Devoir maison numérique
Analyse spectrale et filtrage

Rappels utiles en Python

- pour le calcul numérique, on utilise les bibliothèques numpy et matplotlib ;
- un fichier de données organisées en colonnes peut être importé rapidement avec l'instruction `x, y = np.loadtxt("nom_fichier.dat", skiprows=1, unpack=True)` où skiprows permet de sauter les lignes qui ne sont pas des données (en général les titres des colonnes) ;
- on peut tracer un nuage de points avec `plt.scatter(x, y)` ;
- on peut tracer une courbe continue avec `plt.plot(x, y)` ;
- on peut imposer une échelle logarithmique en x avec `plt.xscale('log')` ;
- on peut passer à une nouvelle figure avec `plt.figure(2)` .

On rappelle que tout signal peut se décomposer selon une somme de Fourier :

$$s(t) = \sum_{\{f_k\}} \hat{s}_k \cos(2\pi f_k t + \varphi_k)$$

Dans la représentation numérique des données, on représentera

- un signal par deux tableaux : temps et valeur du signal ;
- un spectre par trois tableaux : fréquences f_k, amplitudes $\hat{s}_k$ et phases φ_k.

I. Analyse de Fourier : synthèse spectrale

1.1. Construction d'une somme de Fourier

1 – Définir une fonction `synthèse` qui prend en paramètres trois tableaux représentant un spectre, ainsi qu'un nombre de périodes, et qui renvoie un signal (deux tableaux) construit à partir du spectre fourni. On fera attention aux difficultés suivantes :

- la durée du signal : la période du signal est $1/f_1$ où f_1 est la fréquence contenue dans la case 1 (la case 0 étant toujours un harmonique de fréquence nulle) ; ainsi la valeur de durée maximale est $T_{\max} = n_{\text{périodes}}/f_1$;
- le nombre de points pour le temps/le signal : on prendra 50 points pour chaque période de la fréquence la plus élevée du spectre, soit $n_{\text{points}} = T_{\max} f_{\max} \times 50$.

> Le code de cette fonction sera fourni au bout de quelques jours, puisqu'elle est nécessaire à la suite et potentiellement un peu difficile.

1.2. Premier exemple : signal carré

2 – Importer le spectre du signal carré ; le tracer. Commenter.

3 – Construire la somme de Fourier avec ce spectre, pour 2, 5, 20 ou 51 harmoniques. Commenter l'évolution.

1.3. Deuxième exemple : températures

Les fichiers `temperatures_marseille.dat` et `spectre_temperatures.dat` contiennent respectivement des relevés de température à Marseille pendant 4 jours (une mesure par heure, à partir du 30 janvier 2022 à minuit) et le spectre associé (attention : les fréquences y sont en h^{-1}).

4 – Construire la somme de Fourier du spectre de température, et tracer sur un même graphe cette somme et le signal mesuré expérimentalement.

5 – Tracer le spectre en amplitude. Quels sont les deux principaux harmoniques ? Interpréter.

1.4. (facultatif) Troisième exemple : signal mystère

6 – Reconstituer à partir de leur spectre (fichiers `spectre_x.dat` et `spectre_y.dat`) les signaux $x(t)$ et $y(t)$; les tracer. À votre avis, que représentent-ils ?

7 – Tracer $y(x)$. Votre hypothèse précédente était-elle correcte ?

II. Action des filtres

2.1. Passe-bas

8 – Définir une fonction python `passe_bas_1` qui prend en paramètre un spectre (donc trois tableaux) et une fréquence de coupure f_c, et renvoie trois tableaux correspondant au spectre filtré selon la fonction de transfert

$$\underline{H} = \frac{1}{1 + \mathrm{j}f/f_c}$$

ainsi qu'une fonction `passe_bas_2` correspondant aux mêmes spécifications d'entrée et sortie, mais qui applique un passe-base d'ordre deux de facteur de qualité $1/\sqrt{2}$:

$$\underline{H} = \frac{1}{1 - (f/f_c)^2 + \mathrm{j}\sqrt{2}f/f_c} = \frac{\mathrm{j}\sqrt{2}f/f_c}{1 + \mathrm{j}(f/f_c - f_c/f)/\sqrt{2}}$$

9 – Représenter le signal carré (construit avec les 51 harmoniques fournis) et le signal filtré avec un passe-bas de chaque ordre, de fréquence $150\,\mathrm{Hz}$. Commenter.

10 – Représenter le signal carré (construit avec les 51 harmoniques fournis) et le signal filtré avec un passe-bas d'ordre 1 de fréquence $10\,\mathrm{Hz}$. Commenter.

11 – Représenter le signal de températures et sont signal filtré par le passe-bas d'ordre 2 avec une coupure à $f_c = 1{\cdot}10^{-3}\,\mathrm{h}^{-1}$; quelle est l'opération réalisée ?

2.2. Passe-bande

12 – Définir une fonction python `passe_bande` qui prend en paramètre un spectre (donc trois tableaux) et une fréquence de résonance f_r, et renvoie trois tableaux correspondant au spectre filtré selon la fonction de transfert de facteur de qualité 10 :

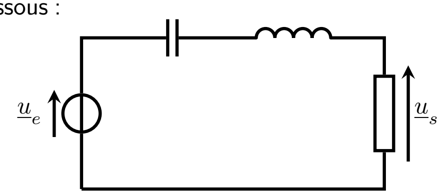
$$\underline{H} = \frac{\mathrm{j}/10f/f_r}{1 - (f/f_r)^2 + \mathrm{j}/10f/f_r} = \frac{1}{1 + 10\mathrm{j}(f/f_r - f_r/f)}$$

13 – Représenter le signal carré (construit avec les 51 harmoniques fournis) et le signal filtré avec un passe-bande de fréquence $372\,\mathrm{Hz}$. Commenter.

 adrien.licari-guillaume@ac-aix-marseille.fr

III. Étude d'un filtre expérimental

On a câblé le circuit ci-dessous :



14 – Établir la fonction de transfert de ce filtre en fonction de R, L et C. On identifiera le type de filtre et ses paramètres canoniques.

15 – Calculer les paramètres canoniques avec les mesures suivantes : $R = 100{,}6\,\Omega$, $L = 0{,}09590\,\mathrm{H}$, $C = 0.5006\,\mu\mathrm{F}$.

16 – On donne dans le fichier `mesures_Bode.dat` les mesures pour différences fréquences des amplitudes U_e et U_s, ainsi que de l'avance de phase **de $\boldsymbol{u_e}$ sur $\boldsymbol{u_s}$** . Tracer les diagrammes de Bode théoriques et expérimentaux.