

M1 info : Bases du Traitement du Signal

TP : Transformée de Fourier Discrète

A la fin de ce TP, vous devez rendre un compte-rendu sur une copie identifiée par votre login de manière visible (pas sous le rabat). Vous aurez par ailleurs 4 figures à sauvegarder (partie 3). La récupération de ces fichiers se fera de manière automatique à la déconnexion.

Ce TP utilise scilab, un logiciel libre de simulation numérique (voir <http://www.scilab.org>). Pour le lancer, sous linux, placez-vous dans votre répertoire de travail et tapez "scilab &" dans une fenêtre terminal. On peut soit taper directement les instructions dans la fenêtre de commande qui apparaît (*scilex*), soit exécuter un script stocké dans un fichier toto.sce. Pour cela, ouvrez ce script dans l'éditeur intégré et faites *exécuter / charger dans scilab*. On peut afficher l'aide en ligne sur une fonction par `help` suivi du nom de la fonction.

Scilab est un langage vectoriel : les données sont représentées par des vecteurs ou des matrices (tableaux à 1 ou 2 dimensions, respectivement) qui sont traités en bloc (pas de boucle échantillon par échantillon). Ainsi, pour multiplier un vecteur ligne x par une fenêtre de Hamming de longueur N , on crée `w = window('hm', N)` ; (le point-virgule évite l'affichage des échantillons de w), puis on effectue `x.*w` (multiplication terme à terme des deux vecteurs).

Les fenêtres rectangulaires, triangulaires, de Hamming et de Hanning sont créées par l'instruction `w = window(type, N)` ; où `type` vaut respectivement 're', 'tr', 'hm' et 'hn'. Pour une fenêtre de Blackman, charger d'abord la fonction `blackman` (`getf("blackman.sci") ;`) puis faire `w = blackman(N)` ;.

Scilab calcule les TFD par FFT (sur un nombre de points puissance de 2). Pour une séquence temporelle x , l'instruction est `fft(x, -1)` pour une TFD, `fft(x, 1)` pour une TFD inverse. La TFD qui résulte de l'instruction `fft` est représentée sur l'intervalle des fréquences normalisées $[0; 1]$. Pour obtenir la représentation habituelle sur $[-1/2; 1/2]$, il faut faire `X=fftshift(X)`.

1 Spectre d'une fenêtre triangulaire

Ouvrez TP_1.sce. Ce programme génère une fenêtre triangulaire de longueur 32, calcule sa TFD et affiche dans la même fenêtre graphique le signal et son spectre d'amplitude en dB.

1) Sachant que la TFD consiste en un échantillonnage du spectre, expliquer pourquoi le spectre obtenu ressemble peu au spectre d'une fenêtre triangulaire.

2) Décommenter la dernière partie du programme (*édition/décommenter*). Le signal `tri2` est composé de la fenêtre triangle précédente complétée par des zéros de manière à porter la durée totale à 512 (zero-padding). Quel est l'intérêt de ceci ? Vérifier sur la figure que l'on retrouve bien les caractéristiques théoriques du spectre d'une fenêtre triangulaire.

Dans cette figure, les échantillons ne sont pas marqués par un cercle mais sont reliés entre eux par une ligne continue. Il en est ainsi dans la suite du TP pour toutes les figures.

2 Analyse spectrale d'un signal sinusoïdal

1) Ouvrez TP_2a.sce. Ce programme scilab génère $N = 32$ échantillons d'un signal sinusoïdal de fréquence 6 kHz échantillonné à 16 kHz, multiplie par une fenêtre triangulaire, calcule la FFT de la séquence ainsi pondérée et représente dans la même fenêtre d'une part les 20 premiers échantillons du signal, d'autre part le spectre d'amplitude de ce signal.

- Quel est le spectre théorique du signal ?
- Après avoir analysé le programme, lancez-le et commentez les figures : pourquoi limite-t-on l'axe des fréquences à l'intervalle ± 8 kHz ? Pourquoi la sinusoïde échantillonnée ressemble-t-elle si peu à une sinusoïde ? En quoi et pourquoi le spectre observé diffère-t-il du spectre théorique ?
- Pourquoi n'observe-t-on pas des lobes ?

2) Ouvrez TP_2b.sce. Ce programme est le même que le précédent, à ceci près que le spectre est en valeurs naturelles (et non plus en dB). Lancez-le. Changez la fréquence d'échantillonnage : 64 kHz au lieu de 16. Augmentez le nombre d'échantillons du signal : $N = 128$. Relancez le programme avec un nouveau numéro de fenêtre graphique (`xset("window", 1)`) pour ne pas effacer la figure précédente. La finesse de l'analyse spectrale est-elle améliorée ? Pourquoi ?

3 Analyse spectrale de signaux harmoniques

1) Ouvrez TP_3.sce. Ce programme génère et analyse un signal x composé de 2 sinusoïdes de fréquences respectives 2000 et 2260 Hz échantillonnées à 16 kHz. On utilise un fenêtrage triangulaire pour l'analyse spectrale. Observez le spectre pour $N = 64$ puis $N = 512$ puis interprétez en calculant la résolution fréquentielle dans les deux cas. Calculer puis tester la valeur minimale de N assurant une résolution fréquentielle suffisante. On appelle N_0 cette valeur. Exporter votre figure au format postscript sous le nom fig_3_1.ps : `file / export`, puis `format : Postscript, Filename : fig_3_1` (sans l'extension).

2) Ajouter à x une troisième sinusoïde, de fréquence 2530 Hz et d'amplitude 0,01. La résolution fréquentielle est-elle toujours suffisante avec N_0 ? Observez le spectre du nouveau signal. Pourquoi ne voit-on pas de pic à 2530 Hz ? Proposez et testez une solution. Exporter votre figure au format postscript sous le nom fig_3_2.ps.

3) Le signal doit être maintenant chargé à partir d'un fichier son : `x=wavread('FSKL0_ask.wav')` ;, qui est un enregistrement du mot "ask". On en prélève N échantillons pour l'analyse : `x=x(1 :N) ;`. Il s'agit du début du [a] prononcé avec une fréquence fondamentale de 270 Hz. Ce signal est donc composé d'harmoniques dont les fréquences respectives sont espacées de 270 Hz, au moins dans sa partie basse fréquence (au delà de 4 kHz, le spectre n'est pas purement harmonique). Les écarts d'amplitude entre les harmoniques sont *a priori* du même ordre qu'à la question précédente.

D'après les résultats précédents, quel fenêtrage (type et valeur minimale de N) permet d'observer la structure harmonique du spectre ? Testez votre proposition. Il pourra être nécessaire d'adapter le cadre de la figure, indiqué par le vecteur $[x_{min} y_{min} x_{max} y_{max}]$ dans l'instruction `plot2d`. Exporter votre figure au format postscript sous le nom fig_3_3a.ps.

Observez x . Dans votre programme, augmentez N jusqu'à la plus forte puissance de 2 possible et observez le spectre. Exporter votre figure sous le nom fig_3_3b.ps. Interprétez.