

Thème I. Ondes et signaux (Électricité) TP n°14 Filtrage d'un son – Corrigé

Vendredi 14 février 2025

I Filtrage analogique

👉 🎵 Copier le fichier son TP_filtrage_son.wav disponible dans la zone Partage → physique → 2024-2025 → TP14 filtrage, dans votre zone personnelle.

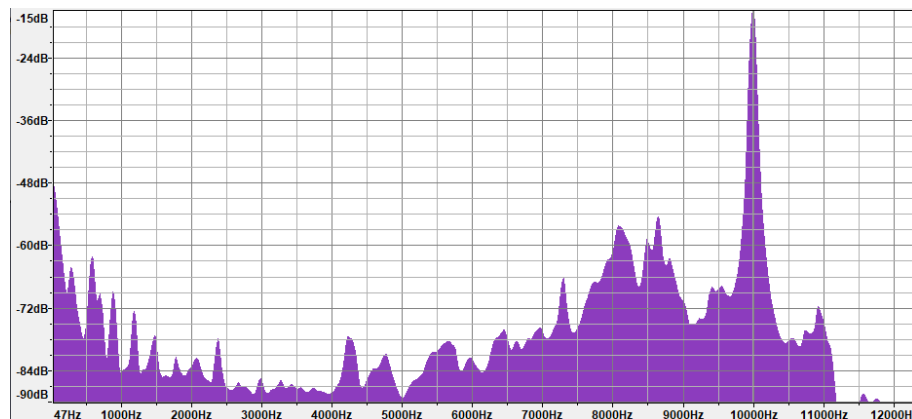
Q1. 🎵 L'écouter, que constatez-vous ?

Solution: On perçoit un son très aigu qui couvre la musique.

Pour rendre le son audible, on souhaite réaliser un filtrage du signal.

Q2. Proposer un cahier des charges adapté.

Solution: Il faut identifier ce qu'on veut garder (comment on veut le garder) et ce qu'on veut atténuer (de combien on veut l'atténuer). On effectue le spectre (avec Audacity) du son :

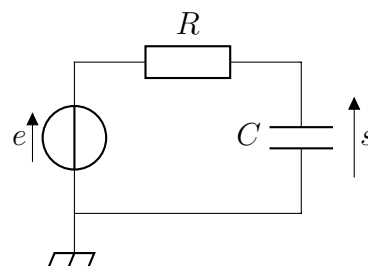


- la musique est de fréquence entre 200 Hz et 800 Hz (environ),
- le bruit désagréable très aigu, est de fréquence 10000 Hz

Je souhaite atténuer le bruit d'un facteur 100, tout en n'atténuant pas trop la musique.

Q3. Proposer un montage répondant au cahier des charges.

Solution: On peut proposer (dans un premier temps, et pour faire simple) un filtre passe-bas du premier ordre, RC aux bornes du condensateur.



Ce filtre est de pulsation de coupure $\omega_c = \frac{1}{RC}$, soit $f_c = \frac{1}{2\pi RC}$.

Pour atténuer d'un facteur 100 le bruit à 10 kHz il doit être situé à 2 décades au-delà de la coupure. Ainsi, il faut que la coupure soit à $f_c = 100$ Hz, soit $RC \approx 1,7 \cdot 10^{-3}$ s

- 1^{re} proposition $C = 100 \text{ nF}$ et $R = 22 \text{ k}\Omega$, soit $f_c = 72 \text{ Hz}$ (vraiment faible, le son utile sera fortement atténué).
- 2^e proposition $C = 100 \text{ nF}$ et $R = 10 \text{ k}\Omega$, soit $f_c = 159 \text{ Hz}$: préférable.
- On pourrait envisager de choisir $C = 100 \text{ nF}$ et $R = 4,7 \text{ k}\Omega$, soit $f_c = 340 \text{ Hz}$, qui pourrait être un bon compromis.

👉 Câbler le montage.

👉 Le signal d'entrée (« semblable » au magnifique morceau de musique bruité précédent) sera généré par la carte d'acquisition.

👉 Récupérer, dans la zone **Partage**, le fichier `signal_entree` contenant le signal d'entrée.

👉 L'ouvrir avec Orphylab.

👉 La carte d'acquisition est connectée par port USB sur l'ordinateur.

👉 La sortie SA est connectée au filtre / à l'oscilloscope.

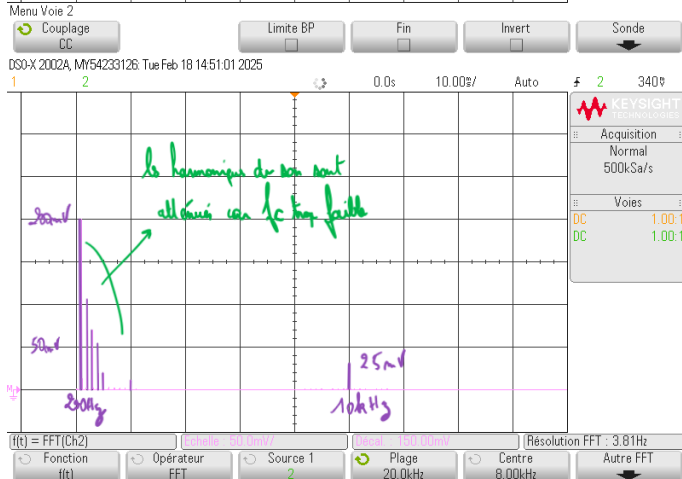
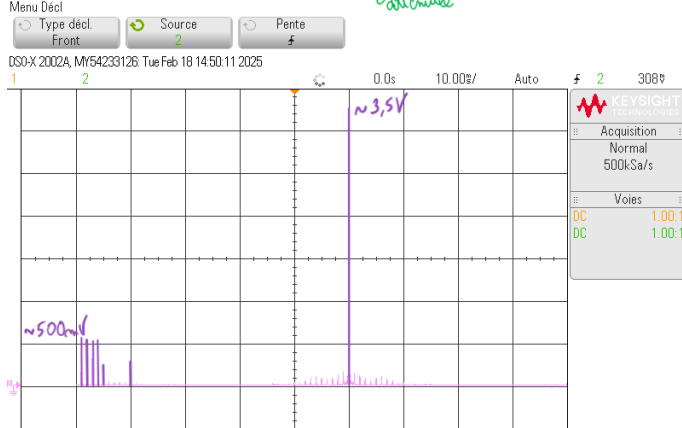
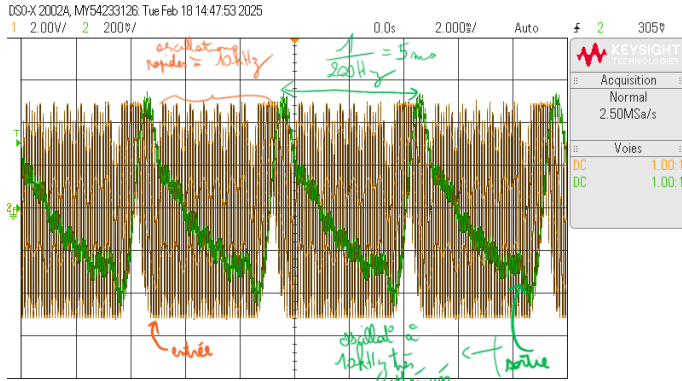
Q4. Étudier quantitativement la qualité du filtrage réalisé par le filtre.

Solution: En comparant le signal d'entrée et le signal de sortie, on observe que les variations rapides à 10 kHz ont été fortement atténuées. On voit après filtrage l'allure du signal utile.

On peut alors jouer sur RC pour voir l'influence du filtrage, et l'améliorer si besoin.

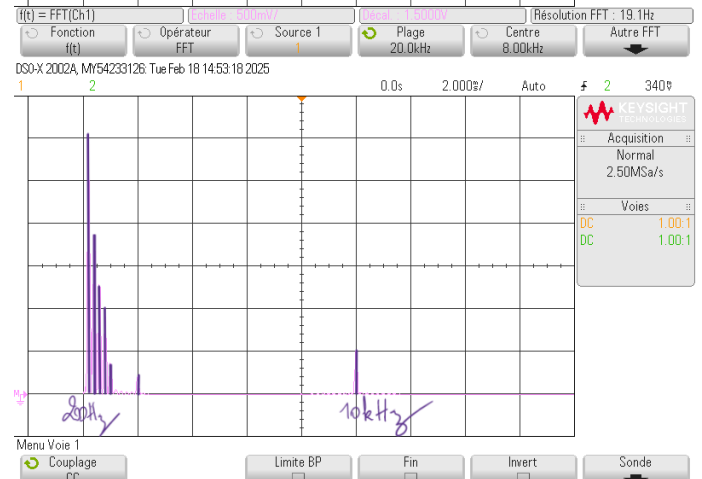
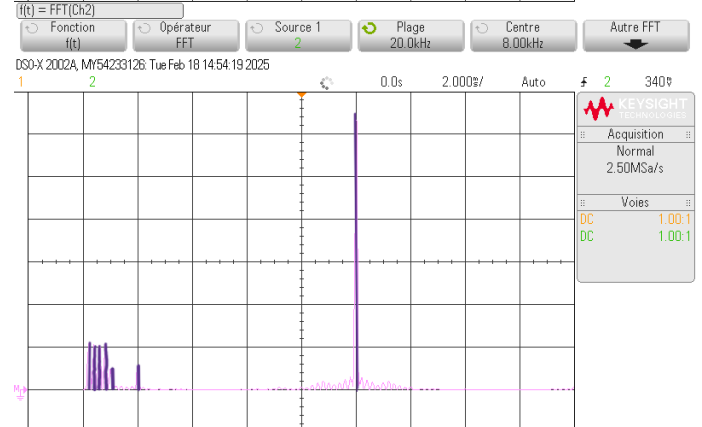
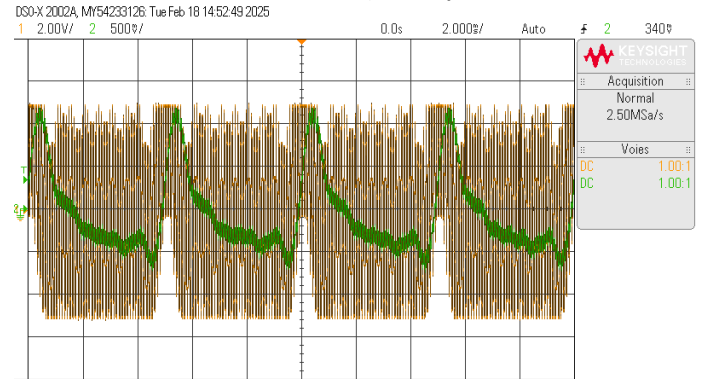
L'observation du spectre (FFT) permet de mesurer quantitativement l'effet du filtrage sur les harmoniques fréquences du son, et sur le bruit à 10 kHz.

$C = 100 \text{ nF}$ et $R = 22 \text{ k}\Omega$, soit $f_c = 72 \text{ Hz}$



Le bruit à 10 kHz est fortement atténué : c'est ce qu'on voulait, MAIS on constate que le son, qui n'est pas à basse fréquence de la coupure, est également atténué et ceux d'autant plus pour les harmoniques de rangs élevés : le son sera donc déformé !

$C = 100 \text{ nF}$ et $R = 10 \text{ k}\Omega$, soit $f_c = 159 \text{ Hz}$



Le bruit à 10 kHz est fortement atténué : c'est ce qu'on voulait, MAIS on constate que le son, qui n'est pas à basse fréquence de la coupure, est également atténué et ceux d'autant plus pour les harmoniques de rangs élevés : le son sera donc déformé, mais l'effet est moins important que dans le cas à gauche.

II Filtrage numérique

On souhaite maintenant réaliser un filtrage numérique à l'aide de python.

On suppose que le signal d'entrée est représenté par une liste et on note f_e la fréquence d'échantillonnage, avec $f_e = 1/T_e$, où T_e est le temps entre deux éléments de la liste.

Q5. À partir de la fonction de transfert du filtre étudié précédemment, obtenir l'équation différentielle reliant les signaux d'entrée e et de sortie s .

Solution: La fonction de transfert du montage précédent est :

$$\underline{H} = \frac{\underline{s}}{\underline{e}} = \frac{1}{1 + j\frac{\omega}{\omega_c}}$$

$$\underline{s} + \frac{j\omega}{\omega_c} \underline{s} = \underline{e}$$

$$\underline{s} + \frac{1}{\omega_c} \frac{ds}{dt} = \underline{e}$$

$$\frac{ds}{dt} + \omega_c s = \omega_c e$$

Q6. À partir de l'équation différentielle reliant les signaux d'entrée et de sortie, montrer que, pour un filtre de fréquence de coupure f_c , la liste représentant le signal de sortie **si** vérifie la relation de récurrence :

$$s_{i+1} = s_i + \frac{2\pi f_c}{f_e}(e_i - s_i)$$

Solution: Avec l'algorithme d'Euler :

$$\frac{s(t_{i+1}) - s(t_i)}{t_{i+1} - t_i} + \omega_c s(t_i) = \omega_c e(t_i)$$

$$s_{i+1} = s_i + \omega_c(e_i - s_i) \times T_e$$

$$s_{i+1} = s_i + \omega_c(e_i - s_i) \times \frac{1}{f_e}$$

$$s_{i+1} = s_i + \frac{2\pi f_c}{f_e}(e_i - s_i)$$

📁 Récupérer dans la zone partage le fichier `filtrage_numerique.py`, le copier dans votre zone personnelle. L'ouvrir avec Spyder.

Q7. 💻 Compléter le fichier.

Solution:

```
1 import scipy.io.wavfile as sci_wav
2 import numpy as np
3 fe, signal = sci_wav.read("TP_filtrage_son.wav") # fe : fréquence d'
    échantillonnage
4 entree = signal[:,0] # tension colonne 0 du tableau signal
5 def convert(entree):
6     sortie = np.zeros(entree.shape)
7     sortie[0] = 0.0
8     for i in range(1, len(entree)):
9         sortie[i] = (entree[i-1] - sortie[i-1]) * 2 * np.pi * 10 / fe + sortie[i-1]
10    return sortie.astype('int16')
11 sortie = convert(entree)
12 sci_wav.write("son_filtre.wav", fe, sortie) # créer le fichier son de
    sortie
```

Q8. Réaliser le filtrage pour plusieurs fréquences de coupures.

Q9. 🎵 Écouter le son issu de ce filtrage.