

Filtre LC passe-bande

1. Introduction

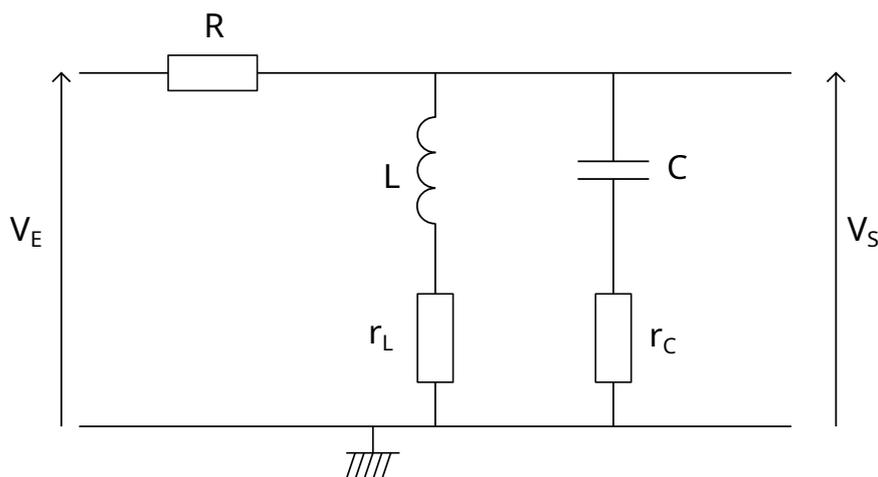
L'objectif est de concevoir et de réaliser un filtre LC passe-bande, constitué d'une bobine, d'un condensateur et d'une résistance. Le filtre devra avoir un facteur de qualité assez grand pour extraire le fondamental d'un signal périodique dont la fréquence est d'environ 10 kHz.

Matériel :

- ▷ Tore en ferrite.
- ▷ Fil souple de 70 cm.
- ▷ Plaque à essai et fils.
- ▷ Condensateur polyester 1 μF .
- ▷ Résistances 1 k Ω et 10 k Ω .
- ▷ Générateur de signaux.
- ▷ Oscilloscope.
- ▷ Carte Sysam SP5.

2. Modélisation du filtre

Le filtre comporte une bobine et un condensateur connectés en parallèle, et une résistance. Afin de tenir compte des pertes, la bobine est modélisée par une auto-inductance L en série avec une résistance r_L , le condensateur et modélisé par une capacité C en série avec une résistance r_C .



En première approche, on néglige les pertes dans la bobine et dans le condensateur : on pose donc $r_L = 0$ et $r_C = 0$.

[1]  Déterminer la fonction de transfert du filtre et la mettre sous la forme :

$$\underline{H}(\omega) = \frac{1}{1 + jQ \left(\frac{\omega}{\omega_0} - \frac{\omega_0}{\omega} \right)} \quad (1)$$

Exprimer ω_0 en fonction de L et C , et démontrer que :

$$Q = R\sqrt{\frac{C}{L}} \quad (2)$$

[2]  On souhaite avoir une fréquence de résonance de 10 kHz. Le condensateur a une capacité $C = 1,0 \mu\text{F}$. Calculer la valeur de L puis le facteur de qualité pour $R = 1000 \Omega$.

Le script suivant (à compléter) permet d'obtenir la courbe du gain en fonction de la fréquence en tenant compte des résistances r_L et r_C . Les valeurs de ces résistances sont données et correspondent à peu près aux valeurs pour la bobine et le condensateur qui seront utilisés (mesurées avec un RLC-mètre à la fréquence de 10 kHz).

[gain.py](#)

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

def transfert(f,R,L,rL,rC):
    # compléter
    return H
R = 1000
L = 253e-6
rL = 0.22
C = 1e-6
rC = 0.15
f = np.linspace(1e3,20e3,1000)
G = np.absolute(transfert(f,R,L,rL,rC))
G_ideal = np.absolute(transfert(f,R,L,0,0))
plt.figure()
plt.plot(f,G)
plt.plot(f,G_ideal)
plt.grid()
plt.xlabel("f (Hz)", fontsize=16)
plt.ylabel('G', fontsize=16)
plt.show()
i = np.argmax(G)
f1 = f[i]
print("f1 = %f"%f1)
G1 = np.absolute(transfert(f1,R,L,rL,rC))
G2 = np.absolute(transfert(2*f1,R,L,rL,rC))
G3 = np.absolute(transfert(3*f1,R,L,rL,rC))
print(G2/G1,G3/G1)
```

[3]  Tracer la courbe pour $R = 1000 \Omega$. Comparer à la courbe de gain du filtre idéal (sans pertes dans la bobine et dans le condensateur). Pour un signal périodique de fréquence 10 kHz en entrée du filtre, quelle est la signification des rapports G_2/G_1 et G_3/G_1 calculés dans le script? Commenter ces valeurs.

[4]  Quel est l'intérêt d'augmenter la valeur de R ? Quel est l'inconvénient? Compléter le script par les lignes suivantes :

```
R = np.logspace(2,5,1000)
G1 = np.absolute(transfert(f1,R,L,rL,rC))
G2 = np.absolute(transfert(2*f1,R,L,rL,rC))
```

```
G3 = np.absolute(transfert(3*f1,R,L,rL,rC))
plt.figure()
plt.plot(R,G2/G1,label='G2/G1')
plt.plot(R,G3/G1,label='G3/G1')
plt.grid()
plt.xscale('log')
plt.xlabel(r"$R\ (\rm\Omega)$", fontsize=16)
plt.legend(loc='upper right', fontsize=16)
plt.show()
```

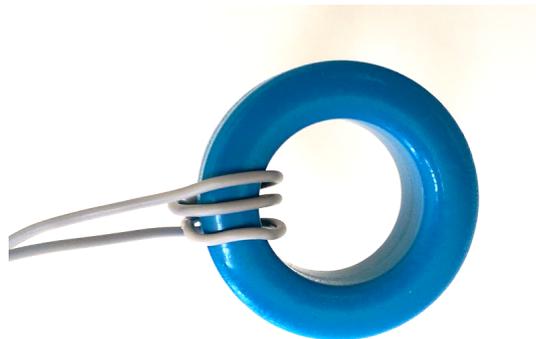
[5]  Commenter les courbes obtenues. L'expression (2) du facteur de qualité est-elle pertinente ? Pourquoi la valeur $R = 1000\ \Omega$ est-elle un bon compromis ?

3. Réalisation de la bobine

Pour réaliser une bobine d'induction possédant une forte inductance pour un faible encombrement, on utilise un noyau en ferrite. La ferrite est une céramique ferromagnétique à base d'oxyde de fer. Par rapport à un noyau en fer, le noyau en ferrite a l'avantage d'être non conducteur, ce qui évite les pertes par courant induit. La présence d'un noyau en ferrite dans une bobine a pour effet d'augmenter la perméabilité magnétique d'un facteur 1000 ou plus.

On utilise un noyau en ferrite sous forme de tore, qui permet de réaliser des bobines ou des transformateurs. La perméabilité magnétique relative est de 4300, ce qui signifie qu'une bobine donnera un flux magnétique 4300 fois plus élevé que la même bobine sans noyau.

La photo suivante montre un enroulement de 2,5 spires sur le noyau en ferrite.



L'autoinductance de la bobine obtenue avec N spires est en principe :

$$L = A_L N^2 \quad (3)$$

La valeur de A_L est donnée par le fabricant du tore : $A_L = 6400\ \text{nH} \pm 30\%$.

[6]  Calculer le nombre de spires N pour obtenir la valeur de L souhaitée. Réaliser l'enroulement.

4. Étude expérimentale du filtre

[7] Réaliser le filtre avec $R = 1000 \Omega$ et vérifier qualitativement son fonctionnement.

[8]  En précisant la méthode utilisée, mesurer avec précision la fréquence de résonance f_1 . Donner sa valeur avec son incertitude.

[9]  Mesurer le gain à la résonance G_1 puis les gains G_2 et G_3 respectivement aux fréquences $2f_1$ et $3f_1$.

[10]  Comparer G_1 à la valeur donnée par le modèle. Comparer aussi les rapports G_2/G_1 et G_3/G_1 .

[11]  Configurer sur le générateur de signaux un signal en dents de scie de fréquence fondamentale f_1 . Envoyer ce signal à l'entrée du filtre puis observer simultanément l'entrée et la sortie. Commenter le résultat.

[12]  Au moyen de la fonction FFT de l'oscilloscope, faire une analyse spectrale de $V_E(t)$ pour ce signal en dents de scie. Reporter dans un tableau les amplitudes des 10 premiers harmoniques en décibel.

[13]  Faire de même l'analyse spectrale de $V_S(t)$.

[14]  Les amplitudes des harmoniques relevées sur ces spectres sont-elles en accord avec les valeurs de G_2/G_1 et G_3/G_1 déterminées précédemment ?

5. Numérisation et analyse spectrale

Afin d'améliorer la précision de l'analyse spectrale, on souhaite réaliser celle-ci en Python à partir d'un signal numérisé. La numérisation est faite avec la carte Sysam SP5 et le logiciel LatisPro. Le nombre d'échantillons d'une acquisition est $N = 2^{17}$ pour $V_S(t)$ et pour $V_E(t)$. On note f_e la fréquence d'échantillonnage et T la durée de l'enregistrement (N échantillons). On rappelle que la fréquence du signal analysé est f_1 , la fréquence de résonance du filtre. La précision de numérisation de la carte Sysam SP5 est de 12 bits.

[15]  On souhaite que la précision en fréquence de l'analyse spectrale soit de 10 Hz. Quelle valeur faut-il choisir pour f_e ?

[16]  Même question pour une précision de 1 Hz. Quel problème ce choix pose-t-il pour l'analyse du signal $V_E(t)$?

[17]  Si une tension sinusoïdale d'amplitude 5 V est numérisée avec le calibre +/- 10 V, quelle est en principe la précision sur les valeurs de tension ?

On fera la numérisation à une fréquence d'échantillonnage $f_e = 1$ MHz. L'analyse spectrale par transformée de Fourier discrète sera faite après avoir appliqué une fonction de fenêtrage au signal puis l'avoir complété par des zéros. La fonction suivante effectue ce calcul.

```

from scipy.signal.windows import get_window
from numpy.fft import fft

def spectre(s, Te, win, nz):
    N = len(s)
    if win=='rect':
        w = np.ones(N)
        a = 1
    elif win=='hamming':
        w = get_window('hamming', N)
        a = 0.54
    elif win=='hann':
        w = get_window('hann', N)
        a = 0.5
    elif win=='blackman':
        w = get_window('blackman', N)
        a = 0.42
    else:
        return None
    p = int(np.log(nz*N)/np.log(2))+1
    Np = 2**p
    u = np.concatenate((s*w, np.zeros(Np-N)))
    tfd = fft(u)/N/a
    freq = np.arange(Np)*1/(Np*Te)
    return freq, tfd

```

s est le tableau contenant le signal échantillonné, T_e est la période d'échantillonnage, win est le nom de la fenêtre (en prendra $win='blackman'$) et nz est le nombre de blocs de N zéros ajoutés (en prendra $nz=5$). La fonction renvoie le tableau des fréquences et le tableau contenant la transformée de Fourier discrète (TFD) normalisée. Le spectre est simplement le module de cette TFD.

[18] Brancher les voies EA0 et EA1 de la carte Sysam SP5 sur respectivement l'entrée et la sortie du filtre. Choisir l'amplitude du signal $V_E(t)$ en dents de scie et le calibre de numérisation en conséquence.

[19] Numériser les deux signaux avec le logiciel LatisPro et exporter dans un fichier texte. Dans un script Python, lire ce fichier au moyen de la fonction `numpy.loadtxt` puis calculer les spectres des deux signaux. Tracer ces spectres en échelle décibel.

[20]  Refaire l'analyse spectrale faite précédemment avec l'oscilloscope.

[21]  Faire l'analyse pour $R = 10\text{ k}\Omega$ (attention au choix du calibre pour la voie EA1).

[22]  Conclusion : dans quelle mesure ce filtre permet-il d'extraire le fondamental d'un signal périodique de fréquence f_1 ?