

TP20 – Filtrage numérique

Erratum : dans le sujet, il faut lire

- $s1(t, fc)$ à la question 8 ;
- $\underline{H}_p = (\underline{H}_1)^p$ dans la section « Effet de l'ordre du filtre ».

Questions préliminaires

1. Plusieurs observations permettent de justifier la non-linéarité du pont de Graëtz :

- sa réponse $e(t)$ à une entrée $u(t)$ sinusoïdale ne l'est pas ;
- la fréquence f_e du signal de sortie $e(t)$ n'est pas la même que la fréquence f_0 du signal d'entrée $u(t)$.

On parle ici d'**enrichissement spectral** : le signal de sortie contient des composantes qui ne sont pas représentées dans le spectre du signal d'entrée. Par ailleurs les diodes sont des dipôles non-linéaires : le comportement d'un système qui en contient présente très souvent des non-linéarités.

Sur le graphe, on constate que la fréquence f_e du signal $e(t)$ est le double de celle du signal $u(t)$, soit $f_e = 2f_0 = 100$ Hz.

2. En définissant au préalable la variable $fe = 2*f_0$, la fonction $e3(t)$ peut s'écrire :

```
1 def e3(t):
2     output = cn[0]/2
3     for n in range(1, len(cn)):
4         output += cn[n] * np.cos(2*np.pi*n*fe * t + phin[n])
5     return output
```

Graphiquement, on constate que la représentation graphique de $e_3(t)$ se superpose convenablement à celle du signal $e(t)$ (Fig. 1).

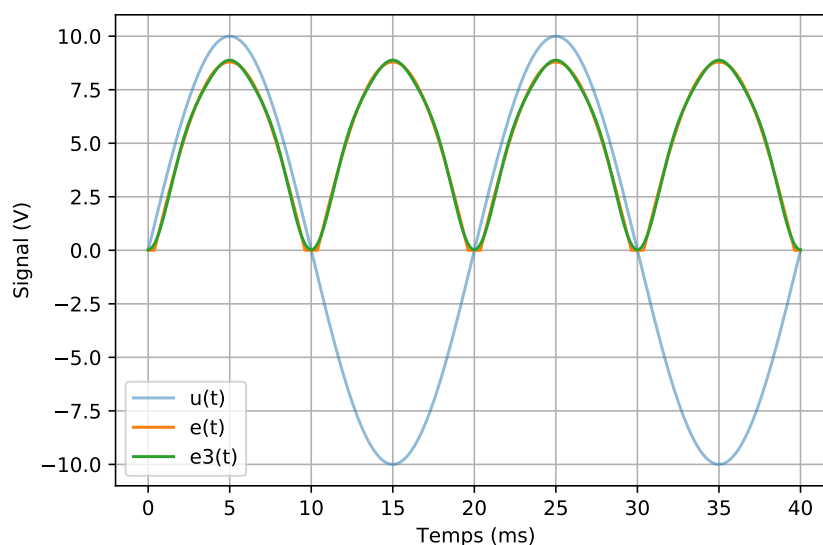
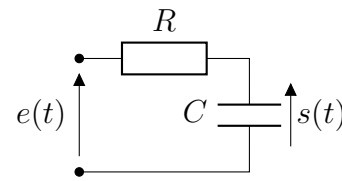


FIGURE 1 – Comparaison entre le signal $e(t)$ approché à l'aide des premières composantes de sa série de Fourier et le signal réel.

3. On souhaite conserver la composante continue associée à la fréquence nulle et atténuer fortement les composantes alternatives de fréquence $f \geq f_e$: il faut utiliser un **filtre passe-bas** de fréquence de coupure $f_c \ll f_e$.



4. Avec $f_c = \frac{1}{2\pi RC}$ on a

$$\underline{H}_1(jf) = \frac{1}{1 + j\frac{f}{f_c}} \quad \text{d'où} \quad G_1(f) = \frac{1}{\sqrt{1 + \left(\frac{f}{f_c}\right)^2}} \quad \text{et} \quad \phi_1(f) = -\arctan\left(\frac{f}{f_c}\right).$$

5. Avec un filtre passe-bas idéal de fréquence de coupure $f_c < f_e$, seule la composante continue est conservée, d'où

$$s(t) = \frac{c_0}{2}.$$

A.N. : $s(t) = 5,2 \text{ V}$.

Filtre passe-bas d'ordre un

6. On peut appliquer le principe de superposition car le filtre utilisé après le pont de Graëtz est **linéaire**.

```
7.
1 def G1(f, fc):
2     return 1 / np.sqrt(1 + (f/fc)**2)
3
4 def phi1(f, fc):
5     return -np.arctan(f/fc)
```

```
8.
1 def s1(t, fc):
2     output = cn[0]/2 * G1(0,fc)
3     for n in range(1, len(cn)):
4         coefn = G1(n*fe,fc) * cn[n]
5         phasen = phin[n] + phi1(n*fe,fc)
6         output += coefn * np.cos(2*np.pi*n*fe * t + phasen)
7     return output
```

L'évolution de l'allure du signal en sortie du filtre en fonction de la fréquence de coupure du filtre (Fig. 2) est cohérente avec le comportement d'un filtre passe-bas : plus la fréquence de coupure du filtre est basse, plus l'amplitude des oscillations du signal de sortie est faible. La sortie se rapproche d'un signal continu : le comportement du filtre tend vers celui d'un **filtre moyennneur**.

Effet de la fréquence de coupure

9. Par définition de la valeur moyenne S_{moy} du signal $s(t)$, on a

$$S_{\text{moy}} = \langle s(t) \rangle = \frac{1}{T} \int_0^T s(t) dt \approx \frac{1}{T} \sum_{k=0}^{N-1} s(k\delta t) \delta t \approx \frac{1}{N} \sum_{k=0}^{N-1} s_k \quad \text{avec} \quad \delta t = \frac{T}{N-1},$$

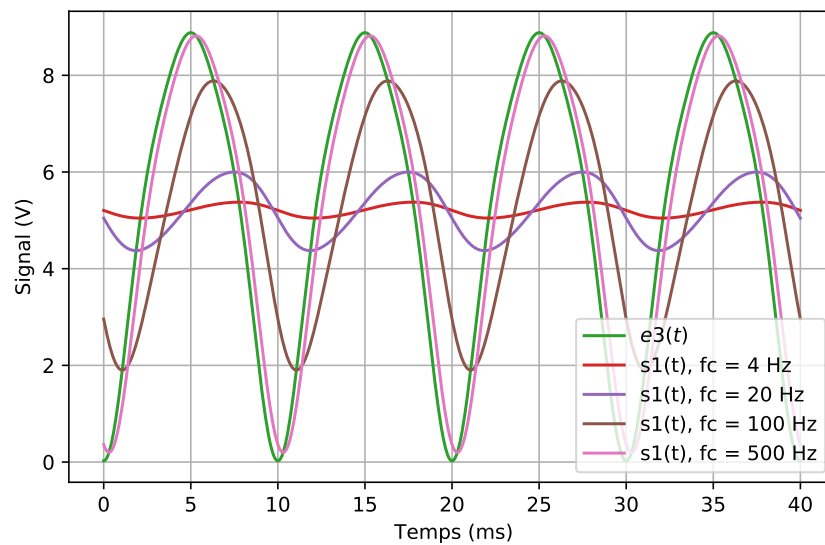


FIGURE 2 – Allure du signal $s(t)$ à la sortie du filtre.

où l'on calcule la valeur approchée de l'intégrale, c'est-à-dire de l'aire sous la courbe de $s(t)$ entre 0 et T , à l'aide de la méthode des rectangles. La dernière approximation est raisonnable tant que l'on dispose d'un nombre $N \gg 1$ d'échantillons $s_k = s(k\delta t)$ de $s(t)$ sur l'intervalle $[0, T]$, ce qui est de toute façon nécessaire pour approcher convenablement l'aire sous la courbe par la méthode des rectangles.

10.

```
def Omega1(fc):
    S = s1(np.linspace(0, 1/fe, 10000), fc)
    Seff = np.sqrt(np.mean(S**2))
    Smoy = np.mean(S)
    return Seff/Smoy - 1
```

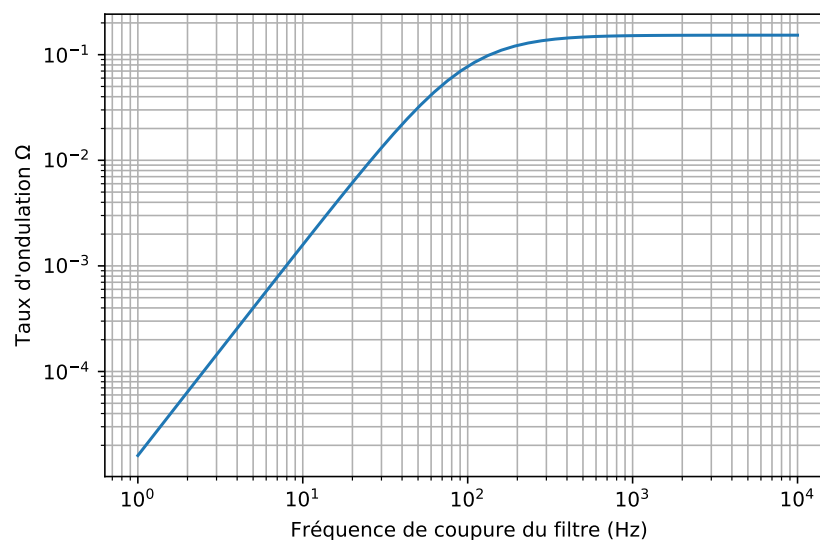


FIGURE 3 – Effet de la fréquence de coupure du filtre passe-bas d'ordre un sur le taux d'ondulation du signal de sortie $s(t)$.

On représente graphiquement l'évolution du taux d'ondulation Ω en fonction de la fréquence de coupure f_c du filtre passe-bas (Fig. 3). Dans la limite où $f_c \gg f_e$, les composantes de $e(t)$ ne sont pas atténuées et on a $s(t) = e(t)$: la valeur limite du taux d'ondulation est celle de $e(t)$. Lorsque la fréquence de coupure est faible devant la fréquence de $e(t)$, le taux d'ondulation diminue et tend vers 0. Les composantes sinusoïdales du spectre de $e(t)$ sont fortement atténuées et la sortie se rapproche d'un signal continu.

11. Le **temps caractéristique τ du régime transitoire** associé au circuit est donné par $\tau = 1/f_c$. En choisissant une fréquence de coupure trop faible, le temps caractéristique peut être très long et ainsi, il faudra attendre que le signal de sortie atteigne la valeur $c_0/2$ en régime permanent. C'est envisageable si $\tau \sim 1$ s, soit $f_c \sim 1$ Hz, ce qui impose une limite basse sur f_c .

Effet de l'ordre du filtre

12.

```

1 def Gp(f, fc, p):
2     return 1 / np.sqrt(1 + (f/fc)**2)**p
3
4 def phip(f, fc, p):
5     return -np.arctan(f/fc) * p
6
7 def sp(t, fc, p):
8     output = cn[0]/2 * Gp(0,fc,p)
9     for n in range(1,len(cn)):
10        coefn = Gp(n*fe,fc,p) * cn[n]
11        phasen = phin[n] + phip(n*fe,fc,p)
12        output += coefn * np.cos(2*np.pi*n*fe * t + phasen)
13    return output

```

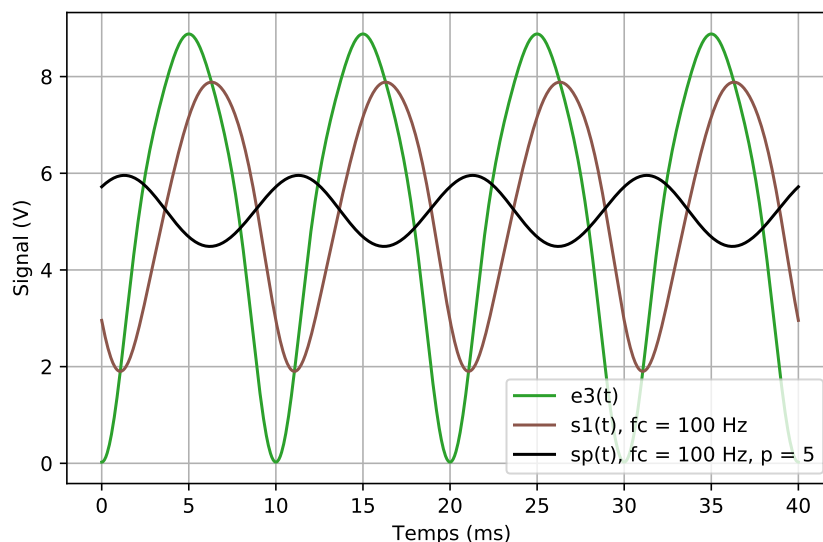


FIGURE 4 – Allure du signal à la sortie d'un filtre d'ordre $p = 5$. L'atténuation du fondamental et des harmoniques est plus importante qu'avec un filtre d'ordre un.

13. Pour mettre en évidence l'effet de l'ordre p du filtre sur l'ondulation du signal de sortie, on peut représenter le taux d'ondulation Ω pour quelques valeurs de p (Fig. 5). On constate bien que, pour f_c fixé, l'atténuation des composantes oscillantes augmente avec l'ordre du filtre. Un filtre d'ordre élevé permet d'obtenir le signal continu souhaité tout en conservant un temps caractéristique raisonnable.

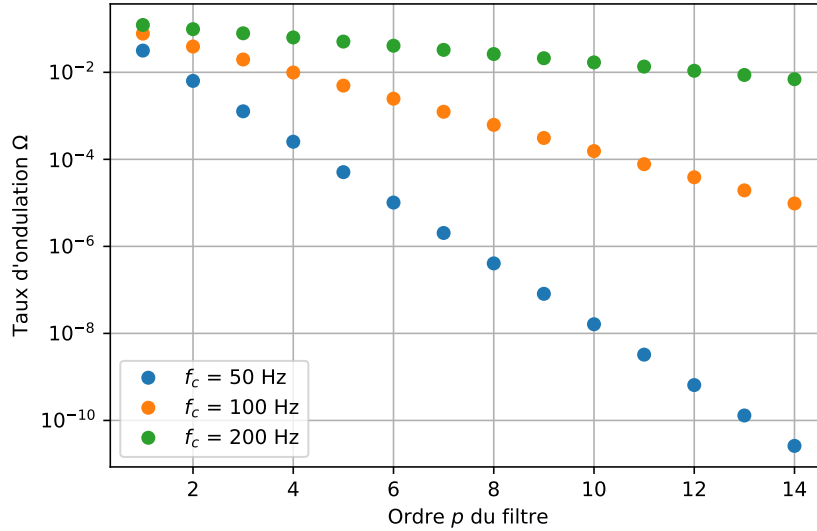


FIGURE 5 – Évolution du taux de modulation du signal de sortie en fonction de l'ordre du filtre utilisé.

14. On trace les diagrammes de Bode en amplitude de quelques filtres (Fig. 6).

La pente de l'asymptote en haute fréquence est d'autant plus raide que l'ordre p est élevé, ce qui confirme ce qui a été vu précédemment. En haute fréquence, on a :

$$\underline{H}_p(jf) = \left(\frac{1}{1 + j\frac{f}{f_c}} \right)^p \underset{\text{HF}}{\sim} \left(\frac{1}{j\frac{f}{f_c}} \right)^p \quad \text{d'où} \quad G(f) \underset{\text{HF}}{\sim} \left(\frac{f}{f_c} \right)^{-p}.$$

Finalement,

$$G_{\text{dB}}(f) \underset{\text{HF}}{\sim} -20p \times \log \left(\frac{f}{f_c} \right),$$

ce qui correspond à une pente de $-20p$ dB/décade. C'est bien ce que l'on constate sur les diagrammes de Bode.

Un filtre d'ordre élevé peut donc être très sélectif : les composantes dont la fréquence est comprise dans la bande coupée sont très rapidement atténuées quand on s'éloigne de la fréquence de coupure du filtre.

Rq : La fréquence de coupure $f_{c,p}$ associée à la mise en cascade de p filtres d'ordre un de fréquence de coupure f_c est plus faible que f_c . On montre que $f_{c,p} = \sqrt[2^{1/p}]{f_c}$.

Lien avec la représentation temporelle

15. En utilisant l'expression de la fonction de transfert \underline{H}_1 en fonction de ω , on déduit pour un signal d'entrée harmonique $e_h(t)$:

$$\left(1 + j\frac{\omega}{\omega_c} \right) \underline{s}_h = \underline{e}_h \quad \text{soit} \quad \left(1 + \frac{1}{\omega_c} \frac{d}{dt} \right) s_h = e_h.$$

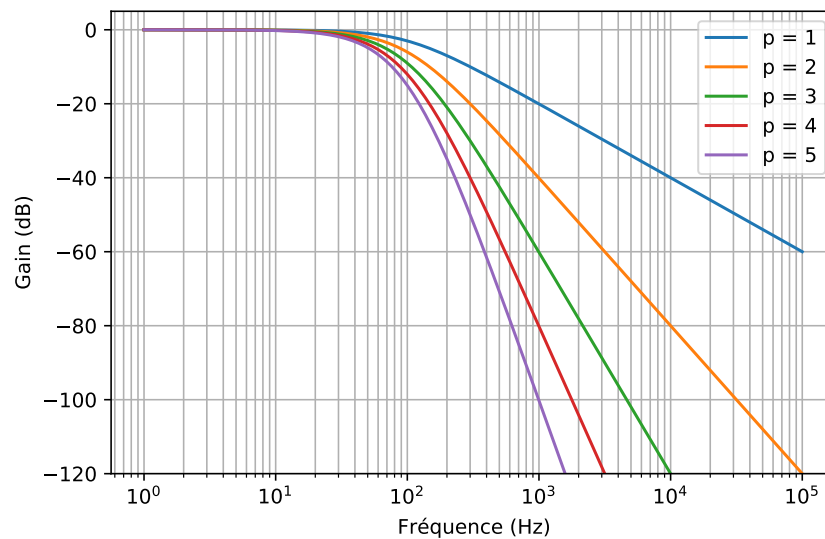


FIGURE 6 – Diagramme de Bode en amplitude de quelques filtres, obtenus en mettant en cascade p filtres d'ordre 1 de fréquence de coupure f_c .

Le système est linéaire, donc pour une entrée $e(t)$ quelconque, on a :

$$\frac{ds}{dt}(t) + \omega_c s(t) = \omega_c e(t).$$

16. On peut intégrer numériquement cette équation différentielle à l'aide de la fonction `odeint` ou avec la méthode d'Euler explicite. L'avantage d'intégrer numériquement l'équation différentielle est que l'on peut utiliser comme signal d'entrée le signal réel $e(t)$ et pas sa décomposition $e_3(t)$ en série de Fourier.¹

```

1 def s1odeint(t, fc):
2     def f(V,t):
3         return (e(t) - V[0]) * 2*np.pi*fc
4     return odeint(f, [e(0)], t)[: ,0]
5
6 def s1euler(t, fc):
7     dt      = t[1] - t[0]
8     omc     = 2*np.pi*fc
9     output  = np.zeros(len(t))
10    output[0] = e(0)
11    for k in range(len(t)-1):
12        output[k+1] = output[k] + dt * omc * (e(t[k]) - output[k])
13    return output

```

Les représentations graphiques (Fig. 7) des différentes solutions se superposent parfaitement, à l'exception et comme attendu, d'un court régime transitoire pendant les premiers instants. On vérifie que la durée du régime transitoire est de l'ordre de $1/f_c$.

1. La bibliothèque `scipy.signal` offre cependant des fonctions performantes adaptées au filtrage numérique de signaux.

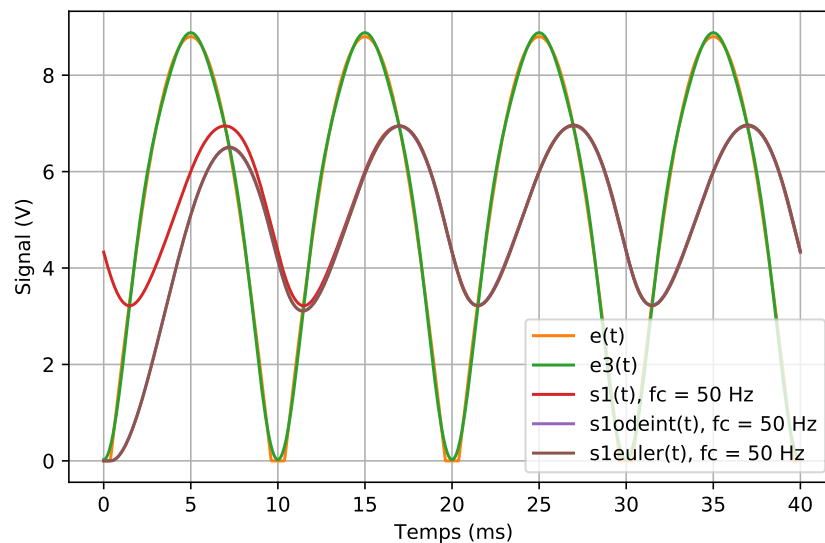


FIGURE 7 – Comparaison entre les solutions obtenues par résolution numérique de l'équation différentielle et celle calculée en exprimant l'action du filtre sur chacune des composantes de la décomposition en série de Fourier.

Annexes

Annexe 1 – Entête du programme tp20-filtrage_numerique.py

```

1 import numpy as np
2 import matplotlib.pyplot as plt
3 from scipy.integrate import odeint
4
5 # CONSTANTES
6 U0 = 10 # amplitude du signal u(t) en volts
7 f0 = 50 # fréquence du signal u(t) en hertz
8 Vs = 0.6 # tension de seuil des diodes en volts
9 fe = 2*f0 # fréquence du signal e(t) en hertz
10
11 cn = [10.4242, 4.1528, 0.7588, 0.2759] # coefficients cn en volts
12 phin = [0, np.pi, np.pi, np.pi] # coefficients phin en radians

```

Annexe 2 – Représentations graphiques

```

1 plt.figure(1) # Représentations temporelles des signaux
2 plt.clf()
3 fc = 100
4 t = np.linspace(0, 2/f0, 1000)
5 plt.plot(t, u(t), label="$u(t)$", alpha=.5)
6 plt.plot(t, e(t), label="$e(t)$")
7 plt.plot(t, e3(t), label="$e3(t)$")
8 plt.plot(t, s1(t, fc), label="$s1(t)$")
9 plt.plot(t, s1odeint(t, fc), label="$s1odeint(t)$")

```

```

10 plt.plot(t, sieuler(t, fc), label="$s_{ieuler}(t)$")
11 plt.plot(t, sp(t, fc, 5), label="$s_p(t)$")
12 plt.title("Évolution temporelle des différents signaux")
13 plt.xlabel("Temps (s)")
14 plt.ylabel("Signal (V)")
15 plt.legend()
16 plt.grid(which="both")
17
18 plt.figure(2) # Effet de la fréquence de coupure
19 plt.clf()
20 fcs = np.logspace(0,4)
21 plt.loglog(fcs, [Omega1(fc) for fc in fcs])
22 plt.title("Taux d'ondulation à la sortie d'un PB d'ordre 1")
23 plt.xlabel("Fréquence de coupure du filtre (Hz)")
24 plt.ylabel("Taux d'ondulation $\Omega$")
25 plt.grid(which="both")
26
27 plt.figure(3) # Effet de l'ordre du filtre
28 plt.clf()
29 n = range(1,20)
30 ond = [Omega1(100, p) for p in n]
31 plt.semilogy(n, ond, "o", label="$f_c = {:.1e}$ Hz".format(100))
32 plt.title("Taux d'ondulation à la sortie d'un PB d'ordre p")
33 plt.xlabel("Ordre $p$ du filtre")
34 plt.ylabel("Taux d'ondulation $\Omega$")
35 plt.grid(which="both")
36 plt.legend()
37
38 plt.figure(4) # Diagramme de Bode
39 plt.clf()
40 fc = 100
41 f = np.logspace(0,5,1000)
42 for p in range(1,6):
43     GdB = 20 * np.log10(Gp(f, fc, p))
44     plt.semilogx(f, GdB, label="p = {:.0f}".format(p))
45 plt.ylim(-120,5)
46 plt.title("Diagramme de Bode en amplitude, $f_c = {:.0f}$ Hz".format(100))
47 plt.xlabel("Fréquence (Hz)")
48 plt.ylabel("Gain (dB)")
49 plt.grid(which="both")
50 plt.legend()
51
52 plt.show()

```