ENSTA-Bretagne STIC	Module Traitement du signal FIPA	Prof. : J.-C. Cexus
TP sous Matlab Introduction à l'analyse Temps-Fréquence		

Dans cette étude, nous proposons d'étudier les limites de l'analyse spectrale. En effet, ce type d'analyse suppose que le signal soit stationnaire sut toute la durée de l'observation. Cela signifie que la fréquence n'évolue pas au cours du temps. Comme nous le verrons lors de la séance, cette hypothèse est extrêmement forte et peut s'avérer fausse dans de nombreux cas.

Pour cela télécharger tous les fichiers Matlab qui se trouve sous Moodle. Par la suite, il s'agit de compléter le code au fur et à mesure de l'avancement de la séance. Attention, le code n'étant pas complet pour le faire fonctionner utiliser l'icône : *'Run section'* afin lancer uniquement les sections de codes utiles (il est aussi possible d'utiliser la touche ***F9*** après sélection du code que l'on souhaite étudier).

Etude 1 : Les limites de Fourier et la notion de fréquence instantanée

1. Synthèse du signal et analyse fréquentielle

Pour cette étude, il n'y a pas de code de départ.

On considère le signal x comme la combinaison de deux signaux x_1 et x_2 (voir figure 1)

$$x = [x_1 , x_1 + x_2 , x_2]$$

avec :

$x_1 = A_1 * cos(2 * \pi * f_1 * t)$, avec $f_1 = 50Hz, A_1 = 1$, de durée $duree = 1s$

$x_2 = A_2 * cos(2 * \pi * f_2 * t)$, avec $f_2 = 100Hz, A2 = 1$, de durée $duree = 1s$

On note $f_e = 1000Hz$ la fréquence d'échantillonnage.

1. Définir les signaux x_1, x_2 et le signal $x = [x_1, x_1+x_2, x_2]$. Donner la représentation temporelle du signal x. Identifier les zones où il existe une unique composante fréquentielle et où le signal est multi-composantes.

Pour vous aider on pourra écrire par exemple :

```
% Signal x
fe = 1000;  % Frequence d'echantillonnage 1000
f1 = 50;
f2 = 100;
duree  = 1; % Durée des signaux x1 et x2
tmp_N  = round(fe*duree);
tmp_t  = (0:tmp_N-1)./fe;
tmp_x1 = cos(2*pi*f1*tmp_t);
tmp_x2 = cos(2*pi*f2*tmp_t);
x  = [tmp_x1, tmp_x1 + tmp_x2, tmp_x2];
N  = length(x);     % Nbre d'echantillons en Temps de x
t  = (0:N-1)./fe;   % Echelle des temps du signal x
```

2. Calculer et afficher le spectre d'amplitude du signal x sur l'intervalle $[-f_e/2, f_e/2]$. Commentaires.

N.B : Pour la définition de l'échelle des fréquences freq entre $[-f_e/2, f_e/2]$, on privilégiera son estimation via la formule suivante : freq = $-f_e/2 : f_e/N_{fft} : f_e/2 - f_e/N_{fft}$;
au lieu de : freq = linspace($-f_e/2$, $f_e/2$, N_{fft}) qui a tendance à décaler les valeurs des échantillons fréquentielles (N_{fft} étant le nombre d'échantillon du spectre).

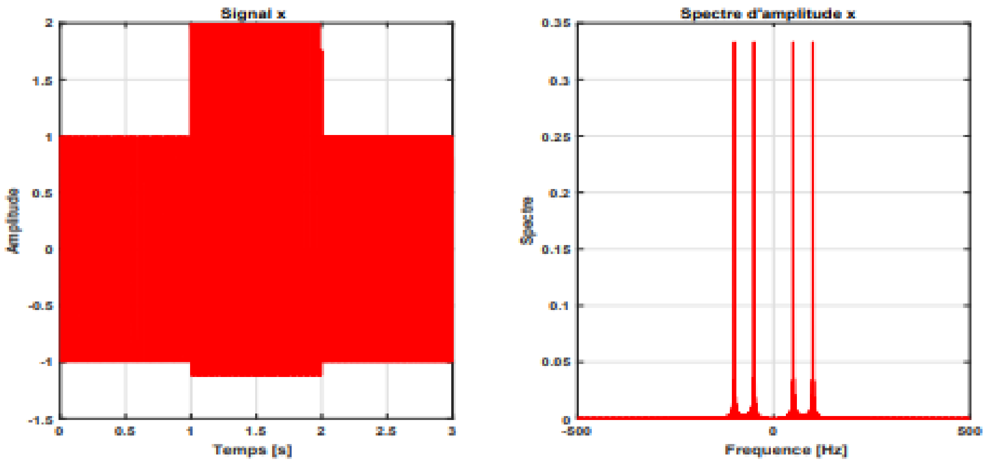

FIGURE 1 – Représentation temporelle et fréquentielle du signal *x*.

2. Étude des grandeurs instantanées via la Transformée de Hilbert

On souhaite estimer les lois instantanées (en fréquence et en amplitude) du signal *x* en utilisant la forme analytique du signal. Pour cela, il est nécessaire d'estimer la Transformée de Hilbert du signal *x* puis d'en déduire la fréquence instantanée (FI) et l'amplitude instantanée (AI) du signal. Ainsi, l'ensemble de l'approche peut se décomposer selon les étapes suivantes :

1. Estimer le signal analytique, noté : *xa*, du signal temporel *x* via la Transformée de Hilbert (voir la fonction *hilbert*()). Noter que le signal analytique, *xa*, est un signal temporel complexe. Pour le fun, afficher le signal analytique *xa* dans le plan complexe (voir https://fr.wikipedia.org/wiki/Plan_complexe).

2. Estimer et afficher le spectre d'amplitude du signal analytique *xa* entre [−fe/2, fe/2]. Commentaires par rapport au spectre d'amplitude de *x*.

3. Estimer l'amplitude instantanée (AI) du signal analytique *xa* : $AI = |xa|$ (module de *xa* à l'aide de la fonction *abs()*). Donner la représentation temporelle de AI. On pourra superposer, sur la même figure, le signal temporel *x* et AI. Commentaires.

4. Estimer la fréquence instantanée (FI) du signal analytique *xa* : FI $= x = \frac{f_e}{2\pi}\frac{\partial\, arg[x_a]}{\partial t}$ (voir les fonctions : *diff*() pour la dérivée partielle et *angle*() pour l'argument). Donner la représentation temporelle de FI (c'est une fonction dépendante du temps). Commentaires.
A noter que la dérivation numérique *diff*() est souvent très délicate et entraine des erreurs d'estimations. Afin de pallier ce problème numérique, il est possible d'appliquer sur le signal FI un filtrage médian de taille 3 par exemple (voir la fonction medfilt1()).

On pourra comparer la précédente approche avec le code suivant (lire l'aide de instfreq pour essayer de comprendre les différences) :

```
% Autre codes (méthode matlab)
instfreq(x, fe, 'Method', 'hilbert')

% ou encore (là c'est un peu plus compliqué - juste pour le fun)
[p,f,t] = pspectrum(x,fe,'spectrogram');
instfreq(p,f,t)
```

3. Analyse conjointe en Temps-Fréquence

On souhaite maintenant mettre en œuvre une représentation temps-fréquence abordé dans le cours : la Transformée de Fourier à Court Terme (TFCT) ou encore le spectrogramme (i.e : le module au carré de la TFCT). Pour mémoire, on obtient la densité d'énergie de x autour du temps t et de la fréquence f. Cette transformation permet d'obtenir une carte d'énergie temps-fréquence de x. En d'autres termes, il est possible de visualiser l'évolution du spectre du signal en fonction du temps.

La syntaxe de cette fonction *specgram* (voir l'aide) est la suivante :

[A, F, T] = specgram(X, NFFT, Fs, WINDOW, NOVERLAP)

avec la matrice A la carte d'amplitude temps-fréquence, le vecteur F l'échelle des fréquences et le vecteur T l'échelle des temps. *Fs* est la fréquence d'échantillonnage.
En utilisant les commandes : mesh() ou imagesc(), il est possible de représenter le spectrogramme du signal en dB (ou en linéaire).

Attention : Dans cette fonction, le paramètre WINDOW doit être un vecteur de point. Le paramètre NOVERLAP est un nombre de point et non un pourcentage. De plus contrairement à ce que suggère le nom de la fonction, la matrice A correspond à la TFCT et non au spectrogramme du signal X !

Remarque : A noter que dans les dernières versions de Matlab, la méthode *specgram* est remplacée par *spectrogram*(), voir l'aide de Matlab pour identifier correctement les paramètres. L'ancienne méthode fonctionne toujours.
Remarque : Afin de lisser la surface, taper l'instruction *shading interp* après l'affichage temps-fréquence.

Cela peut donner avec notre signal précédent :

```
%% TFR avec deux méthodes de Matlab
nfft     = 512;
win      = hann(nfft);
noverlap = ceil(nfft/3);
figure; hold on; grid on;
specgram(x,nfft,fe,win,noverlap);          % ici affichage directe de la TFR
xlabel('Temps'); ylabel('Fréquence (Hz)');
axis([0.08 2.81 0 round(fe/2)]);

% ou encore avec récupération de la matrice TFR pour manipulation par la suite)
[TFR,ftfr,ttfr] = spectrogram(x,win,noverlap,nfft,fe,'yaxis');
figure; hold on; grid on;
imagesc(ttfr,ftfr,10*log10(abs(TFR)));     % affichage en log
xlabel('Temps'); ylabel('Fréquence (Hz)');
axis xy; axis tight; colormap(jet);
```

Appliquer le spectrogramme sur le signal précédent. Conclure sur les caractéristiques du signal (mono-composante ou multi-composantes) et les limitations importantes de la FI estimée à partir du signal analytique.
S'il vous reste du temps faire varier le recouvrement (dernier argument de la méthode *specgram*) et le type de la fenêtre (*boxcar*, *hanning*, *bartlett*, *blackman*).

On pourra tester d'autres méthodes comme la commande *wvd* permettant d'estimer la Wigner-Ville Distribution et la Smoothed Pseudo Wigner-Ville distribution.
Pour finir, je vous invite à aller voir la Time-Frequency Gallery (à taper dans l'aide) de Matlab (pour écouter et voir les chants des baleines ! Commande (*soundsc()* pour écouter – voir étude 2 pour son utilisation).

Etude 2 : Visualisation du repliement de spectre

Pour cette étude, un code de départ est proposé : Etude_2.m.
Considérons le signal à temps continu donné par :

$$x(t) = Acos[\theta(t)]$$

On appelle fréquence instantanée, $F_i(t)$, la fonction dépendant du temps défini par :

$$F_i(t) = \frac{1}{2\pi}\frac{d\Theta}{dt}$$

Considérons par exemple, le cas où la fréquence instantanée varie linéairement en fonction du temps, ce qui s'écrit :

$$F_i(t) = F_0 + \lambda t, \quad \text{avec } \lambda \text{ s'exprime en Hz/s.}$$

En observant le signal par tranche de temps de durée suffisante, on doit constater qu'entre les instants 0 et T, la fréquence varie linéairement entre F_0 et $F_1 = F_0 + \lambda t$.

En intégrant $F_i(t)$ et en supposant que $F_i(0)=0$, on obtient :

$$\Theta(t) = 2\pi F_0 t + \pi\lambda t^2$$

- On désire créer le signal numérique x avec **A=1** l'amplitude du signal, **Fe=8000Hz** la fréquence d'échantillonnage, **T=2s** la durée, **F0 = 1000Hz** et **λ = 1000Hz/s**. Quel sera le nombre **N** d'échantillons de ce signal ?

- Dans un fichier script, créez le vecteur temps noté **t** à partir de ce nombre d'échantillons puis le signal numérique associé à x(t). Afficher le signal x en fonction du temps.

Pour trois valeurs de **λ= {1000, 2000, 3000}**, calculer les intervalles de fréquence **[F0, F1]** sur lequel la fréquence varie.

- Construire les 3 signaux **{x1, x2, x3}** pour les trois valeurs de **λ** ci-dessus.

- Tracer les représentations temporelles des 3 signaux.

- Tracer les représentations fréquentielles des 3 signaux.

- Faire l'écoute du signal reconstruit (*A faire si une carte son est installée*), la fonction *soundsc(x,Fe)* reconstruit, à partir des échantillons **x** et de la fréquence d'échantillonnage **Fe**, un signal à temps continu qui est envoyé sur la sortie audio.

- Obtient-on des signaux dont la fréquence varie linéairement ? Que se passe-t-il ?

- Pour les trois valeurs de **λ={1000, 2000, 3000}**, appliquer une Transformée Temps-Fréquence (TFTC ou spectrogramme) pour les trois valeurs de **λ**. Obtient-on un signal dont la fréquence varie linéairement ? Que se passe-t-il ? On pourra modifier les paramètres de la fonction pour améliorer l'image temps-fréquence.

Etude 3 : Visualisation temps-fréquence et musique (en plus mais c'est beau !)

Pour cette étude, il n'y a pas de code de départ.
A partir du fichier *Callas.wav*, écrire un petit programme permettant successivement :

1. De charger le signal et d'afficher sa représentation temporelle (voir la fonction *audioread*). Faire attention à la fréquence d'échantillonnage.

2. D'écouter le signal si carte son (voir la fonction *sound*).

3. D'estimer sa représentation fréquentielle. On pourra chercher à afficher l'analyse fréquentielle dans une bande de fréquence précise à l'aide de la commande *xlim([fmin, fmax])*.
   Une norme s'est imposée selon laquelle l'oreille humaine percevrait les sons dans des fréquences comprises entre 20 Hz et 20 000 Hz
   Lire par exemple http://www.cochlea.org/entendre/champ-auditif-humain
   Dans notre cas, il semble que la partie pertinente soit entre 20 et 5000 Hz.

4. De faire une analyse temps-fréquence en utilisant le spectrogramme (figure 2). Visualiser alors la voix de la chanteuse dans le plan temps-fréquence en modifiant les différents paramètres du spectrogramme. On pourra chercher à afficher l'analyse temps-fréquence dans une bande de fréquence précise [fmin, fmax]. Commentaires.
   Pour vous aider :
   ```
   nfft     = 512;
   window   = hanning(nfft);
   noverlap = ceil(nfft/3);
   ```

5. Pensez-vous qu'une analyse des grandeurs instantanées via la Transformée de Hilbert soit pertinente ? Expliquer (voir votre cours si besoin 😉).

6. Reprendre l'étude avec le fichier gong.wav ou prendre des fichiers à vous.

Attention : Ne pas utiliser de trop gros fichier sinon l'analyse risque de devenir impossible.
La taille de la matrice temps-fréquence risque, en effet, de faire planter Matlab !

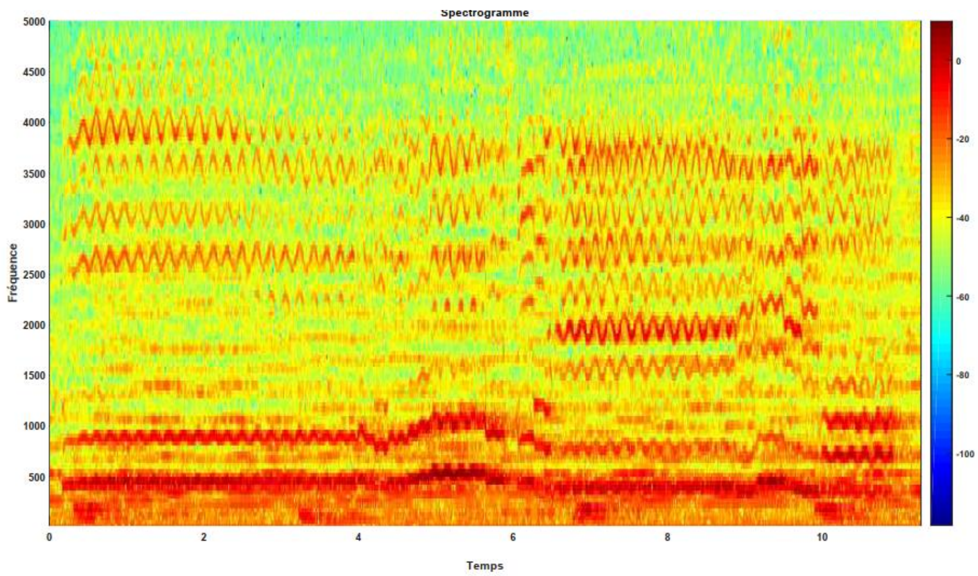

Figure 2 : Spectrogramme de l'enregistrement Callas.wav.