

Détermination d'une inductance

On souhaite déterminer l'impédance d'une bobine modélisée par une inductance L et une résistance r . On dispose pour cela d'un générateur de signaux continus et variables de résistance interne nulle et d'une boîte à décade de résistance notée R_0 .

Montage 1 : le générateur, R_0 et la bobine sont en série :

- Une première mesure est effectuée en continu. Aux bornes de $R_0 = 2\Omega$ on relève une tension deux fois plus faible que celle du générateur
- Une deuxième mesure est effectuée en régime sinusoïdal établi. A la fréquence $f=1\text{kHz}$ avec $R_0 = 10\Omega$ on relève les tensions suivantes aux bornes de R_0 et de la bobine

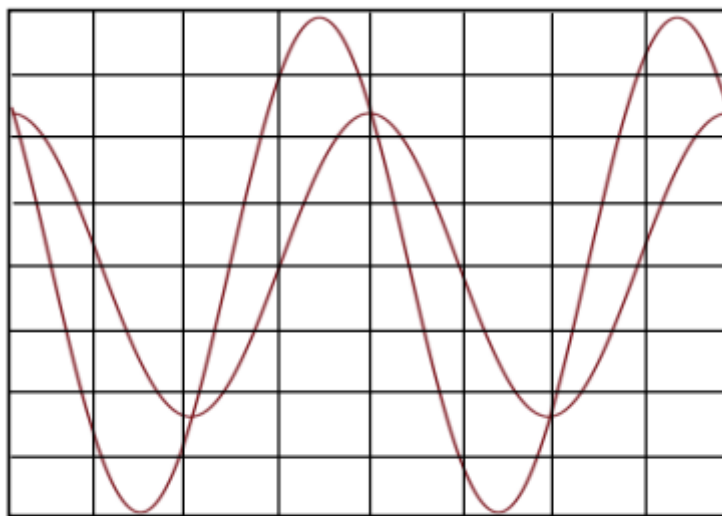
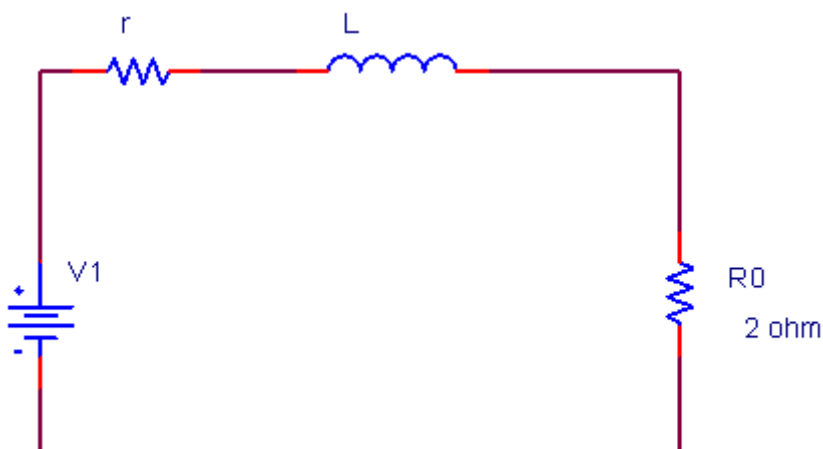


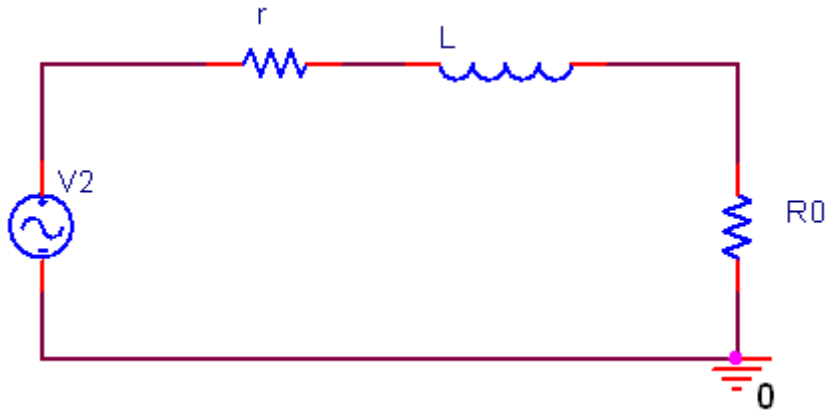
Figure 1

1) Faire un dessin du montage des deux mesures et préciser les difficultés expérimentales à éviter.



En continu, l'inductance est équivalente à un fil et le schéma se ramène à un pont diviseur de tension, ainsi $r = R_0 = 2\Omega$

- Mesurer la tension aux bornes du générateur (Voie1) et de la résistance (Voie2). A l'oscilloscope on visualise (Voie1-Voie2) et (Voie2)
- Utiliser un transformateur d'isolement
- Utiliser des sondes différentielles



2) Déterminer les valeurs expérimentales de L et r.

La tension aux bornes de la bobine est en déphasage avec le courant qui la traverse :

$$\underline{u} = (r + jL\omega) \underline{i}$$

La phase de l'impédance donnant le déphasage ϕ entre la tension et le courant, on trouve :

$$\tan\phi = \frac{L\omega}{r}$$

La lecture de l'oscillogramme donne 4 carreaux = 1ms, le déphasage est voisin de 0,5 carreaux soit 45° .
Donc :

$$L = \frac{r}{2\pi f} \tan\phi \approx 0.31\text{mH}.$$

3) Critiquer la mesure. Comment peut-on procéder autrement pour mesurer L et R ?

La mesure du déphasage est peu précise (on peut l'améliorer avec un oscilloscope numérique donnant une lecture directe, la méthode des neufs carreaux, utiliser directement un RLC-mètre...)

On utilise la bobine précédente dans le montage 2 ci-dessous où $R = 4\Omega$ et $C=1\mu\text{F}$.

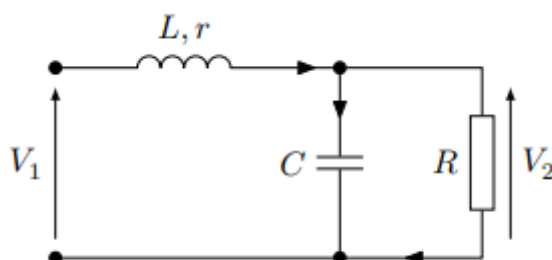


Figure 2

4) Obtenir la fonction de transfert $\underline{H} = \frac{V_2}{V_1}$ sous la forme :

$$\underline{H} = \frac{V_2}{V_1} = \underline{H} = \frac{H_0}{1 + j\frac{\omega}{Q\omega_0} - \left(\frac{\omega}{\omega_0}\right)^2}$$

.

On exprimera Q, ω_0 et H_0 .

$$\begin{aligned} \underline{H} &= \frac{Z_2}{Z_1 + Z_2} = \frac{\frac{R}{RjC\omega + 1}}{\frac{R}{RjC\omega + 1} + (r + jL\omega)} = \frac{R}{R + (r + jL\omega)(RjC\omega + 1)} = \underline{H} = \frac{R}{R + rRjC\omega + r + jL\omega +} \\ &= \frac{H_0}{1 + j\frac{\omega}{Q\omega_0} - \left(\frac{\omega}{\omega_0}\right)^2} \end{aligned}$$

Par identification, on a :

$$H_0 = \frac{R}{R+r} \approx 0,6, \omega_0 = \sqrt{\frac{r+R}{LRC}} \approx 70000 \text{ rad/s}, Q\omega_0 = \frac{r+R}{rRC+L} \text{ soit } Q = \frac{\sqrt{(r+R)RLC}}{rRC+L} \approx 2,7 \times 10^{-1}$$

5) Obtenir le diagramme de Bode en gain en complétant et en utilisant la fonction `diagramme()` ci-dessous. Retrouver les propriétés caractéristiques de ce filtre sur le diagramme de Bode.

In [2]:

```
1 import matplotlib.pyplot as plt
2 import numpy as np
3 def diagramme():
4     def G(w, H0, Q, w0):
5         return (20.0 * np.log10(abs(H(w, H0, Q, w0))))
6     w = np.linspace(1000, 1e7, 10000)
7     plt.title('diagramme de Bode en gain')
8     plt.grid()
9     plt.semilogx(w, G(w, H0, Q, w0))
10    plt.show()
```

In [4]:

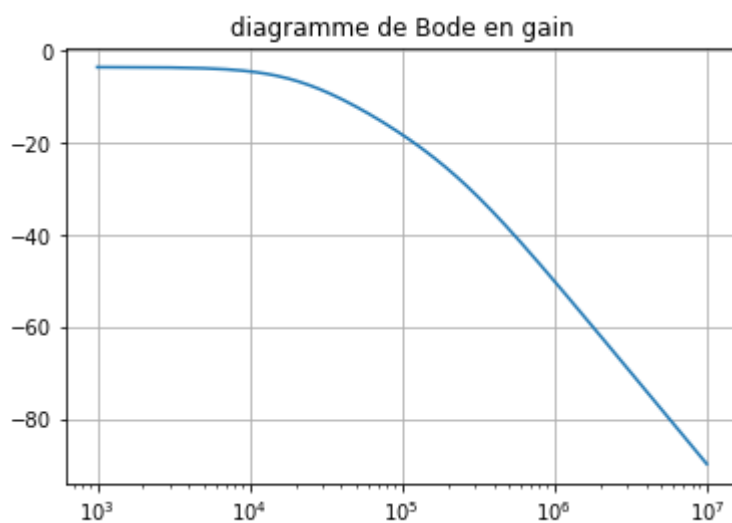
```

1 L=0.00031
2 r=2
3 C=1e-6
4 R=4
5 H0=R/(R+r)
6 w0=((R+r)/(R*L*C))**0.5
7 print(w0/(2*np.pi))
8 Q=(R+r)/((L+r*R*C)*w0)
9 print(Q)
10 def H(w,H0,Q,w0):
11     return H0/(1+1j*w/(Q*w0)-(w/w0)**2)
12 diagramme()

```

11070.950634631195

0.27124350506726824



La fréquence propre est donc de 11kHz avec un facteur de qualité ne permettant aucune résonance (la pente est de -40dB/décade). On peut aussi voir la combinaison de deux filtres d'ordre 1 :

$$H \approx \frac{H_0}{1+j\frac{w}{w_1}} \times \frac{1}{1+\frac{jw}{w_2}} \approx \frac{H_0}{1+j\left(\frac{w}{w_1}\right) + \left(\frac{jw}{w_1 w_2}\right)^2}$$

On soumet le filtre à un signal carré d'amplitude de 1V et de fréquence F_e . On donne :

- la fonction créneau permettant de générer ce signal carré
- la fonction spectre permettant de calculer les coefficient de Fourier d'un signal périodique
- La fonction trace_spectre permettant d'obtenir le spectre d'un signal de fréquence F_e

In [5]:

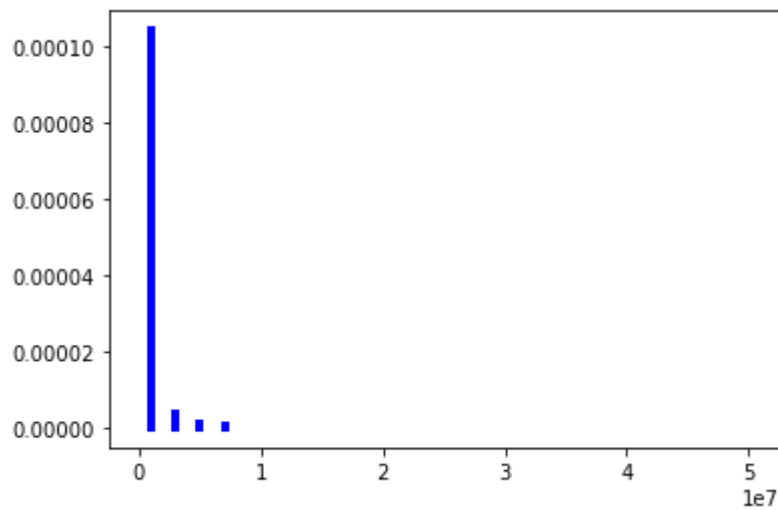
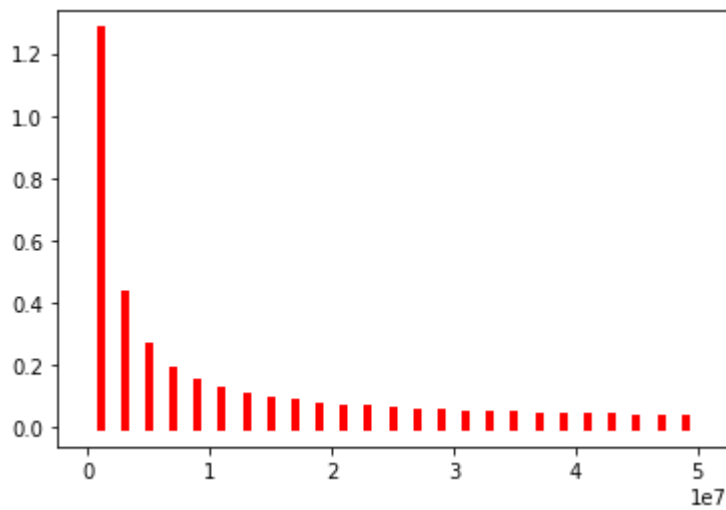
```

1 import scipy.integrate as integr
2 def creneau(t):
3     if t<0: return(-1.0)
4     else: return(1.0)
5
6 def spectre(signal):
7     # coefficients de Fourier réels du signal
8     #creation d'une liste de 50 zéros
9     bnSignal = np.zeros(51)
10    for k in range(1,51):
11        #pour modifier la kème élément de liste:bnSignal[k]
12        #pour intégrer la fonction simple signal(t)*sin(2*pi*k*t) en fonction
13        # de k à chaque intégration d'où le args=(k) entre -0.5 et +0.5
14        # le [0] permet d'intégrer le calcul d'intégral dans la liste
15        bnSignal[k] = 2.0 * integr.quad(lambda t,k: signal(t)*np.sin(2*np.pi*k*t),-0.5,0.5)[0]
16
17    anSignal = np.zeros(51)
18    anSignal[0] = integr.quad(lambda t: signal(t),-0.5,0.5)[0] #valeur moyenne
19    for k in range(1,51):
20        anSignal[k] = 2.0 * integr.quad(lambda t,k: signal(t)*np.cos(2*np.pi*k*t),-0.5,0.5)[0]
21    return anSignal,bnSignal
22 def trace_spectre(signalentree,Fe):
23     #on applique la fonction spectre qui renvoie 2 valeurs renotée an et bn
24     an,bn=spectre(signalentree)
25     for k in range(0,51):
26         #tracé des raies du spectre du signal d'entrée
27         plt.plot([k*Fe,k*Fe],[0,np.sqrt((an[k])**2+(bn[k])**2)],'r',linewidth=2)
28     plt.show()
29     for k in range(0,51):
30         cn=np.sqrt((an[k])**2+(bn[k])**2)
31         #tracé du spectre du signal de sortie
32         plt.plot([k*Fe,k*Fe],[0,cn*abs(H(2*np.pi*k*Fe,H0,Q,w0))],'b',linewidth=2)
33     plt.show()
34

```

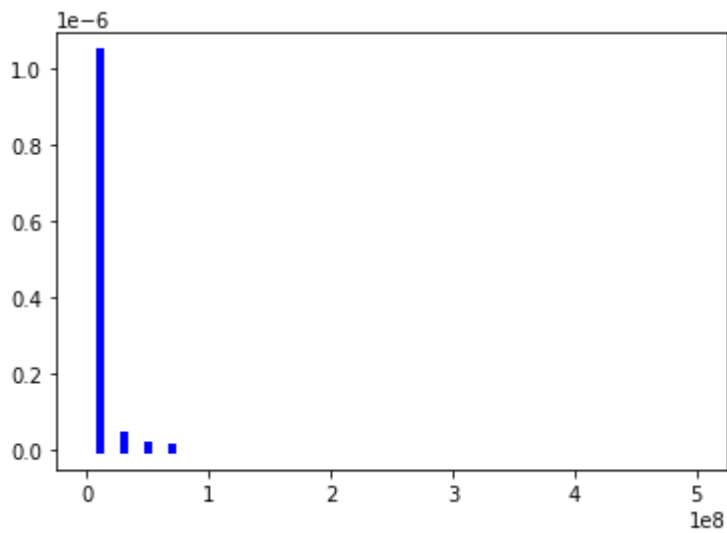
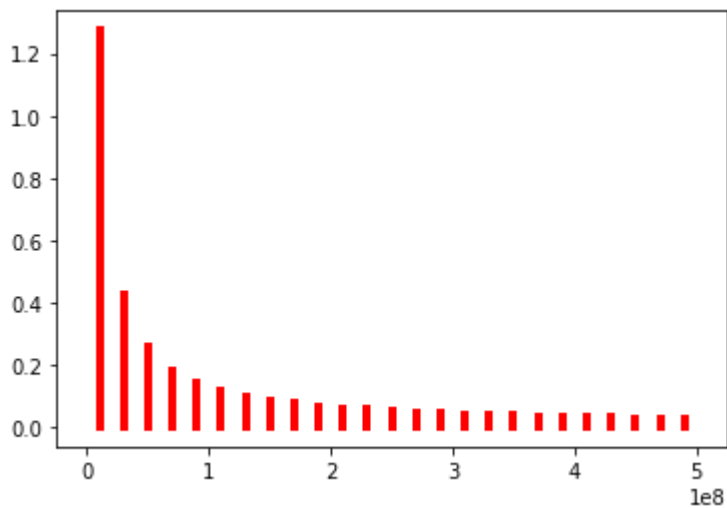
In [6]:

```
1 trace_spectre(creneau,10**6)
```



In [7]:

```
1 trace_spectre(creneau, 10**7)
```



Pour le spectre de raies, on apprécie l'effet du filtre. On peut comparer l'atténuation entre deux fréquences dans un rapport dix qui s'effectue dans un rapport 100.

In []:

```
1
```

In []:

```
1
```

In []:

```
1
```