

TD5

Traitement Numérique du Signal avec Matlab (Signal Processing Toolbox)

Objectif :

Savoir échantillonner un signal et élaborer son spectre par FFT.

Savoir analyser et filtrer un signal numérique à partir des méthodes classiques.

A) Echantillonnage d'un signal porte

Soit $x(t)$ le signal porte d'amplitude 1 de largeur 1seconde (-0,5s à +0,5s).

1. Détermination de la fréquence d'échantillonnage F_e en appliquant Shannon.

2. Elaborer le signal numérique $x(n)$ obtenu par échantillonnage de $x(t)$.

3. Calculer le spectre de $x(n)$ par et tracer son spectre. Comparer par superposition avec celui obtenu en continu. Expliquer les différences (repliement de spectre, effet de fenêtrage).

Prendre le soin de paramétrer correctement le programme pour pouvoir changer facilement la fréquence d'échantillonnage et donc le nombre de point de $x(n)$ pour montrer l'influence de F_e sur le spectre (à durée constante) ainsi que l'effet du zéro-padding (augmentation de la durée du signal par injection de valeurs nulles avant et après) à F_e constante.

B) Traitement numérique d'un signal par filtrage IIR et FIR

1. **Importer de Arche** dans Matlab le fichier suivant : **mesure.dat**

Ce fichier inclut le vecteur appelé **mesure** de taille 2x1024.

La première ligne donne les temps, la seconde les valeurs correspondantes du signal échantillonné.

2. Analyse du signal :

Analyser le signal par FFT. Faire apparaître les différentes composantes en explicitant les fréquences et les amplitudes réelles des différentes composantes ainsi. Essayer de caractériser le bruit.

3. Filtrage du signal :

Filtrer le signal afin d'extraire la composante de plus haute fréquence. Deux méthodes sont à mettre en œuvre : l'une procédant par filtre RII et l'autre par filtre RIF.

Technique RII :

Principe : passage continu-numérique par transformation bilinéaire (Tustin). On prend un filtre de type Butterworth.

Commande Matlab : **butter** (Butterworth digital filter design).

Technique RIF :

Principe : technique dite par fenêtrage et troncature. On spécifie la réponse fréquentielle désirée $H(e^{j\omega})$ comme étant celle d'un filtre idéal (gain unitaire dans la BP, nul ailleurs). Par définition cette fonction est continue et périodique, on calcule par Fourier inverse les coefficients $h(n)$, qui sont identiques à ceux de la série (infinie!) de Fourier. Enfin, troncature et régularisation sur N points par une fonction fenêtre $w(n)$ (type Hamming par défaut) tel que $b(n)=h(n).w(n)$.

Noter que sous la forme RIF, les coefficients h_n sont les mêmes que ceux du filtre $H(z)$.

Commande Matlab : **fir1** (finite impulse response filter design).

4. Travail demandé :

Mise en œuvre de ce qui est indiqué ci-dessus. Etude des filtres obtenus, de leur réponse impulsionnelle, de leur transmittance en z . Etude de l'influence des paramètres de réglage (ordre, bande passante, retard) du filtre sur le résultat final (observer les régimes transitoires et établis).

Un compte rendu est exigé.

Autres commandes Matlab utiles

stem : équivaut à un plot avec bâtonnet (évite l'interpolation entre points)

fft : 1D fast Fourier transform

fftshift : rearrange the outputs of fft

filter : digital filtering

freqz : frequency response of digital filters

impz ou **dimpulse** : impulse response of digital filters