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

Les deux séances de TP seront consacrées à la mise en place et à la bonne compréhension des outils qui ont été vus en cours et qui seront par la suite utilisés en télécommunication et lors du projet signal/telecom. **Aucun compte-rendu ne vous sera demandé sur les TPs mais vous aurez à réaliser un test de 15 mn en fin de chaque séance qui sera noté.**

Un fichier vous est fourni : TP1_a_completer.m. Il vous servira de base pour vos implantations et de modèle pour vos codes à venir : un en-tête avec vos noms, prénoms et date de réalisation, au moins un commentaire par action réalisée, les différentes parties du code bien identifiées, le tracé des figures avec des labels sur les axes, des titres, des légendes lorsque plusieurs tracés sont superposés. Vous y trouverez également les fonctions Matlab nécessaires pour réaliser vos implantations.

Vous partirez de ce fichier, le modifierez pour réaliser ce qui vous est demandé et expliquerez les résultats obtenus en utilisant ce qui a été vu en cours et/ou qui se trouve dans les diapos de cours et le polycopié fourni sur moodle ("signaux et systèmes à temps discret"). Vous pouvez, bien entendu, réaliser plusieurs fichiers à partir de celui fourni pour davantage de clarté (un par exercice réalisé par exemple).

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 $k^{ième}$ 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 avec une fréquence d'échantillonnage $F_e = 10000$ Hz.
2. Tracer le cosinus généré précédemment **avec une échelle temporelle en secondes**. Retrouvez, à partir du tracé, la fréquence et l'amplitude du cosinus.
3. Générer 90 échantillons d'un cosinus (fonction *cos.m* sous matlab), d'amplitude 1 (V), de fréquence $f_0 = 1100$ Hz et échantillonné avec une fréquence d'échantillonnage $F_e = 1000$ Hz.
4. 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. Expliquez pourquoi et d'où vient la valeur de la fréquence observée.

2 Représentation fréquentielle

2.1 Transformée de Fourier discrète (TFD)

La transformée de Fourier numérique, ou discrète, 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 $n^{ième}$ 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 $n^{ième}$ élément $X(n)$ du tableau représentant la TFD du signal x est alors $X\left(n\frac{F_e}{N}\right)$.*

1. 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, calculez sa transformée de Fourier numérique, 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*.
   – en échelle log grâce à la fonction *semilogy.m* et en utilisant la fonction fftshift.
   Dans les deux cas, **L'échelle fréquentielle devra être en Hz.** Retrouvez-vous les fréquences des cosinus générés ?
3. Choisissez un des cosinus générés précédemment et calculez sa transformée de Fourier numérique 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, 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. Choisissez un des cosinus générés précédemment et estimez sa transformée de Fourier en utilisant un fenêtrage. Vous testerez l'utilisation des fenêtres de Blackman (fonction blackman.m) et de Hamming (fonction hamming.m). Tracez, superposées sur une même figure, la TFD non fenêtrée, la TFD avec fenêtrage de Hamming, et la TFD avec fenêtrage de Blackman, le tout avec une échelle fréquentielle en Hz, un titre, des labels sur les axes, une légende et en utilisant fftshift. Quel est l'intérêt d'utiliser une transformée de Fourier fenêtrée ?

2.2 Densité spectrale de puissance (DSP)

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). Plusieurs estimateurs de la densité spectrale de puissance existent (voir diapositives de cours et, pour plus de détails, le chapitre 5 *du polycopié "signaux et systèmes à temps discret").*
Nous vous demanderons ici d'implanter, dans un premier temps, une estimation par périodogramme de la DSP d'un des cosinus générés précédemment. Vous la tracerez et devrez retrouver la fréquence du cosinus généré.
Vous estimerez ensuite la DSP de ce même cosinus en utilisant un périodogramme de Welch (fonction pwelch.m de Matlab). Quel est l'avantage d'utiliser ce dernier estimateur ?

3 Fonction d'autocorrélation

La fonction d'autocorrélation d'un signal peut s'estimer de différentes manières en numérique, dans le domaine temporel ou bien dans le domaine fréquentiel (voir diapositives de cours et, pour plus de détails, le chapitre 4 *du polycopié "signaux et systèmes à temps discret"). Nous vous proposons d'utiliser, dans le code donné, un estimateur de la fonction d'autocorrélation donné par la fonction xcorr de Matlab*

1. Générer un bruit gaussien (fonction *randn.m* de Matlab), dont vous déterminerez la puissance afin de l'ajouter à un des cosinus générés précédemment pour obtenir un rapport signal à bruit de 0 dB.
2. Ajouter le bruit au cosinus et calculer la fonction d'autocorrélation du signal bruité, en utilisant la fonction *xcorr.m* de Matlab.
3. Tracer le cosinus bruité et sa fonction d'autocorrélation. En déduire un intérêt du calcul de la fonction d'autocorrélation du signal.
4. Quel est l'estimateur qui a été utilisé ici pour calculer la fonction d'autocorrélation numérique ?
5. Que représente un rapport signal sur bruit de 0 dB ?
6. Que représente la valeur en 0 de la fonction d'autocorrélation ?
7. A partir de la valeur de l'autocorrélation en 0 retrouver le rapport signal sur bruit.
8. On pourra retrouver, par transformée de Fourier de sa fonction d'autocorrélation, la densité spectrale de puissance du signal (estimateur du corrélogramme).