h = (0.2)*10**(-3)
c = (0.25)*10**(-3)
b = 2*10**(-3)
l = 40*10**(-3)
density = (8.9)*10**3
m1 = density*h*l*b
m0 = 0.0215
m = m0+m1
d = -274*10**(-12)
eps = (3.01)*10**(-8)
sig = 61*10**9
wn = 45
u = 2*h*d*sig/eps
Rb = 1.8*10**(-4)
L = 10**(-4)
g = 9.8
Y0 = 10*10**(-2)
R1 = 82.5*10**3
R2 = 70
Xs = 1/(wn*cs)
I = h*b**3/12
def zp(R1):
    return (R1*wn*((d*(h+2*c)*h*l)/(I*eps))**2)/(8*(Xs+R1)**2)

def zm(R2):
    return (N*B*L)**2/(2*m*wn*(Rb+R2))

```

Annexes

```
34
35  def z(R1,R2):
36      return zm(R2)+zp(R1)
37
38  def z0(R1,R2,w):
39      return (Y0*(w/wn)**2)/((1-(w/wn)**2)**2 +(2*z(R1,R2)*(w/wn))**2)**.5
40
41
42  def P(R1,R2,w):
43      return ((R1*(z0(R1,R2,w)**2)*(m*(wn**2)*d*(h+2*c)*h*l)**2)/(2*((Xs+R2)*2*I*eps)**2))
44              + (R2*((N*B*L*w)**2)*(z0(R1,R2,w)**2))/(2*(Rb+R2)**2)
45
46  R1 = 82.5*10**3
47  R2 = 70
48
49  def Pout(w):
50      return P(R1,R2,w)
51
52  W = np.linspace(0, 100, 50000)
53  Z1 = P(R1,R2,W)
54  Z2 = P(2*R1,R2,W)
55  Z3 = P(5*R1,R2,W)
56  Z4 = P(R1,5*R2,W)
57  Z5 = P(R1,10*R2,W)
58  plt.plot(W,Z1,label ="R1, R2")
59  plt.plot(W,Z2,label ="2*R1, R2")
60  plt.plot(W,Z3,label ="5*R1, R2")
61  plt.plot(W,Z4,label ="R1, 5*R2")
62  plt.plot(W,Z5,label ="R1, 10*R2")
63  plt.legend(loc="upper left")
64  plt.xlabel('w(Hz)')
65  plt.ylabel('P(W)')
66
```

Annexes

```
68
69
70  T = np.array([0.06,0.12,0.18,0.24,0.301,0.361,0.421,0.481,0.541,0.601,0.661,0.721,0.782
71                ,0.842,0.902,0.962,1.022,1.082,1.142,1.202,1.263,1.323,1.383,1.443,1.503,
72                1.563,1.623,1.683,1.743,1.804,1.864,1.924,1.984,2.044,2.104,2.164,2.224,
73                2.285,2.345,2.405,2.465,2.525,2.585,2.645,2.705,2.766,2.826,2.886,2.946,
74                3.006,3.066,3.126,3.186,3.246,3.307,3.367,3.427,3.487,3.547,3.607,3.667,
75                3.727,3.788,3.848,3.908,3.968,4.028,4.088,4.148,4.208,4.269,4.329,4.389,
76                4.449,4.509,4.569,4.629,4.689,4.749,4.81,4.87,4.93,4.99,5.05,5.11,5.17,
77                5.23,5.291,5.351,5.411,5.471,5.531,5.591,5.651,5.711,5.772,5.832,5.892,
78                5.952,6.012,6.072,6.132,6.192,6.253,6.313,6.373,6.433,6.493,6.553,6.613,
79                6.673,6.733])
80  U = np.array([-0.004,-0.13,-0.25,0,0.002,0,0,0.004,0.25,0.175,0.015,0,-0.021,-0.25,0.024,
81                -0.003,0.001,0,0,0.049,0.25,0.074,-0.001,0,-0.047,-0.25,0,-0.001,0.001,0,
82                -0.002,0.164,0.25,0.035,-0.001,-0.006,-0.182,-0.152,-0.002,0.001,0.001,0,
83                0.006,0.25,0.156,0.011,-0.001,-0.018,-0.25,0.01,-0.002,0,-0.001,-0.001,
84                0.058,0.25,0.076,-0.002,0,-0.058,-0.25,-0.002,-0.002,0.001,0,-0.002,0.169,
85                0.247,0.036,0.001,-0.01,-0.204,-0.114,-0.004,0,0,0.001,0.011,0.25,
86                0.144,0.007,-0.001,-0.026,-0.25,0.007,0,0.001,0,0.002,0.088,0.25,
87                0.053,-0.001,0,-0.09,-0.25,0.001,0,0.001,-0.00,-0.001,0.21,0.221,0.029,0,
88                -0.012,-0.25,0.003,-0.002,0,0,0])
89  plt.plot(T,U)
90  plt.xlabel('t(s)')
91  plt.ylabel('u(V)')
92
```

Annexes

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

def plot_signal_spectrum(signal):
    # Calculer le spectre du signal en utilisant la transformée de Fourier discrète (DFT)
    spectrum = np.fft.fft(signal)
    spectrum = spectrum[:56]

    # Calculer les fréquences correspondantes au spectre
    frequencies = np.fft.fftfreq(len(signal))
    frequencies = frequencies[:56]
    # Créer une figure pour afficher le spectre du signal
    plt.figure()
    plt.plot(frequencies, np.abs(spectrum))
    plt.xlabel('Fréquence(Hz)')
    plt.ylabel('Amplitude(V)')
    plt.title('Spectre du signal')
    plt.grid(True)
    plt.show()

signal = [-0.004,-0.13,-0.25,0,0.002,0,0,0.004,0.25,0.175,0.015,0,-0.021,
          -0.25,0.024,-0.003,0.001,0,0,0.049,0.25,0.074,-0.001,0,-0.047,
          -0.25,0,-0.001,0.001,0,-0.002,0.164,0.25,0.035,-0.001,-0.006,
          -0.182,-0.152,-0.002,0.001,0.001,0,0.006,0.25,0.156,0.011,
          -0.001,-0.018,-0.25,0.01,-0.002,0,-0.001,-0.001,0.058,0.25,
          0.076,-0.002,0,-0.058,-0.25,-0.002,-0.002,0.001,0,-0.002,
          0.169,0.247,0.036,0.001,-0.01,-0.204,-0.114,-0.004,0,0,0.001,
          0.011,0.25,0.144,0.007,-0.001,-0.026,-0.25,0.007,0,0.001,0,
          0.002,0.088,0.25,0.053,-0.001,0,-0.09,-0.25,0.001,0,0.001,
          -0.00,-0.001,0.21,0.221,0.029,0,-0.012,-0.25,0.003,-0.002,0,0,0]
plot_signal_spectrum(signal)
```