



BLAISE PASCAL
PT 2022-2023

TP 3 – Séquence 1 : Électronique

Filtrage d'un extrait musical

Techniques et méthodes

- ▷ Mise en œuvre d'un filtrage analogique ;
- ▷ Mise en cascade de blocs fonctionnels, adaptation d'impédance entre blocs ;
- ▷ Mise en œuvre d'un filtrage numérique ;
- ▷ Visualiser l'effet d'un filtre sur un signal réel.

Matériel sur votre paillasse :

- ▷ Un PC portable avec Python ;
- ▷ Un casque audio ;
- ▷ Adaptateurs JACK/fiches banane ;
- ▷ Un ALI et son alimentation +15/-15 V ;
- ▷ Une résistance variable (boîte à décade) ;
- ▷ Condensateur 100 nF ;
- ▷ Oscilloscope ;
- ▷ Plaquette de branchement.

Ce TP a pour objectif de rendre plus concrètes les opérations de filtrage en les appliquant à des signaux « réels », en l'occurrence deux extraits sonores à récupérer sur l'ENT. Le premier provient du prélude de la *Deuxième Suite pour Violoncelle* de Jean-Sébastien Bach, composée autour de 1720, « détériorée » par un bruit parasite de fréquence 10 kHz. Le second est un extrait de *Dolce Vita*, single de Ryan Paris datant de 1983 et maintes fois repris dans des films ou des spots publicitaires. Le fichier son sera joué à l'ordinateur, filtré, puis écouté dans le casque.

I - Filtrage analogique

I.A - Dimensionnement du filtre



Dimensionnons le filtre pour éliminer le bruit parasite de la suite de Bach. On souhaite qu'il soit atténué d'un facteur 100 en sortie du filtre, et pour des questions de simplicité de mise en œuvre, on s'impose de travailler avec un filtre passe-bas du premier ordre.

✍ Déterminer sans calcul excessif la fréquence de coupure à donner au filtre. En déduire la valeur de résistance à choisir.

Une atténuation d'un facteur 100 correspond à un gain de -40 dB. Comme le gain d'un filtre passe-bas du premier ordre décroît de 20 dB/décade, on en déduit que la fréquence de coupure du filtre doit être deux décades avant celle du bruit, soit

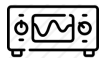
$$f_{br} = 10^2 f_c \quad \text{d'où} \quad f_c = 100 \text{ Hz}.$$

La pulsation de coupure d'un filtre RC vaut $\omega_c = 1/RC$, d'où $f_c = 1/2\pi RC$, et ainsi

$$R = \frac{1}{2\pi C f_c} = 16 \text{ k}\Omega.$$

Espace 1

I.B - Premier montage, nécessité d'une adaptation d'impédance



Câbler le montage, en branchant dans un premier temps le casque directement sur la sortie du filtre, comme schématisé figure 1.

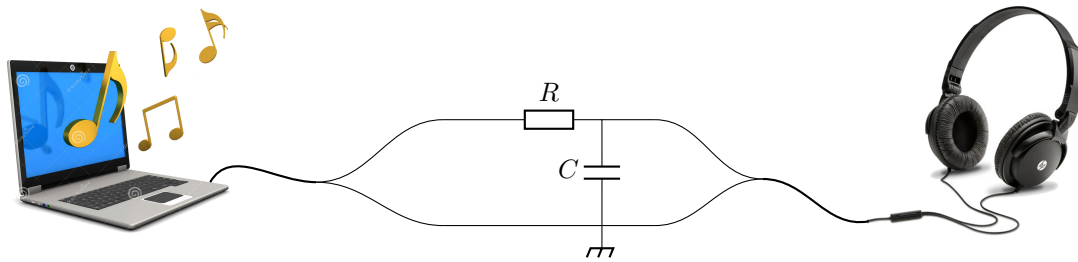


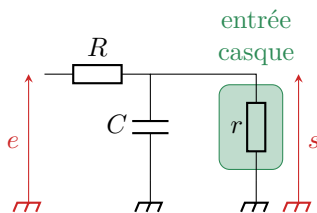
Figure 1 – Schéma du premier montage.

- Comparer l'intensité sonore des signaux entendus directement en sortie du PC et en sortie du filtre. on n'entend presque rien dans le casque! ... aucune fréquence ne semble correctement transmise

Espace 2



Interprétons ce phénomène inattendu, en se rappelant que tout composant électronique peut être décrit par son impédance d'entrée. Dans le cas d'un casque audio aux fréquences intermédiaires, elle vaut environ 60Ω . Cela signifie que tout se passe comme si une résistance $r = 60 \Omega$ était branchée en parallèle de la sortie du filtre.



- Établir la fonction de transfert du filtre modifié par l'entrée du casque et l'écrire sous la forme

$$\underline{H} = \frac{H_0}{1 + j\omega/\omega_c}$$

Après des calculs classiques, on trouve

$$\underline{H} = \frac{\frac{r}{R+r}}{1 + j\frac{rRC}{r+R}\omega}$$

Espace 3

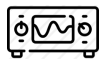
- Avec les valeurs utilisées, la fréquence de coupure modifiée est de l'ordre de 15 kHz. Interpréter alors l'observation initiale.

Quasiment tout le spectre audible correspond au domaine basse fréquence du filtre, où le signal est atténué d'un facteur $r/(R+r) \simeq 1/150$, ce qui explique qu'on n'entende ... presque rien!

Espace 4

I.C - Montage amélioré par ajout d'un bloc suiveur à ALI

Pour résoudre le problème, il est nécessaire d'inclure un bloc spécifique entre la sortie du filtre et l'entrée du casque : un bloc suiveur à ALI, qui possède une impédance d'entrée infinie ... et permet donc de retrouver exactement les mêmes signaux qu'avec le filtre RC utilisé en sortie ouverte.



Câbler le montage comme indiqué figure 2, sans oublier qu'un ALI doit être alimenté par une alimentation continue +15/-15 V.

Attention ! Ne pas se tromper sur le sens des branchements sous peine de griller l'ALI!

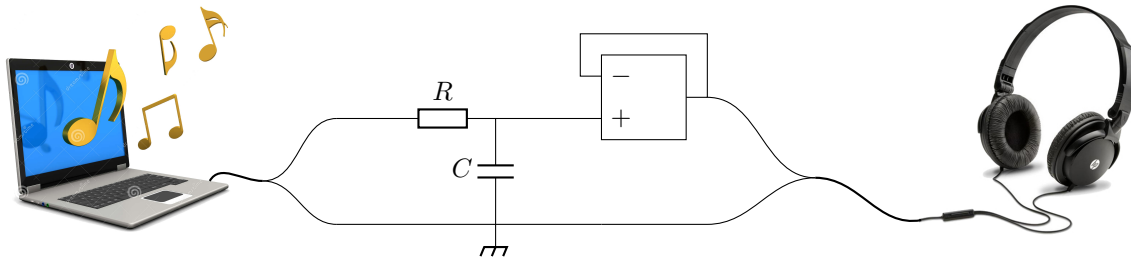


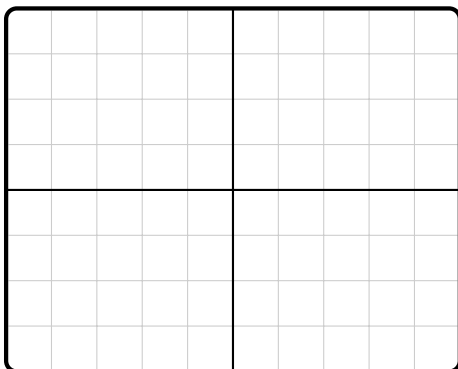
Figure 2 – Schéma du montage complet, avec suiveur à ALI.

🔗 Comparaison « acoustique » des sons filtré et non-filtré :

Le filtre est un passe-bas, qui élimine les harmoniques haute fréquence. En termes acoustiques, elles correspondent aux sons aigus : le bruit parasite de la *Suite* de Bach s'entend nettement moins, et on a l'impression de ne conserver quasiment que les basses de *Dolce Vita*.

Espace 5

🔗 Comparaison « électronique » des signaux filtré et non filtré : observer les tensions d'entrée et de sortie du montage à l'oscilloscope, et reproduire l'allure des signaux sur un extrait bien choisi.



Les fluctuations rapides du signal d'entrée, associées aux hautes fréquences du spectre, sont lissées par le filtre, ce qui se voit très bien sur l'oscilloscope.

Espace 6

II - Filtrage numérique

Plus aucun filtrage de signaux sonores ne se fait à l'heure actuelle avec un filtre RC et un montage suiveur ! De nombreux logiciels professionnels ou amateurs permettent de faire ce filtrage de manière numérique. C'est l'objet de la deuxième partie de ce TP.

II.A - Algorithme de filtrage



Le but est de réaliser l'analogie numérique du filtre de la partie précédente, c'est-à-dire un passe-bas du premier ordre de fréquence de coupure $f_c = 100$ Hz, décrit par la fonction de transfert

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

avec la pulsation de coupure $\omega_c = 2\pi f_c$.

Pour construire numériquement le signal de sortie, on pourrait imaginer calculer le spectre du signal numérisé par transformation de Fourier, le multiplier par la fonction de transfert, puis réaliser une transformation de Fourier inverse pour obtenir le signal dans le domaine temporel. Cependant, cette méthode devient rapidement trop coûteuse en temps de calcul. Il est donc préférable de travailler directement dans le domaine temporel.

✎ Relation différentielle entre s et e :

$$\left(1 + \frac{j\omega}{\omega_c}\right) \underline{s} = \underline{e} \quad \text{d'où} \quad s + \frac{1}{\omega_c} \frac{ds}{dt} = e \quad \text{soit} \quad \boxed{\frac{ds}{dt} = -\omega_c s + \omega_c e.}$$

Espace 7

Pour transformer cette relation différentielle en une équation sur les échantillons, il faut approximer la dérivée. Le plus simple est d'utiliser le schéma d'Euler, en prenant la période d'échantillonnage comme pas de temps. Toutefois, pour des questions de précision des calculs, il est nécessaire d'utiliser le schéma d'Euler *implicite* au lieu du schéma *explicite* : la dérivée est toujours approximée par un taux de variation entre les instants t_n et t_{n+1} , mais toutes les autres grandeurs sont à considérer au pas de temps $n + 1$, au lieu du pas de temps n dans le schéma explicite.

✎ Traduction du schéma d'Euler en relation de récurrence : on posera $a = \omega_c T_e$

$$\frac{s_{n+1} - s_n}{T_e} = -\omega_c s_{n+1} + \omega_c e_{n+1}$$

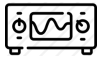
$$s_{n+1} - s_n = -a s_{n+1} + a e_{n+1}$$

$$(1 + a) s_{n+1} = s_n + a e_{n+1}$$

$$\boxed{s_{n+1} = \frac{1}{1 + a} s_n + \frac{a}{1 + a} e_{n+1}}$$

Espace 8

II.B - Mise en œuvre



Ouvrir le script Python associé au TP (à récupérer sur l'ENT), et compléter la fonction `filtrage_PB1`. Cette fonction prend comme argument un tableau `numpy` `e` contenant le signal d'entrée sous forme d'entiers codés sur 16 bits, et un flottant `fc` correspondant à la fréquence de coupure exprimée en hertz. Elle renvoie un tableau `numpy` `s`, contenant le signal filtré, là encore converti en entiers codés sur 16 bits.

Remarque 1 : La procédure appliquée à la Suite pour violoncelle renvoie un avertissement, car certaines méta-données du fichier WAV ne sont pas correctement comprises par les fonctions utilisées pour l'importation sous Python : c'est sans importance pour ce TP, il est donc inutile d'en tenir compte.

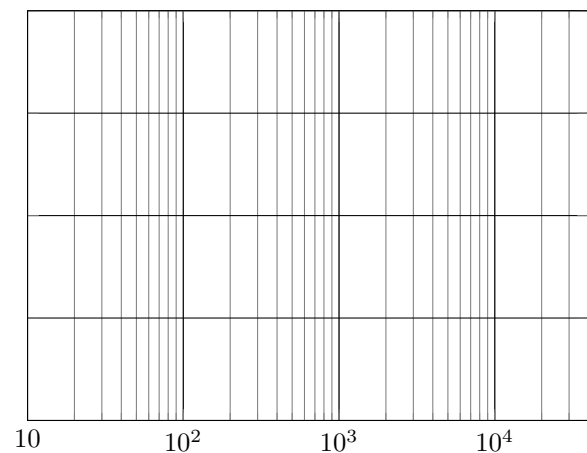
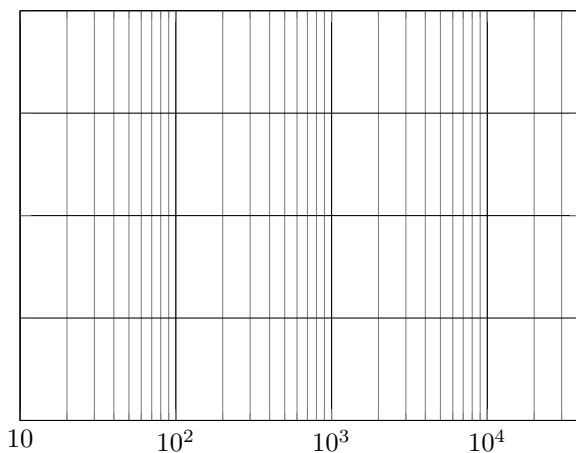
Remarque 2 : L'utilisation de boucles rend les calculs assez longs. Une règle générale en Python pour améliorer l'efficacité des calculs est de privilégier systématiquement les fonctions issues de bibliothèques plutôt que de coder des boucles à la main. Dans le cas présent, la boucle `for` intervenant dans la fonction `filtrage_PB1` peut avantageusement être remplacée par

```
1 | s = signal.lfilter([a/(1+a)], [1, -1/(1+a)], e)
```

Cela demande au préalable d'avoir importé le module `signal` de la bibliothèque `scipy`. Il faut alors ajouter au préambule « `from scipy import signal` ».

Outre la vitesse d'exécution, la gestion des arrondis numériques n'est pas la même, si bien que le signal de sortie obtenu peut différer sensiblement de celui produit par la fonction codée à la main dans certains cas où ces arrondis jouent un rôle important.

🔗 Avec le logiciel Audacity, ouvrir l'extrait sonore original et l'extrait filtré. Afficher le spectre de ces deux signaux (onglet « Analyse » puis « Tracer le spectre »). Comparer l'allure des spectres et les reproduire qualitativement ci-dessous.



II.C - Diagramme de Bode, effet de l'échantillonnage



Tout comme l'échantillonnage a un impact sur le spectre d'un signal (réplication spectrale), il est naturel qu'un filtre numérique ne puisse pas avoir le même diagramme de Bode que son équivalent analogique. Il existe des outils permettant d'y accéder directement, mais pour rester dans le cadre du programme de PT on décide plutôt de le construire « point par point » avec la fonction `bode` ci-dessous.

```
1 | def bode(filtrage, fc, fmin, fmax):
2 |     freq = np.linspace(fmin, fmax, num=1000)
3 |     H = np.empty_like(freq)
4 |     for i in range(len(freq)):
5 |         f = freq[i]
6 |         t = np.arange(start=0, stop=1000/f, step=1/fe)
7 |         phi = 2*np.pi*np.random.random()
8 |         e = (2**15 - 1) * np.cos(2*np.pi * f * t + phi)
9 |         e = e.astype('int16')
10 |        s = filtrage(e, fc)
11 |        H[i] = np.max(s)/np.max(e)
12 |    return freq, H
```

✍ Expliquer le fonctionnement de la fonction `bode`.

On génère d'abord un jeu de 1000 fréquences réparties entre `fmin` et `fmax`.

Pour chacune de ces fréquences, on génère un signal sinusoïdal `e` dont la phase est tirée aléatoirement, destiné à être converti sous forme d'entiers codés sur 16 bits, ce qui explique le terme d'amplitude inhabituel.

Enfin, on lui applique la fonction de filtrage pour obtenir le signal de sortie `s`.

En assimilant l'amplitude du signal à sa valeur maximale, on calcule pour chacune des fréquences l'équivalent du module de la fonction de transfert, ce qui donne naturellement accès au diagramme de Bode.

Espace 9

✍ Le digramme de Bode obtenu pour $f_c = 100$ Hz est représenté ci-dessous, à gauche avec un axe des fréquences logarithmique, à droite avec un axe en échelle linéaire.

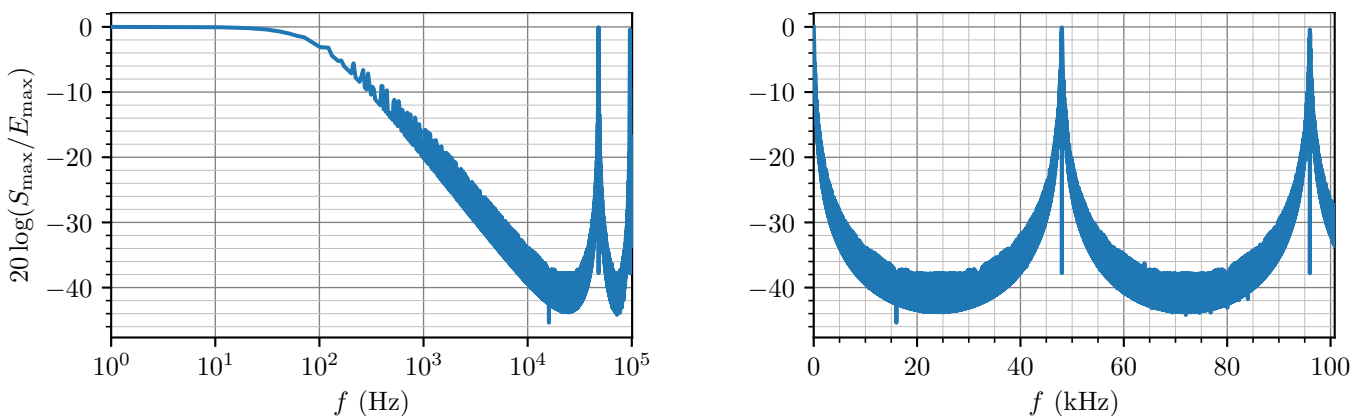


Figure 3 – Diagramme de Bode du filtre numérique. Diagramme construit point par point à l'aide de la fonction `bode`. À gauche, l'axe des fréquences est logarithmique alors qu'à droite il est linéaire. L'épaisseur des courbes est due au choix aléatoire de phase, qui joue en pratique un rôle crucial sur le gain du filtre aux fréquences qui ne vérifient pas (ou mal) le critère de Shannon.

✍ Identifier et interpréter les analogies et différences le diagramme de Bode du filtre analogique correspondant. Conclure sur le domaine d'utilisation du filtre numérique : quelle gamme de fréquence de coupure peut-on choisir ?

On retrouve un gain nul en basse fréquence, puis la partie haute fréquence démarre bien avec une pente de -20 dB/décade ... avant de faire apparaître des pics, qui ne sont pas présents dans le diagramme de Bode d'un filtre analogique.

Le diagramme de Bode est périodique, de période égale à la fréquence d'échantillonnage $f_e = 48$ kHz, ce qui est naturel compte tenu du processus d'échantillonnage à cette fréquence. Il y a également une symétrie par rapport à $f_e/2$. On retrouve en fait sur le diagramme de Bode tout ce qui a été constaté précédemment sur la réplication de spectre.

Ainsi, pour que le filtre ait un intérêt, il faut que la fréquence de coupure soit (nettement) inférieure à $f_e/2$.

Espace 10