Traitement Numérique du Signal
Département Sciences du Numérique - Première année
TP 1
Génération et étude d'un signal numérique : le cosinus

Les deux premières séances de TP seront consacrées à la mise en place des outils nécessaires à la réalisation du projet et à la bonne compréhension de leur utilisation. Aucun compte-rendu ne vous sera demandé sur ces parties là.

1 Représentation temporelle

Un signal numérique, x, est représenté sous Matlab par un tableau contenant un nombre fini, N, de valeurs représentant des échantillons de signal prélevés toutes les T_e secondes (échantillonnage uniforme) : $[x(0)\ x(1)...\ x(N-1)]$, le $k^{\text{ième}}$ élément $x(k)$ représentant en réalité $x(kT_e)$.

1. Générer 90 échantillons d'un cosinus (fonction *cos.m* sous matlab), d'amplitude 1 (V), de fréquence $f_0 = 1100$ Hz et échantillonné à $F_e = 10000$ Hz.
2. En utilisant la fonction *plot.m* de Matlab, tracer le cosinus généré précédemment avec une échelle temporelle en secondes. On doit pouvoir retrouver, à partir du tracé, la fréquence et l'amplitude du cosinus. Pensez à ajouter des labels sur vos axes en utilisant les fonctions *xlabel.m* et *ylabel.m* de Matlab, éventuellement un titre à votre figure en utilisant la fonction *title.m*.
3. Générer 90 échantillons d'un cosinus (fonction *cos.m* sous matlab), d'amplitude 1 (V), de fréquence $f_0 = 1100$ Hz et échantillonné à $F_e = 1000$ Hz.
4. En utilisant la fonction *plot.m* de Matlab, tracer le cosinus généré précédemment avec une échelle temporelle en secondes. La fréquence mesurée sur le cosinus tracé n'est pas $f_0 = 1100$ Hz. Vous devez être capables d'expliquer pourquoi et d'où vient la valeur de la fréquence observée.

2 Représentation fréquentielle

La transformée de Fourier numérique du signal x devra être estimée à partir du tableau de points représentant le signal numérique. Elle sera, elle même, représentée par un tableau, contenant un nombre fini de valeurs, représentant des échantillons de la TFD du signal calculés avec un certain pas. On obtient, en effet, sous Matlab, en utilisant la fonction *fft.m* sur le signal x, un tableau : $[X(0)\ X(1)...\ X(N-1)]$, le $n^{\text{ième}}$ élément $X(n)$ représentant en réalité $X(n\Delta f)$, si Δf désigne le pas de calcul de la TFD. En considèrant que l'on calcule N points pour la TFD entre 0 et F_e, le $n^{\text{ième}}$ élément $X(n)$ du tableau représentant la TFD du signal x est alors $X\left(n\frac{F_e}{N}\right)$.

1. Vous devez être capable de répondre à la question suivante : qu'est-ce qui peut justifier que l'on calcule la transformée de Fourier numérique entre 0 et F_e ?
2. Pour chaque cosinus généré précédemment, estimez sa transformée de Fourier, en utilisant la fonction *fft.m* de Matlab, et tracez en le module (fonction *abs.m* de Matlab), en échelle log grâce à la fonction *semilogy.m*. L'échelle fréquentielle devra être en Hz. Vous devez être capables d'expliquer le tracé obtenu : retrouvez-vous les fréquences des cosinus générés ?
3. Choisissez un des cosinus générés précédemment et estimez sa transformée de Fourier sur un nombre de points, N', supérieur au nombre de points de signal, en utilisant la technique du zero padding : $X = fft(x, N')$. Tracez le module de la transformée de Fourier obtenue (en échelle log avec une échelle fréquentielle en Hz) pour plusieurs valeurs de N'. En comparant les résultats obtenus, vous devez être capable d'en déduire l'intérêt de la technique dite du Zero Padding. Attention afin d'utiliser l'algorithme de calcul rapide de la TFD (FFT) on devra utiliser un nombre de points N' égal à une puissance de 2.
4. Un autre outil permet de visualiser la représentation fréquentielle d'un signal, notamment pour les signaux aléatoires. Il s'agit de la Densité Spectrale de Puissance (DSP). Choisissez un des cosinus générés précédemment, estimez, puis tracez, sa DSP. Vous devez pouvoir retrouver, sur le tracé effectué, la fréquence du cosinus généré. Vous pourrez tester une estimation par périodogramme (fenêtré et non fenêtré), par corrélogramme, puis par périodogramme de Welch (en utilisant la fonction pwelch.m de Matlab de la manière suivante : $DSP\_x = pwelch(x, [], [], [], Fe, 'twosided')$). Quels sont les avantages et inconvénients de ces différents estimateurs de la DSP ?