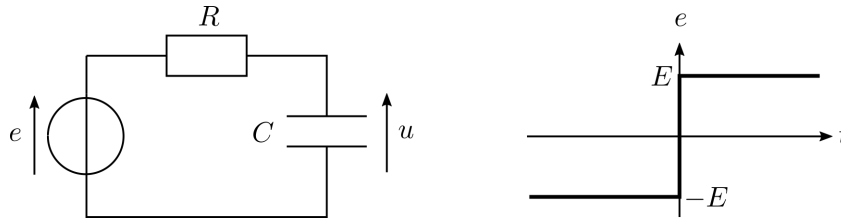


Temps de réponse d'un circuit RC

On considère un circuit RC soumis à un échelon de tension.



1. Établir la loi $u(t)$ pour $t \geq 0$
2. Tracer l'allure du graphe $u(t)$.
3. Établir l'expression du temps de réponse à 50%, t_{50} .

On branche aux bornes d'un GBF, un condensateur de capacité C (variable) et une résistance $R = 100 \Omega$ (boîte AOIP), en série. Le GBF délivre un signal carré de fréquence 100 Hz et d'amplitude 5 V. La précision sur la résistance de la boîte AOIP est de 0,2%.

4. Sur un schéma indiquer comment brancher l'oscilloscope pour visualiser la tension aux bornes du GBF sur la voie 1 et la tension aux bornes du condensateur sur la voie 2.
5. Mesurer le temps de réponse à 50%, pour des valeurs de C allant de 0,1 μF à 1 μF . On utilisera les curseurs de l'oscilloscope.
6. En utilisant python, tracer le graphe de t_{50} en fonction de C . Commenter.
7. En utilisant python, calculer la résistance totale du circuit \bar{R}_{tot} et son incertitude-type $u(\bar{R}_{\text{tot}})$, par une évaluation de type A.
8. En déduire la valeur de la résistance interne r du GBF, ainsi que l'incertitude-type $u(r)$ sur cette valeur.
9. Calculer l'écart normalisé avec la valeur annoncée par le constructeur : $r_c = 50 \Omega$ avec une précision de 3%. Commenter.
10. Modifier le montage pour visualiser l'image de $i(t)$ sur la voie 2 de l'oscilloscope. Pourquoi la tension aux bornes du GBF n'est-elle pas tout à fait un signal carré?

Temps de réponse d'un circuit RC

1. $i(0^-) = C \frac{du}{dt}(0^-) = 0$ (régime stationnaire)

D'après la loi des mailles, $u(0^-) = e(0^-) - Ri(0^-) = -E$

La tension aux bornes du condensateur est continue, donc $u(0^+) = u(0^-) = -E$

Loi des mailles à $t > 0$: $E - Ri - u = 0$, avec $i = C \frac{du}{dt}$, donc $E - RC \frac{du}{dt} - u = 0$

$$\frac{du}{dt} + \frac{u}{RC} = \frac{E}{RC}$$

On pose $\tau = RC$

Les solutions de l'équation homogène sont les fonctions : $u_h(t) = Ae^{-t/\tau}$

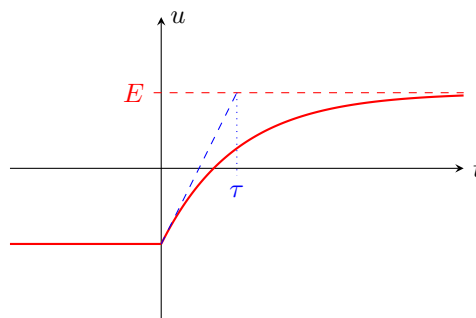
On cherche une solution particulière constante : $u_p = E$

$$u(t) = Ae^{-t/\tau} + E$$

$$u(0^+) = A + E = -E \text{ d'où } A = -2E$$

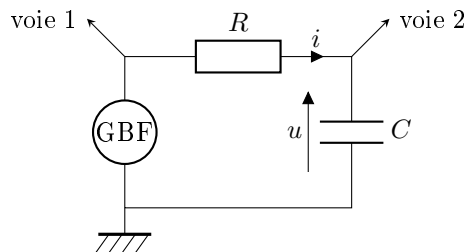
Ainsi, $u(t) = E(1 - 2e^{-t/\tau})$ avec $\tau = RC$

- 2.



3. t_{50} est défini par $u(t_{50}) - u(+\infty) = \frac{50}{100}(u(0) - u(+\infty))$, soit $u(t_{50}) = 0$
donc $1 - 2e^{-t_{50}/\tau} = 0$, donc $t_{50} = \tau \ln(2) = RC \ln(2)$

4. Le condensateur doit avoir une borne à la masse, donc commune avec le GBF.

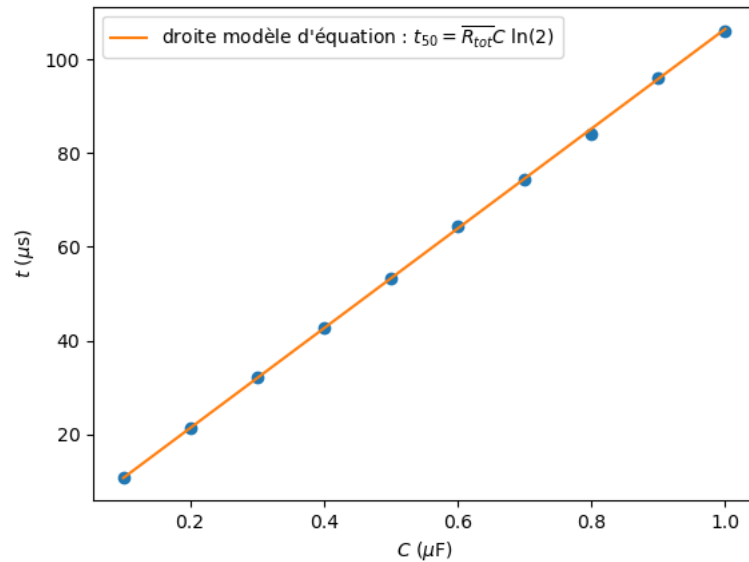


5. `import numpy as np`
`import matplotlib.pyplot as plt`

`C=np.arange(0.1, 1.1, 0.1) #uF`

`t50=np.array([10.7, 21.2, 32.2, 42.8, 53.2, 64.4, 74.4, 84, 96, 106]) #us`

6. `plt.plot(C,t50,'o')`
`plt.show()`



On obtient une droite passant par l'origine ce qui est cohérent avec l'expression $t_{50} = R_{tot}C \ln(2)$.

- On calcule les valeurs de R_{tot} obtenues pour chaque point dans un tableau. Puis on calcule la moyenne et l'incertitude-type par une évaluation de type A.

```
Rtot=t50/C/np.log(2)
Rtotmoy=np.mean(Rtot)
uRtotmoy=np.std(Rtot,ddof=1)/np.sqrt(len(C))
```

On obtient $\bar{R}_{tot} = 153,65 \Omega$ et $u(\bar{R}_{tot}) = 0,13 \Omega$

- $\bar{R}_{tot} = R + r$ donc $r = \bar{R}_{tot} - R = 53,65 \Omega$

On suppose que la distribution des valeurs de R suit une loi uniforme de demi-étendue $u(R) = \frac{0,2\%R}{\sqrt{3}} = 0,12 \Omega$

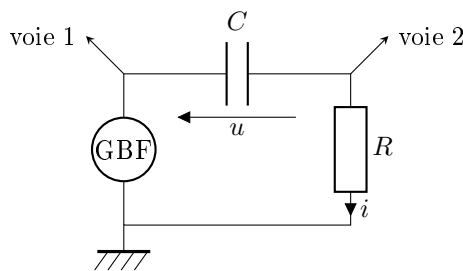
Par propagation de l'incertitude $u(r) = \sqrt{u(\bar{R}_{tot})^2 + u(R)^2} = 0,33 \Omega$

- On suppose que la distribution des valeurs de r_c suit une loi uniforme de demi-étendue $u(r_c) = \frac{3\%r_c}{\sqrt{3}} = 0,87 \Omega$

$$E_N = \frac{|r - r_c|}{\sqrt{u(r)^2 + u(r_c)^2}} = 3,9$$

$E_N > 2$ donc les 2 valeurs ne sont pas compatibles. La raison n'est pas évidente : le condensateur réel n'est probablement pas bien modélisé par une simple capacité.

- Il faut visualiser la tension aux bornes de R qui est proportionnelle à i . Pour cela la résistance doit avoir une borne à la masse, donc commune au GBF.



Le GBF n'est pas une source idéale de tension, mais peut être modélisé par un générateur de Thévenin. La tension à ses bornes est $e(t) - ri(t)$, où $e(t)$ est un signal carré. Comme $i(t)$ n'est pas un signal carré, la tension aux bornes du GBF n'est pas tout à fait un signal carré.

Régime transitoire d'un circuit RL

On branche aux bornes d'un GBF, une bobine de 1000 spires et une boîte AOIP ($\times 10 \Omega$) de résistance R , en série. Le GBF délivre un signal carré de fréquence 100 Hz et d'amplitude 10 V. Le GBF a une résistance de sortie $R_s = 50 \Omega$ avec une précision de 3%. On modélise la bobine par l'association série d'une inductance L et d'une résistance r .

1. Établir l'expression de la constante de temps du circuit en fonction de R , r , L et R_s .
2. Sur un schéma indiquer comment brancher l'oscilloscope pour visualiser la tension aux bornes du GBF sur la voie 1 et l'intensité du courant dans la bobine sur la voie 2.
3. En utilisant la carte d'acquisition et le logiciel *LatisPro*, acquérir la tension aux bornes de la résistance pour des valeurs de R allant de 10Ω à 100Ω . On paramétera l'acquisition pour acquérir 10000 points sur une durée totale d'une période.

Pour chaque valeur de R , modéliser la réponse à un échelon (ou le régime libre) par la fonction appropriée et relever la valeur de la constante de temps τ du circuit.

4. En utilisant python, tracer le graphe de R en fonction de $1/\tau$. Modéliser par une fonction affine. Dédire du coefficient directeur et de l'ordonnée à l'origine, l'inductance L et la résistance interne r de la bobine.

La précision sur R , fournie par le constructeur, est de 1%. On néglige les incertitudes sur τ .

5. En utilisant python, calculer les demi-étendues Δ_i et les incertitudes-types $u(R_i)$ sur chaque valeur R_i de R .
6. Tracer le graphe de R en fonction de $1/\tau$, avec les barres d'incertitude sur R . Commenter.
7. On définit les résidus normalisés par rapport à la droite de régression d'équation $R = a\frac{1}{\tau} + b$ par

$$R_{Ni} = \frac{R_i - a\frac{1}{\tau_i} - b}{u(R_i)}$$

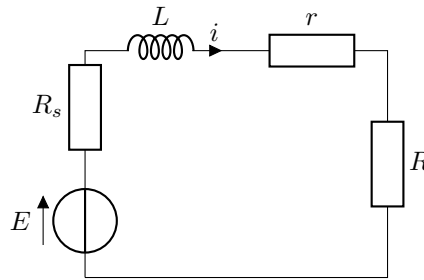
Tracer le graphe des résidus normalisés R_N en fonction de R . Commenter.

8. Pour estimer les incertitudes-types sur a et b , on réalise une simulation Monte-Carlo. On simule un grand nombre N de séries de 10 valeurs de R , distribuées selon des lois uniformes. Pour chaque série de R , on calcule le coefficient directeur a et l'ordonnée à l'origine b que l'on stocke dans deux listes. Finalement, les écarts-types des listes de N valeurs de a et de b donnent des estimations des incertitudes-types $u(a)$ et $u(b)$.

Réaliser la simulation Monte-Carlo. En déduire les incertitudes-types sur r et L . Conclure.

Régime transitoire d'un circuit RL

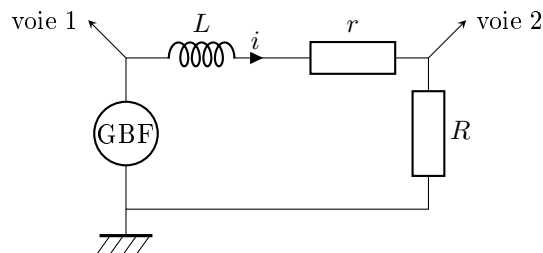
1.



Loi des mailles : $E = (R_s + r + R)i + L \frac{di}{dt}$, d'où $\frac{di}{dt} + \frac{R_s+r+R}{L}i = \frac{E}{L}$.

On identifie le temps caractéristique du circuit $\tau = \frac{L}{R_s+r+R}$

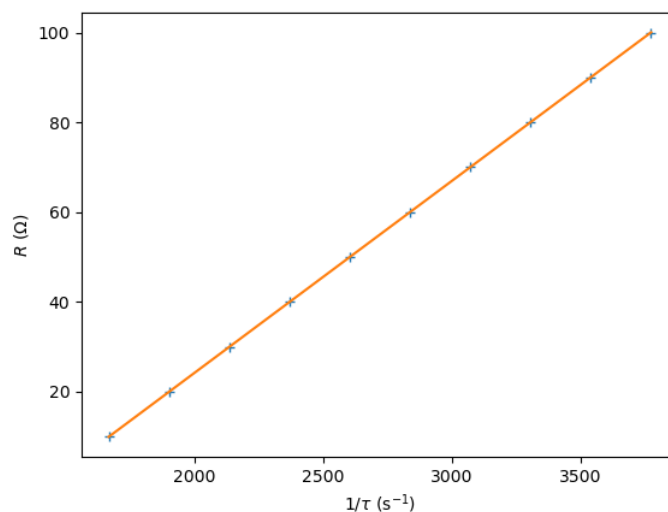
2. La tension qui donne une image de l'intensité est la tension aux bornes de la résistance (Ri). Il faut donc que la résistance est une borne à la masse.



3. La période du signal est $T = \frac{1}{100 \text{ Hz}} = 10 \text{ ms}$.

On choisit donc une période d'échantillonnage $T_e = \frac{T}{10000} = 1 \mu\text{s}$.

```
4. import numpy as np
import matplotlib.pyplot as plt
R=np.arange(10,110,10)
tau=1e-6*np.array([600,526.021,468.278,422.052,383.981,352.795,325.776,302.535,282.847,265.063])
plt.plot(1/tau,R,'+')
a,b=np.polyfit(1/tau,R,1) # régression linéaire
plt.plot(1/tau,a/tau+b) # droite de regression
plt.show()
```

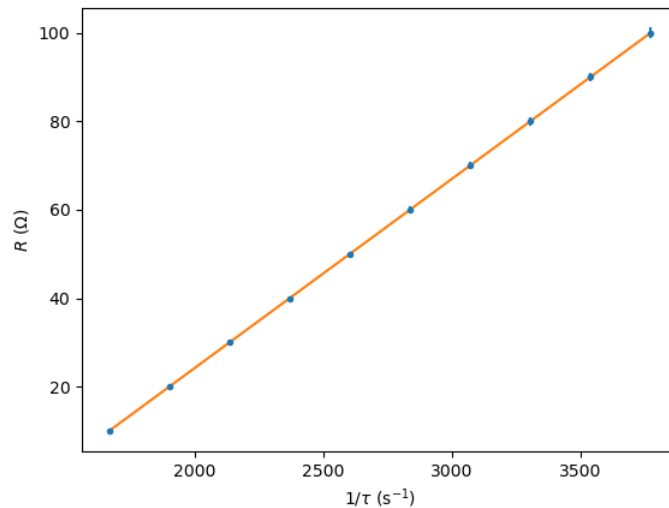


$\tau = \frac{L}{R_s+r+R}$ donc $R = \frac{L}{\tau} - r - R_s$.

On identifie le coefficient directeur $a = L = 42,77 \text{ mH}$ et l'ordonnée à l'origine $b = -r - R_s = -61,32 \Omega$, d'où $r = 11,32 \Omega$

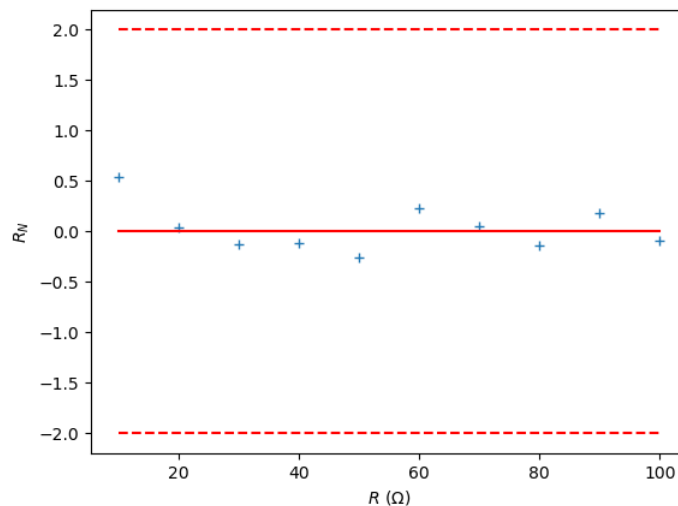
```
5. Delta=1/100*R
uR=Delta/np.sqrt(3)
```

```
6. plt.errorbar(1/tau,R,yerr=2*uR,fmt='b')
```



Les barres d'erreur sont trop petites pour être visibles. Difficile de juger si toutes les barres d'erreur coupent la droite modèle! Il faut tracer le graphe des résidus normalisés.

```
7. RN=(R-a/tau-b)/uR
plt.close()
plt.plot(R,RN,'+')
plt.show()
```



Les résidus normalisés sont tous compris entre -2 et 2, donc toutes les mesures sont compatibles avec le modèle.

```
8. import numpy.random as rd
N=10000 #Nombre de tirages Monte-Carlo
aMC=[]
bMC=[]
for i in range(N):
    RMC=R+rd.uniform(-Delta,Delta)
    p=np.polyfit(1/tau,RMC,1)
    aMC.append(p[0])
    bMC.append(p[1])
ua=np.std(aMC)
ub=np.std(bMC)
```

On obtient $u(a) = 1,8 \times 10^{-4}$ H et $u(b) = 0,41 \Omega$.

$u(L) = u(a) = 0,18$ mH

$r = -b - R_s$ d'où $u(r) = \sqrt{u(b)^2 + u(R_s)^2}$ avec $u(R_s) = \frac{3\%50 \Omega}{\sqrt{3}} = 0,87 \Omega$ (hypothèse d'une distribution uniforme), soit $u(r) = 0,96 \Omega$