

Exercice 2 : Listes et exploitations d'un relevé expérimental

Présentation

On étudie le régime libre d'un circuit RLC série. La tension aux bornes du condensateur est enregistrée à intervalles réguliers et fournie sous la forme d'une liste de flottants `signal`, la mesure d'indice i étant relevée à l'instant $t_i = i T_e$, où T_e est la période d'échantillonnage. L'enregistrement comporte plusieurs pseudo-périodes et se prolonge jusqu'au retour à l'équilibre.

L'objectif est d'extraire de ce relevé la pseudo-période T , la pulsation propre ω_0 et le facteur de qualité Q de l'oscillateur, en n'utilisant que des listes. Aucune bibliothèque n'est autorisée, à l'exception du module `math` :

```
from math import log, pi, sqrt
```

Partie A — Premières mesures sur le relevé

1. Écrire une fonction `indice_maximum(signal)` renvoyant l'indice du premier maximum global de la liste.
2. L'oscillateur revient vers une position d'équilibre x_∞ qui n'est pas nécessairement nulle, l'acquisition pouvant présenter un décalage. Écrire une fonction `valeur_finale(signal, p)` renvoyant la moyenne des p dernières mesures. Pourquoi moyenner plutôt que prendre la dernière valeur ?
3. Écrire une fonction `ecarts(signal, x_inf)` renvoyant la liste des écarts à la position d'équilibre. La liste `signal` ne doit pas être modifiée.
4. Le temps de réponse à ε près est l'instant à partir duquel l'écart reste, en valeur absolue, inférieur à ε jusqu'à la fin de l'enregistrement. Écrire une fonction `temps_de_reponse(ec, epsilon, te)` qui le renvoie. Expliquer pourquoi il serait faux de s'arrêter au premier indice où l'écart passe sous ε .

Partie B — Lissage par moyenne glissante

Le bruit d'acquisition crée de nombreux maxima parasites, qu'il faut éliminer avant toute recherche d'extremum. On remplace pour cela chaque mesure par la moyenne de ses voisines sur une fenêtre de $2p + 1$ points.

1. Écrire une fonction `moyenne_locale(signal, i, p)` renvoyant la moyenne des valeurs d'indices compris entre $i - p$ et $i + p$. Lorsque la fenêtre déborde de la liste, on la tronquera aux indices valides ; la moyenne portera alors sur un nombre réduit de points.
2. En déduire `moyenne_glissante(signal, p)`, qui renvoie le signal lissé, de même longueur que l'original. Quelle est la complexité de cette fonction en fonction de n et de p ?
3. Que se passe-t-il si la largeur de la fenêtre approche la pseudo-période du signal ? Quel critère de choix de p proposer ?

Partie C — Pseudo-période et décrétement logarithmique

1. Écrire une fonction `indices_maxima(signal)` renvoyant la liste croissante des indices des maxima locaux stricts, les deux extrémités étant exclues.
2. Écrire une fonction `pseudo_periode(indices, te)` estimant T par la moyenne des écarts entre maxima consécutifs. Simplifier ensuite l'expression obtenue : quelles mesures interviennent réellement dans le résultat ?
3. Le décrétement logarithmique est défini, sur deux maxima successifs d'écarts x_k et x_{k+1} , par $\delta = \ln(x_k/x_{k+1})$. Écrire une fonction `decrement(ec, indices)` renvoyant sa valeur moyenne sur l'enregistrement.
4. On rappelle que les deux paramètres caractérisant un oscillateur amorti sont : la pulsation propre ω_0 et le facteur de qualité Q . On montre que $\delta = 2\pi/\sqrt{4Q^2 - 1}$ et que $\frac{2\pi}{T} = \omega_0\sqrt{1 - \frac{1}{4Q^2}}$. Écrire une fonction `parametres(delta, t_pseudo)` renvoyant le couple (ω_0, Q) .