Note sur la traduction de transformée de Welch en javascript

13 mai 2025

Il n'existe pas de bibliothèque fournissant la transformée de Welch en javascript. Donc je l'écris en utilisant `https://github.com/indutny/fft.js/` (fft.js-master) pour la fft. J'implémente que ce qui est utilisé dans le programme python initial, *ie* fenêtre de hann et ... La référence de la fonction welch est https ://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.welch.html

Dans le javascript les paramètres passés sont

— `data,`
— `-f fs` (TEIs.getModule(TEImodule).AdcSamplingRate),
— `-s` len(data)//1024
— `-m` nombre de segments

les paramètres affectés

— window='hann',
— nperseg=nbperseg
— scaling='density'

Dans le python la commande scipy lancée est :

signal.welch(data, fs, 'hann',nperseg=nbperseg, scaling='density') avec fs=2000000 nbperseg=len(data)

La preuve est le log suivant (les lignes apparaissent deux fois car on a appelé une fois avec seg=1 et une fois avec seg=2) :

PY : commande lancee : signal.welch(data, fs, 'hann',nperseg=nbperseg, scaling='density') avec fs=2000000 nbperseg=16384 len(data)=16384

PY : commande lancee : signal.welch(data, fs, 'hann',nperseg=nbperseg, scaling='density') avec fs=2000000 nbperseg=32768 len(data)=32768

PY : commande lancee : signal.welch(data, fs, 'hann',nperseg=nbperseg, scaling='density') avec fs=2000000 nbperseg=65536 len(data)=65536

fenêtre de Hann :

$$w(n) = \frac{1}{2} - \frac{1}{2}\cos\left(\frac{2\pi n}{M-1}\right) \qquad 0 \leq n < M$$

On lance toujours le cas nperseg=len(data) et si on a demander sec>1 on relance avec nperseg=len(data)/seg. Le nombre de frquence sera alors (len(data)/2+1)/seg

Questions

D'après la commande lancée nbperseg=len(data)/seg, mais les segments **se recouvrent par défaut de moitié** donc seg n'est pas le nombre de segments. Par exemple si $n = 2^k$ et seg= 2 nbperseg= 2^{k-1} et il y aura 3 segments : $\left[0, 2^{k-1}\right]$, $\left[2^{k-2}, 3\,2^{k-2}\right]$ et $\left[2^{k-1}, 2^k\right]$

Les segments doivent aussi être des puissances de 2. Ci-dessus on a vu 2^{k-1}. Pour 2^q le débuts des segments seront de $\alpha 2^{q-1}$ et le dernier segment sera tel que $\alpha^* 2^{q-1} = 2^k$ donc $\alpha^* = 2^{k-q+1}$.

$$\text{nps} = \frac{N}{2^k} \Longrightarrow \text{nbs} = 2^{k+1} - 1$$