

*Amortisseurs accordés pour la
protection des tours.*

BEN-NAJIM SAMI

SCEI : 28084



Introduction



Figure-1



Figure-2



Figure-3

- ① **Figure -1** : Séisme enregistré en Turquie , le 6 février 2023 . Magnitude 7,8 sur l'échelle de Richter.
- ② **Figure -2** : La tour Taipei 101 en Taiwan.
- ③ **Figure -3** : Image de Burj Al Arab aux UAE , il est soumis aux rafales de vent les plus intenses .

Problématique



Figure-4 : la masse pendulaire PTMD 101.

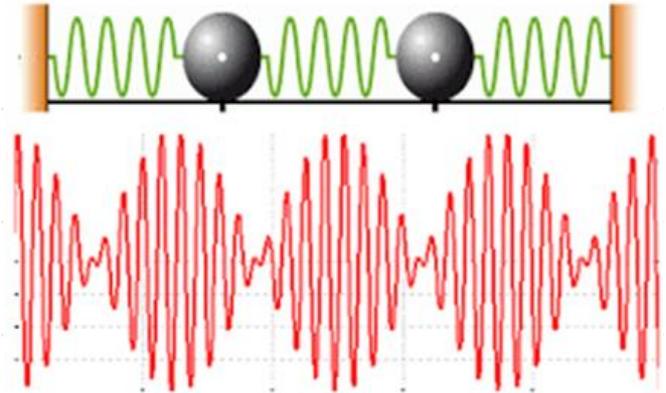


Figure-5 : Couplage des oscillateurs.

Quel est l'effet du couplage mécanique entre la tour et le pendule sur les oscillations subies par la structure ? Quelles sont alors les valeurs optimales des paramètres de ce pendule pour assurer une bonne atténuation des oscillations de la tour ?

Sommaire

- 1 Étude théorique du système mécanique à pendule accordé.
- 2 Étude expérimentale et modélisation informatique.
- 3 Amélioration du système et synthèse.
- 4 Annexe. (scripts phyton, circuits électroniques, mécanismes)

Effet du vent turbulent

- Les rafales génèrent un courant turbulent qui impose un régime forcé sur la tour en action:



Figure-6 : Burj Khalifa , UAE.

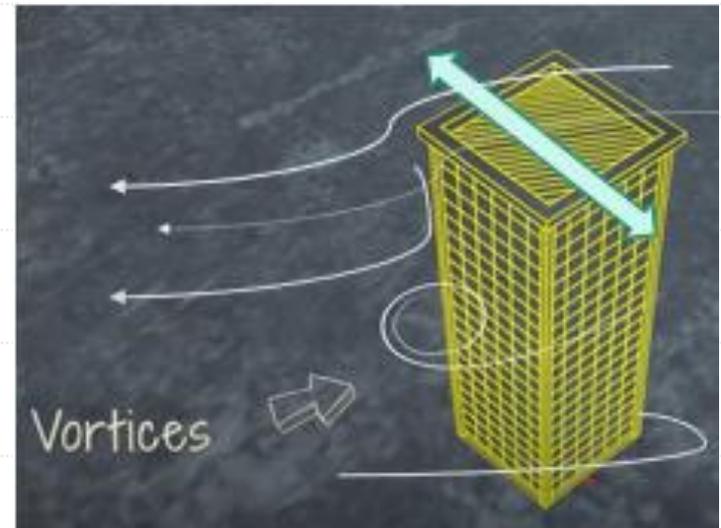


Figure-7 : Turbulence et oscillation de la tour .

Effet de séisme

- Les séismes engendrent par leur effet inertiel le tangage des structures élancées:

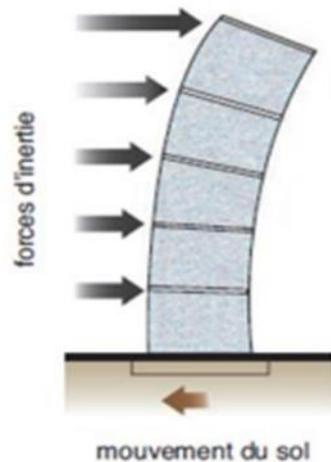


Figure-8 : Tangage inertiel engendré par le mouvement du sol.



Figure-9 : Séisme , Turquie 2023 de 7,8 en magnitude selon l'échelle de Richter.

Résultat préliminaire (hypothèse du problème plan)

- Hypothèse du problème plan: (Oxy)

L'angle maximal de déviation horizontale est autours de **15 degrés** ce qui donne une variation de cote relative de **3,2 %** .

$$\frac{\Delta H}{H} = 1 - \cos(\alpha)$$

- α : L'angle d'inclinaison du bâtiment.

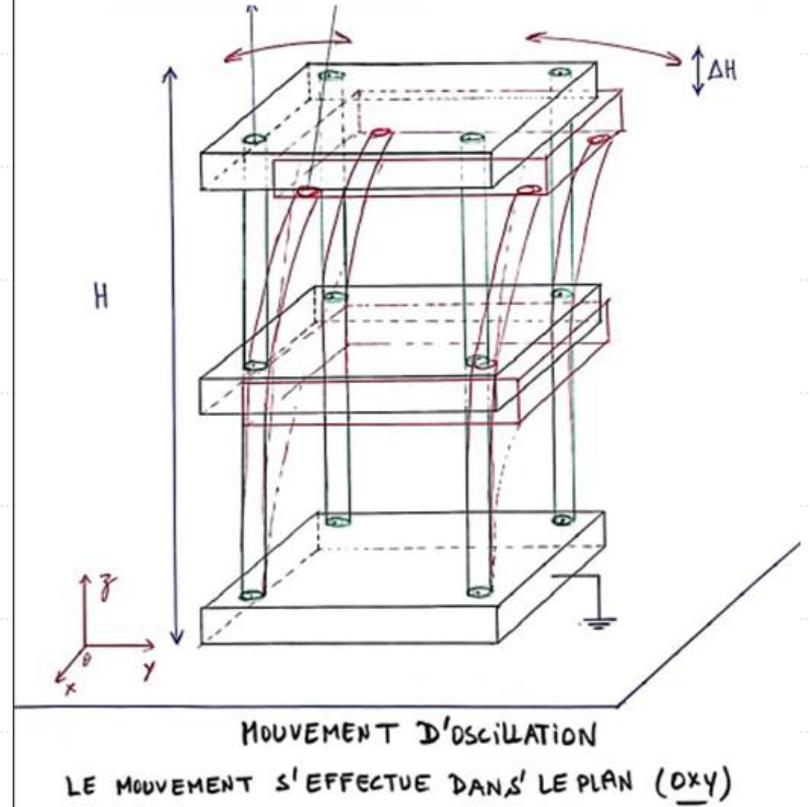


Figure-10: tangage d'une structure élancée.

Flexion d'une poutre (analogie avec un ressort)

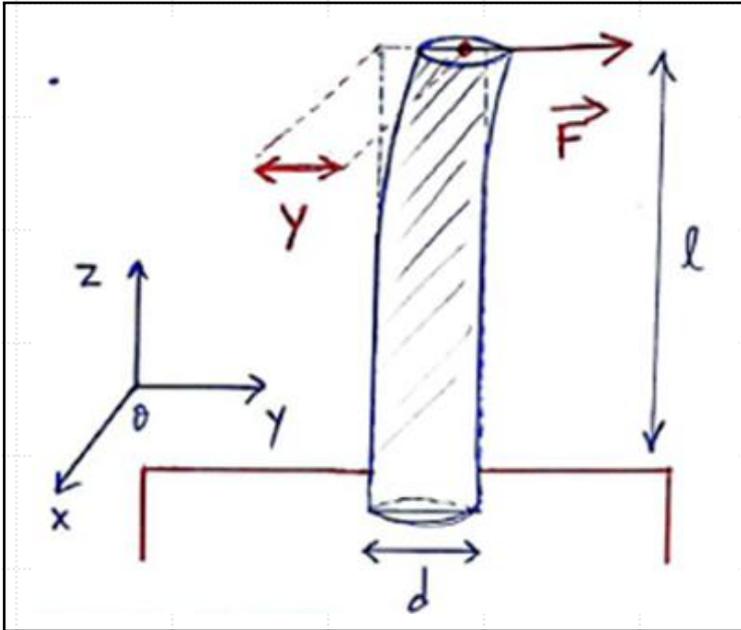


Figure-11: flexion d'une poutre.



Figure-12: isolation standard par butées d'isolation.

- La protection standard est imposée par la norme « [Eurocode 8](#) » élaborée en mars 2010.

Flexion d'une poutre (analogie avec un ressort)

- Analogie entre flexion de poutre et déformation d'un ressort:

$$\vec{F} = \frac{E d^4}{7l^3} y \vec{e}_y$$



$$k_0 = \frac{E d^4}{7l^3}$$

E : module de Young de la poutre en (Pa). **d** : diamètre de la poutre en (m). **l** : longueur de la poutre en (m).

La poutre et la butée sont montées en série donc la raideur équivalente est donnée par :

$$k_{eq} = \frac{1}{\frac{1}{k_0} + \frac{1}{k_b}}$$

Modélisation de la structure

1 Modélisation physique et paramétrage de la structure de l'immeuble :

R^* : Repère non galiléen lié au bâtiment.

R_0 : Repère galiléen lié au sol.

h_1 : Coefficient de frottement (bâtiment).

h_2 : Coefficient de frottement (pendule).

k : Raideur propre de la structure.

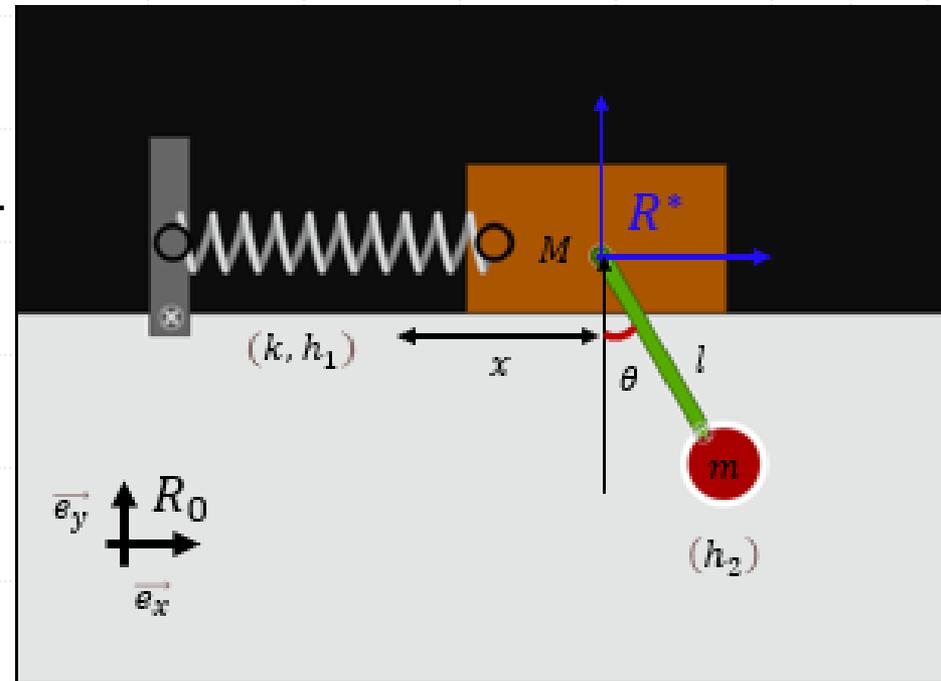
M : Masse du bâtiment.

m : Masse du pendule.

θ : Abscisse angulaire du pendule.

x : Déplacement horizontale du bâtiment.

l : Longueur du pendule.



Mise en équation

- Bilan des forces sur la pendule accordée :

$$\vec{p}_1 = m\vec{g} \quad \text{Le poids de la pendule de masse } m \text{ et de centre d'inertie } G.$$

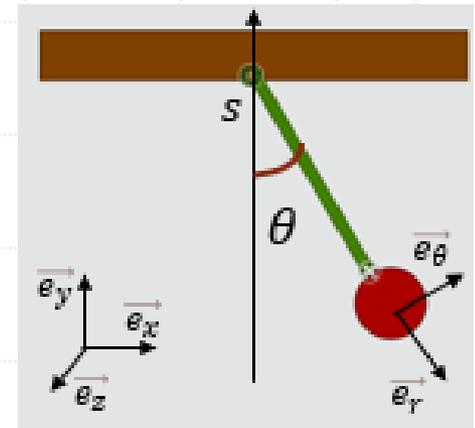
$$\vec{f}_i = -h_2\dot{\theta}l\vec{e}_\theta \quad \text{La force des frottements visqueux appliqués sur la pendule.}$$

$$\vec{f}_{ie} = -m\ddot{x}\vec{e}_x \quad \text{La force d'inertie d'entrainement associé au déplacement du bâtiment.}$$

$$\vec{T} = -T\vec{e}_r \quad \text{La tension de la tige appliquée sur la pendule.}$$

- On isole la pendule et on applique Th. Moment cinétique en projection sur (Sz) :

$$\frac{d\vec{\sigma}}{dt}(S/R^*) = \sum_{R^*} M_s(\vec{F})$$



Mise en équation

- Bilan des forces sur le système { Bâtiment } :

\vec{P} : Le poids du bâtiment.

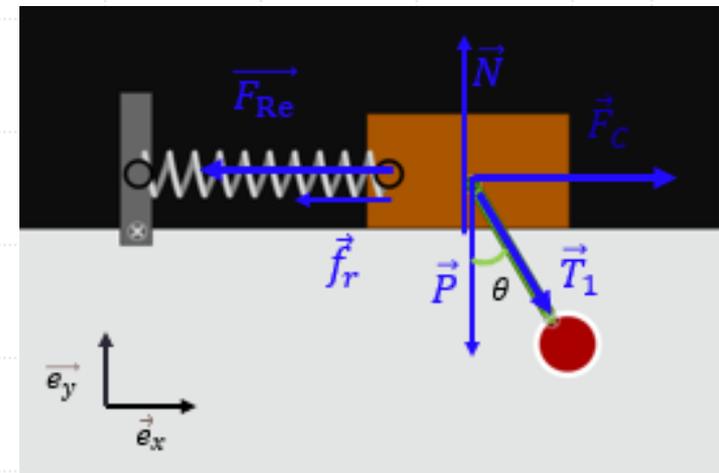
\vec{f}_r : La force des frottements fluides sur le bâtiment.

\vec{F}_{Re} : La force de rappel associée à la raideur propre.

\vec{N} : La réaction normale au plan de contact modélisant l'appui entre les étages du bâtiment.

\vec{T}_1 : La tension de la tige sur le bâtiment.

\vec{F}_C : C'est la force combinée par le séisme et les rafales de vent.



Mise en équation (démonstration dans l'annexe)

- **Le théorème du moment cinétique en projection sur l'axe (Sz) donne l'équation (A) :**

$$\ddot{\theta} + \frac{h_2}{m} \dot{\theta} + \frac{g}{l} \sin(\theta) + \frac{1}{l} \ddot{x} \cos(\theta) = 0$$

- **Le théorème de la résultante dynamique en projection sur l'axe (Ox) donne l'équation (B) :**

$$-h_1 \dot{x} - kx + T \sin \theta + F_c = M \ddot{x}$$

- **Le théorème de la quantité de mouvement en projection sur l'axe (Ox) donne l'équation (C) :**

$$ml(\ddot{\theta} \cos(\theta) - \dot{\theta}^2 \sin(\theta)) + m\ddot{x} = -T \sin(\theta) - h_1 l \dot{\theta} \cos(\theta)$$

Mise en équation

- En combinant les équations B et C on obtient l'équation suivante :

$$(M + m)\ddot{x} + h_1\dot{x} + kx + h_2l\dot{\theta} \cos(\theta) + ml(\ddot{\theta} \cos(\theta) - \dot{\theta}^2 \sin(\theta)) = F_C$$

- On obtient le système couplé suivant :

$$\left\{ \begin{array}{l} \ddot{\theta} + \frac{h_2}{m}\dot{\theta} + \frac{g}{l}\sin(\theta) + \frac{1}{l}\ddot{x} \cos(\theta) = 0 \\ (M + m)\ddot{x} + h_1\dot{x} + kx + h_2l\dot{\theta} \cos(\theta) + ml(\ddot{\theta} \cos(\theta) - \dot{\theta}^2 \sin(\theta)) = F_C \end{array} \right.$$

Mise en équation

- On procède à une linéarisation des équations différentielles trouvées précédemment :
- Le mouvement de la pendule est borné à faibles angles donc :

$$\left\{ \begin{array}{l} \ddot{\theta} + \frac{h_2}{m} \dot{\theta} + \frac{g}{l} \theta + \frac{1}{l} \ddot{x} = 0 \\ \ddot{x} = -\frac{h_1}{M} \dot{x} - \frac{K}{M} x + \frac{mg}{M} \theta + \frac{F_c}{M} \end{array} \right.$$

$$\{s\} \left\{ \begin{array}{l} (M + m)\ddot{x} + h_1\dot{x} + Kx + h_2l\dot{\theta} + ml\ddot{\theta} = F_c \\ \ddot{\theta} = -\frac{h_2}{m} \dot{\theta} - \frac{g}{l} \left(1 + \frac{m}{M}\right) \theta + \frac{h_1}{Ml} \dot{x} + \frac{k}{Ml} x - \frac{F_c}{Ml} \end{array} \right.$$

Perturbation mono-fréquentielle

- C'est lorsque la perturbation causée par le séisme ou par une rafale de vent est de la forme :

$$\vec{F} = F_m \cos(\omega_p t) \vec{u}$$

- On écrit les équations différentielles trouvées dans le domaine de Laplace :

$$\left\{ \begin{array}{l} \frac{\bar{x}}{\bar{\theta}} = \frac{l}{\omega^2} \left(\frac{g}{l} - \omega^2 + j \frac{h_2}{m} \omega \right) \quad \text{On note :} \quad H_1 = \frac{\bar{x}}{\bar{\theta}} \\ -\omega^2 (M + m)x + jh_1 \omega x + kx + jh_2 l \omega \theta - \omega^2 ml \theta = F_m \end{array} \right.$$

Perturbation mono-fréquentielle

- D'après ce qui précède on aboutit à ce qui suit :

$$\bar{X} = \frac{F_m}{k - (M + m)\omega^2 + jh_1\omega + H_1(jh_2l\omega - \omega^2lm)}$$

- Étude qualitative sur H_1 : 1ère Condition sur l, m, h_2

$$H_1 = \frac{\bar{x}}{\theta} \text{ Prend sa valeur minimale pour : } \omega_p^2 = \frac{g}{l} \Rightarrow l = \frac{g}{\omega_p^2} \text{ et } h_2 \ll m\omega_p$$

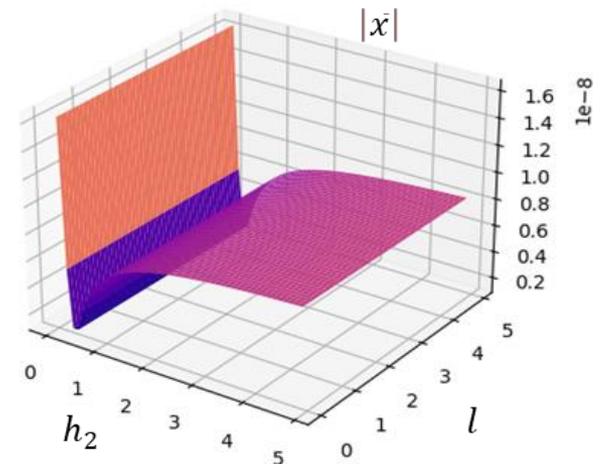
- En respectant ses conditions on vérifie le phénomène d'**Antirésonance** :

$$H_1 \rightarrow 0 \Rightarrow \bar{x} \rightarrow 0$$

Perturbation mono-fréquentielle

$$\bar{X} = \frac{F_m}{k - (M + m)\omega^2 + jh_1\omega + H_1(jh_2l\omega - \omega^2lm)}$$

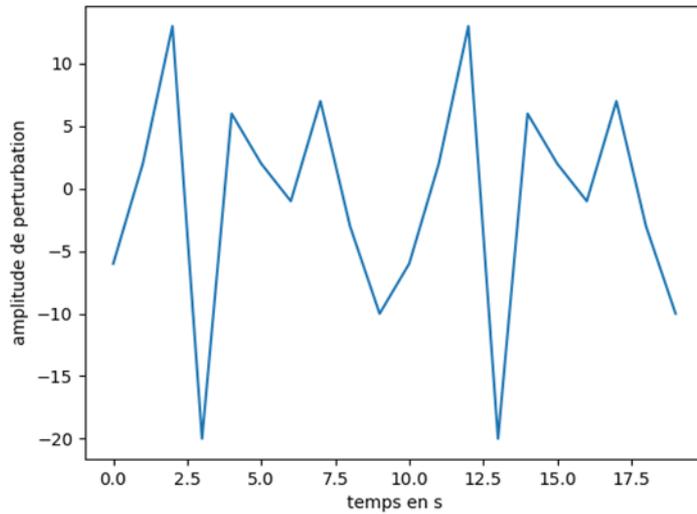
- À l'aide de l'algorithme (Annexe) « Particulieroptimal(w) » on trace la courbe surface dépendante de deux des trois "h2", "l", "m". Et on détermine le triplet optimal (m, l, h2) :
- Pour : w=10rad/s , M=1000t , m=3,5t, h1=150si , k=1000si on obtient :
- **Limitation de la méthode** : Cette démarche présente un inconvénient , elle ne permet pas l'étude une perturbation dont le spectre est riche.



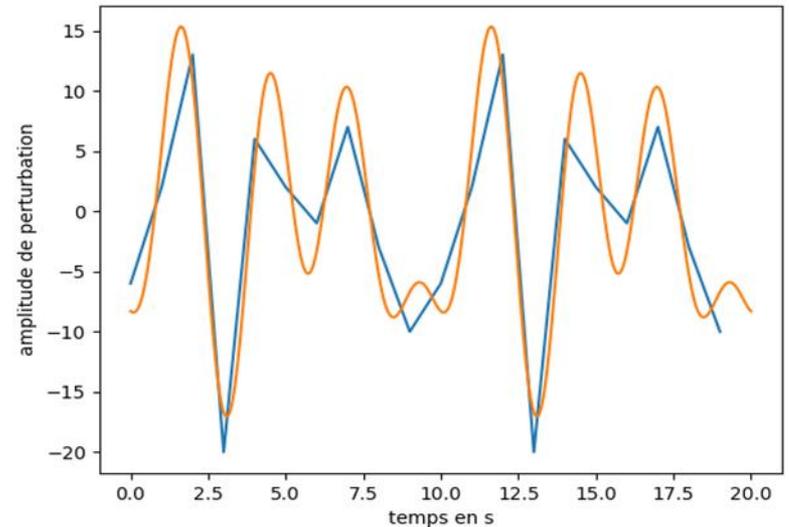
Perturbation multi-fréquentielle

- À l'aide des algorithmes (Annexe) « `Fourierlist(wave_list)` », « `meilleurordre(wave_list)` », « `traçage(wave_list)` » :
- On trouvera la décomposition finie de Fourier qui reflète le plus possible l'allure de la perturbation :

Enregistrement de perturbation



Approximation par `traçage(wave_liste)`



Perturbation multi-fréquentielle

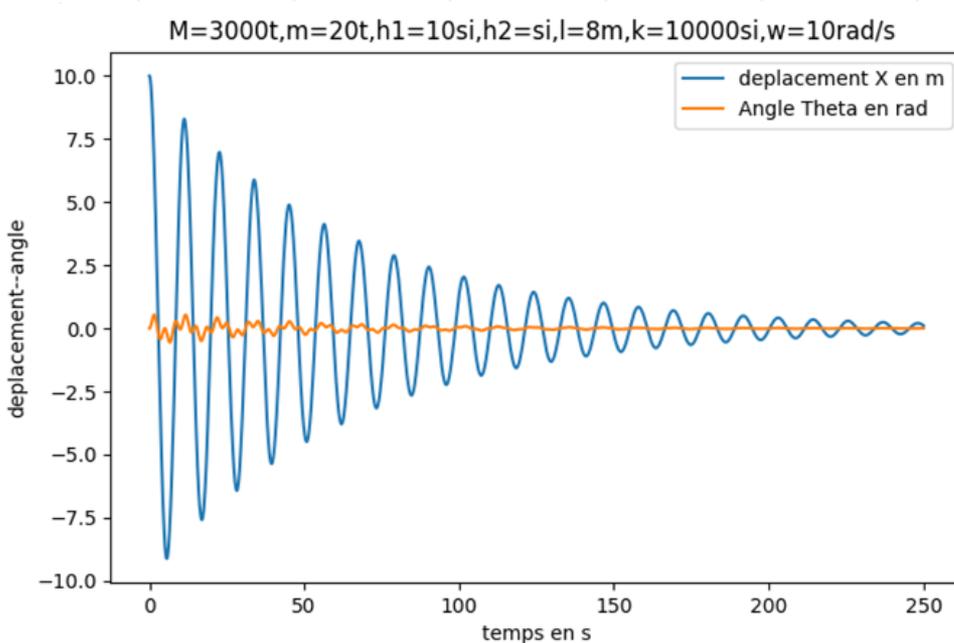
- À l'aide de ses trois algorithmes , on obtient une expression mathématique approchée à la perturbation de la forme :

$$\vec{F}(t) = \left(\sum_{k=0}^N a_k \cos(k\omega t) + b_k \sin(k\omega t) \right) \vec{u}$$

- Avec : $\left\{ \begin{array}{l} N : C'est\ l'ordre\ de\ fourrier\ issue\ de\ "meilleurordre(wave_list)" \\ \omega : C'est\ la\ fondamentale\ de\ la\ perturbation \end{array} \right.$

Le triplet (m, l ,h2) optimal

- Grâce à l'algorithme (annexe) « generaloptimal() » on obtient le triplet (m , l , h2) optimal .
- Cet algorithme se base sur un tri des possibilités sur le critère de temps de réponse à 80% (temps nécessaire pour amortir 80% de la perturbation).

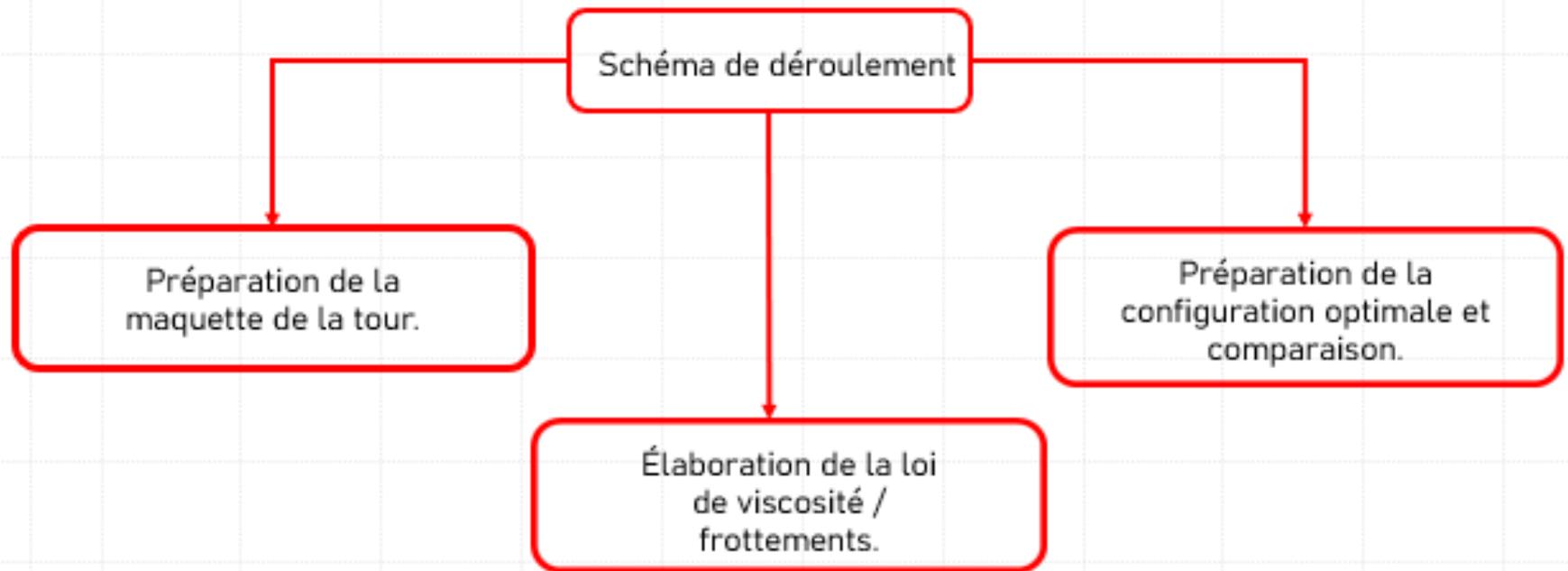


```
>>> (executing file "officiel gain plot for tipe.py")
```

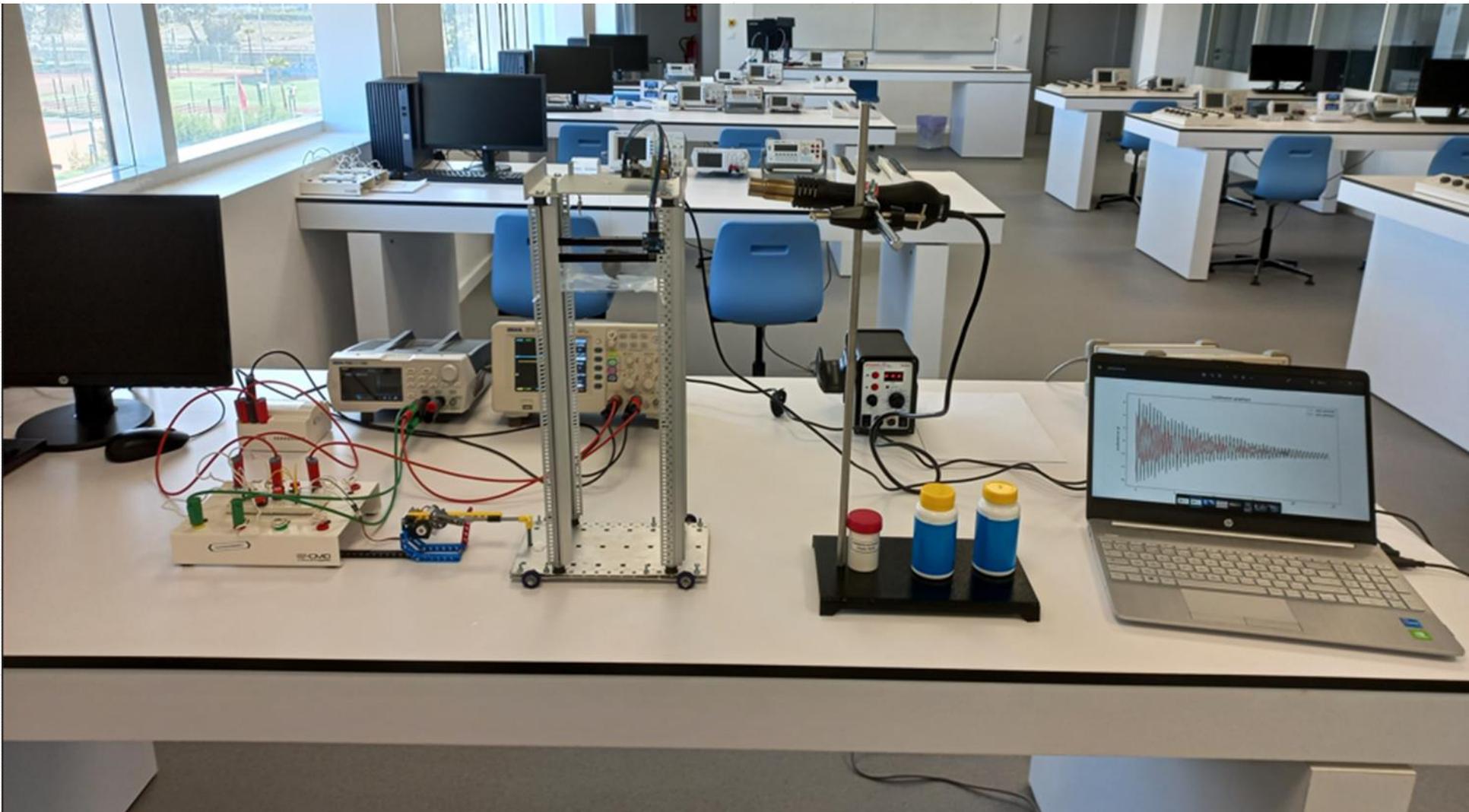
```
>>> generaloptimal()  
[26, 18.0, 15, 77.25]
```

↑ masse ↑ longueur ↑ h2 ↑ t80%

Expérience



Expérience



Expérience phase 1 (matériel expérimental)

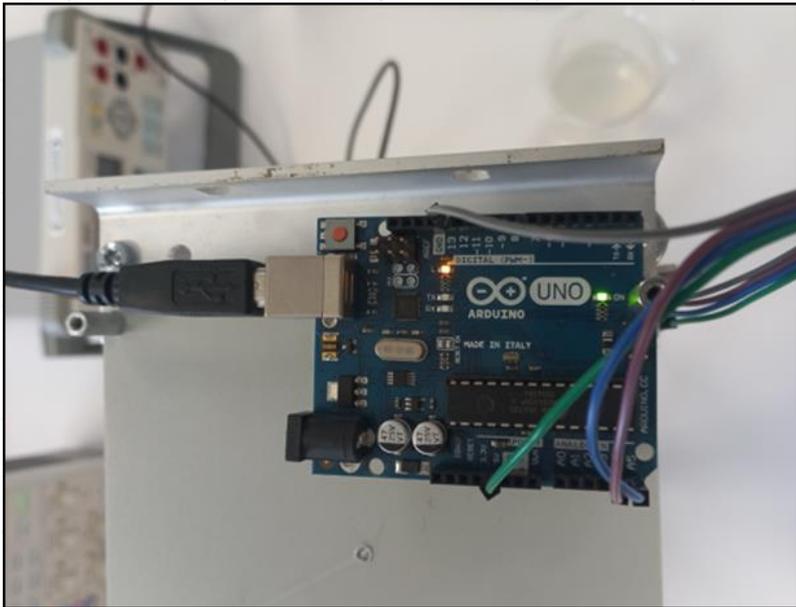


Figure-13: Carte ARDUINO-UNO.

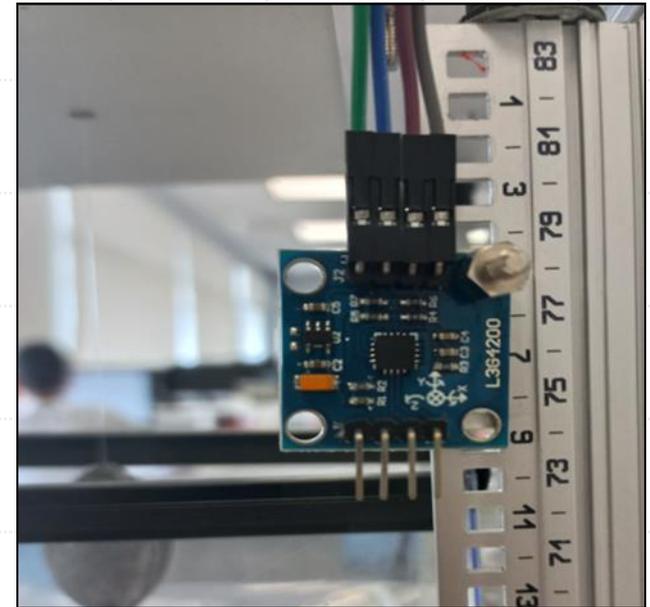


Figure-14: Gyromètre L3G4200.

Figure-20: préparation des échantillons de différentes concentrations à base d'Alginates de sodium.

Expérience phase 1 (matériel expérimental)

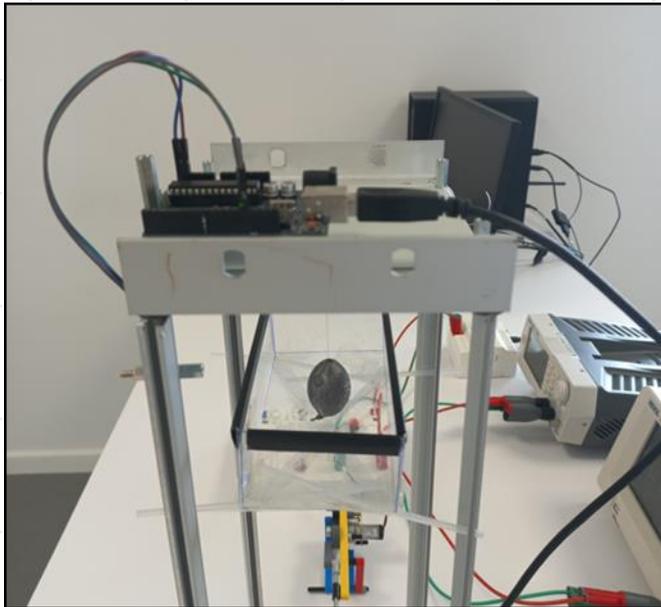


Figure-15: Unité d'amortissement inertielle « Tuned mass damper ».

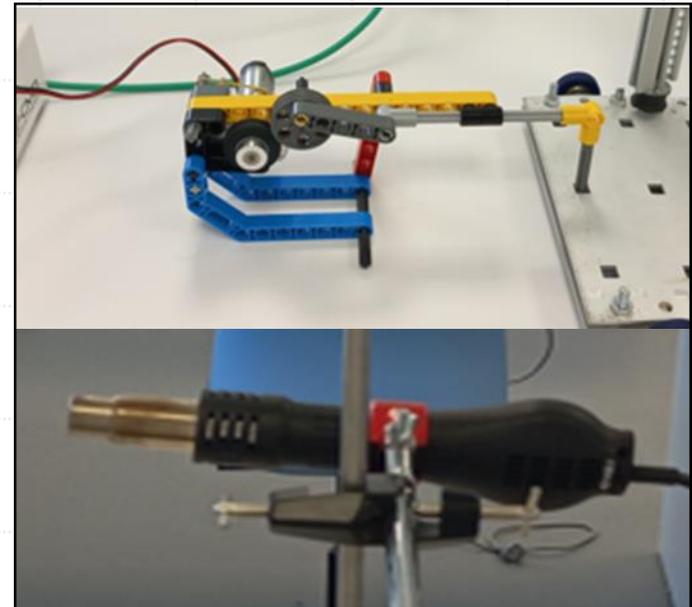


Figure-16: Vibreur sismique et souffleur électrique.

Expérience phase 1 (matériel expérimental)

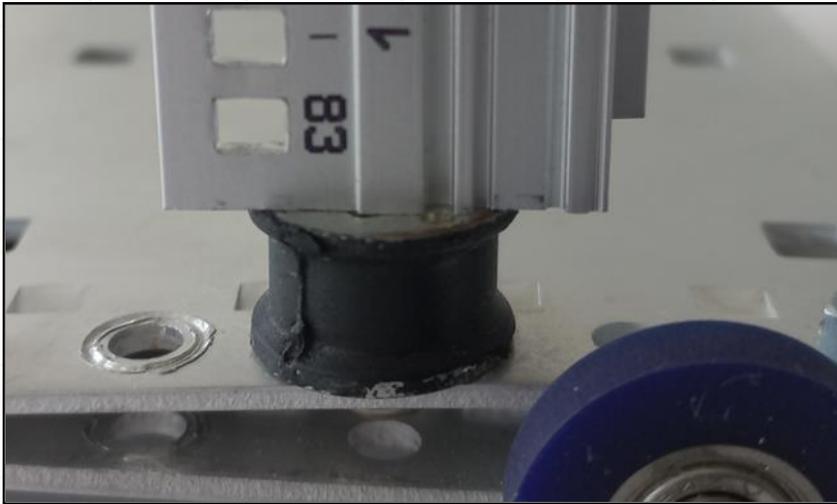


Figure-17: Butée d'isolation installée sur la maquette.



Figure-18: Butée d'isolation réelle.

Expérience phase 2

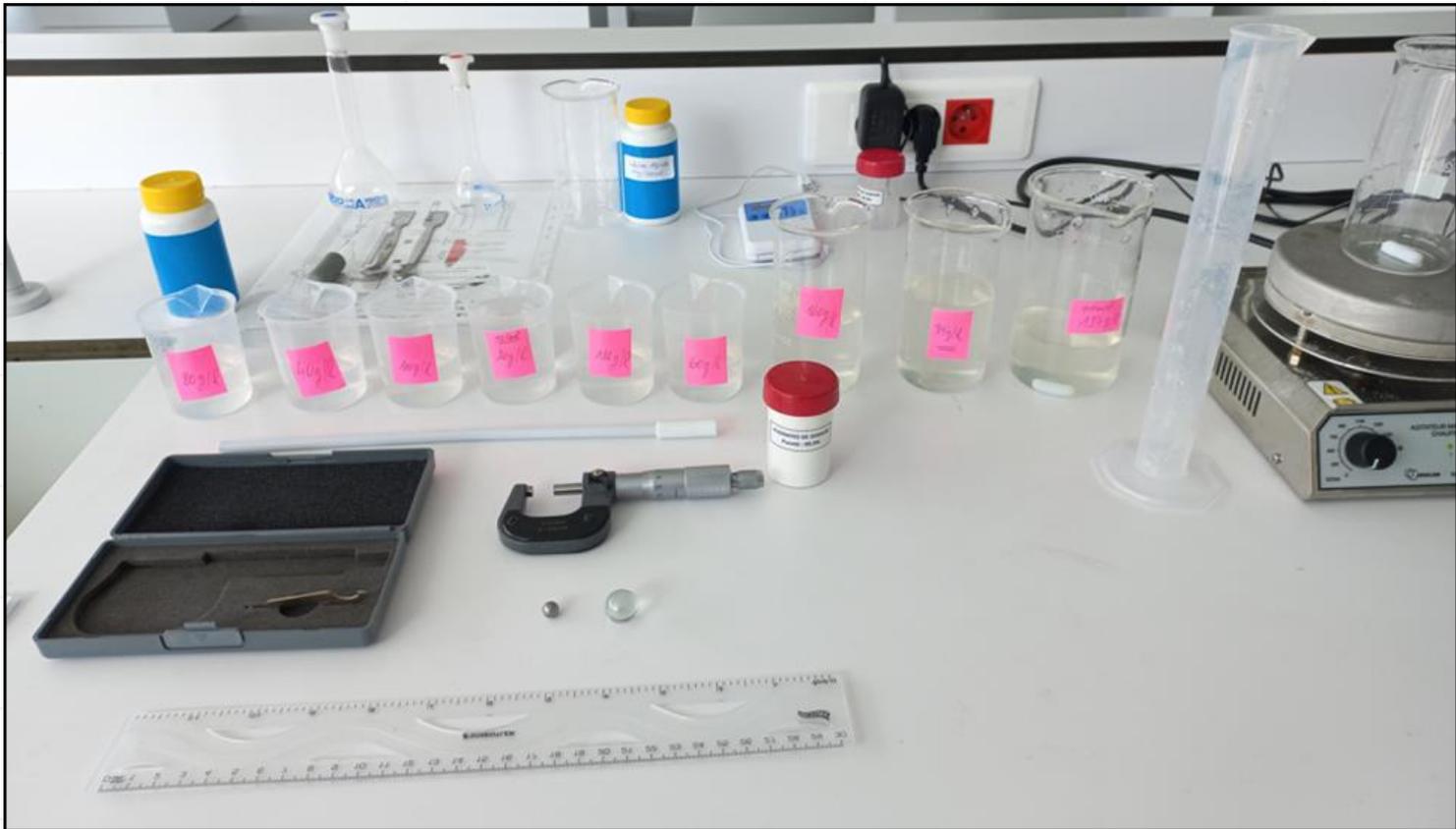


Figure-19: verrerie et matériel pour préparation des solutions visqueuses.

Expérience phase 2



Figure-20: préparation des échantillons de différentes concentrations à base d'Alginate de sodium.

Expérience phase 2



Figure-21: remplissage du tube de chute.



Figure-22: Chronométrage de la chute de la bille dans le tube.

Expérience phase 2

Concentration g/l	Temps de chute ms	Viscosité en Pa/s	Coef.frottement Kg/s
0	48	45,4	12,8 / 17,11 / 21,4
20	54	51,1	14,4 / 19,3 / 24,1
40	65	61,5	17,3 / 23,2 / 29
60	72	68,2	19,2 / 25,7 / 32,1
80	86	81,4	23,1 / 30,7 / 38,4
100	109	103,2	29,2 / 38,9 / 48,6
120	123	116,4	32,8 / 43,8 / 54,8
140	139	131,5	36,7 / 49,5 / 61,9
160	157	148,5	42,1 / 56,2 / 69,9

- Grâce au script « **assistantTPchimie(C)** » et **aux calculs** faits (**annexe**) on obtient le tableau suivant pour les rayons : **R1=1,5cm** , **R2=2cm** , **R3=2,5cm**.

Expérience phase 2

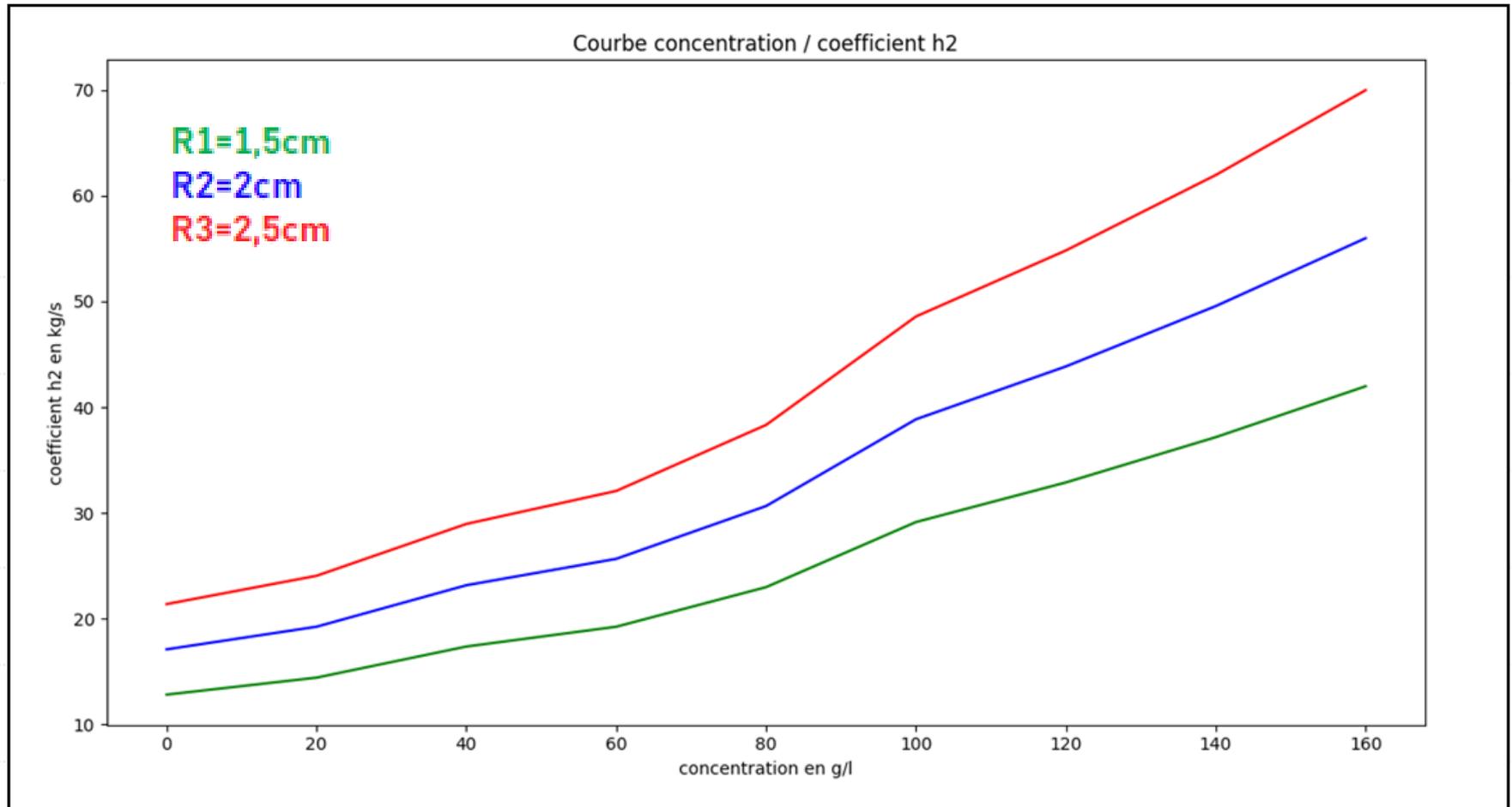
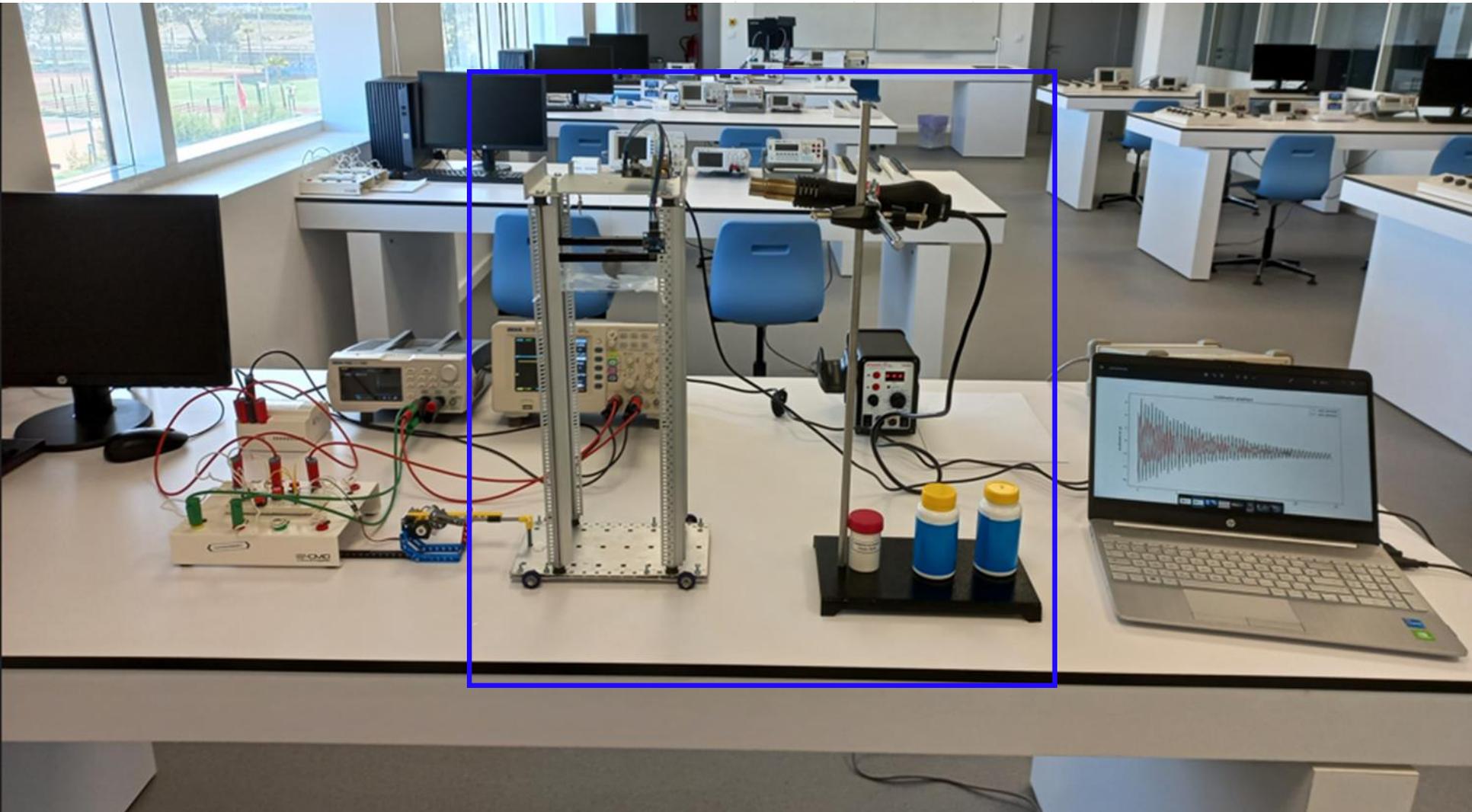
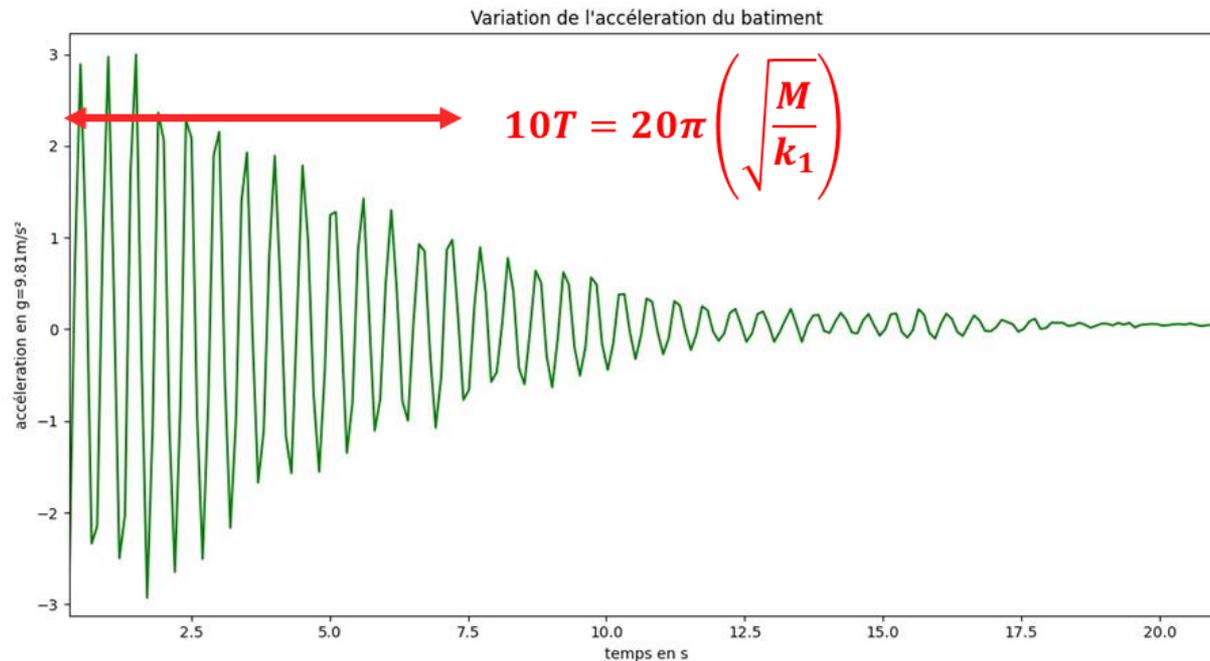


Figure-23: évolution de h2 en fonction de la concentration d'alginate de sodium.

Simulation d'une rafale de vent



Expérience phase 2



$$k_1 = \frac{4\pi^2 M}{T^2}$$

$$k_1 = 2,3N/m$$

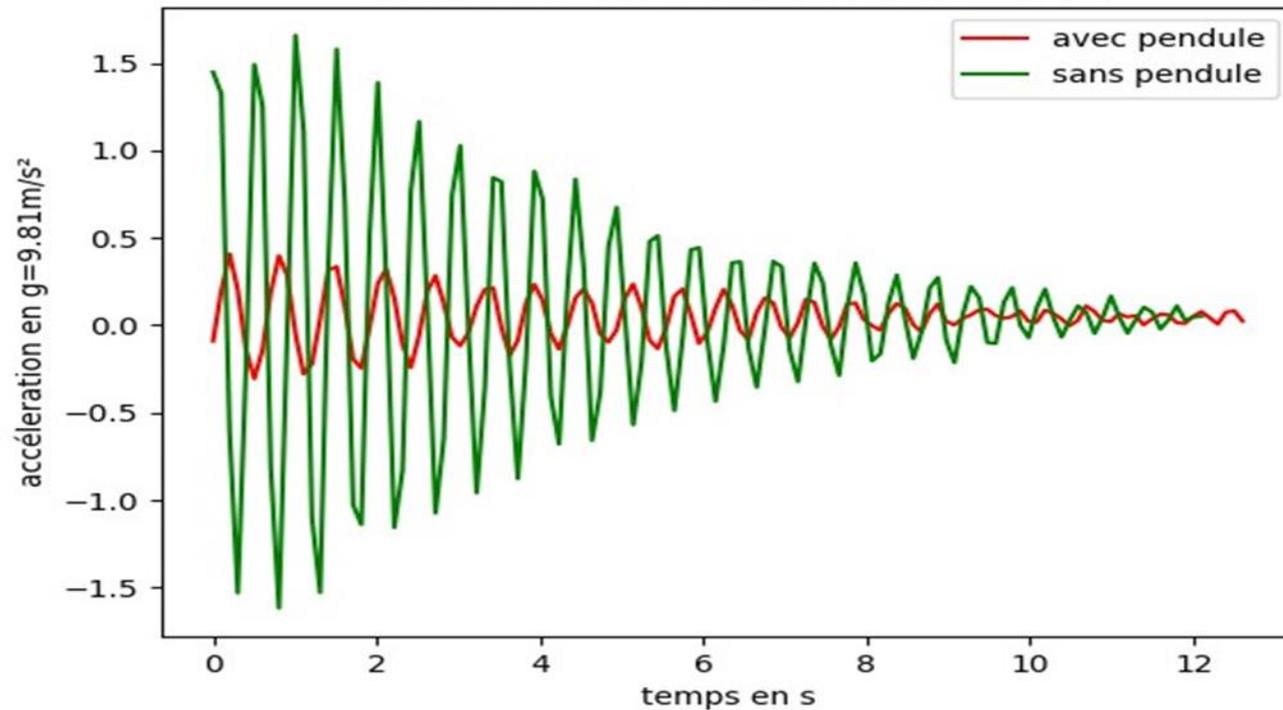
$$\lambda = 2M \frac{\ln(a_1/a_2)}{t_2 - t_1}$$

$$\lambda = h_1 \approx 1,3 \text{ kg/s}$$

Figure-24: mesure de h_1 et k_1 en régime libre d'oscillations de la tour .

Traçage en temps réel (régime libre)

Variation de l'accélération du bâtiment



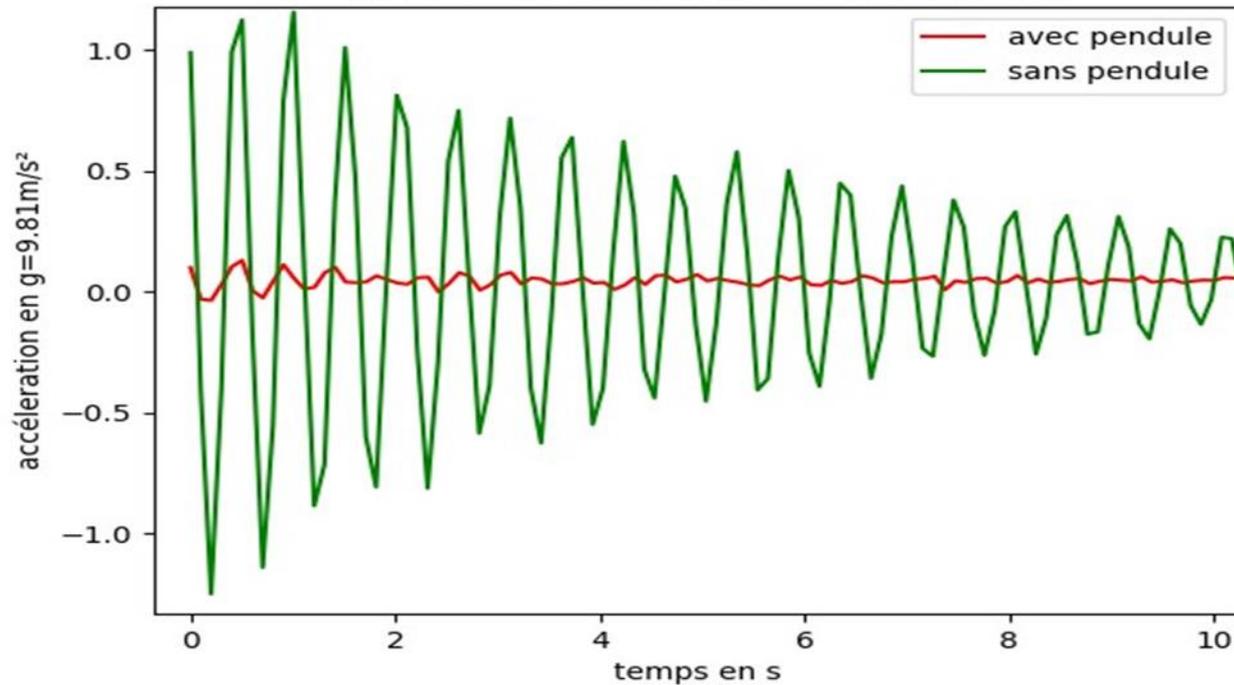
$$l = 5\text{cm}$$

$$m = 100\text{g}$$

$$h_2 = 40\text{kg/s}$$

Traçage en temps réel (régime libre)

Variation de l'accélération du bâtiment

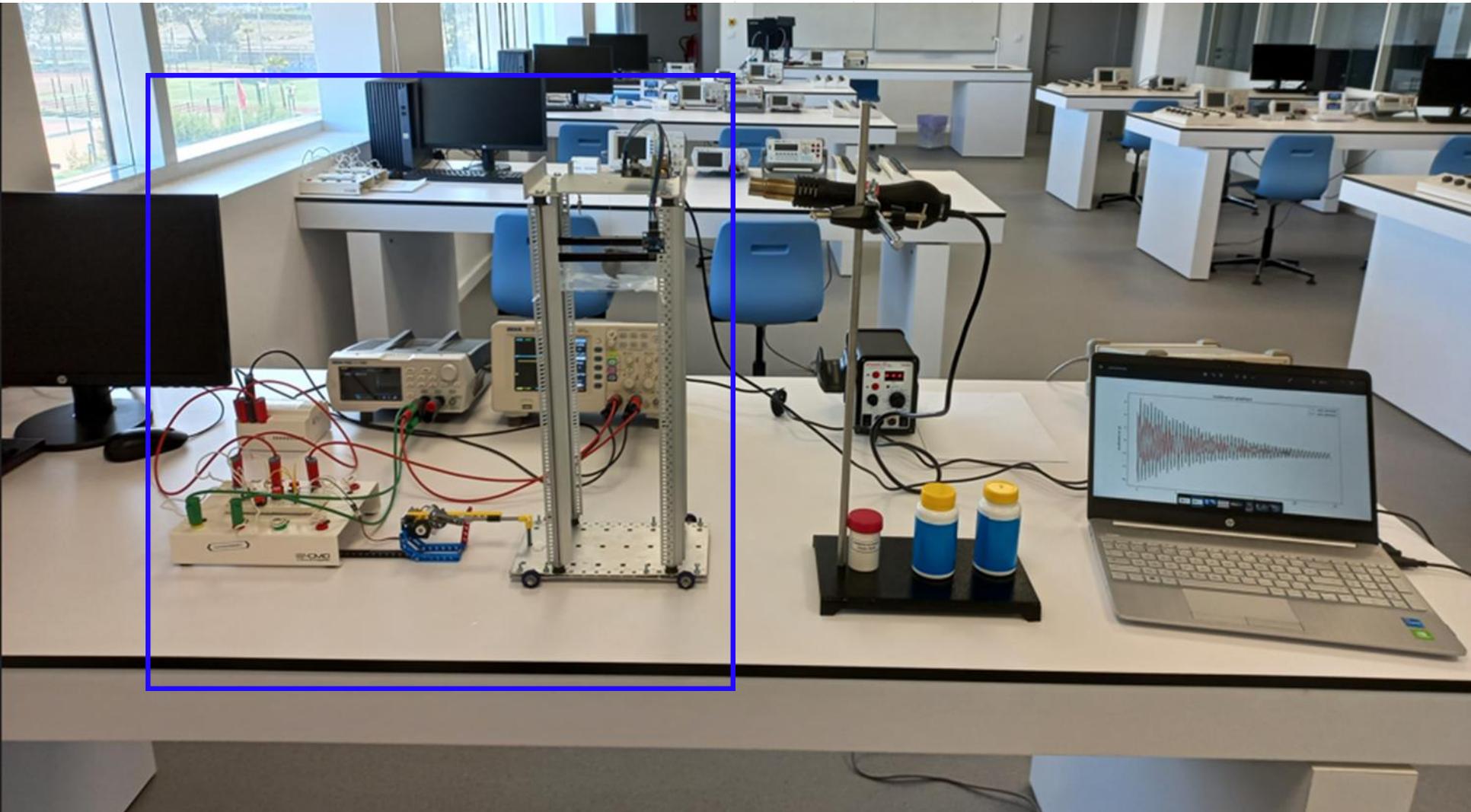


$$l = 6\text{cm}$$

$$m = 100\text{g}$$

$$h_2 = 40\text{kg/s}$$

Simulation d'une excitation sismique



Expérience phase 3

- L'algorithm « `generaloptimal()` » donne le triplet théorique optimal:

```
>>> (executing file "all codes in one file.py")
```

```
>>> generaloptimal()  
[0.1, 0.04, 40]
```

$h_2 = 40 \text{ kg/s}$
 $m = 100\text{g}$
 $l = 4\text{cm}$

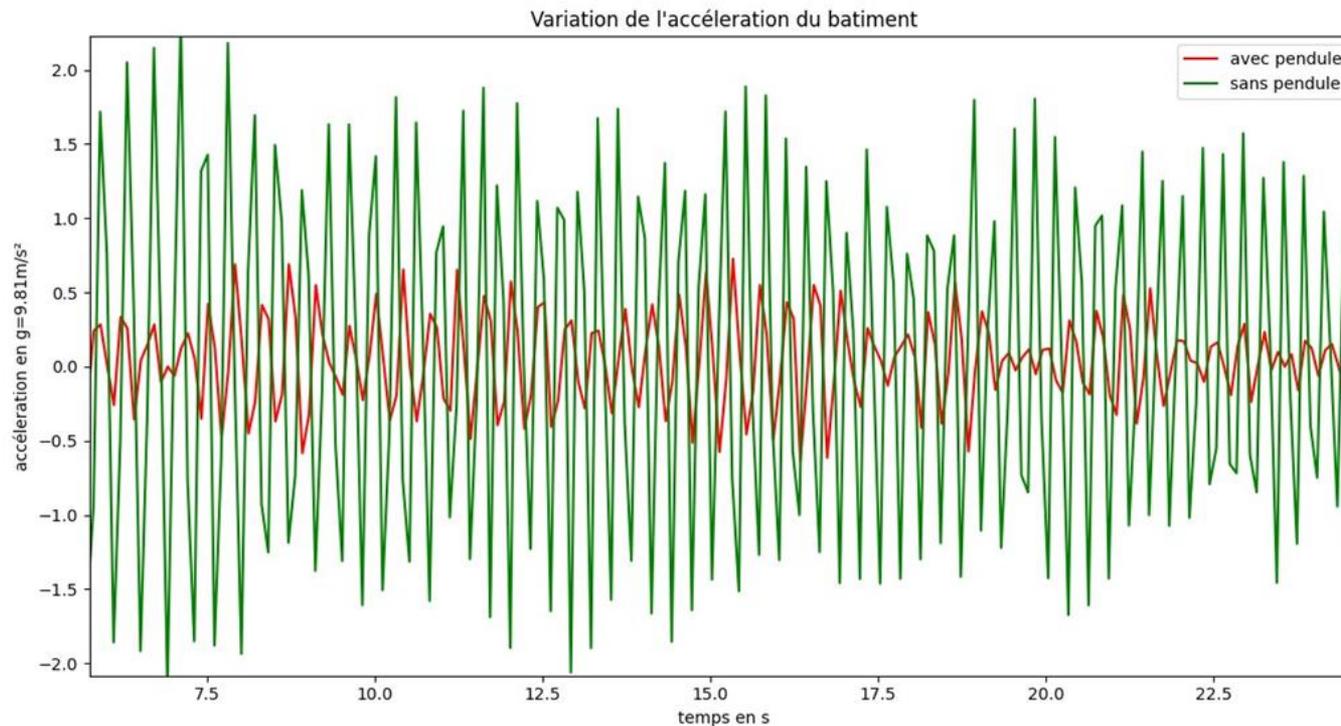
- Correspond à la concentration $C = 106\text{g/l}$

```
masse=np.arange(0.05,0.15,0.05)  
length=np.arange(0.01,0.08,0.01)  
friction=np.arange(1,70,5)
```



Figure-25: préparation de la configuration théorique optimale.

Traçage en temps réel (régime forcé)

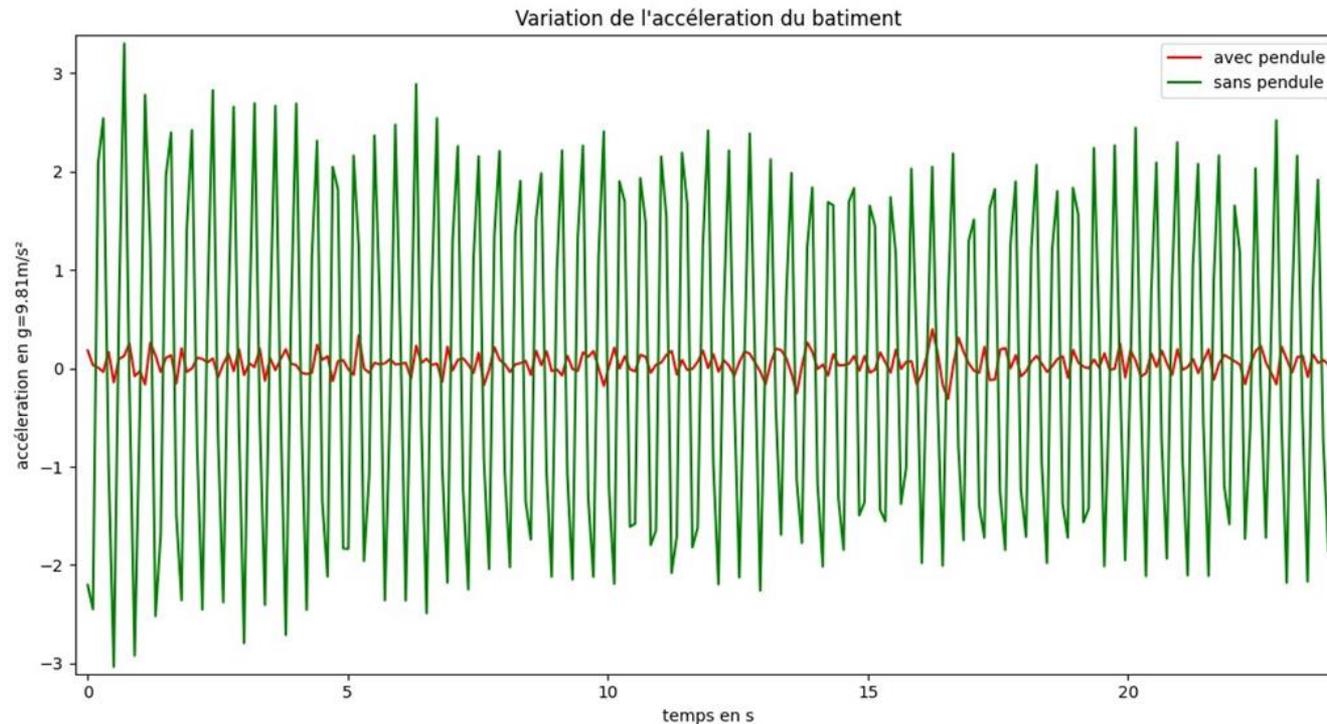


$$l = 6cm$$

$$m = 150g$$

$$h_2 = 40kg/s$$

Traçage en temps réel (régime forcé)



$$l = 6cm$$

$$m = 100g$$

$$h_2 = 40kg/s$$

Tableau de mesures (rapport d'amortissement en %)

- On obtient le tableau suivant pour : $M=3\text{kg}$, $h_1=1,3\text{kg/s}$, $k_1=2,3\text{N/m}$, $h_2=40\text{Kg/s}$, $w= 3,5\text{rad/s}$

		Longueur du pendule accordé							
		1cm	2cm	3cm	4cm	5cm	6cm	7cm	8cm
Masse du pendule	50g	10%	25%	30%	46%	44%	36%	27%	18%
	100g	8%	16%	27%	30%	33%	64%	52%	42%
	150g	5%	16%	21%	31%	42%	39%	28%	15%

Figure-26: tableau récapitulatif des pourcentages d'amortissement.

- Expérimentalement on trouve que la longueur optimale vérifie : $6\text{cm} \leq l_+ \leq 7\text{cm}$

Conclusion

- Suite à l'étude faite , je remarque qu'il y a **un écart** entre les paramètres optimaux abouties par la théorie et ceux trouvés par le modèle expérimental .
- Pour justifier cet écart , Il faut remettre en question :
 - 1) L'incrémentation discrète (**parcours fini de valeurs**) dans les scripts python.
 - 2) Le glissement des roues (**maquette**) et du galet (**moteur**).
 - 3) Les liaisons entre composants n'est pas parfaite (**jeu et frottements**).
 - 4) Difficultés d'ordre technique et financier.

Le système pendulaire à masse accordée permet de réduire les oscillations des structures élancées soumis à l'excitation d'un séisme ou d'une rafale de vent, il est indispensable d'adapter ses paramètres pour chaque structure.

Merci pour votre attention

Annexe



- [1] MR. PASCAL HEMON : Vibrations couplées avec le vent : *LadHyX Département de Mécanique Ecole Polytechnique F-91128 PALAISEAU CEDEX Mél* : pascal.hemon@ladhyx.polytechnique.fr <https://www.off-ladhyx.polytechnique.fr/people/pascal/pdf/poly-aeroelasticite.pdf>
- [2] MR. XAVIER AMANDOLESE : Low speed flutter and limit cycle oscillations of a two-degree-of-freedom flat plate in a wind tunnel : *Journal of Fluids and structures 2013. LadHyX, CNRS-Ecole Polytechnique, F-91128 PALAISEAU, France.* <http://yakari.polytechnique.fr/Django-pub/documents/amandolese2013rp-2pp.pdf>
- [3] MR. NASSIM DJEDOUÏ : Modélisation et analyse du contrôle hybride(actif-passif) des vibration des bâtiments sous sollicitations sismiques en champ proche : *Université Mohamed khider-Biskra . Thèse de doctorat soutenue en juin 2018.* <https://thesis.univ-biskra.dz/4000/1/These%20Complete%20Finale%20corrige%20apres%20soutenance.pdf>
- [4] MINISTÈRE DE L'HABITAT ET DE LA POLITIQUE DE LA VILLE : Le règlement marocain de la construction parasismique : *Le règlement marocain de construction parasismique R.P.S.2000*
- [5] MR. MUSTAPHA RGUIG : Calcul dynamique et sismique des structures , troisième année Génie Civil : *École Hassania des travaux publics EHTP .*
- [6] MME. ISABELLE GOUJON : Les alginates, excipients d'origine marine utilisés dans L'industrie pharmaceutique: applications à la synthèse d'un gel chimique : *Université Paris Saclay . Version de 23 mars 2016* https://docnum.univ-lorraine.fr/public/SCDPHA_T_2004_GOUJON_ISABELLE.pdf

Expérience de chute libre

- La force des frottements visqueux :

$$\vec{f} = -6\pi \eta_g R \vec{v},$$

- La force globale appliquée sur la bille :

$$\vec{P}_{\text{app}} = \vec{P} + \vec{\Pi} = \frac{4}{3}\pi R^3 \underbrace{(\rho_a - \rho_g)}_{\rho'} \vec{g}$$

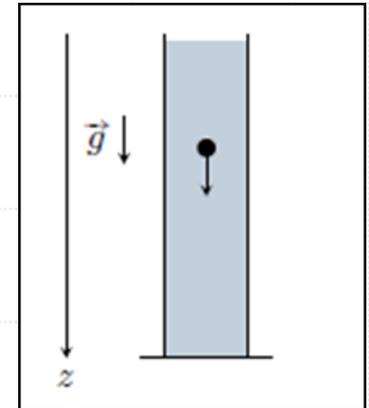
- D'après le théorème de quantité de mouvement en projection sur (Oz):

$$\frac{4}{3}\pi R^3 \rho_a \frac{dv}{dt} = \frac{4}{3}\pi R^3 \rho' g - 6\pi \eta_g R v$$

$$\eta = \frac{2 R^2 \rho' g}{9 v_{\text{lim}}}$$



$$h_2 = 6\pi \eta R'$$



Circuit électrique amplificateur

Pour faire varier la vitesse de rotation du moteur j'ai monté un circuit amplificateur qui donne accès à 3 vitesses : $w_1=2,7\text{rad/s}$ $w_2=4,2\text{rad/s}$ $w_3=6,5\text{rad/s}$

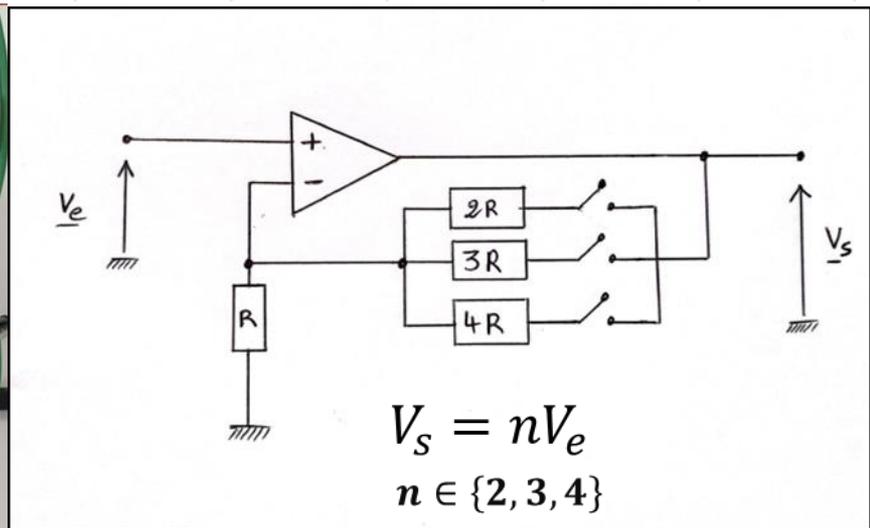
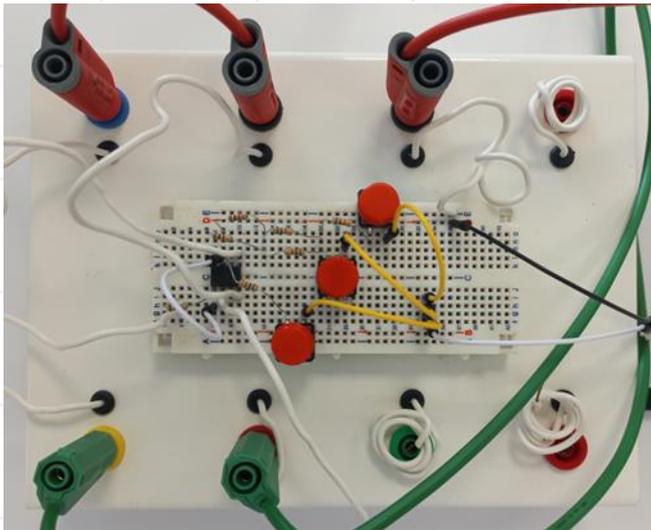


Figure-a1: pont multiplicateur.

À l'aide du GBF on injecte à l'entrée une tension continue de 6V et on obtient à la sortie respectivement 12V , 18V , 24V.

Moteur vibreur simulant un séisme

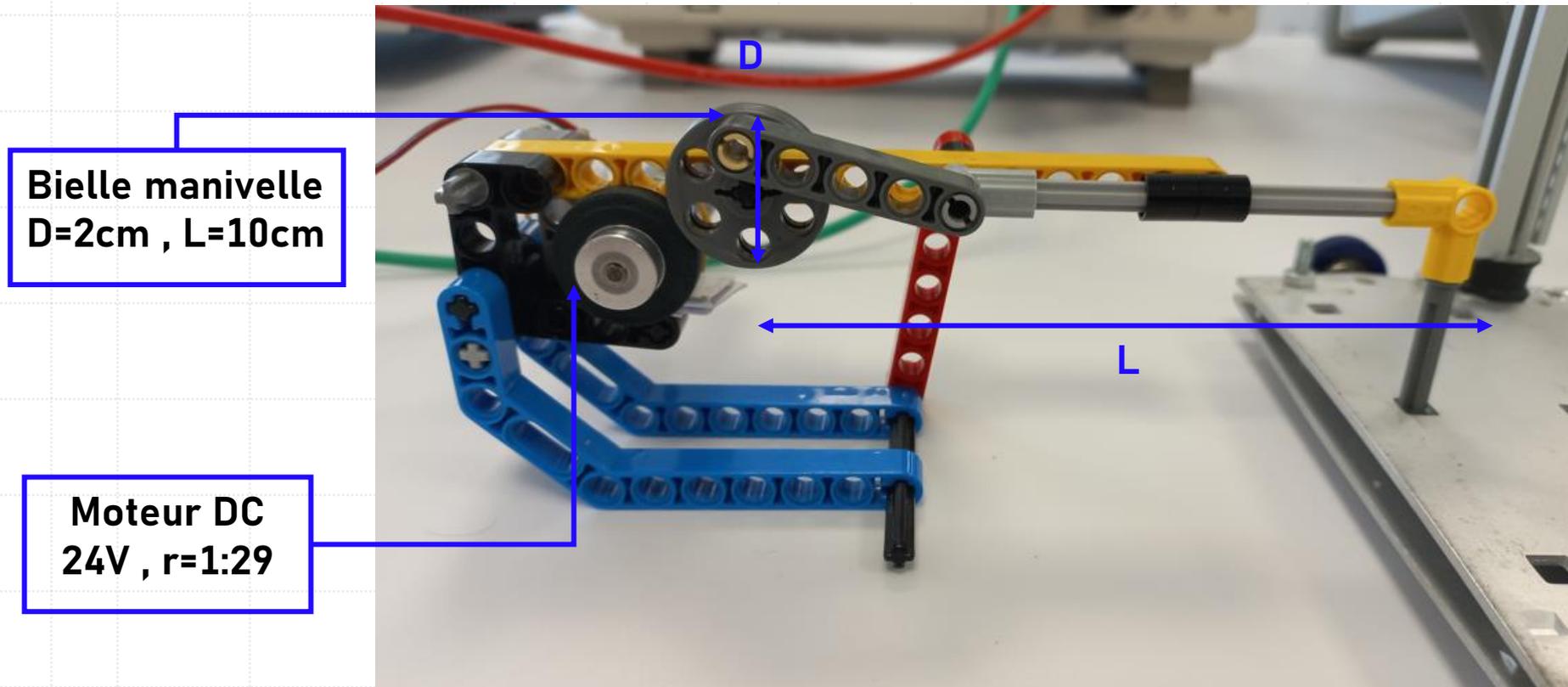


Figure-a2: unité de vibration sismique.

Bassin de frottements

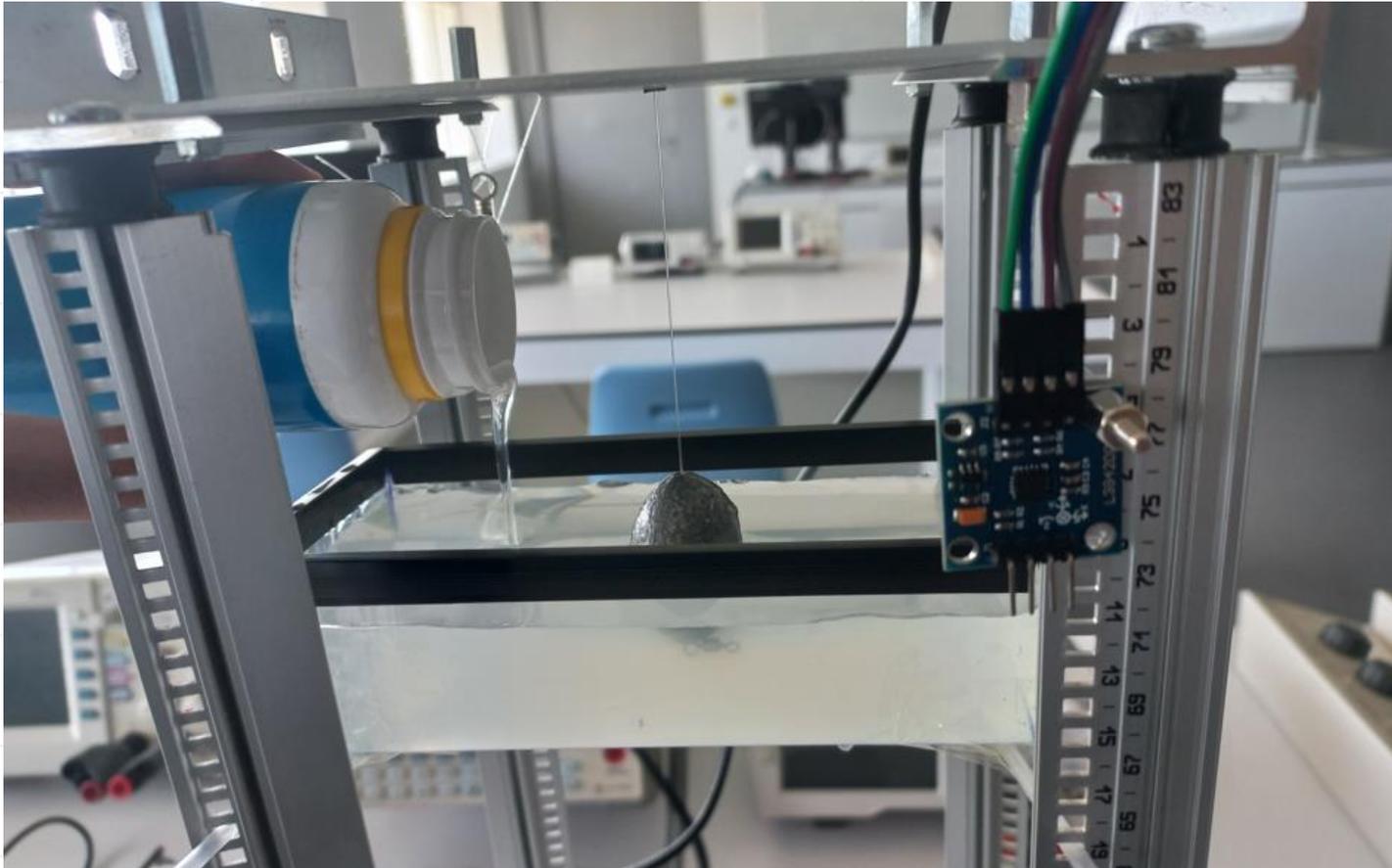


Figure-a3: bassin rempli de solution d' « Alginate de sodium » de concentration $C=106\text{g/l}$

Les masses pendulaires utilisées



$m=150\text{g}$, $R= 2,5\text{cm}$



$m=100\text{g}$, $R= 2\text{cm}$



$m=50\text{g}$, $R= 1,5\text{cm}$

Démonstration de l'équation (A)

$$\frac{d\vec{\sigma}}{dt}(S|R^*) = \frac{d}{dt}(l\vec{e}_r \otimes ml\dot{\theta}\vec{e}_\theta) = ml^2\ddot{\theta}\vec{e}_z$$

$$\vec{M}(\vec{P}_1) = \vec{SG} \otimes m\vec{g} = -mgl \sin(\theta)\vec{e}_z$$

$$\vec{M}(\vec{f}_i) = \vec{SG} \otimes (-h_2l\dot{\theta}\vec{e}_\theta) = -h_2l^2\dot{\theta}\vec{e}_z$$

$$\vec{M}(\vec{f}_{ie}) = \vec{SG} \otimes (-m\ddot{x}\vec{e}_x) = -ml\ddot{x} \cos(\theta)\vec{e}_z$$

$$\vec{M}(\vec{T}) = \vec{SG} \otimes \vec{T} = \vec{0}$$

équation A : issue de Th. Moment cinétique

$$\ddot{\theta} + \frac{h_2}{m}\dot{\theta} + \frac{g}{l}\sin(\theta) + \frac{1}{l}\ddot{x}\cos(\theta) = 0$$

Démonstration de l'équation (B)

- Application du théorème de la résultante dynamique sur { Bâtiment } qu'on note {B}:

$$\overrightarrow{R_d}(\{B\}/R_0) = \sum_{i \in \Omega} \overrightarrow{F_i} \quad (\text{avec } \Omega \text{ est l'ensemble des forces appliquées sur le bâtiment})$$

- On obtient : $\overrightarrow{R_d}(\{B\}/R_0) = \overrightarrow{P} + \overrightarrow{N} + \overrightarrow{f_r} + \overrightarrow{T} + \overrightarrow{F_R} + \overrightarrow{F_C} = M\ddot{x}\overrightarrow{e_x}$

- Par projection sur l'axe (Ox) on trouve l'équation scalaire suivante :

équation B :
$$-h_1\dot{x} - kx + T \sin \theta + F_C = M\ddot{x}$$

Démonstration de l'équation (C)

- On va maintenant trouver l'expression de T en fonction des paramètres utiles :
- On isole {la pendule} notée $\{P\}$ et on applique le théorème de quantité de mouvement :

$$\frac{d\vec{P}}{dt}(\{P\}|R_0) = \frac{d}{dt}(m \cdot \vec{v}(\{P\}, R_0)) = \sum_{i \in \Omega} \vec{F}_i$$

- On obtient : $\vec{p}(\{S\}, R) = m \cdot (\vec{v}(\{s\}/R^*) + \vec{v}(\{B\}/R_0)) = ml\dot{\theta}\vec{e}_\theta + m\dot{x}\vec{e}_x$

$$\vec{e}_x \cdot \frac{d\vec{p}}{dt}(\{S\}/R_0) = ml(\ddot{\theta} \cos(\theta) - \dot{\theta}^2 \sin(\theta)) + m\ddot{x}$$

équation C : $ml(\ddot{\theta} \cos(\theta) - \dot{\theta}^2 \sin(\theta)) + m\ddot{x} = -T \sin(\theta) - h_1 l \dot{\theta} \cos(\theta)$

Scripts python

```
# TIPE/ BEN-NAJIM SAMI / SCEI 28084
#Scripts_phyton

# Permet l'acquisition / le filtrage / représentation des traçés réels
import numpy as np
import matplotlib.pyplot as plt
def filtrer(L):# filtrage des données ARDUINO.
    return[i for i in L if i!='']
def homogenise(L):
    return[int(i)/2250 for i in L ]
def database():# acquisition des données ARDUINO.
    import serial
    L=[]
    S=[]
    portserie=serial.Serial('COM3',baudrate=9600,timeout=0.1)
    for i in range(1,35):
        L.append(portserie.readline().decode('ascii'))
    F=filtrer(L)
    for k in F:
        S=S+k.split(",")
    M=filtrer(S)
    return homogenise(M)
```

Scripts python

```
def oscillate():# trace la réponce gyroskopique de la maquette.
    import time
    Y1=database()
    print("test sans pendule terminée avec succès")
    time.sleep(5)
    print("ok SAMI attachez la pendule ")
    time.sleep(15)
    print("Go for 2nd shot")
    time.sleep(1)
    Y2=database()
    n1=len(Y1)
    n2=len(Y2)
    X1=np.linspace(0,0.1*n1,n1)
    X2=np.linspace(0,0.1*n2,n2)
    plt.plot(X2,Y2,'r')
    plt.plot(X1,Y1,'g')
    plt.title("Variation de l'accélération du batiment")
    plt.xlabel('temps en s')
    plt.ylabel('accélération en g=9.81m/s2')
    plt.legend(['avec pendule','sans pendule'])
    plt.show()
```

Scripts python

```
# Permet de fournir le triplet optimal théorique :
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import odeint

def F1(Y,t,m,l,h2):# u est un angle et x le déplacement.
    M=3
    h1=2
    k=2.3
    g=9.81
    o=1+m/M
    x,dx,u,du=Y
    return [dx, -dx*h1/M-x*k/M+u*m*g/M+Fc(t)/M,du, -du*h2/m-u*g*o/l+dx*h1/
(M*l)+x*k/(M*l)-Fc(t)/(M*l)]
```

- L'algorithme « generaloptimal() » tri les différentes configurations obtenues par « F1(Y,t,m,l,h2) » en se basant sur le tri (croissant) du taux :

$$\text{taux} = \frac{t80\%}{S_{\max} - S(+\infty)}$$

Scripts python

```
def donnelist(): # donne la liste des quadriplets (m,l,h2,t80%).
    list=[]
    T=np.linspace(0,250,1000)
    masse=np.arange(0.05,0.15,0.05)
    lenght=np.arange(0.01,0.08,0.01)
    friction=np.arange(1,70,5)
    for m in masse:
        for l in lenght:
            for h2 in friction:
                Y=odeint(F1,[10,0,0,0],T,args=(m,l,h2))
                x=Y[:,0]
                val1=max(x[600:])
                val2=max(x[:50])
                t80=temps80(x)
                selectionrate=t80/(val2-val1)
                list.append([m,l,h2,t80,selectionrate])
    return list

def temps80(L): # determine le temps de réponse à 80%.
    mag=max(L)
    perm=max(L[600:])
    n=1
    while max(L[n:])>0.2*(mag-perm) and n<len(L)-2:
        n+=1
    return n*250/1000

def generaloptimal(): # donne le triplet (m,l,h2) optimal.
    L=donnelist()
    M=sorted(L,key= lambda x:x[4])
    return M[0][:3]
```

Scripts python

```
# Permet d'approcher une excitation par méthode de Fourier :
import numpy as np
import matplotlib.pyplot as plt
def fourierlist(wave_list,n): # donne la décomposition de fourrier.
    a0 = sum(wave_list)/len(wave_list)
    an = lambda n: 2/len(wave_list) * sum([wave_list[i]*np.cos(2*np.pi*n*i/
len(wave_list)) for i in range(len(wave_list))])
    bn = lambda n: 2/len(wave_list) * sum([wave_list[i]*np.sin(2*np.pi*n*i/
len(wave_list)) for i in range(len(wave_list))])
    x = np.linspace(0, len(wave_list), 500)
    s = np.zeros_like(x)
    s += a0/2
    for i in range(1, n+1):
        s += an(i)*np.cos(2*np.pi*i*x/len(wave_list)) +
bn(i)*np.sin(2*np.pi*i*x/len(wave_list))
    return s
def tracage(wave_list): # trace la perturbation brute et celle de fourrier.
    n=meilleurordre(wave_list)
    s=fourierlist(wave_list,n)
    x = np.linspace(0, len(wave_list), 500)
    plt.plot(range(len(wave_list)), wave_list)
    plt.plot(x, s)
    plt.xlabel('temps en s')
    plt.ylabel('amplitude de perturbation')
    plt.show()
```

Scripts python

```
def meilleurordre(wave_list): # donne l'ordre minimal assurant une bonne
approximation.
    wave=np.array(wave_list)
    n=0
    s=fourierlist(wave_list,n)
    while max(wave)-max(np.array(s))>0 :
        n+=1
        s=fourierlist(wave_list,n)
    else:
        return n
# Permet de résoudre le système différentiel :
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import odeint

def Fc(t):
    return 10*cos(2*t)

def F2(Y,t):# u est un angle et x le déplacement
    M=3000
    m=30
    h1,h2=12,100
    k=10
    l=10
    g=9.81
    o=1+m/M
    x,dx,u,du=Y
    return [dx, -dx*h1/M-x*k/M+u*m*g/M+Fc(t)/M, du, -du*h2/m-u*g*o/l+dx*h1/
(M*l)+x*k/(M*l)-Fc(t)/(M*l)]
```

Scripts python

```
# Permet de calculer la viscosité lors de la manipulation TP chimie :
import matplotlib.pyplot as plt
from math import *
import numpy as np
def assistantTPchimie(CT):
    H=0.25
    g=9.81
    pi=3.14
    R=4.16e-3
    m=2.06
    R2=0.015
    R3=0.020
    R4=0.025
    raubille=m/(4*pi*R**3/3)
    eau=1000
    visco=[2*R**2*(raubille-(x[0]+eau))*x[1]/H for x in CT]
    friction2=[6*pi*R2*n for n in visco]
    friction3=[6*pi*R3*n for n in visco]
    friction4=[6*pi*R4*n for n in visco]
    X=[x[0] for x in CT]
    plt.plot(X,friction2,"g")
    plt.plot(X,friction3,"b")
    plt.plot(X,friction4,"r")
    plt.xlabel("concentration en g/l")
    plt.ylabel("coefficient h2 en kg/s")
    plt.title("Courbe concentration / coefficient h2")
    plt.show()
# c'est le résultat exp CT=[[0,0.048],[20,0.054],[40,0.065],[60,0.072],
[80,0.086],[100,0.109],[120,0.123],[140,0.139],[160,0.157]]
```

Scripts python

```
def comportement():  
    T=np.linspace(0,2000,1000)  
    Y=odeint(F2,[10,0,0,0],T)  
    x=Y[:,0]  
    u=Y[:,2]  
    plt.plot(T,x,'g')  
    plt.plot(T,u,'r')  
    plt.show()
```

```
from matplotlib.pyplot import *  
from numpy import *  
from pylab import *  
from mpl_toolkits import mplot3d
```

```
def H(w,l,m,h2):  
    M,h1,k,g=10000,150,100,9.8  
    a,b,c,d=(m*l)**2,(h2*l)**2,(M*l**2-m*g*l),h1-h2  
    return (k-(M+m)*w**2)+(a*w**4-b*w**2)/c+1j*(d*w+(2*m*b*w**3)/c)
```

Scripts python

```
def Module(w,l,m,h2):  
    return abs(1/H(w,l,m,h2))  
def tracer1():  
    m=35  
    w=100  
    ax=plt.axes(projection="3d")  
    x_data=np.arange(0,5,0.1)  
    y_data=np.arange(0,5,0.1)  
    X,Y= np.meshgrid(x_data,y_data)  
    Z= Module(w,X,m,Y) # on peut placer X,Y dans les variables de tracage  
    ax.plot_surface(X,Y,Z,cmap="plasma")  
    plt.show()  
def G(w):  
    M,m,h1,h2,l,g,k=1000,1000,10,10,30,9.8,1000  
    a,b,c,d=(m*l)**2,(h2*l)**2,(M*l**2-m*g*l),h1-h2  
    return (k-(M+m)*w**2)+(a*w**4-b*w**2)/c+1j*(d*w+(2*m*b*w**3)/c)
```

Scripts python

```
def bode():
    omega=np.arange(0,10,0.01)
    Y=10000/abs(G(omega))
    plt.plot(omega,Y)
    plt.show()
def Particulieroptimal(w):
    h2=15
    x=np.linspace(-1,1,500)
    y=np.linspace(-1,1,500)
    X, Y = np.meshgrid(x,y)
    Z=Module(w,X,Y,h2)
    fig = plt.figure()
    ax = plt.axes(projection='3d')
    ax.plot_surface(X, Y, Z, cmap="plasma", linewidth=0, antialiased=False,
alpha=0.65)
    ax.set_xlabel('x')
    ax.set_ylabel('y')
    ax.set_zlabel('z')
    plt.show()
```