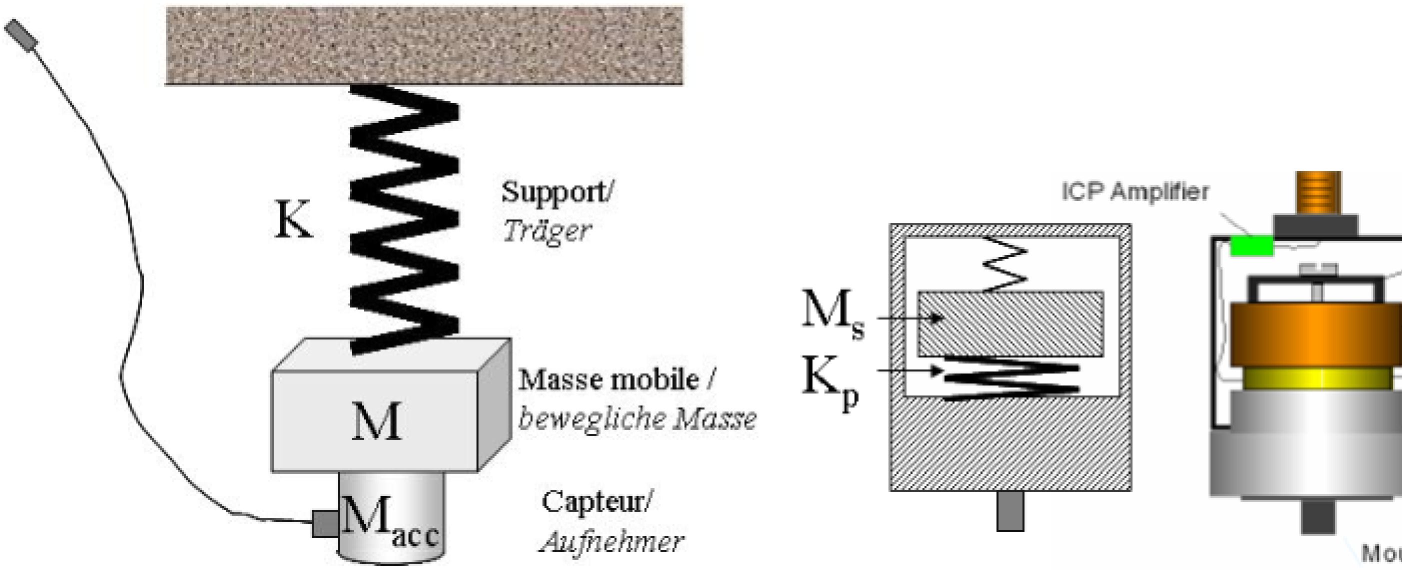
ex_accelerometre_piezo_sol

November 10, 2023

1 Accéléromètre Piézoélectrique

Un accéléromètre piézoélectrique (voir schéma de principe) est utilisé pour détecter les oscillations d'une masse M suspendue à un ressort de constante élastique K.



On connaît :

[28]:
```
vK=100 # [N/m]
```

Le cristal a les propriétés suivantes :

[29]:
```
# NOTE : On ajoute un 'v' devant les variables qui sont des valeurs pour les distinguer des variables
#        symboliques

vD=3 # diamètre en [mm]
vH=1 # hauteur en [mm]
vE=8e12 # Module d'élasticité [N/m^2]
vBeta = 2.26e-12 # coefficient piézoélectrique [C/N]
vEpsilon0 = 8.85e-12 # [A s / V m]
vEpsilon_r = 4.5
```

Les connecteurs du capteur on une capacité en parallèle avec le cristal

[30]:
```
vCstray=2 # pF
```

L'accéléromètre mesure une accélération subie par la masse sismique M_s qui vaut :

[31]:
```
vMs=1 # poids de la masse sismique du capteur [g]
```

1.0.1 Q1 Calcul de la sensibilité de l'accéléromètre

- Quelle est la sensibilité du capteur S en $[V/(m/s^2)]$ en prenant en compte la capacité parasite C_{stray} ?

On cherche la tension qui apparaît sur la sortie lorsqu'on applique une accélération de $1[m/s^2]$

[32]:
```
# Import du module pour calcul litteral
import sympy as sp
sp.init_printing()

# Fonction utilisée pour afficher la valeur d'une expression sous la forme "nom␣
↪= valeur"
def deq(name, value):
    return sp.Eq(sp.Symbol(name),value)

# Declaration des variables litterales
Ms,a, beta,D,H,epsilon_0, epsilon_r = sp.symbols('M_s,a, beta,D,H,epsilon_0,␣
↪epsilon_r')
Cstray=sp.symbols('C_stray')
E=sp.symbols('E')

# Equations
F = Ms * a          # Newton
Q = beta * F       # equation du piézo

# Calcul de la capacité du cristal
S=sp.pi *(D/2)**2 # surface du piézo
Cq =  S /H * epsilon_0 * epsilon_r # Capacité du piézo

Ctot=Cq + Cstray  # Capacité totale
Uc = Q / Ctot     # Relation tension-charge de la capacité
deq('Uc',Uc) # Expression de U
```

[32]:

$$Uc = \frac{M_s a \beta}{C_{stray} + \frac{\pi D^2 \epsilon_0 \epsilon_r}{4H}}$$

[33]:
```
# Valeurs numériques de l'exercice
valeurs={a:1,Ms:vMs*1e-3,beta:vBeta, Cstray:vCstray*1e-12, D:vD*1e-3, H:
↪vH*1e-3, epsilon_0:vEpsilon0, epsilon_r:vEpsilon_r, E:vE}
```

[34]:
```
# Valeur numérique pour la tension
# On a posé a=1, si bien que ceci correspond à la *sensibilité*
vS=Uc.subs(valeurs).evalf()
print("la sensibilité est de ",vS,"[V/(m/s^2)]")
```

```
la sensibilité est de  0.000990573623938871 [V/(m/s^2)]
```

[35]:
```
deq("C_q",Cq.subs(valeurs).evalf())
```

[35]:

$$C_q = 2.8150633671573 \cdot 10^{-13}$$

[36]:
```
print("Cq vaut {:.2f} [pF]".format(Cq.subs(valeurs).evalf()*1e12))
```

```
Cq vaut 0.28 [pF]
```

1.0.2 Q2 Fréquence propre du capteur

Quelle est la fréquence propre du capteur, (celle de la masse sismique suspendue au quartz comme ressort) ?

La fréquence propre est donnée par :

[37]:
```
# Constante d'élasticité du piézo
Kf=E*S/H # [N/m]=[N/m2]*S[m2]/H[m]

# Relation entre masse, constante et pulsation propre
w=sp.sqrt(Kf/Ms)

# Fréquence propre
f=w/(2*sp.pi)
deq("f",f)
```

[37]:

$$f = \frac{\sqrt{\frac{D^2 E}{H M_s}}}{4\sqrt{\pi}}$$

[38]:
```
deq("K_f",Kf)
```

[38]:

$$K_f = \frac{\pi D^2 E}{4H}$$

[39]:
```
# Valeur numérique
print("La fréquence propre est de {:.2f} MHz".format(f.subs(valeurs).
 ↪evalf()*1e-6))
```

```
La fréquence propre est de 1.20 MHz
```

1.1 Q3

Le capteur complet (avec la masse sismique, le quartz et le boîtier) pèse 5g. Après avoir fixé le capteur sur la masse M on observe que la fréquence propre de l'ensemble suspendu au ressort K s'élève à $f = 10Hz$.

- Déterminer la masse M ?
- Si on enlève l'accéléromètre, que devient la fréquence d'oscillation libre de la masse M suspendue au ressort de rigidité K ?
- Calculer l'erreur relative de fréquence introduite par l'installation de l'accéléromètre.

[40]:
```
# On cherche la masse à partir de la relation entre K,M et la pulsation
# Déclaration des variables littérales
Mtot, fp, Macc, M, K = sp.symbols('M_tot, f_p, M_acc, M, K')

# Equations
Eq1=sp.Eq(fp,sp.sqrt(K/Mtot)/(2*sp.pi)) # Pulsation propre
Eq2=sp.Eq(Mtot, M+Macc)                 # Masse totale

# Résolution des equations pour M, on élimine Mtot
sol=sp.solve([Eq1,Eq2],[M, Mtot])
deq("M",sol[0][0])
```

[40]:

$$M = \frac{K}{4\pi^2 f_p^2} - M_{acc}$$

[41]:
```
# On a les valeurs numériques suivantes :
valeurs2={fp:10, Macc:5e-3, K:vK}

# Ce qui permet de calculer la valeur numérique
vM=sol[0][0].subs(valeurs).subs(valeurs2).evalf()
print("La masse pèse {:.1f} [g]".format(vM*1e3))
```

```
La masse pèse 20.3 [g]
```

[42]:
```
# Calcul de la fréquence d'oscillation sans la masse du capteur
ff=(sp.sqrt(K/M)/(2*sp.pi)).subs(valeurs).subs(valeurs2).subs(M,vM)
print("la fréquence d'oscillation libre est {:.2f} [Hz]".format(ff.evalf()))
```

```
la fréquence d'oscillation libre est 11.16 [Hz]
```

[43]:
```
# L'erreur de fréquence est donnée par
e=((10-ff)/ff).evalf()
print("L'erreur est de ",e*100,"[%]")
```

```
L'erreur est de  -10.4116128073391 [%]
```

1.2 Q4

La tension produite par l'accéléromètre est enregistrée avec un instrument d'une résolution $R_u = 10uV$. - En régime d'oscillation libre, quelle est l'amplitude minimale des oscillations qui peut être mesurée (ou en d'autres termes : quelle est l'amplitude minimale en [m] des vibrations que le capteur est capable de détecter) ?

[44]:
```
# a * Uc = Ru = 10 [uV]
vRu=10e-6 # Tension minimale
```

```python
va0 = (vRu/vS) # accélération minimale
print("Le capteur a une sensibilité minimale S={:.3f} [m/s^2]".format(va0))
```

```
Le capteur a une sensibilité minimale S=0.010 [m/s^2]
```

[45]:
```python
# Calcul de l'amplitude minimale des oscillations.

# Déclaration des variables
a0,t=sp.symbols('a0,t')

# on indique que w est positif pour simplifier la solution
w=sp.symbols('w', positive=True)

# Equation de l'accélération et calcul de la position dans le temps
a=a0*sp.sin(w * t)
v=a.integrate(t)
x=v.integrate(t)
print("La position comme fonction du temps est x(t)=")
deq("x(t)",x)
```

```
La position comme fonction du temps est x(t)=
```

[45]:

$$x(t) = -\frac{a_0 \sin(tw)}{w^2}$$

[46]:
```python
# L'amplitude du mouvement est donc
x0=-x/a*a0
deq("x_0",x0)
```

[46]:

$$x_0 = \frac{a_0}{w^2}$$

[47]:
```python
deq("x0",x0.subs(a0, va0).subs(w,10*2*sp.pi).evalf())
```

[47]:

$$x_0 = 2.55713409871164 \cdot 10^{-6}$$

[ ]:
```python

```