TP 5 – Transformée de Fourier - Signaux de synthèse

S.Gibet

Année 2022-2023

1 Transformée de Fourier discrète sur des signaux de synthèse

Le spectre d'un signal échantillonné peut être obtenu en utilisant la transformée de Fourier discrète (TFD). Dans cet exercice on généralise la notion de série de Fourier (applicable sur un signal périodique) à n'importe quel signal échantillonné, pas forcément périodique.
Dans ce TP, vous reprendrez quelques signaux de synthèse vus en cours (voir fichier `fft-M1.py`). L'implémentation de la TFD est ici la fft (Fast Fourier Transform).

1.1 Exercices de base vus en cours

1. Signal impulsionnel (ou Dirac) – Faites tourner le code en Python pour générer l'impulsion de Dirac et pour calculer la fft de ce signal impulsionnel ainsi que pour visualiser ses parties réelle et imaginaire (signal généré au TP précédent).

2. Signal "marche unitaire" – Reprenez le signal de synthèse "marche unitaire" défini dans le TP précédent. Générez à l'aide du code fourni ce signal (20 échantillons) et visualisez le. Calculez la transformée de Fourier de ce signal et visualisez ses parties réelle et imaginaire.

1.2 TFD d'un signal sinusoidal

1. La suite du code permet de générer un signal sinusoïdal `x` de 16 Hz (vecteur de 256 échantillons) et de le tracer avec matplotlib. La fréquence d'échantillonnage est de 256 Hz. Vous pourrez utiliser la fonction `genSin` du TP3 pour générer ce signal sinusoïdal ou bien reprendre la fonction fournie.

2. La suite du code permet de calculer la transformée de Fourier du signal `x` en utilisant la fonction `fft`. Quelle est la taille du vecteur `X` ainsi calculé et le typage de ses éléments ?

3. Il est possible de calculer et d'afficher les parties réelle et imaginaire du vecteur `X` calculées à la question précédente.

   ```
   Re = np.real(X)                   # partie reelle de X
   Im = np.imag(X)                   # partie imaginaire X
   ```

   Vérifiez l'affichage donné par le code fourni. La fonction `grid` permet d'afficher une grille. Quelles sont les grandeurs et les valeurs limites sur les axes x et y.

4. Plutôt que de calculer la partie réelle et imaginaire, on peut également calculer l'amplitude et la phase du spectre (préférable aux parties réelle et imaginaire).

```
A = np.abs(X/N)                    # amplitude du spectre
An = A/A.max()                     # amplitude normalisée
Ph = np.angle(X/N)                 # phase normalisée
```

Observez attentivement ce code et reportez les plages de variations sur l'axe fréquentiel. Pourquoi 128 points ?

5. Le code fourni permet également de calculer l'amplitude normalisée en dB :

```
AdB = 20*np.log10(An)          #amplitude en dB
```

   Reportez les variations sur l'axe y.

6. Vous reprendrez les questions précédentes de cet exercice en variant la fréquence f (f = 20Hz par exemple). Qu'observez-vous ?
7. Puis vous modifierez le nombre de points (N = 512). Que devez-vous modifier ? Qu'observez-vous ?
8. Puis vous modifierez la fréquence d'échantillonnage (fe = 400). Qu'observez-vous ?

1.3 Repliement de spectre

1. Créez un signal numérique sinusoïdal de 128 Hz et vérifier son allure en le traçant. Vous prendrez pour fréquence d'échantillonnage 200 Hz, et construirez un vecteur de 256 échantillons.
2. Visualisez la transformée de Fourier de ce signal. Qu'observez-vous ?
3. Expliquez le phénomène observé.
4. Modifiez les paramètres pour vous retrouver dans l'hypothèse de Shanon-Nyquist.

1.4 TFD d'un signal combinaison linéaire de plusieurs sinusoides

1. En choisissant soigneusement la fréquence d'échantillonnage, créez le signal numérique suivant :
$$s(t) = 3.cos(50.\pi.t) + 10.sin(300.\pi.t) - sin(100.\pi.t) \tag{1}$$
2. Tracez son spectre en amplitude (en dB) et en phase. Vous pourrez vous inspirer du code fourni.
3. Qu'observez-vous ?