Méthodes numériques

Action d'un filtre sur un signal périodique

Objectifs :

- Tracer un signal si on se donne la valeur des coefficients de son spectre.
- Simuler l'action d'un filtre sur ce signal (on se donnera la fonction de transfert du filtre).

Sur le site de la classe, rubrique TP, DS, maths ... : lien vers un compilateur Python en ligne (https://trinket.io/embed/python3/a5bd54189b). Et aussi fiche formulaire Python (distribuée plus tôt dans l'année).

Étape 1 : construction du signal d'entrée

On considère un signal d'entrée $e(t)$, périodique, de fréquence f_e (par exemple $f_e = 1\,\text{kHz}$). Sa pulsation est $\omega_e = 2\pi\, f_e$.

La décomposition de Fourier de ce signal est : $\boxed{e(t) = c_0 + \sum_{n=1}^{+\infty} c_n \cos(n\omega_e t + \varphi_n).}$

1 - Que représente c_0 ?

Comment s'appelle le premier terme de la somme (pour $n = 1$) ? Quelle est sa pulsation et sa fréquence ?

Comment s'appellent les termes pour $n > 1$? Quelle est leur pulsation et leur fréquence ?

2 - On considère le signal harmonique suivant : $e(t) = 0{,}5 + \cos(\omega_e t)$.

Donner la valeur des coefficients c_n et φ_n pour ce signal.

En Python, on se donne la liste des c_n et celle des φ_n, pour le signal d'entrée $e(t)$:

```
c_entree   = [0.5,1]
phi_entree = [0,0]
```

On souhaite ensuite tracer le signal $e(t)$. On dispose pour cela, dans le code, d'une fonction `signal(c,phi,t)`, déjà écrite, qui permet d'obtenir le signal $e(t)$. Lire pour cela sa description dans le code. Pour l'utiliser :

- On crée une liste d'instants `t[i]` avec `t = np.linspace(0,5*Te,500)` (cf code).
- On fait l'affectation : `signal_entree = signal(c_entree,phi_entree,t)`

  `signal_entree` est alors une liste qui contient les valeurs de $e(t)$.

3 - Copier-coller le code du fichier source `info_filtrage_etape1.py` dans votre éditeur Python.

Compléter le code pour pouvoir tracer la liste `signal_entree` en fonction de la liste `t`. On légendera l'axe des abscisses (t (s)) et des ordonnées (signal), ainsi que la courbe, on mettra une grille. Se référer à la fiche python distribuée en début d'année.

Une fois que cela fonctionne, recopier vos lignes de code (qui concernent la figure) sur votre compte-rendu pour en garder une trace.

4 - Le signal tracé est-il cohérent avec un cosinus d'amplitude 1 et de valeur moyenne 0.5 ?

5 - On souhaite tracer un autre signal.

Modifier les listes `c_entree` et `phi_entree` pour tracer le signal :

$e(t) = 0{,}5 + 2{,}5\cos(\omega_e t) + 1{,}5\cos(2\omega_e t) + 1{,}5\cos(3\omega_e t + \pi/2)$.

Recopier sur votre compte rendu ce que vous avez écrit pour `c_entree` et `phi_entree`.

Étape 2 : signal de sortie du filtre

On rappelle l'action d'un filtre sur le signal d'entrée :

$$e(t) = c_0 + \sum_{n=1}^{+\infty} c_n \cos(n\omega_e t + \varphi_n) \quad \xrightarrow{\text{sys}} \quad s(t) = c'_0 + \sum_{n=1}^{+\infty} c'_n \cos(n\omega_e t + \varphi'_n),$$
$$\text{avec} \quad c'_n = c_n \times |\underline{H}(n\omega_e)|, \quad \text{et} \quad \varphi'_n = \varphi_n + \arg(\underline{H}(n\omega_e)).$$

On commence par l'exemple d'un filtre passe-bas. Sa fonction de transfert, son gain et son argument sont :

$$\underline{H}(\omega) = \frac{1}{1 + \mathrm{j}\dfrac{\omega}{\omega_0}}, \quad G = |\underline{H}(\omega)| = \frac{1}{\sqrt{1 + \omega^2/\omega_0^2}}, \quad \arg(\underline{H}(\omega)) = -\arctan\frac{\omega}{\omega_0}.$$

6 - Copier-coller le code du fichier source `info_filtrage_etape2.py` dans votre éditeur Python, à la suite de votre code précédent.

Compléter les codes des fonctions `G(w)` et `argH(w)` pour le filtre passe-bas. Recopiez les sur votre compte rendu.

7 - Notons c'_n et φ'_n les coefficients du spectre du signal de sortie, et c_n et φ_n ceux du signal d'entrée.

Donner l'expression de c'_n en fonction de c_n et du gain G (à quelle pulsation ce gain doit-il être pris ?).

Donner l'expression de φ'_n en fonction de φ_n et de l'argument $\arg(\underline{H})$ (à quelle pulsation cet argument doit-il être pris ?).

8 - Compléter alors les deux lignes `c_sortie.append` et `phi_sortie.append` en indiquant ce qu'il faut ajouter aux listes `c_sortie` et `phi_sortie` pour les remplir (d'après la question précédente). Recopiez les sur votre compte rendu.

9 - Compléter enfin la ligne `signal_sortie =` qui calcule le signal de sortie à partir de ses coefficients de Fourier `c_sortie` et `phi_sortie` .

10 - La suite du code contient déjà les lignes qui permettent de tracer le signal de sortie. L'exécuter : ceci doit afficher les signaux $e(t)$ et $s(t)$.

Actuellement dans le code, que vaut la fréquence du signal d'entrée ? Et la fréquence $f_0 = \omega_0/(2\pi)$ du filtre passe-bas ? Interpréter alors l'allure de la courbe de $s(t)$ par rapport à celle de $e(t)$.

Essayer en changeant la fréquence de coupure f_0 du filtre.

Étape 3 : interprétation

11 - Copier-coller le code du fichier source `info_filtrage_etape3.py` dans votre éditeur Python, à la suite de votre code précédent. Il s'agit de lignes, déjà complètes, qui permettent de tracer le diagramme de Bode du filtre sur la figure du haut, et les coefficients c_n et c'_n sur la figure du bas (pour $n = 1,2,...$, mais c_0 n'est pas tracé).

Conservez $f_e = 1\,\text{kHz}$, faire un essai avec $f_0 = 1\,\text{kHz}$, puis $f_0 = 10\,\text{kHz}$ et $f_0 = 100\,\text{Hz}$. Interpréter à chaque fois.

Filtre passe-bande

12 - À l'aide de votre cours, écrire la fonction de transfert d'un filtre passe-bande (sous forme canonique), de son gain G et de $\arg(\underline{H})$.

13 - Modifier alors, dans le code, les fonctions `G(w)` et `argH(w)` pour qu'elles correspondent au filtre passe-bande. Il faudra introduire une variable Q. Prendre $Q = 10$ par exemple.

Essayer. Normalement le code retourne une erreur, car en $\omega = 0$ le terme en $1/\omega$ n'est pas défini. Il faut donc, dans les fonctions `G(w)` et `argH(w)`, faire un test :

```
if w=0:
    return  # valeur de G ou de arg(H) à retourner si w=0 (à compléter)
else:
    return  # expression complète de G(w) ou de arg(H(w)) (à compléter)
```

Le faire et essayer.

14 - Si tout fonctionne, vous pouvez faire plusieurs essais (en variant Q et f_0) et interpréter à chaque fois.

En particulier, essayer de choisir Q et f_0 pour ne sélectionner que l'harmonique $n = 2$ du signal $e(t)$.

Action sur un signal triangle

Pour étudier l'action du filtre sur un signal triangle, il faut d'abord en construire un. Ses coefficients de Fourier sont les suivants :

$$\begin{cases} \text{si } n \text{ pair :} & c_n = 0, \qquad \varphi_n = -\dfrac{\pi}{2} \\ \text{si } n \text{ impair :} & c_n = (-1)^{(n-1)/2} \times \dfrac{8}{\pi^2 n^2}, \qquad \varphi_n = -\dfrac{\pi}{2} \end{cases}$$

En Python, le symbole `%` permet d'obtenir le reste dans la division euclidienne. On peut donc tester si n est pair avec `n%2 == 0` : ceci est vrai si et seulement si n est pair.

15 - Reprendre, au début de votre code, la définition des listes `c_entree` et `phi_entree` afin de les définir à l'aide de la définition ci-dessus. On utilisera une boucle du type suivant :

```
c_entree   = []
phi_entree = []
for n in range(20):
    phi_entree.append(... à compléter
    if n%2==0:
        c_entree.append( ... à compléter
    else:
        c_entree.append( ... à compléter
```

Tester pour $N = 3$, 5, 20... et voir que votre signal d'entrée est de plus en plus triangulaire. Pour la suite on prendra par exemple 20 termes.

Recopier ce que vous avez écrit sur votre compte rendu.

On s'intéresse ensuite à l'action du filtre passe-bande sur le signal triangle. On souhaite se placer dans trois configurations différentes :

16 - Choisir les caractéristiques du filtre pour ne sélectionner que le fondamental du signal triangle.

17 - Dans quel domaine du diagramme de Bode un filtre a-t-il une action de dérivateur ? Choisir les caractéristiques du filtre pour réaliser la dérivée du signal triangle. (on prendra Q assez faible pour éviter une résonance)

18 - Dans quel domaine du diagramme de Bode un filtre a-t-il une action d'intégrateur ? Choisir les caractéristiques du filtre pour réaliser une primitive du signal triangle.

Ceux qui ont terminé peuvent chercher comment construire un signal créneau (chercher son spectre sur internet), ou bien peuvent câbler un circuit RLC pour réaliser un passe-bande et reproduire (en vrai, avec un GBF et Latis Pro) les observations du TP.