TP4: Analyse spectrale des signaux numériques.

Prof. HES Maurizio Tognolini
Institut d'Automation Industrielle (iAi) HEIG-VD

17 mai 2017 rev2.3

1 But

Ce TP débutera par le calcul de la transformé de Fourier d'un signal discret en appliquant la définition vue dans le cours que l'on comparera au résultat obtenus avec l'algorithme de calcul rapide (fft) de la transformée de Fourier. La deuxième partie concerne l'application de l'analyse spectrale d'un signal sonore dans le but de le filtrer dans le domaine fréquentiel.

2 Analyse spectrale d'un signal numérique.

Pour les signaux discrets de durée finie la transformé de Fourier peut être aproximé comme monté dans le cours par :

$$X(jf) = \frac{T_e}{N} \sum_{n=0}^{N-1} x[n] e^{-j2\pi f n T_e} \tag{1}$$

Pour l'exemple on va prendre un signal $x[n]$ simple et bien connu, l'impulsion rectangulaire centré en $n = 0$ d'amplitude $A = 1$ et de longueur d'impulsion $\Delta t = 11$ [s] (premier échantillon à $n = -40$ et dernier échantillon à $n = +39$ comme montré dans la figure suivante).
On prendra pour période d'échantillonnage $T_e = 1$ [s].
Ce signal discret est le résultat de l'échantillonnage du signal continu $x(t) = A \cdot rect(t/\Delta t)$

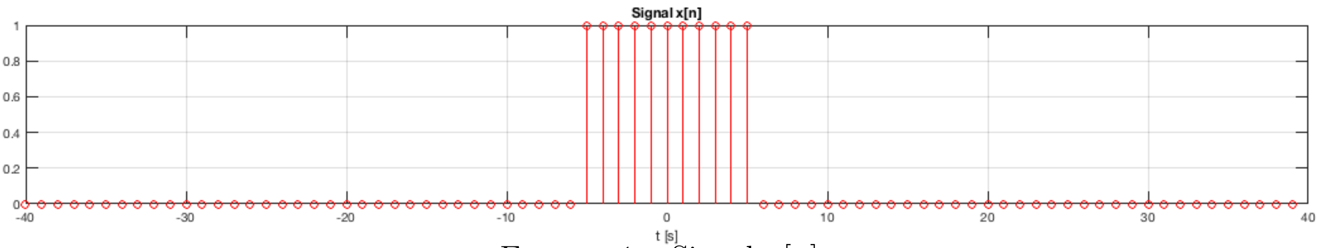


FIGURE 1 – Signal $x[n]$

La transformé de Fourier théorique du signal continu $x(t)$ est donné par les tables et dans le cas présent vaut :

$$X_{th}(jf) = \frac{A\Delta t}{N} sinc(f\Delta t) \tag{2}$$

Dans cette relation la variable f est continue.

La formule de la transformé de Fourier définie en (1) produit comme résultat un spectre continu pour toute valeur de fréquence f. Pratiquement pour calculer la transformé de Fourier comme défini en (1) il est nécessaire de définir le pas fréquentiel Δf à utiliser qui peut être arbitrairement petit, on aimerai voir les passages par zéro du spectre qui se trouvent à $Fz = 1/11$, il faut donc choisir un sous multiple entier de Fz . Pour l'exemple prenons $\Delta f = 1/44$

Pour calculer la formule (1) dans ce cas il est pratique de multiplier le vecteur ligne du signal $x[n]$ par un vecteur ligne $w_f[n] = \frac{T_e}{N} e^{-j2\pi f_1 n T_e}$ pour une fréquence particulière f_1 qui nous intéresse. Donc on aura :

$$X[jf_1] = x[n] \cdot w_{f1}^T[n] \tag{3}$$

Pour le calcul de tout le spectre il suffit alors de calculer (2) pour toutes les fréquences désirées de $[-Fe \cdots + Fe]$ par pas de Δf

Si vous affichez le module du spectre obtenu par la formule de la TFD (équation (3)) vous devez alors obtenir la figure suivante. Vous remarquerez que le spectre est périodique de période F_e, alors que le spectre théorique du signal continu, ne l'est pas.

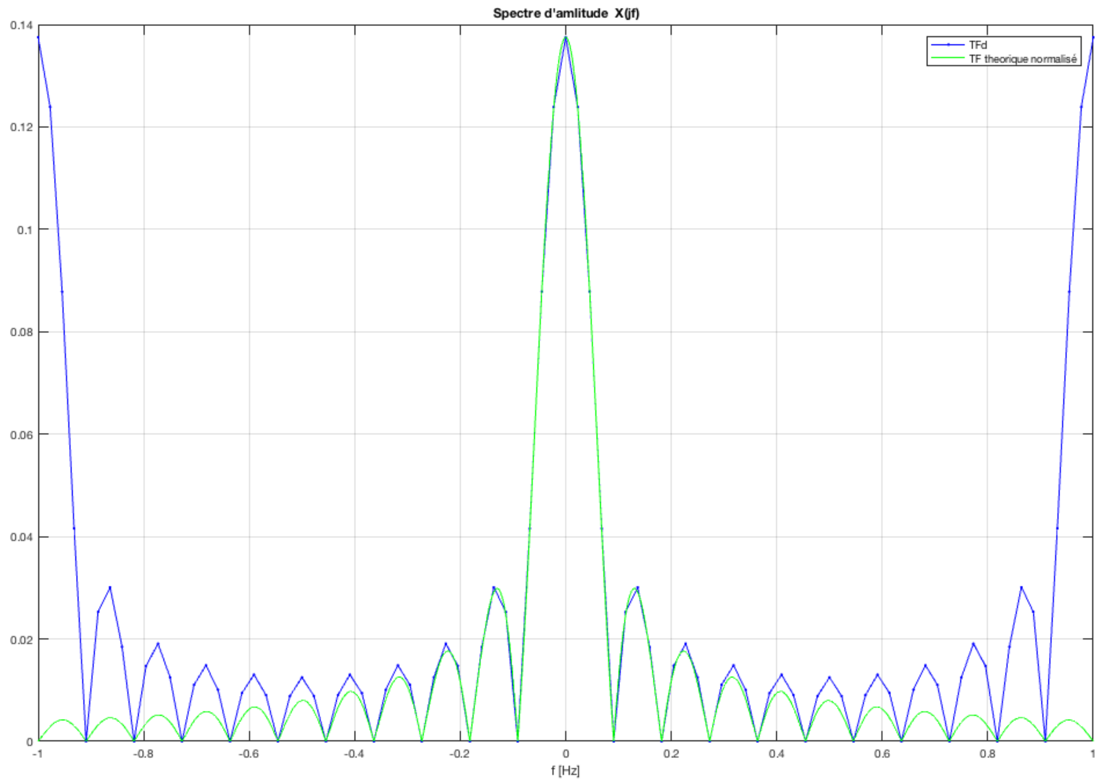


FIGURE 2 – Spectre d'amplitude $X[jf]$

La fonction `fft()` permet le calcul rapide de la transformé de Fourier mais avec la contrainte $\Delta f = Fe/N$ dans notre cas $\Delta f = 1/80 = 0.0125$ [Hz]. Entre autre le nombre de points du résultat est N et la plage de fréquence est $[-Fe/2 \cdots + F_e/2 - \Delta f]$. Sa définition est la suivante :

$$X_D[jk] = \sum_{n=0}^{N-1} x[n] e^{-j2\pi kn/N} \tag{4}$$

Où k est l'index de fréquences $f = k\Delta f$, n est l'index du temps $t = nT_E$ et N la longueur du signal. Les valeurs de X_D ne sont pas les mêmes que ceux de $X(jf)$, un facteur d'échelle doit être appliqué celui ci vaut $1/N$.

L'utilisation est la suivante avec Matlab ou Octave :

```
Xfft = fftshift(fft(xn));
df1 = Fe/N;
ff = -Fe/2:df1:Fe/2-df1;
```

Si vous comparez le spectre trouvé par la fft et celui calculé par l'équation (1) vous obtenez la figure 3. Comme vous pouvez le remarques les points ne sont pas placés aux mêmes fréquences, si bien que on n'a pas les passages par zéro du spectre avec la fft.

1. Comment faire pour que les passages par zéro du spectre calculé avec la fft coïncident exactement avec les passages par zéro du spectre théorique ?
2. Que se passe t'il si on allonge le signal $x[n]$ de part et d'autre avec des échantillons nuls ?
3. A quoi correspond la valeur $X(0)$?

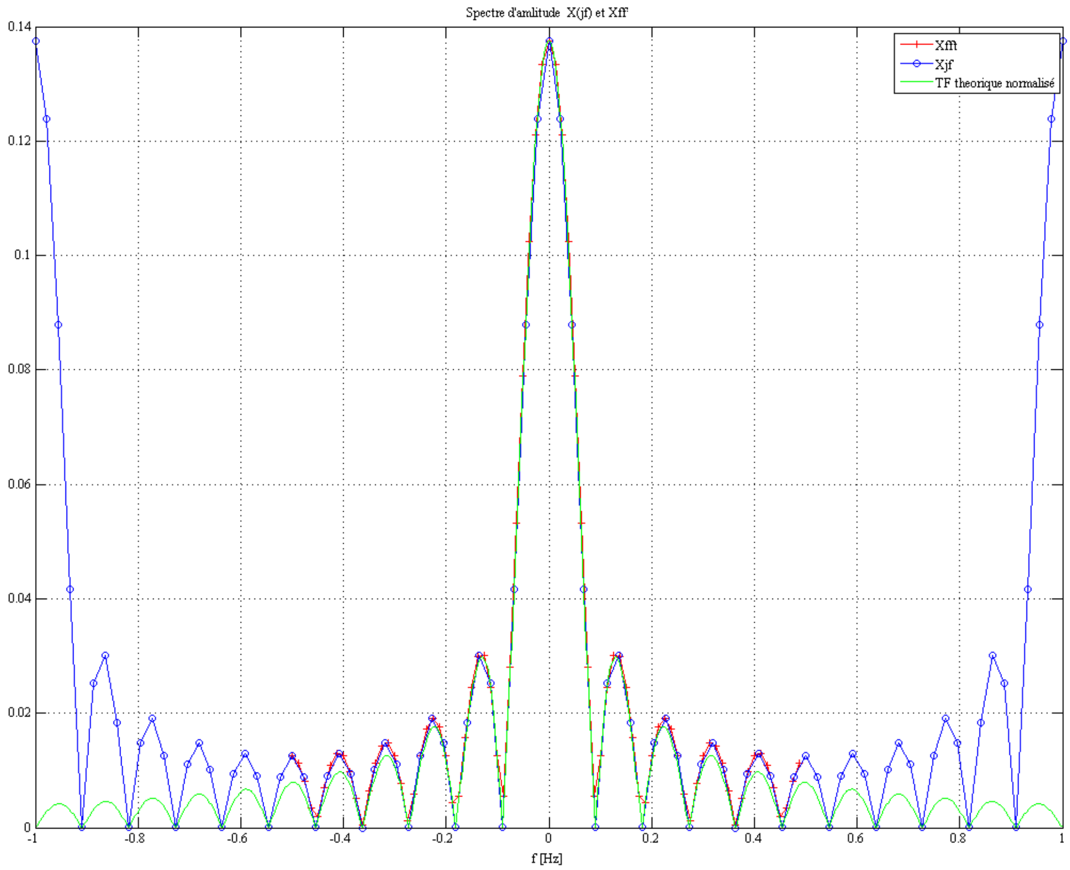


FIGURE 3 – Spectre d'amplitude $X[jf]$ calculé avec la fft (rouge) et avec la formule (1) (bleu), théorique (vert)

3 Filtrage dans le domaine fréquentiel

3.1 Signal carré

Créez un signal carré de fréquence $f_0 = 100\,Hz$ et de durée 1 seconde avec une fréquence d'échantillonnage $f_e = 1/T_e = 20\,kHz$:

```
N = duree * fe;
tt = (0:N-1) * Te;
xt = sign(sin(2*pi*tt*f0));
```

Puis :

1. Tracez 10 périodes du signal `xt` dans la fenêtre `subplot(2,2,1)`. Écoutez le signal $x(t)$ avec :
   ```
   soundsc(xt, fe);
   ```
2. Calculez son spectre `Xjf` et le vecteur fréquence `ff` avec les commandes :
   ```
   Xjf = fftshift(fft(xt)/N);
   df = fe/N;
   ff = -fe/2:df:fe/2 - df;
   ```
   A quoi sert l'instruction fftshift ?
3. Tracez son module dans la fenêtre `subplot(2,2,2)`.
4. Initialisez le spectre que vous allez filtrer :
   ```
   Xjf_f = Xjf;
   ```
   Modifiez ce spectre en annulant les composantes spectrales supérieures à $f_c = 500\,Hz$. La fonction `find` est très utile dans ce cas. Tracez le module de ce spectre dans la fenêtre `subplot(2,2,4)`.
5. Construisez le signal temporel correspondant `xt_f` avec les commandes :
   ```
   xt_f = real(ifft(fftshift(Xjf_f * N)));
   ```
   Tracez ce signal filtré dans `subplot(2,2,3)` et écoutez le résultat du filtrage.
6. Modifiez la fréquence de coupure ; écoutez et commentez.
7. Ouvrez une nouvelle figure et tracez les spectrogrammes des deux signaux :
   ```
   Nspec = 2048;   % longueur de la fenetre spectrale
   subplot(2,1,1); spectrogram(xt, hamming(Nspec),Nspec/2,N, fe,'yaxis');
   axis([0,duree,0,1000]);
   subplot(2,1,2); spectrogram(xt_f, hamming(Nspec),Nspec/2,N, fe,'yaxis');
   axis([0,duree,0,1000]);
   ```
   Observez et commentez.

3.2 Signal sonore

Ces notions d'analyse fréquentielle et de filtrage connues, vous allez les appliquer à un signal sonore créé par vos soins. Pour ce faire, enregistrez une phrase de courte durée (1 à 3 secondes) avec une fréquence d'échantillonnage $f_e = 20\,kHz$

```
recorder = audiorecorder(fe,16,1);
recordblocking(recorder,3);
xt = getaudiodata(recorder);
play(recorder);
```

Si vous n'avez pas la possibilité de enregistrer un son vous pouvez prendre le son `comment.wav` qui est un message de bienvenue. Si vous intéressez à l'analyse de sons de machine prenez le fichier `motor.wav` qui est un enregistrement d'un moteur à explosion. Pour lire les fichier audio utilisez :

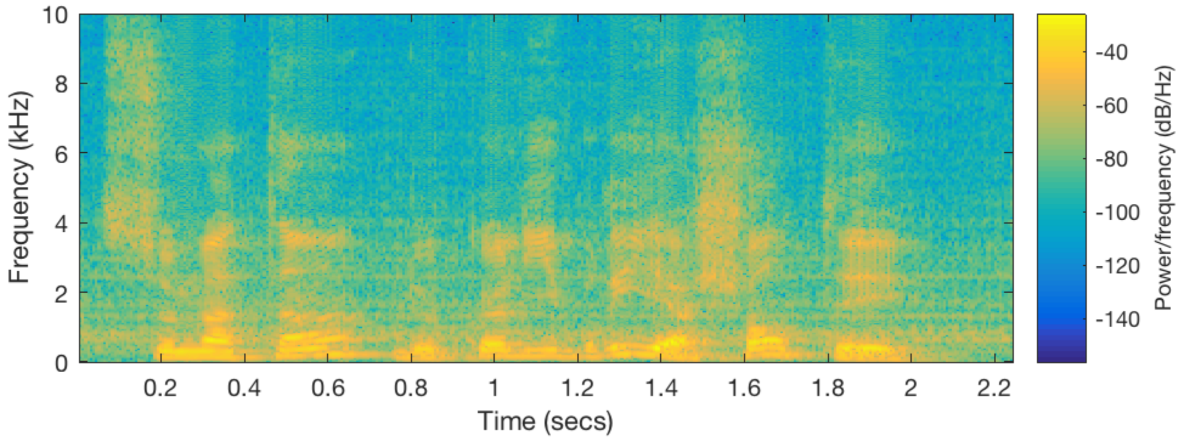
```
[y,Fs] = audioread('comment.wav');
```

Puis :

1. Tracez le signal (`plot(xt)`). Avec l'aide du zoom, relevez le domaine intéressant (`n1, n2`) du signal enregistré puis éliminez les parties inutiles et sauvegardez le signal dans un fichier `.wav` après l'avoir normalisé à 1 :

```
xt = xt(n1:n2);                        % partie utile
N = length(xt);                        % longueur du vecteur
xt = xt/max(abs(xt));                  % normalisation
audiowrite( 'phrase.wav',xt, fe);      % sauvegarde de la phrase
```

2. Calculez le vecteur temps `tt` et tracez `xt` dans la fenêtre `subplot(2,2,1)`. Écoutez le signal.
3. Répétez sur votre phrase les opérations 2) à 7) appliquées préalablement au signal carré. Dans le cas de ce signal sonore, vous réaliserez un filtrage passe-bande plutôt qu'un simple passe-bas en annulant les composantes spectrales qui n'appartienent pas au domaine des fréquences téléphoniques $f_i = 300\,Hz$, $f_s = 3.5\,kHz$.
4. Comme le message vocal n'est pas stationnaire (il varie constamment en amplitude et en fréquence), il est nécessaire d'écourter la fenêtre d'analyse du spectrogramme. Vous prendrez donc `Nspec = 256`.
5. Observez, analysez et commentez.



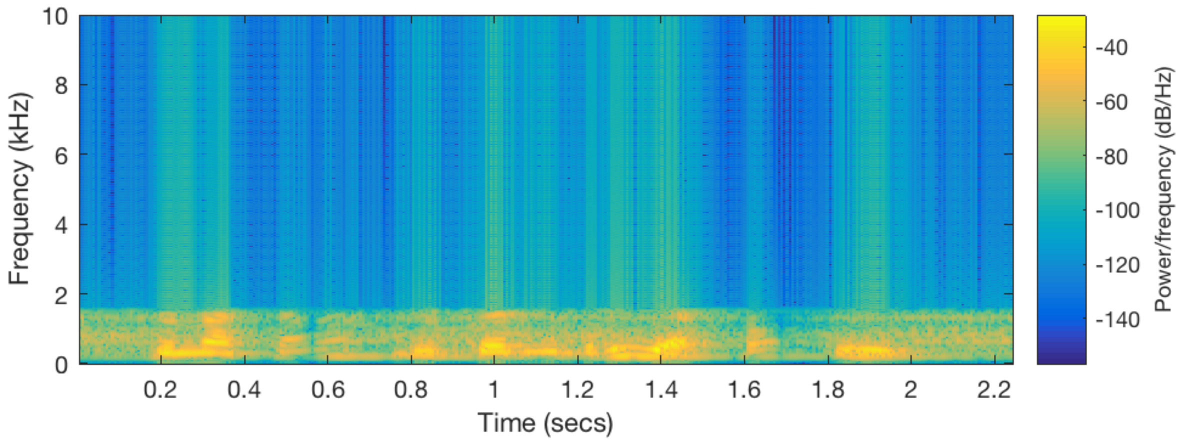


FIGURE 4 – Spectrogramme du signal audio non filtre (haut) et filtré (bas)

4 Dé-bruitage par reconstruction d'un signal périodique

Un capteur fournit un signal fortement bruité, on sait que ce signal contient un signal périodique à $f_0 = 50Hz$ (fréquence fondamentale) et plusieurs harmoniques. L'échantillonnage du signal analogique se fait à une fréquence $f_e = 5$ [KHz]. La durée de l'enregistrement est de $T = 510$ [ms]. Le signal est montré dans la figure ci dessous.

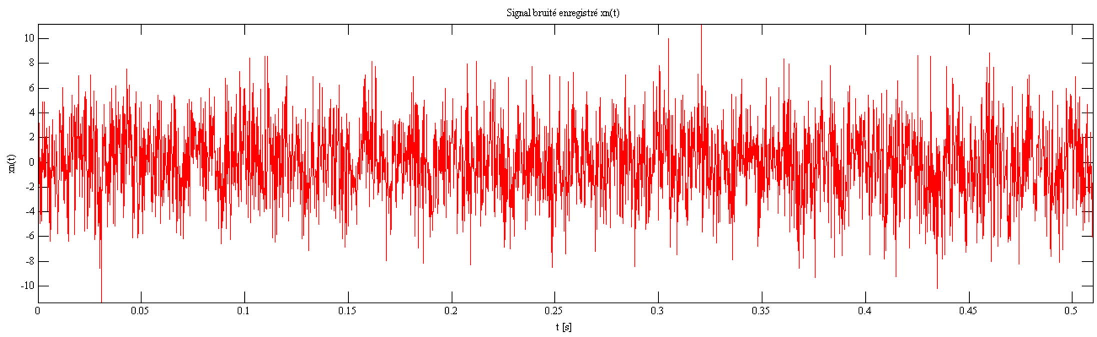


FIGURE 5 – Signal enregistré comprenant le signal périodique et le bruit de mesure

Le but de cette expérience est de retrouver le signal périodique sans bruit, sans utiliser les techniques de filtrage temporel ni fréquentiel vu précédemment. La technique que l'on va employer consiste à faire une analyse spectrale du signal pour retrouver les amplitudes et les phases de chacune des harmoniques, et ensuite reconstituer le signal périodique par synthèse numérique.

Si on utilise sans précaution la TFD on remarque plusieurs pics qui ne correspondent pas exactement aux fréquences recherchés comme montré dans la figure ci dessous :

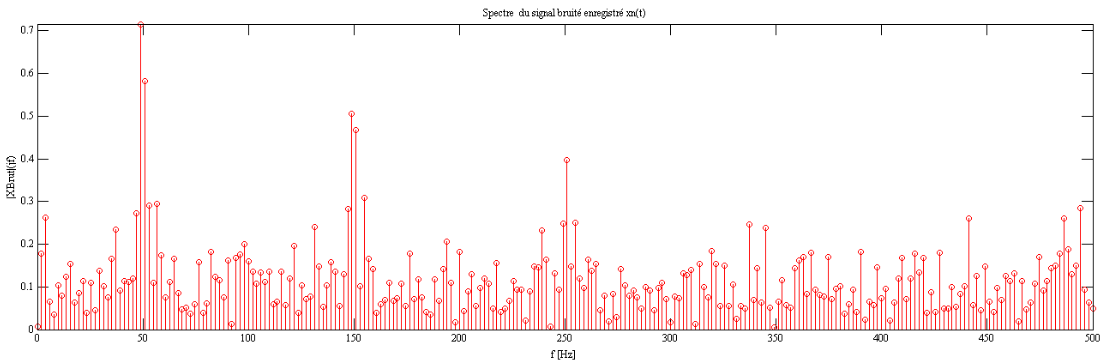


FIGURE 6 – Spectre du signal bruité : les pics ne sont pas exactement aux fréquences voulues !

Le travail à réaliser est le suivant :

1. A l'aide de la FFT analysez le signal, veillez à choisir un pas fréquentiel Δf en rapport au fréquences du signal périodique dans le but de retrouver dans le spectre les 3 pics correspondants. Affichez le spectre unilatéral du module et de la phase, et détectez automatiquement les composantes de chaque harmonique.
2. A partir du module et de la phase de chaque harmonique reconstruisez le signal non bruité et affichez le en superposition avec le signal bruité.
3. Calculez la valeur efficace du signal, celle du bruit (signal bruité - signal non bruité), et calculez le rapport signal sur bruit en [dB].