

---

# Densité spectrale de puissance

déc. 12, 2018

## 1 Densité spectrale du bruit d'une machine à laver

Nous avons enregistré le bruit d'une machine à laver le linge pendant la phase d'essorage. Les fichiers `.wav` dans le répertoire `data` correspondent à trois enregistrements. On entend un son aigu correspondant au moteur. Le tambour tourne beaucoup plus lentement.

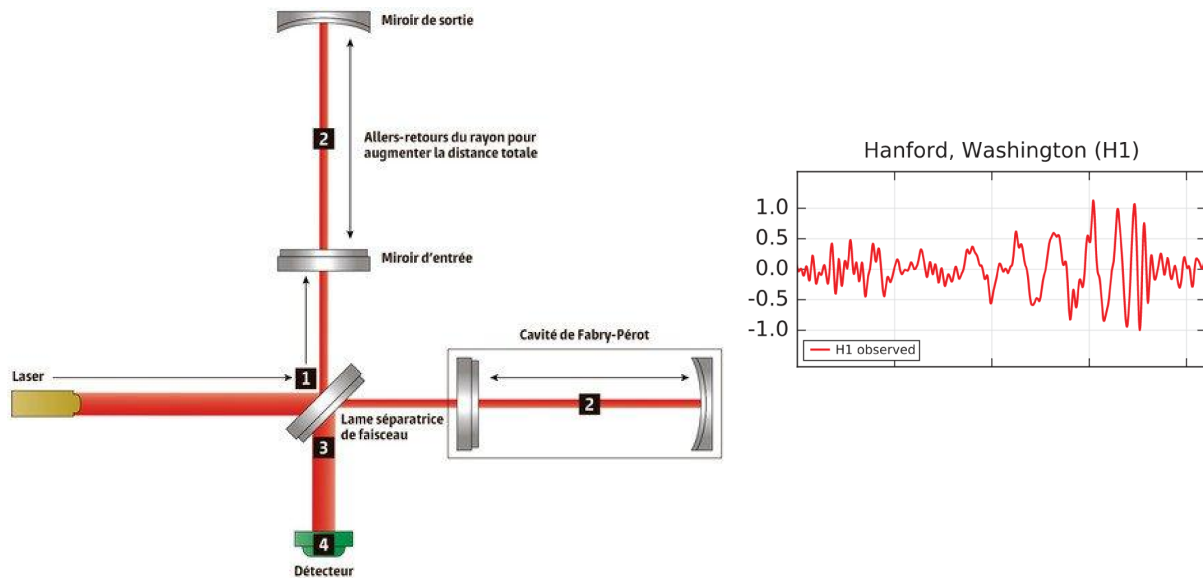
1. Quel est la fréquence typique de rotation du tambour d'une machine à laver la linge ? (Il n'est pas interdit d'utiliser internet pour répondre à cette question).
2. On va lire les fichiers `wav` en utilisant la fonction `read` de la librairie `scipy.io.wavfile`. On enregistrera l'amplitude du son dans la variable *amplitude*

```
from scipy.io.wavfile import read
samplerate, amplitude = read('data/machine_a_laver.wav')
```

Tracer l'amplitude du son pendant la dernière seconde.

3. On va utiliser la fonction `periodogram` du module `scipy.signal`. Le périodogramme est le nom de la méthode utilisée pour estimer la densité spectrale de puissance en se basant sur la transformée de Fourier. Déterminer la fréquence de rotation du tambour en zoomant entre 10 et 50 Hz.
4. On suppose que après la transformation de la question 2, les unités de `amplitude` sont des volts (V). Quelle est l'unité de la densité spectrale de bruit ? Quelle est sa valeur typique entre 10 et 100 Hz ?
5. Lorsque l'on fait l'analyse spectrale du son, nous distinguons le signal que l'on veut voir (ici la rotation du tambour) du bruit ambiant. Le signal est défini par son amplitude (ou puissance instantanée) et le bruit par sa densité spectrale de puissance. En utilisant l'option `scaling='spectrum'` de la fonction `periodogram`, calculer l'amplitude du signal. Estimer le rapport signal à bruit.
6. Sur quelle bande passante faut-il filtrer pour avoir un signal à bruit raisonnable ? Filtrer le signal et le tracer.

## 2 Onde gravitationnelle



L'objectif de cette partie est d'analyser les signaux provenant des deux détecteurs d'onde gravitationnelle de LIGO (Hanford et Livingston). Le signal qui est enregistré est de celui de l'élongation relative (strain) des deux bras de l'interféromètre de Michelson.

1. En utilisant la fonction `loaddata` de la librairie `readligo`, extraire les données d'élongation relative et de temps des deux fichiers `H-H1_LOSC_4_V1-1126259446-32.hdf5` et `L-L1_LOSC_4_V1-1126259446-32.hdf5`.
2. L'unité de temps est la seconde. Quelle est la durée de la mesure, le taux d'échantillonnage ?
3. Ces données sont centrées autour du moment où une onde gravitationnelle a été détectée, le 14 septembre 2015, vers 9h50. Quelle est la date de l'origine du temps ?
4. Tracer le signal pendant 10 secondes autour du passage de l'onde (autour du centre des données). Le passage de l'onde gravitationnelle a duré 100 ms environ. Peut-on voir ce signal ?
5. Tracer la densité spectrale de puissance entre 10 Hz et 2kHz. On utilisera la fonction `periodogram`. Puis on utilisera la fonction `welch` (toujours dans le module `scipy.signal`). La méthode de Welch effectue la moyenne de plusieurs périodogrammes pris consécutivement sur des durées plus courtes (argument `nperseg`). On prendra une largeur de bande de 1 Hz (i.e. on prendra `nperseg` égal au taux d'échantillonnage) :

```
f, psd = welch(strain_H1, samplerate, nperseg=samplerate)
```

6. Quel est la valeur typique de la densité vers 100 Hz. Estimez la variance du signal après un filtre entre 30 et 300 Hz. Qu'en pensez-vous ?
7. Le problème est qu'il y a des pics de résonance qui représentent la majeure partie du bruit. Pour les supprimer, on va appliquer un filtre dont le gain en puissance sera inversement proportionnel à la densité spectrale de puissance.

Voici comment ce filtre sera appliqué

```
from scipy.interpolate import interp1d

def whiten(strain, dt):
    freqs, psd_welch = welch(strain, fs=1/dt, nperseg=int(1/dt))
    interp_psd = interp1d(freqs, psd_welch)
```

```

N = len(strain)
freqs = np.fft.rfftfreq(N, dt)

strain_tilde = np.fft.rfft(strain)
gain = 1 / np.sqrt(interp_psd(freqs))
gain = gain/gain.max()
white_strain_tilde = strain_tilde * gain
white_strain = np.fft.irfft(white_strain_tilde)
return white_strain

```

8. Il reste encore du bruit à haute et basse fréquence. Appliquer un filtre passe bas de fréquence 300 Hz et un filtre passe haut de fréquence 20 Hz. Peut-on voir le signal ? Tracer le signal des deux détecteurs l'un à côté de l'autre, en changeant le signe d'un des deux signal. Quel est le délai entre les deux signaux ?

9. Sur cet événement, on peut facilement voir le signal sur chaque détecteur car il est supérieur au bruit après filtrage. Cependant, il est aussi possible de corrélérer les deux détecteurs. On peut alors appliquer un filtre passe bas de plus basse fréquence car après corrélation, le signal ne se moyenne plus à zéros.

Effectuer le produit des deux signaux avec un délai  $\tau$  et filtrer avec un filtre passe bas de fréquence de coupure à 10 Hz. Donner un ordre de grandeur du bruit RMS et de l'amplitude du signal. Quel est le facteur d'amélioration sur le rapport signal sur bruit ?

Tracer sur une image en fausse couleur la corrélation en fonction de  $\tau$ .

**Note :** Nous avons fait une analyse a posteriori du bruit. Cette analyse doit en principe être faite a priori afin de prévoir le sensibilité de l'instrument. C'est par exemple ce qui est fait pour les améliorations prévues sur ce détecteur.

