Signal, TP1

27 novembre 2019

1 Stationnarité et ergodicité

Exercice 1 Qu'est ce qu'une variable aléatoire ? Donnez la définition mathématique. Si vous ne l'avez jamais vu en cours, demandez à l'encadrant.

Exercice 2 Qu'est ce qu'un signal aléatoire ? Donnez la définition mathématique.

Exercice 3 En utilisant la fonction **loadtxt()** du module **numpy**, chargez les données du fichier s.csv dans un tableau numpy **s**. L'indice du tableau **s** est interprété comme une discrétisation temporelle.

Affichez le signal à l'aide du code suivant :

```
import matplotlib.pyplot as plt
plt.figure()
plt.plot(s)
plt.show()
```

Sommes nous en mesure de dire que nous avons à faire à un signal déterministe ? Justifiez votre réponse à l'aide de la définition.

Exercice 4 Chargez maintenant le fichier S.csv avec **loadtxt()** dans un tableau **S**. Dans la suite du TP, chaque lignes correspond à une nouvelle acquisition expérimentale de notre signal et les columnes correspondent à la discrétisatoin temporelle. Affichez différentes lignes en utilisant le code de l'exercice précédant.

Que peut-on dire du caractère déterministe ou aléatoire de notre signal ?

Exercice 5 Rappelez la définition de la stationnarité d'un signal. Chargez maintenant le fichier S.csv et écrivez un programme qui analyse la stationnarité de S. Le signal S semble-t-il être stationnaire ?

Exercice 6 Rappelez la définition de l'ergodicité d'un signal. Ecrivez un programme qui analyse l'ergodicité de S. Le signal S semble-t-il ergodique ?

Exercice 7 Donnez un exemple de signal stationnaire non ergodique.

2 Introduction au filtrage : réduction de bruit

2.1 Moyennes locales et convolution

Exercice 8 Générez un signal porte

$$x(t) = \mathrm{Porte}(t - t_0) + b(t)$$

où Porte est une fonction porte et b un bruit blanc gaussien. Vous choisirez la longeur du tableau, la largeur de la porte, le paramètre t_0 et la variance du bruit. A votre avis, comment est-il possible de réduire l'influence du bruit sur notre signal en manipulant les valeurs de **x** ? Expliquez pourquoi votre méthode est intéressante.

Exercice 9 Nous allons effectuer un moyennage local des valeurs de **x**. Ecrivez une fonction **moyenneLocale()** qui prend en argument un tableau **y** et un entier **n**, et qui renvoie un tableau **yMoy** tel que

$$\mathbf{yMoy}[\mathbf{k}] = \frac{1}{2n+1} \sum_{i=-n}^{n} \mathbf{y}[\mathbf{k+i}]$$

quand k n'est pas proche du bord. Appelez votre fonction avec le tableau **x** et différents entiers **n**, affichez et commentez les résultats.

Exercice 10 Rappelez la formule de la convolution de signaux à temps continu. La fonction **convolve** de **numpy** effectue la convolution discrète. Vous pouvez vérifier la formule utilisée dans la documentation de **numpy**, ainsi que les différentes options proposées. On va maintenant effectuer le moyennage local en utilisant la fonction **convolve**. Appelez la fonction **convolve** avec en argument le tableau **x** et un tableau valeurs constantes égales à $1/n$ et de longeur **n**, pour différentes valeurs de **n**. Affichez les résultats.

Exercice 11 Construisez vous-même de nouveaux noyaux de convolutions et filtrez le signal **x** à l'aide de ces noyaux.

Exercice 12 Bonus. A l'aide de la fonction **fft** du module **numpy.fft**, observez l'impacte du filtrage sur la transformée de Fourier de **x**.

2.2 Filtre adapté

Exercice 13 Quelle est la réponse impulsionnelle du filtre adapté qui permet de détecter la porte dans le signal x ?

Exercice 14 Quelle est l'expression mathématique de la réponse impulsionnelle h du filtre adapté qui permet de détecter la porte dans le signal x ?

Exercice 15 Ecrivez on programme qui calcule la convolution entre le signal x et la réponse impulsionnelle h. Retrouvez la valeur du paramètre t_0 choisie précédement.

Exercice 16 Même question mais utilisez maintenant la fonction **np.convolve**. La difficulté principale est de choisir correctement la convention de correspondance entre valeurs de temps et valeurs d'indice, et d'interpréter de manière adaptée le résultat du produit de convolution.

Exercice 17 Faites varier la largeur de la porte et la variance du bruit, et regardez comment évolue le signal filtré.

Exercice 18 Effectuez maintenant le filtrage adapté dans le domaine fréquentiel. On utilisera pour cela les fonctions de transformée de Fourier tu TP de python puis la fonction **fft** de **numpy**.