Thomas Dagier 31.01.2020

Quentin Rod

Rapport Travail Pratique sur Fourrier

Dans le cadre du cours de mathématiques appliquées à l'ingénierie, nous devons analyser un signal et décoder un message secret en appliquant les séries de Fourrier. Cela nous permet de mieux comprendre le cours étant donné que l'analyse de signaux n'est pas simple à visualiser. Nous avons 4 fichiers de une seconde qui sont samplés à 8192Hz dans lesquels du bruit a été ajouté aléatoirement. Nous savons que le bruit est représenté, dans chaque fichier, par 3 paires de sinus de fréquences proches soit 6 sinus en tout. De plus, du bruit a été rajouté en dehors de l'intervalle 100Hz – 1900Hz que nous ne devrons pas prendre en compte.

L'objectif principal est de recréer le signal complet à l'identique en enlevant le bruit. Pour ce faire, nous devons récupérer tous les signaux du fichier ce qui revient à calculer les coefficients Ak et Bk de ces derniers. Par la suite, nous devons appliquer une transformée de Fourrier inverse pour reconstruire le signal sur chacun des samples.

La première étape a été d'utiliser un fichier simple dont nous connaissions les fréquences des signaux. Une fois que nous avons correctement reconstruit le signal Troissinus, nous avons testé sur un des fichiers dans lequel une partie du message secret est à trouver. Pour visualiser le signal original et celui reconstruit, nous avons utilisé la librairie MatPlotLib. Cela dit, il n'y avait pas de bruit sur Troissinus mais des fréquences à isoler donc nous avons du légerment changer notre approche pour trouver le message en calculant tous les Ak et les Bk grâce aux formules vues en cours :

$$a_k = \sum_{l=0}^{l=fs-1} \cos(2\pi . k . \frac{l}{fs}) . s(\frac{l}{fs}) . \frac{1}{fs}$$

$$b_k = \sum_{l=0}^{l=fs-1} \sin(2\pi . k . \frac{l}{fs}) . s(\frac{l}{fs}) . \frac{1}{fs}$$

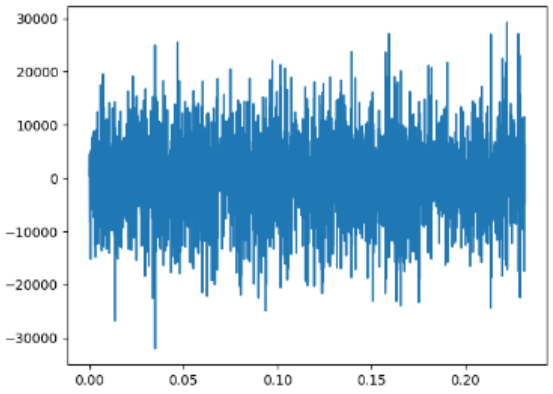
Pour réaliser cela, nous utilisons deux dictionnaires sous la forme: clef = k, valeur = Ak et clef = k , valeur = Bk. Nous pouvons stocker dans nos dictionnaires l'ensemble des Ak et Bk dont le k est compris entre 100 et 1900Hz.

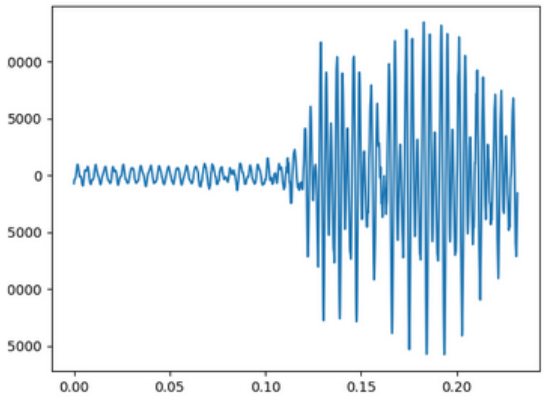
Pour identifier le bruit, nous avons affiché tous les coefficents Ak et Bk dans le terminal et avons remarqué que certains étaient beaucoup plus grands que les autres. Nous avons aussi remarqué sur MatPlotLib que l'amplitude des signaux qui se distinguent est beaucoup plus elevée. Nous en concluons que ces sinus sont du bruit car ils prennent le dessus sur le message caché. Nous avons définit un seuil maximum pour les coefficients qui est de 100 (défnit arbitrairement selon les fréquences du bruit). Pour les 6 sinus dont les coefficients sont au-dessus, nous les ajoutons quand meme aux dictionnaires mais avec une valeur de 0 pour qu'ils influencent le moins possible la voix.

L'ensemble des Ak et Bk récupérés correspondent à la série de Fourrier du message final. Grace aux formules du cours, nous pouvons alors reconstruire le signal filtré. Pour cela, nous devons trouver les images de l'ensemble des points qui constituent le signal soit les 8192 points qui correspondent à un échantillon d'une seconde.

$$a_0 + \sum_{k=1}^{\infty} (a_k . \cos(2\pi . k . t) + b_k . \sin(2\pi . k . t))$$

Pour chaque point on calcule donc son image, celle ci est la somme des images de chaque sinuosoides composant le signal filtré. Il suffit donc de parcourir chaque échantillon du sample soit 8192 au total et parcourir tous les Ak et Bk pour trouver l'image correspondante. Nous avons également constaté que le signal filtré était très peu audible. Pour remedier à ce problème, nous avons augmenté le niveau sonore en chaque point en le multipliant par 100. L'ensemble des points sont stockés dans une liste. Nous pouvons observer sur les images suivantes respectivement le 1er signal donné et celui filtré.





Nous avons répété ces opérations autant de fois qu'il y a de morceaux sonores. Nous avons alors une liste composée de l'ensemble des points des 4 fichiers. Afin d'obtenir un fichier sonore à partir de cette liste nous avons utilisé une méthode fournie qui consiste à transformer une liste en un fichier au format WAV.

Grâce à ce travail pratique, nous avons pu assimiller concrêtement les notions théoriques abordées durant le cours de mathématiques. Nous avons rencontré quelques difficultés pour identifier les signaux à supprimer et pour écouter le message après reconstruction car le signal était trop faible. Cet exercice nous a également permis de renforcer nos compétences en programmation Python et en théorie des signaux puisque nous avons dû comprendre comment fonctionnent les séries de Fourier afin de supprimer le bruit et de garder l'essentiel... hélas, le message n'est pas très interessant.