

# Effet d'un filtre sur un signal

## Techniques et méthodes à acquérir

- ▶ Choisir les paramètres d'acquisition pour numériser un signal;
- ▶ Calculer le spectre d'un signal numérisé;
- ▶ Écrire un code Python permettant de déterminer le signal de sortie d'un filtre connaissant le spectre de son signal d'entrée.

## I - Allure qualitative du signal de sortie

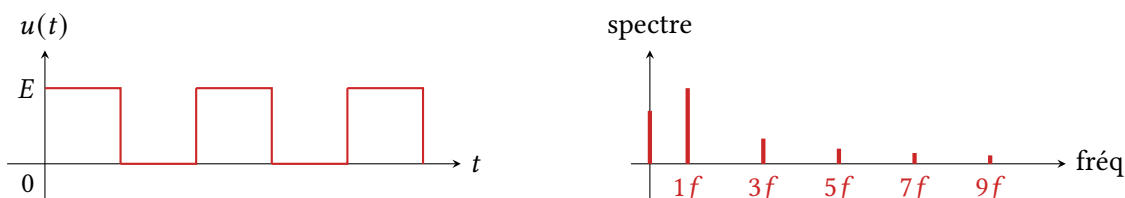
### I.A - Filtre passe-bas du premier ordre

1 - Proposer et réaliser un montage permettant de réaliser un filtre passe-bas du premier ordre de fréquence de coupure de  $f_c = 1 \text{ kHz}$  à partir d'une résistance  $R = 1 \text{ k}\Omega$ . Pourquoi faut-il choisir  $R$  élevée? Vérifier (expérimentalement!!) que le filtre que vous avez réalisé a bien la fréquence de coupure souhaitée.

On envoie en entrée de ce filtre un signal créneau de fréquence  $f$  réglable, et on souhaite interpréter l'allure du signal de sortie. Le développement de Fourier d'un signal créneau de fréquence  $f$  et d'amplitude crête-à-crête  $E$  s'écrit

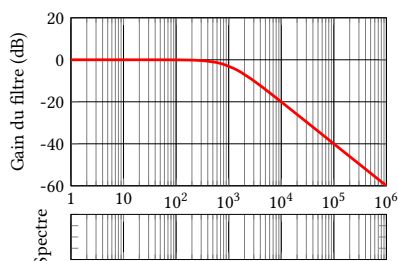
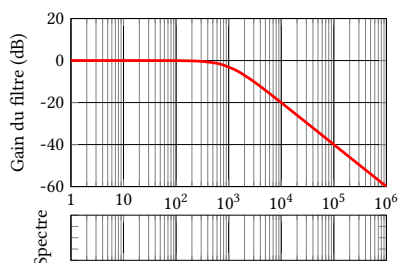
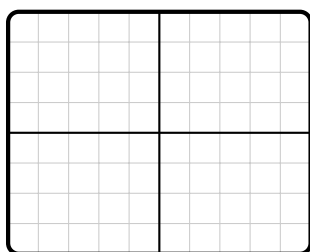
$$u(t) = \langle u \rangle + \frac{2E}{\pi} \sum_{n=1}^{\infty} \frac{1}{2n+1} \sin(2\pi n f t).$$

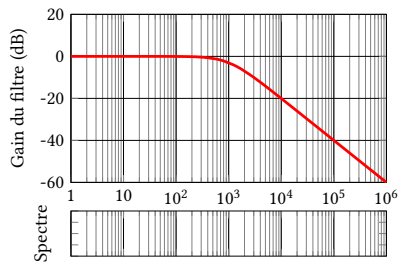
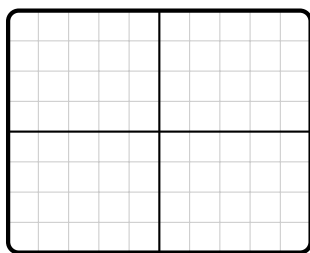
Il ne contient donc que des harmoniques impaires, dont l'amplitude est inversement proportionnelle à  $n$ .



2 - Envoyer en entrée du filtre RC construit précédemment un signal créneau et, en s'appuyant sur le diagramme de Bode, interpréter l'allure du signal de sortie pour les fréquences 10 Hz, 10 kHz et enfin 300 Hz.

Allure des signaux observés :      Diagramme de Bode et spectre :      Interprétation :





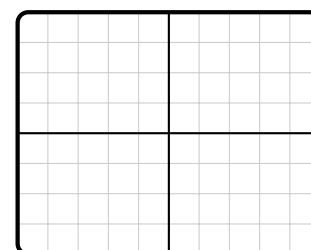
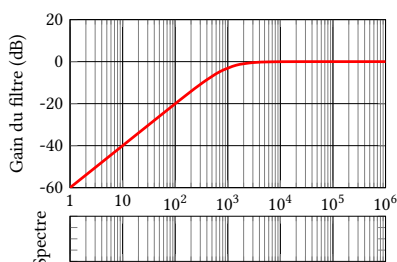
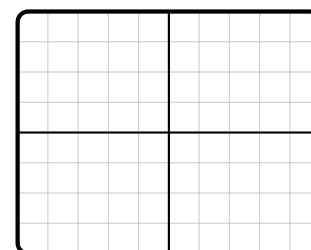
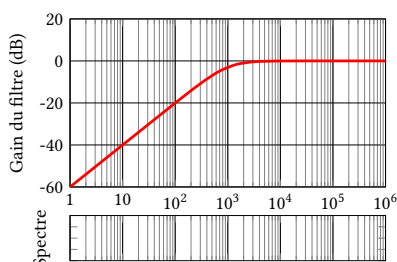
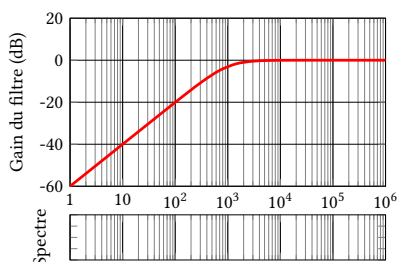
### I.B - Filtre passe-haut du premier ordre

3 - Modifier le montage pour obtenir un filtre passe-haut. Reproduire l'expérience précédente, mais cette fois en **anticipant** les résultats.

Diagramme de Bode et spectre :

Prévision :

Signal observé :



## II - Calcul numérique du signal de sortie

On souhaite désormais aller plus loin que cette approche qualitative, et calculer de manière exacte le signal de sortie d'un filtre et son spectre. Récupérer le code à compléter TP-E9\_effet-filtre\_script-eleve.py. Ce code est structuré en une première partie dans laquelle doivent être renseignées toutes les fonctions utiles, le programme proprement dit basé sur ces fonctions venant dans un second temps.

Les différentes fonctions sont les suivantes :

- ▶ Les fonctions dont le nom commence par `spectre` renvoient le spectre d'un signal donné sous forme d'une liste de listes à trois éléments  $f$ ,  $A$  et  $\phi$  contenant respectivement les fréquences présentes dans le spectre du signal, l'amplitude de l'harmonique correspondante et sa phase initiale.
- ▶ Une fonction `spectre_to_signal`, permet de construire la représentation temporelle du signal à partir de son spectre.
- ▶ Les fonctions dont le nom commence par  $H$  renvoient la fonction de transfert (complexe) d'un filtre pour une fréquence prise en argument, et pour des caractéristiques de filtre à préciser. Noter une petite subtilité : pour pouvoir être interchangées par la suite, les fonctions pour les filtres du premier et du deuxième ordre

doivent prendre le même nombre d'arguments, d'où la présence d'un argument optionnel `Q` (de valeur par défaut `None`) pour les filtres du premier ordre.

- Une fonction `calc_spectre_sortie` permettant de déterminer le spectre du signal de sortie d'un filtre à partir du spectre en entrée.
- Une fonction d'affichage `affiche_spectre` permettant d'afficher un spectre sous sa forme usuelle d'un diagramme en bâtons.

4 - Compléter la fonction `spectre_to_signal`. En utilisant cette fonction, afficher la courbe représentant un signal créneau de fréquence 1 kHz construit à partir de 10, 100 et 1000 harmoniques.

5 - En utilisant la fonction `affiche_spectre`, afficher dans une deuxième figure le spectre correspondant. Dans le cas d'un grand nombre d'harmoniques, il se peut que la figure semble vide en raison d'un trop grand nombre d'harmoniques d'amplitude très faible tracées en traits trop fin : redéfinissez dans ce cas les limites de l'axe des abscisses avec `plt.xlim(mini, maxi)`, par exemple en vous limitant à l'intervalle `[0, 20 kHz]`.

6 - Compléter la fonction `calc_spectre_sortie`.

7 - En combinant les différentes fonctions, reproduire numériquement les expériences du paragraphe I.A. Vérifier d'une part que l'allure des signaux correspond bien à celle que vous avez observée, et d'autre part que le spectre du signal est modifié comme vous l'aviez prévu.

### III - Détermination expérimentale du spectre d'un signal

L'observation de spectres est possible mais malaisée à l'oscilloscope, on utilisera donc la carte d'acquisition et le logiciel LatisPro.

8 - Dans le cas du filtre passe-bas et d'une entrée créneau de fréquence 300 Hz, acquérir le signal d'entrée et le signal de sortie du filtre. Choisir une acquisition sur 10 000 avec une période d'échantillonnage  $T_e \ll T = 1/f$  et une durée d'acquisition  $\Delta t \gg T$ .

9 - Réaliser l'analyse spectrale du signal d'entrée puis du signal de sortie (*Traitements* → *Calculs spécifique* → *Analyse de Fourier*). Comparer aux spectres calculés numériquement.

### IV - Bonus : sélection d'une harmonique unique

10 - Proposer un filtre transmettant en sortie une sinusoïde de fréquence  $3f$  à partir d'un signal créneau d'entrée de fréquence  $f$ . Commencer par tester numériquement vos idées avant de réaliser le montage.

#### Bilan à retenir :

- Les basses fréquences encodent l'allure générale du signal, les hautes fréquences ses variations rapides et notamment ses discontinuités. Ainsi, un filtre passe-bas tend à lisser ces discontinuités et un filtre passe-haut à les accentuer par rapport au reste du signal.
- Pour qu'une acquisition numérique d'un signal conduise à une estimation correcte du spectre, la période d'échantillonnage doit être faible et la durée d'acquisition aussi grande que possible comparativement à la période du signal.

