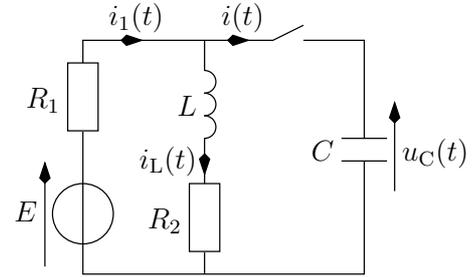


# DM n° 6 de Physique

## Régime transitoire d'ordre 2

### Étude de la charge d'un condensateur

On considère le circuit ci-contre et on souhaite y étudier l'évolution transitoire de la tension  $u_C$ . Au début de l'expérience que l'on modélise, le condensateur est chargé à la tension  $u_C = u_0$  et l'interrupteur est ouvert *depuis très longtemps*.



#### A Valeurs initiales et finales

On souhaite dans un premier temps établir les valeurs initiales et finales des courants et tensions.

1. À l'aide d'un nouveau schéma, déterminer la valeur des courants  $i(t=0^-)$ ,  $i_L(t=0^-)$  et  $i_1(t=0^-)$  avant la fermeture de l'interrupteur.
2. En déduire la valeur de  $u_C(t=0^+)$ ,  $i_1(t=0^+)$ ,  $i_L(t=0^+)$  et  $i(t=0^+)$  avant la fermeture de l'interrupteur.
3. À l'aide d'un schéma supplémentaire, déterminer la valeur de la tension  $u_C(t=\infty)$  en régime permanent, au bout d'un temps très long.

#### B Équation différentielle

On cherche désormais à établir l'équation différentielle régissant l'évolution de la tension  $u_C$ .

4. Donner les quatre équations électriques reliant  $i_1$ ,  $i_L$ ,  $i$  et  $u_C$ . Ces équations peuvent faire apparaître les dérivées de ces fonctions et bien sûr les données du problème ( $E$ ,  $R_1$ ,  $R_2$ ,  $L$  et  $C$ ).
5. À partir d'une de ces équations, exprimer  $i_1$  en fonction  $u_C$  et des données du problème.
6. À partir de deux équations parmi celles de la question 4, établir une relation entre  $i_1$ ,  $i$ ,  $\frac{di}{dt}$  et  $\frac{di_1}{dt}$ .
7. Déduire des deux questions précédentes une relation reliant  $u_C$ ,  $i$  et leurs dérivées, faisant intervenir  $E$ ,  $R_1$ ,  $R_2$  et  $L$ .
8. En déduire finalement l'équation différentielle régissant l'évolution de la tension  $u_C$ .
9. Montrer que la pulsation propre  $\omega_0$  et le facteur de qualité  $Q$  du système s'écrivent

$$\omega_0 = \sqrt{\frac{R_1 + R_2}{R_1 L C}} \quad \text{et} \quad Q = \frac{\sqrt{R_1 (R_1 + R_2) L C}}{L + R_1 R_2 C}$$

#### C Résolution de l'équation différentielle

On choisit  $R_1 = 470 \Omega$ ,  $R_2 = 10 \Omega$ ,  $L = 8 \text{ mH}$ ,  $C = 2200 \text{ nF}$ ,  $E = 20 \text{ V}$  et  $u_0 = 1 \text{ V}$ .

10. Application numérique : calculer  $Q$ . De quel régime s'agit-il ?
11. En déduire l'expression complète de  $u_C(t)$ , sans déterminer les constantes d'intégration.
12. Exprimer les deux conditions initiales  $u_C(0)$  et  $\dot{u}_C(0)$  et les calculer (en V et en V/ms).
13. Tracer qualitativement l'évolution de  $u_C(t)$ . Il faut pour cela calculer numériquement la constante de temps de l'enveloppe exponentielle, la valeur finale, et la pseudo-période.
14. Déterminer les constantes d'intégration. Conseil : utiliser une forme  $A \cos(\omega_p t) + B \sin(\omega_p t)$  pour la partie pseudo-périodique et noter  $\dot{u}_C(0)$  pour ces calculs, que l'on remplacera au dernier moment.

## D Cadeau bonus

Voilà le script qui trace  $u(t)$ . Mais il faut le télécharger et le compléter pour avoir droit à la figure!

```

1  # Importation des modules nécessaires
2  import numpy as np
3  import matplotlib.pyplot as plt
4
5  # Script à compléter. Inutile de le lancer sans avoir complété !
6
7  # Données
8  E =
9  u0 =
10 r1 =
11 r2 =
12 l = 8e-3 # Éviter 8*10**(-3), bien moins précis. 8e-3 est un simple nombre
13 c =
14
15 # Constantes définies dans le sujet et permettant de réaliser le tracé
16 Q = np.sqrt()/()
17 omega0 = ...
18 tau = 2*Q/omega0
19 omegap = omega0/(2*Q) * np.sqrt(...)
20 up0 = 1/(r1*c) * ( r2/(r1+r2)*E - u0 )
21 ufinal = ...*E
22 # Affichage
23 print("Q      =",round(Q,2))
24 print("omega0 =",int(omega0),'rad/s')
25 print("tau     =",round(1000*tau,3),'ms')
26 print("omegap  =",int(omegap),'rad/s')
27 print("taup    =",round(1000*.../omegap,3),'ms')
28 print("up0     =",round(up0/1000,3),'V/ms')
29 print("ufinal  =",round(ufinal,3),'V')
30
31 # Constantes d'intégration
32 A = u0-ufinal
33 B = ( up0 + A/tau ) / omegap
34
35 # Fonctions temporelles
36 def enveloppe(t):
37     return np.exp(-t/tau)
38 def u(t):
39     return ufinal + enveloppe(t) * ( A*np.cos(omegap*t) + B*np.sin(...) )
40
41 # Tracé de l'évolution du système
42 t = np.linspace(0, ..., 2001) # de 0 à 5*tau, 2001 points : dt = 5*tau/2000
43 plt.plot(1000*t, ...) # temps en abscisses en millisecondes
44 plt.xlabel('Temps (ms)')
45 plt.ylabel('Tension u_c (V)')
46 plt.grid()
47 # Enveloppe exponentielle
48 plt.plot(1000*t, ufinal + np.sqrt(A**2+B**2) * enveloppe(t), 'r--')
49 plt.plot(..., ufinal - np.sqrt(A**2+B**2) * ..., 'r--')
50 plt.show()

```