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

Nous allons illustrer un certain nombre de concepts abordés lors du cours d'analyse spectrale. Pour cela télécharger tous les fichiers Matlab qui se trouve sous Moodle. Le fichier principal est ***Analyse_Spectrale_Eleve.m***. 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 de 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).

Dans le cadre de cette étude, on considère un signal de la forme :

$$\boldsymbol{x = x_1 + x_2 + x_3 + bruit}\,,$$

avec pour $\boldsymbol{i \in 1,2,3 : x_i = A_i \cos(2\pi F_i t)}$:

	Signal x_1	Signal x_2	Signal x_3
A_i	1	0.1	1
F_i [Hz]	20	30	40

Lorsque cela sera nécessaire, le bruit sera supposé être un bruit blanc gaussien de moyenne nulle et de variance ***0.2*** (commande Matlab : *randn()*)

La fréquence d'échantillonnage $\boldsymbol{f_e}$ est de **510 Hz** et le nombre d'échantillons en temps : $\boldsymbol{N = 2048}$. **Dans un premier temps, on suppose qu'il n'y a pas de bruit (variance nulle).**

Première partie

1. Compléter la section *'GENERATION du signal'* afin de générer le signal ***x***. Tracer le signal.

2. Compléter la section '*SPECTRE d'amplitude*' afin de représenter son spectre d'amplitude entre $\left[\frac{-f_e}{2}, \frac{+f_e}{2}\right]$.
   Pour mémoire, il est possible de représenter l'échelle des fréquences[1] par la commande suivante : $freq = \frac{-f_e}{2} : \frac{f_e}{N_{fft}} : \frac{f_e}{2} - \frac{f_e}{N_{fft}}$ avec N_{fft} le nombre d'échantillons en fréquence. Dans cette étude, $\boldsymbol{N_{fft}}$ sera toujours égale à ***N***.
   Le spectre d'amplitude sera représenté d'une part en échelle linaire et d'autre part en dB (pour mémoire : $20log10(spectre_{amplitude})$).
   Commandes utiles : *fft(), abs(), fftshift(), linspace(), log10(), length() …*

3. Faire une analyse des deux représentations fréquentielles. Identifier les trois fréquences $\boldsymbol{F_i}$.

4. Modifier la section *'GENERATION du signal'* afin de prendre en compte une réalisation de bruit blanc gaussien de variance ***0.2***. Réaliser une analyse des deux représentations fréquentielles. Commentaires. Est-il toujours facile d'identifier les trois fréquences $\boldsymbol{F_i}$?

[1] On pourrait utiliser $freq = linspace(\frac{-f_e}{2}, \frac{f_e}{2}, N_{fft})$. Cependant, il s'avère que cette approche peut occasionner des erreurs sur la création des échantillons associé à l'échelle des fréquences.

5. Relancer plusieurs fois le code afin d'étudier l'effet du bruit. Regarder notamment les valeurs de fréquences que retourne la fonction *AffichеMaxFreq(fftshift(fftx), freq, 3).* Cette méthode permet de retourner automatiquement les trois principales fréquences détectées dans le signal. Identifier la fréquence qui est la moins bien identifiée ? Pourquoi ?

6. **Etude annexe : estimation du rapport signal à bruit**.
   La qualité d'un signal est souvent représentée par le '*rapport signal/bruit*' (S*ignal-to-Noise Ratio* (SNR) en anglais). Ainsi, pour un signal $x(t) = s(t) + n(t)$, le rapport signal sur bruit peut être défini par la relation suivante [2] :

$$\xi = \frac{W_s}{W_n},$$

   avec W_s l'énergie du signal $s(t)$ et W_n l'énergie du bruit $n(t)$. Pour rappel, l'estimation de l'énergie d'un signal discret est $W_s = \sum_{i=1}^{N} s(i)^2$ avec *s(i)* les *N* échantillons décrivant le signal *s.* Le SNR est souvent représenté avec une échelle logarithmique appelés décibels (dB) :

$$\xi_{dB} = 10log_{10}(\xi).$$

   On vous demande d'estimer le SNR en dB du signal simulé précédemment.

Deuxième partie

Par la suite, nous allons tenter d'affiner la localisation des trois fréquences en programmant quelques approches simples d'analyse spectrale.

1. **Etude du périodogramme simple**
   a. Ouvrir le fichier ***periodogram.m***. L'objectif est de compléter le code en s'inspirant du cours. Indications :
      - Pour $\boldsymbol{N_{fft}}$, on prendra la longueur du signal : $\boldsymbol{N_{fft} = N}$.
      - La grandeur ***Y*** est : $\boldsymbol{Y(f) = FFT\{x(t)\}}$ avec $\boldsymbol{FFT}$ la Transformée de Fourier du signal $\boldsymbol{x(t)}$.
      - **Px** est la quantité : $\gamma_{xx}(f) = \frac{|Y(f)|^2}{N}$ avec ***N*** le nombre d'échantillons en temps.

   b. Revenir dans le code ***Analyse_Spectrale_Eleve.m*** et compléter la section '*PERIODOGRAMME simple*' afin de mettre en place l'analyse spectrale par périodogramme simple sur le signal construit dans la première partie.
   On affichera la DSP en dB sur l'intervalle $\left[\frac{-f_e}{2}, \frac{+f_e}{2}\right]$. Pour info, l'échelle des fréquences est identique à la première partie.

   c. Toujours dans la section '*PERIODOGRAMME simple*', compléter afin d'identifier automatiquement les trois fréquences (s'inspirer de la première partie avec la fonction *AffichеMaxFreq.m*). Commentaires. Retrouve-t-on les trois fréquences ? On pourra relancer plusieurs fois le code pour étudier l'effet du bruit.

2. **Etude du périodogramme modifié**

[2] Dans le cas de signaux discret, il est possible de prendre une autre définition équivalente qui est le rapport des variances entre le signal et le bruit : $\xi_d = \frac{{\sigma_s}^2}{{\sigma_s}^2}$

a. Ouvrir le fichier ***periodogram_Modifie.m***. L'objectif est de compléter le code en s'inspirant du cours. Indications :
   - Pour $\boldsymbol{N_{fft}}$, on prendra la longueur du signal en temps: $\boldsymbol{N_{fft} = N}$.
   - Pour la fenêtre de pondération $\boldsymbol{w(t)}$ de longueur **N**, on pourra définir plusieurs formes : rectangulaire, triangle, hamming, hanning, bartlett, blackman…
   - La grandeur ***Y*** est : $\boldsymbol{Y(f) = FFT\{xw(t)\}}$ avec $\boldsymbol{FFT}$ la Transformée de Fourier du signal $\boldsymbol{xw(t) = x(t).* w(t)}$
   - ***Px*** est la quantité : $\boldsymbol{\gamma_{xx}(f) = \frac{|Y(f)|^2}{N*U}}$ avec ***N*** le nombre d'échantillons en temps et ***U*** le paramètre de normalisation associé à la fenêtre de pondération $\boldsymbol{w(t)}$.

b. Revenir dans le code ***Analyse_Spectrale_Eleve.m*** et compléter la section '*PERIODOGRAMME modifié* afin de mettre en place l'analyse spectrale par périodogramme modifié sur le signal construit dans la première partie.
   On affichera la DSP en dB sur l'intervalle $\left[\frac{-f_e}{2}, \frac{+f_e}{2}\right]$. Pour info, l'échelle des fréquences est identique à la première partie.

c. Toujours dans la section '*PERIODOGRAMME modifié*', compléter afin d'identifier automatiquement les trois fréquences (s'inspirer de la première partie avec la fonction *AffficheMaxFreq.m*). Commentaires. Retrouve-t-on les trois fréquences ? On pourra relancer plusieurs fois le code pour étudier l'effet du bruit.

d. Pour étudier l'effet des fenêtres sur l'estimation de la DSP. On prendra le même signal mais avec moins d'échantillons ***N=256*** et, on supposera **qu'il n'y a pas de bruit**. Modifier la première section pour prendre en compte les nouvelles caractéristiques du signal et, compléter la section '*EFFET FENETRAGE pour le Periodogramme modifié*' pour l'effet du fenêtrage lors de l'analyse spectrale via la périodogramme modifié. Commentaires.

3. Etude de l'approche proposé par Bartlett

Avant toute chose, revenir à la configuration initiale pour le signal : ***N=2014*** avec un bruit blanc gaussien de variance ***0.2***. On pourra aussi mettre en commentaires la section 'EFFET FENETRAGE pour le Periodogramme modifié' en encadrant cette section de code par **%{** et **%}**

a. Ouvrir le fichier ***Bart.m***. L'objectif est de compléter le code en s'inspirant du cours. Indications :
   - Pour $\boldsymbol{N_{fft}}$, on prendra la longueur du signal en temps : $\boldsymbol{N_{fft} = N}$.
   - Pour ***L***, on pendra de manière à vérifier la relation ***N=KL*** avec **K** le nombre de bloc qui est connu à l'avance.
   - Pour estimer la DSP pour chaque bloc ***i*** de signal, $\boldsymbol{\gamma^i_{x_b loc}(f)}$, on utilisera le périodogramme simple qui a été écrite précédemment.
   - ***Px*** est la quantité vérifiant la relation : $\boldsymbol{\gamma_{xx}(f) = \frac{\sum \gamma^i_{x_{bloc}}(f)}{K}}$.

b. Revenir dans le code ***Analyse_Spectrale_Eleve.m*** et compléter la section '*ETUDE BARTLETT*' afin de mettre en place l'analyse spectrale par Bartlett sur le signal construit dans la première partie.
   On affichera la DSP en dB sur l'intervalle $\left[\frac{-f_e}{2}, \frac{+f_e}{2}\right]$. Pour info, l'échelle des fréquences doit être reconstruite en fonction du nombre d'échantillons de la nouvelle DSP.

c. Toujours dans cette section *'ETUDE BARTLETT'*, compléter afin d'identifier automatiquement les trois fréquences (s'inspirer de la première partie avec la fonction *AffichеMaxFreq.m*).

d. En prenant différentes valeurs pour K, retrouve-t-on les trois fréquences ? On pourra relancer plusieurs fois le code pour étudier l'effet du bruit. Y a-t-il intérêt à prendre ***K=1*** ? Que se passe-t-il si ***K*** est trop grand ? Proposer une valeur de ***K*** qui serait selon vous intéressante.

4. Etude de l'approche proposé par Welch

a. Ouvrir le fichier ***welch.m***. L'objectif est de compléter le code en s'inspirant du cours. Indications :
   - Pour N_{fft} , on prendra la longueur du signal en temps : $N_{fft} = N$.
   - Pour ***L***, on pendra de manière à vérifier la relation ***N=KL*** avec **K** le nombre de bloc qui est connu à l'avance.
   - Pour estimer la DSP pour chaque bloc ***i*** de signal, $\gamma^{i}_{x_b loc}(f)$, on utilisera le périodogramme simple qui a été écrite précédemment.
   - ***Px*** est la quantité vérifiant la relation : $\gamma_{xx}(f) = \frac{\sum \gamma^{i}_{x_{bloc}}(f)}{K}$.

b. Revenir dans le code ***Analyse_Spectrale_Eleve.m*** et compléter la section *'ETUDE WELCH'* afin de mettre en place l'analyse spectrale par Welch sur le signal construit dans la première partie.
   On affichera la DSP en dB sur l'intervalle $\left[\frac{-f_e}{2}, \frac{+f_e}{2}\right]$. Pour info, l'échelle des fréquences doit être reconstruite en fonction du nombre d'échantillons de la nouvelle DSP.

c. Toujours dans cette section *'ETUDE WELCH*, compléter afin d'identifier automatiquement les trois fréquences (s'inspirer de la première partie avec la fonction *AffichеMaxFreq.m*).

d. En prenant différentes valeurs pour **L**, retrouve-t-on les trois fréquences ? On pourra relancer plusieurs fois le code pour étudier l'effet du bruit. Que se passe-t-il si ***L*** est trop petit ? On prendra un ***overla*p** de ***50%***. Proposer une valeur de ***L*** qui serait selon vous intéressante.

e. Déterminer les paramètres tels que Welch soit équivalant à Bartlett ?

f. Proposer une configuration qui serait selon vous intéressante.

g. Tester votre configuration sur le même signal mais avec une fréquence d'échantillonnage de $f_e = 80\ Hz$ (à noter que nous sommes en limites du théorème de Shannon). Modifier si besoin votre configuration de Welch. Ne pas hésiter à relancer plusieurs fois l'ensemble du code pour voir l'effet du bruit.

Troisième partie

Dans cette dernière partie, nous allons aborder l'analyse spectrale paramétrique afin de localiser les trois fréquences via la mise en œuvre d'un modèle AR(32) puis d'un modèle ARMA(32,8).

Afin d'estimer un modèle AR à partir de notre signal ***x*** via la méthode de Yule-Walker (ou équation normal), il est nécessaire d'estimer la fonction de covariance (ACF) (via son expression biaisée) afin de pouvoir obtenir la Matrice de covariance R_n nécessaire à l'estimation des paramètres AR :

$$\begin{bmatrix} r(1) \\ \vdots \\ r(n) \end{bmatrix} + R_n \begin{bmatrix} a_1 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}$$

Ou encore sous une autre forme :

$$r_n + R_n \theta_n = 0$$

$$\theta_n = [\, a_1, \ldots, a_n \,]^T$$

Matrice de Covariance R_n :

$$R_n = \begin{bmatrix} r(0) & \cdots & r(-n+1) \\ \vdots & \ddots & \vdots \\ r(n-1) & \cdots & r(0) \end{bmatrix}$$

Estimateur biaisé de la fonction de covariance d'un signal x(k) :

$$\hat{r}(k) = \frac{1}{N} \sum_{t=k+1}^{N} x(t) x^*(t-k) \text{ avec } 0 \leq k \leq N-1$$

Pour estimer directement la Matrice R_n, il est possible d'utiliser une fonction sous Matlab :

```
K = 32 ; % on suppose que l'on connait l'Ordre du modèle AR(K)
[X, Rn_tmp] = corrmtx(x, K,'autocorrelation'); % ! Dim (K+1)x(K+1)
Rn = Rn_tmp(1:K, 1:K); % Matrice doit être (K)x(K)
r  = Rn_tmp(:, 1);     % Vecteur de dimension (K+1)x(1)
```

Par la suite, pour estimer les paramètres $[\, a_1, \ldots, a_n \,]^T$, il est nécessaire d'inverser la matrice R_n afin de résoudre l'équation normale :

$$\hat{\theta}_n = [\, \hat{a}_1, \ldots, \hat{a}_n \,]^T = - R_n^{-1} r_n$$

Pour cela, il est possible d'écrire en Matlab :

```
a_tilde  = -inv(Rn2)*r(2:K+1); % Ar coef ou -Rn\r(2:K+1);
ar_tilde = [1; a_tilde].';     % il manque le 1 !
```

A noter qu'une estimation de la variance du bruit peut être estimée en écrivant :

```
var_tilde  = real(r(1)+a_tilde.'*r(2:end));  % Variance du bruit
```

En réalité, plus simplement (on pourra comparer les deux variances ainsi que les coefficients du modèle trouvé par les deux approches) :

```
[a_estim, var_estim] = aryule(x, K);
```

A partir de là, il est possible de simuler une réalisation de notre processus. Pour cela, il suffit **de filtrer un bruit blanc** de variance estimée précédemment à travers notre modèle AR estimé :

```
w     = sqrt(var_estim)*randn(N,1); % UNE réalisation de bruit blanc
x_mod = filter(1, a_estim, w);  % UNE réalisation du processus AR(P)
```

On pourra afficher le signal x et notre réalisation. Commentaires.

Afin d'estimer la DSP est il judicieux de calculer le périodogramme (simple) directement sur notre unique réalisation `x_mod ?` :

```
Px0_mod = Periodogram_Moi(x_mod);
figure; hold on; grid on;
plot(freq, 20*log10(fftshift(Px0_mod)));
```

Ou alors de passer plutôt par une estimation de la DSP via les coefficients AR ? :

```
Lfft = N;
H2   = var_w ./ abs(fft(a_estim, Lfft)).^2;
F2   = -fe/2 : fe/Lfft : (fe/2-fe/Lfft); % Echelle des fréquences

figure; hold on; grid on;
plot(freq, 20*log10(fftshift(Px0))); % Périodogramme simple de x
plot(F2,   10*log10(fftshift(H2)), 'r');
xlabel(' Freq.'); ylabel(' DSP (dB/Hz)');
```

Malheureusement, il apparait que notre modèle AR, bien que performant, n'est pas capable de prendre en compte la fréquence de 30Hz (elle est d'amplitude trop basse et coincée entre deux fréquences d'amplitude très importantes).
Pour arriver à construire un modèle permettant d'avoir les trois fréquences, il est nécessaire d'utiliser à un modèle ARMA(32, 8). La partie MA(8) va permettre de créer "des vallées" susceptibles de faire apparaitre cette fréquence de 30Hz d'amplitude (très faible par rapport aux deux autres) :

```
na = 32; nc = 8;
sys           = armax(x.',[na nc]);
a_estim_arma  = sys.a;
b_estim_arma  = sys.c;
var_estim2    = sys.NoiseVariance;

% Estimation de la DSP à partir des parametres ARMA
[H_tildearma, F] = freqz(b_estim_arma, a_estim_arma,[],fe);
plot(F, 10*log10(abs(var_estim2.*H_tildearma)));
legend(' Periodogramme', 'AR(32)', 'ARMA(32,8)');
```