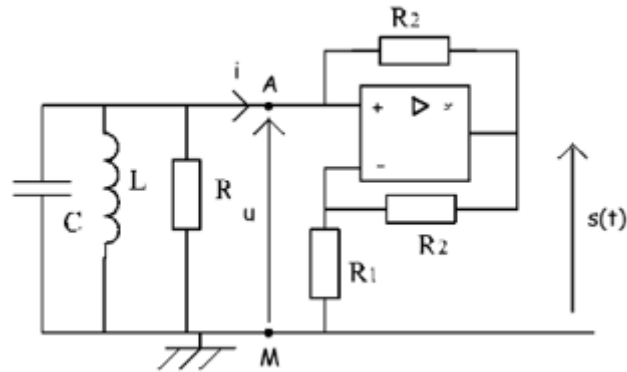


Oscillateur quasi-sinusoïdal

On considère l'oscillateur suivant dans lequel l'AO est idéal et en régime linéaire :



1) Montrer que $u = -R_1 i$

L'AO étant idéal (pas de courants de polarisation) :

$$u = \frac{R_1 s}{R_1 + R_2}$$

D'après la loi d'Ohm :

$$u - s = R_2 i$$

$$u - \frac{R_1 + R_2}{R_1} u = R_2 i$$

$$u = -R_1 i$$

2) Ecrire les relations tension-courant pour chaque dipôle de la cellule RLC.

$$u = -Ri$$

$$i = -C \frac{du}{dt}$$

$$u = -L \frac{di}{dt}$$

3) Ecrire la loi des noeuds en A et en déduire l'équation différentielle vérifiée par la tension $u(t)$

$$-\frac{u}{R} - C \frac{du}{dt} - \int \frac{u}{L} dt = -\frac{u}{R_1}$$

On dérive une fois :

$$\frac{d^2 u}{dt^2} + \left(\frac{1}{RC} - \frac{1}{R_1 C} \right) \frac{du}{dt} + \frac{u}{LC} = 0$$

4) A quelle condition sur R_1 cette équation différentielle admet-elle une solution parfaitement sinusoïdale ?

La condition théorique d'entretien des oscillations est $R = R_1$

5) Retrouver ce résultat en proposant une modélisation en schéma bloc de cette structure et en utilisant la condition de Barkhausen.

Sous forme de fonction de transfert, on a : $u = \frac{R_1 s}{R_1 + R_2}$

$$u = \frac{Z_{eq} s}{Z_{eq} + R_2}$$

Donc la fonction de transfert en BO :

$$T = \frac{R_1}{R_1 + R_2} \frac{Z_{eq} + R_2}{Z_{eq}}$$

L'absence de partie imaginaire impose la même pulsation d'oscillation.

Et un module unitaire $R = R_1$

Pour observer un démarrage des oscillations, on fixe $R_1 < R$. Dans ces conditions l'AO idéal atteint le régime saturé.

6) Donner l'équation différentielle vérifiée par $u(t)$ en régime saturé.

$$-\frac{u}{R} - C \frac{du}{dt} - \int \frac{u}{L} dt = \frac{u - V_s}{R_2}$$

On dérive une fois :

$$\frac{d^2 u}{dt^2} + \left(\frac{1}{RC} + \frac{1}{R_2 C} \right) \frac{du}{dt} + \frac{u}{LC} = 0$$

On donne les variables globales ci-dessous :

In [5]:

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 C=100*10**-9
4 L=10*10**-3
5 R=10**3
6 R2=10**3
7 w0=(L*C)**-0.5
8 T0=2*np.pi/w0
9 Te=T0/1000
10 t=np.arange(0,10*T0,Te)
11 u=np.zeros((2,len(t)))
12 u[0,0]=0.1
13 Vsat=15
14
```

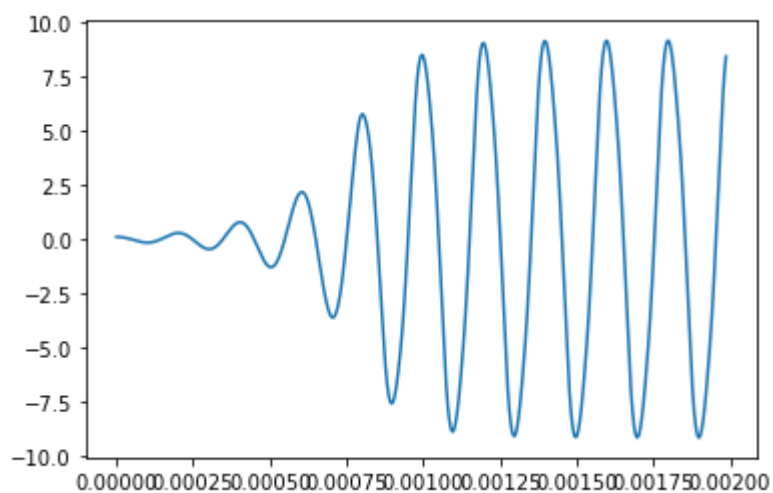
7) Proposer un programme permettant d'apprécier le démarrage des oscillations en fonction des valeurs de R_1

In [11]:

```

1 def oscillo(R1):
2     for i in range(len(t)-1):
3         u[0,i+1]=u[0,i]+Te*u[1,i]
4         if abs(u[0,i])<(R1)/(R1+R2)*Vsat:
5             K=1/(R*C)-1/(R1*C)
6             u[1,i+1]=u[1,i]-Te*K*u[1,i]-Te*w0**2*u[0,i]
7         else:
8             K=1/(R*C)+1/(R2*C)
9             u[1,i+1]=u[1,i]-Te*K*u[1,i]-Te*w0**2*u[0,i]
10    return u[0],u[1]
11 u[0],u[1]=oscillo(0.5*R)
12 plt.plot(t,u[0])
13 plt.show()

```



In []:

1