Signaux et systèmes électroniques
$2^{\text{ème}}$ année

Introduction pratique à l'analyse de Fourier

Daniel Oberson

24 septembre 2018

Version 1.0

1 Introduction

En raison de ses remarquables propriétés, la sinusoïde occupe une place importante dans la modélisation des signaux et des systèmes linéaires. La sinusoïde est une valeur propre des systèmes linéaires : elle fait partie de leur réponses naturelles et transitoires. De plus, la sinusoïde est une fonction « isomorphe » : la somme, la dérivée, l'intégrale d'une sinusoïde restent une sinusoïde et la réponse d'un système linéaire à une sinusoïde permanente est une sinusoïde permanente d'amplitude et de phase modifiée. Enfin, la modélisation de la sinusoïde à l'aide de l'exponentielle complexe permet de transformer les équations différentielles des systèmes linéaires en équations algébriques complexes, simplifiant ainsi considérablement le calcul de la réponse des systèmes en régime sinusoïdal.

L'ensemble des techniques d'analyse et de modélisation des signaux et des systèmes en régime sinusoïdal portent le nom de Fourier.

Jean-Baptiste Joseph Fourier

Mathématicien et physicien français, né à Auxerre en 1768, mort à Paris en 1830. Il est connu pour ses travaux sur la décomposition des fonctions périodiques en séries trigonométriques convergentes appelées séries de Fourier.

Une manière intuitive de voir ce qu'est une analyse fréquentielle par Fourier est représentée par la Figure 1. En effet, un signal est représenté dans le temps, mais peut également être représenté et caractérisé du point de vue fréquentiel.

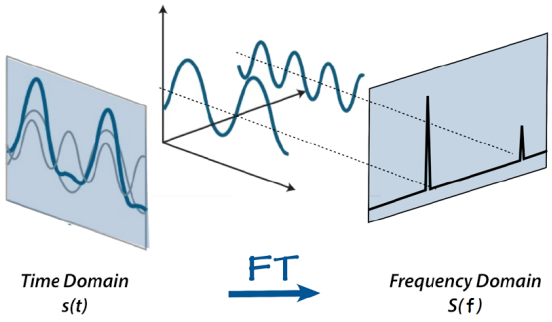

FIGURE 1 – Représentation temporelle vs fréquentielle (Fourier).

2 Analyse de Fourier

Deux domaines d'analyse fréquentielle peuvent être associés aux analyses de Fourier, soient le domaine des signaux continus et celui des signaux discrets. De plus deux analyses de Fourier existent soit la série de Fourier pour les signaux périodiques et la transformées de Fourier pour les signaux non-périodiques. En résumé, il y a donc 4 formes d'analyse fréquentielle présentées dans le tableau récapitulatif ci-dessous.

Signal	Périodiques	Non-périodiques
Continu t	Série de Fourier $x(t) = \sum_{k=-\infty}^{+\infty} X[k]\ e^{jk2\pi f_0 t}$ $X[k] = \frac{1}{T}\int_0^T x(t)\ e^{-jk2\pi f_0 t} dt$	Transformée de Fourier $x(t) = \int_{-\infty}^{+\infty} X(f)\ e^{j2\pi ft} df$ $X(f) = \int_{-\infty}^{+\infty} x(t)\ e^{-j2\pi ft} dt$
Discret n	Série de Fourier discrète $x[n] = \sum_{k=0}^{N-1} X[k]\ e^{jk2\pi \frac{1}{N} n}$ $X[k] = \frac{1}{N}\sum_{n=0}^{N-1} x[n]\ e^{-jk2\pi \frac{1}{N} n}$	Transformée de Fourier discrète $x[n] = \int_{-1/2}^{1/2} X(f/f_e)\ e^{j2\pi \frac{f}{f_e} n} df$ $X(f/f_e) = \sum_{n=-\infty}^{+\infty} x[n]\ e^{-j2\pi \frac{f}{f_e} n}$
	où f_0=fréquence fondamentale N=période	où f_e=fréquence d'échantillonnage

TABLE 1 – Récapitulatif de Fourier.

Le domaine des signaux continus permet de couvrir tous les phénomènes physiques classique alors que le domaine des signaux discrets a été initialement exploré pour la physique quantique puis largement utilisé pour le traitement de signal par informatique. Il est à noter que les valeurs de $X[k]$, $X(f)$ ou $X(f/f_e)$ sont complexes et sont généralement représentées par deux graphiques distincts, soit leur module et leur phase en fonction de k, f ou f/f_e.

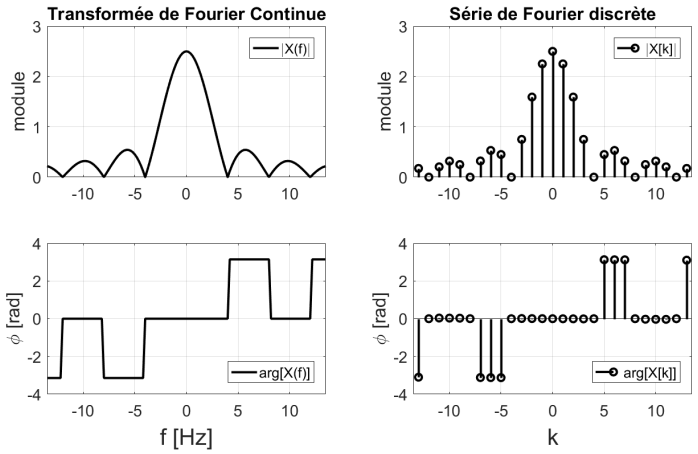

FIGURE 2 – Exemple de spectres de la série $X[k]$ et de la transformée $X(f)$ de Fourier.

Les amplitudes des fréquences présentent dans le signal sont représentées en Figure 2 correspondant à un axe des fréquences ayant des valeur positives et négatives. Bien que que ce soit contre intuitif, les fréquences négatives ont un sens réel en analyse fréquentiel et traitement de signal. Néanmoins, pour cette introduction à l'analyse fréquentielle, aucune interprétations fine sera faite sur ces fréquences négatives. Il suffit de s'accorder avec le faite qu'une composante fréquentielle est "divisée" en deux parties, soit une partie positive et une partie négative. La phase ϕ est également représentée en fonction des fréquences.

Etant donné que les analyses fréquentielles sont utilisées dans tous les domaines de la science et de la technique, de multiple recherches ont été effectuées afin d'optimiser ces calculs. L'algorithme qui est ressorti de toutes ces recherches est la *FFT* pour *Fast Fourier Transform*. Bien que la traduction littérale soit la *Transformée* de Fourier Rapide, c'est en fait le calcul d'une *série* de Fourier discrète. Malgré cet abus de langage mondialement généralisé et moyennant quelques connaissances ainsi que des ajouts exposés par la suite, il est tout a fait possible de ressortir les informations fréquentielle utiles de cet algorithme appelé *FFT*.

3 Implémentation dans Matlab

L'algorithme de la *FFT* étant dans bientôt toutes les *library* des processeurs, il est donc évident qu'il est également implémentée dans les applications de calcul de haut niveau telles que Matlab. Néanmoins, il est important pour l'utilisateur de bien être au fait du retour de ces fonctions pré-implémentées avant de les mettre en oeuvre. En effet, il n'est pas rare qu'une mauvaise interprétation de ces fonctions *FFT* soit faite et amène des problèmes dans les développements. Deux choses sont spécifiquement importantes à regarder de près dans la documentation associée et doivent être contrôlés par quelques tests préliminaires :

- le format des paramètres d'entrée et de sortie de la fonction
- contrôler l'équation implémentée [1]

3.1 Mise en oeuvre de la FFT

En regardant donc dans la documentation fournie par Matlab et après quelques tests, ces deux points doivent être parfaitement maîtrisés. En considérant le cas de l'utilisation la plus simple de la fonction *FFT*, soit le calcul de la *FFT* d'un signal numérisé, il ressort que le paramètre d'entrée est un simple vecteur et que le paramètre retourné est un vecteur colonne de valeurs complexes de longueur identique au vecteur d'entrée. Correspond à chaque valeur du résultat de la *FFT* une fréquence normalisée à $0 \leq {}^{f}/_{f_e} \leq 1 - {}^{1}/_{N}$. La première valeur correspondant à la composante continue et la dernière proche de la fréquence d'échantillonnage.

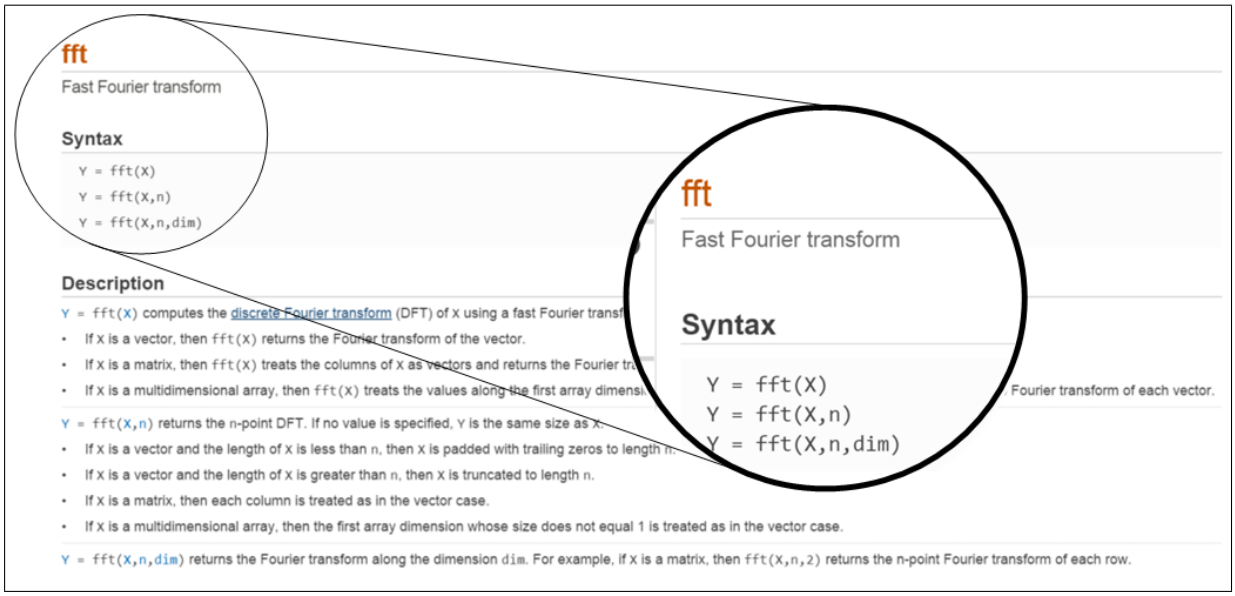

FIGURE 3 – Documentation Matlab sur la fonction *FFT*, paramètres d'entrée/sortie.

1. Si l'algorithme et le calcul de fond est toujours le même, quelques facteurs d'échelles peuvent exister.

De plus, le calcul implémenté dans la fonction *FFT* est donné (Figure 4). En comparant ce calcul avec celui de la série de Fourier discrète du tableau 1, il ressort que l'implémentation du calcul de la *FFT* n'est pas totalement identique. En effet, la normalisation $\frac{1}{N}$ est absente dans Matlab.

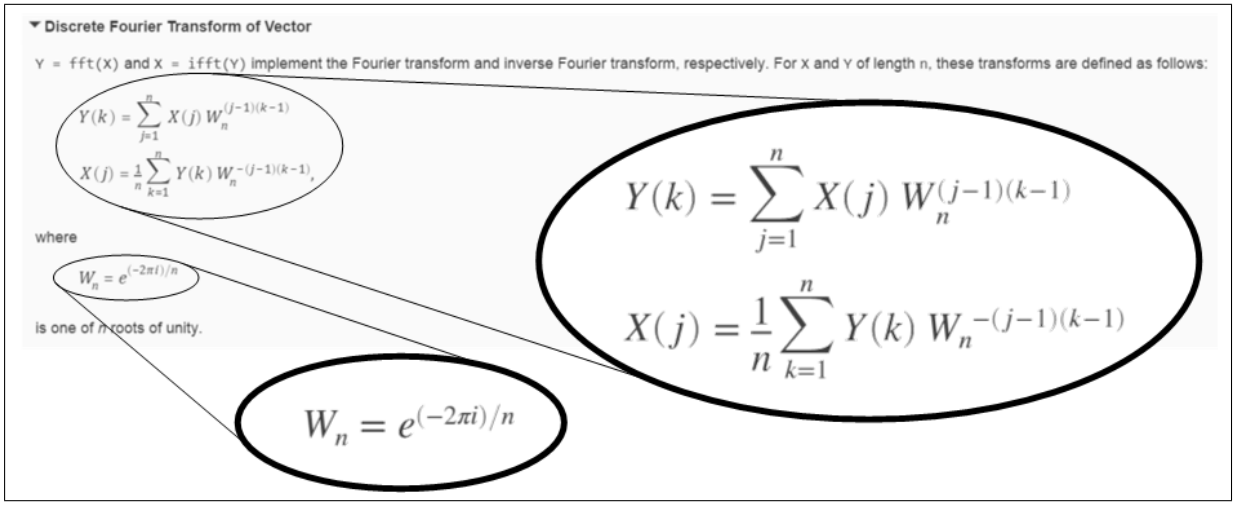

FIGURE 4 – Documentation Matlab, fonctions *FFT* et *iFFT* implémentées.

Afin d'avoir les amplitudes qui correspondent, il faut donc ajouter la normalisation $\frac{1}{N}$ sur le résultat obtenu par la *FFT* de Matlab. Il ne faut également pas oublier que le spectre de l'amplitude est le module [2] des valeurs de la *FFT* en fonction de l'axe des fréquences. Pour représenter totalement la *FFT*, il faut donc ajouter un graphique avec la phase qui correspond cette fois à l'argument [3] des valeurs de la *FFT* en fonction de l'axe des fréquences.

Il reste encore à déterminer correctement l'axe des fréquences afin de pouvoir interpréter aisément les résultats obtenus. Comme cité précédemment, à chaque valeur du vecteur de sortie de la fonction *FFT* correspond une fréquence normalisée par la fréquence d'échantillonnage. Cela implique une représentation graphique avec des abscisses de $f = 0$ à $f = f_e$ qui n'est généralement pas la représentation choisie. De fait, la représentation la plus usuelle est de $-f_e/2$ à $+f_e/2$. Pour ce faire, Matlab offre la fonction *fftshift()* permettant cette représentation préférée (Figure 6).

```
%% Calcul de la FFT
fftx1=fftshift(1/N*fft(x1));
```

FIGURE 5 – Utilisation de la fonction *fftshift()* et normalisation des amplitudes.

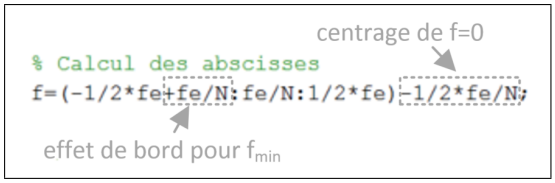

FIGURE 6 – Calcul des abscisses avec correction.

Le calcul des valeurs des abscisses se fait en créant un vecteur de taille N, N étant la longueur de la *FFT* de $-f_e/2$ à $+f_e/2$ en faisant attention aux effets de bords ainsi qu'au centrage de la composante continue où $f = 0$. Un exemple de calcul des abscisses dans Matlab est montré dans la figure 6.

2. fonction abs() dans Matlab.
3. fonction angle() dans Matlab.

L'affichage peut se faire alors dans Matlab avec la fonction *plot()*.

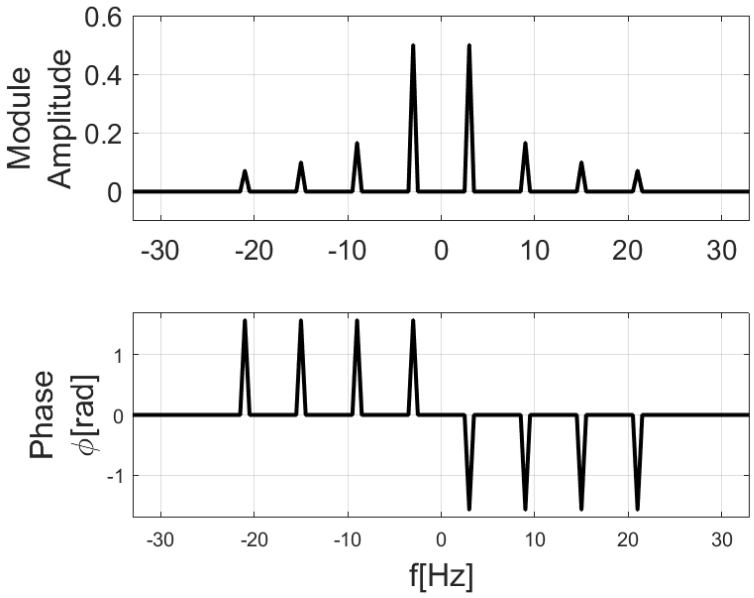

FIGURE 7 – Exemple d'*FFT* avec les abscisses en fréquence.

3.2 Fenêtrage

L'exemple de la figure 7 fonctionne parfaitement car le signal de base est un signal périodique synchronisé à la fenêtre d'acquisition, soit qu'il y a un multiple entier de période qui est analysé. Mais dans la plus grande majorité des cas, le signal n'est pas du tout synchronisé et le signal analysé est constitué d'un multiple non-entier de période. Cette désynchronisation amène une déformation du résultat de la *FFT* qui doit être corrigée.

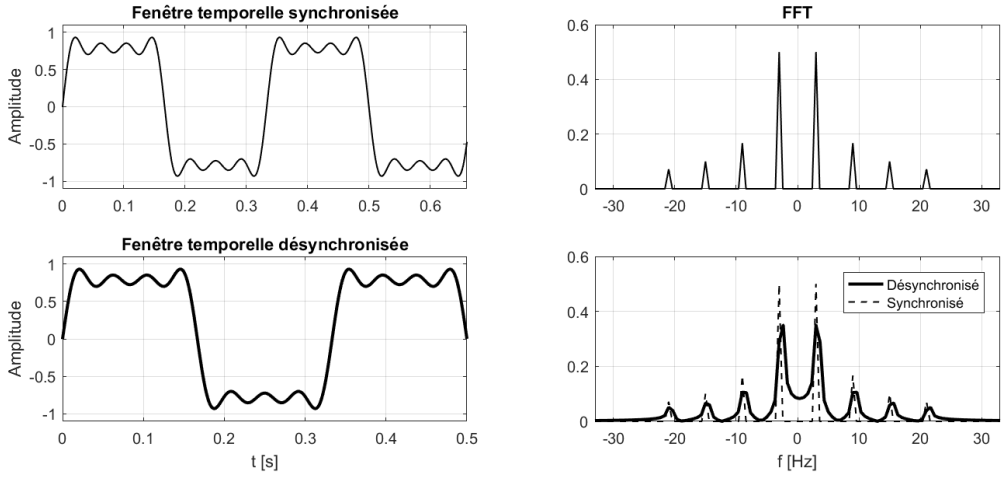

FIGURE 8 – Fenêtre temporelle synchronisée vs désynchronisée.

La correction faite à ce problème est d'appliquer une fenêtre pondérée à toutes les valeurs du signal analysé de base. Sans entrer dans les détails de la théorie sur le fenêtrage pour les analyses fréquentielles, dans Matlab, il est possible d'appliquer la fenêtre *flat-top* qui permet de faire une analyse fréquentielle correcte dans la plupart des cas. Pour ce faire, la fonction *flattopwin()* de Matlab permet de faire un vecteur de pondération de chaque échantillon de base du signal. Un exemple de l'utilisation de cette fonction est faite dans la Figure 9.

```
% Avec Fenêtre Flat-top
w=flattopwin(N);
fftxw1=fftshift(1/sum(w)*fft(x1.*w'));
```

FIGURE 9 – Calcul de la *FFT* avec fenêtre de pondération *flat-top*.

Le résultat de cette pondération est une nette perte de résolution fréquentielle comme on peut le constater aisément par la Figure 10. En effet, les "pointes" des fréquences présentes dans le spectre résultat de la *FFT* sont épaissies en fréquence. Cette perte de résolution en fréquence est très souvent un moindre mal par rapport au fait que les amplitudes ont des valeurs correspondant réellement au signal de base.

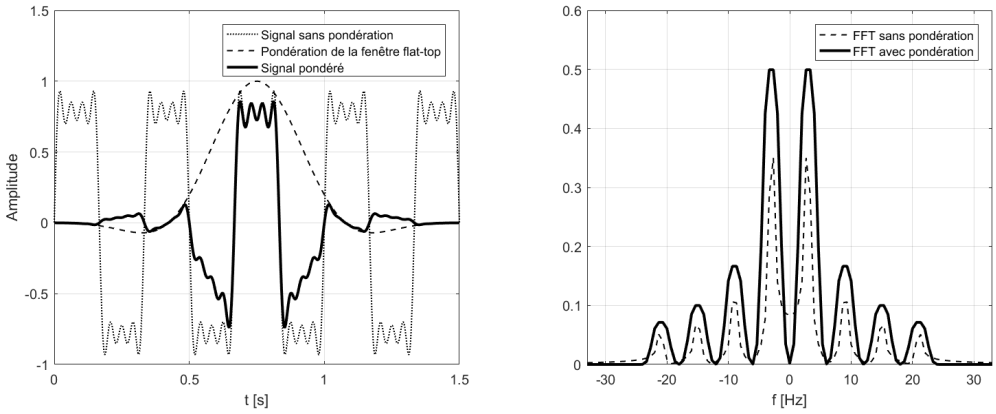

FIGURE 10 – Calcul de la *FFT* avec et sans fenêtre de pondération *flat-top*.

En plus de la *flat-top*, plusieurs fenêtres existent (*Blackman-Harris*, *Von Hann*, *Hamming*, etc.) et sont souvent disponibles sur les analyseur de spectre. La fenêtre *Uniform* étant l'option généralement donnée pour ne pas utiliser de fenêtre, ce qui peut être utile dans le cas d'un signal que l'on peut synchroniser à l'acquisition.