TP N 03 : Analyse spectrale paramétrique AR et/ou ARMA de signaux sonore (exemple de signaux non-stationnaire)

But de TP :

Ce TP consiste à étudier sous MATLAB les méthodes d'analyse spectrale non paramétrique, basée sur la théorie de Fourier, notamment **le périodogramme** et **la méthode de Welch** avec différente **fenêtre d'analyse (rectwin, Hamming, Hanning, kaiser………).**

- Estimer la densité spectrale de puissance (PSD) d'un signal en utilisant des modèles paramétriques (AR et ARMA).
- Comparer les résultats obtenus avec des méthodes non paramétriques comme la méthode de Welch.
- Observer l'impact du choix du modèle sur l'estimation de la PSD.

Introduction :

L'analyse spectrale consiste à déterminer le contenu fréquentiel d'un signal donné, plusieurs méthode d'estimation spectrale existent ayant chacune ses avantages et ses inconvénient. Ces méthodes se regroupent dans deux grandes catégories, les méthodes classiques basées sur **la transformation de fourier** et **les méthodes modernes** comme celles basée sur l'estimation des paramètres de modèle Auto-Régressif (**AR)**, de Moyenne Ajustée (**MA)** ou Auto-Régressif à Moyenne-Ajustée **(ARMA).**

II. Estimation spectrale

Le problème d'estimation spectrale est un problème classique en statistique et en traitement du signal. Le problème est le suivant: on dispose d'un signal de longueur finie, que l'on suppose être une observation de longueur finie d'un signal aléatoire faiblement stationnaire. La question est la suivante: **comment estimer le spectre de celui-ci (qui est une quantité moyennée sur l'ensemble des réalisations possibles) à partir de cette unique observation**. La détermination précise du spectre d'un signal, donne nécessite qu'il soit périodique, ou de longueur finie, et non contaminé par un bruit. Ces méthodes d'analyse ne donnent qu'une estimation du véritable spectre.
Les diverses approches d'analyse spectrale permettent d'améliorer l'estimation de **la Densité Spectrale de Puissance (DSP).**

Cette partie consiste à étudier sous MATLAB pour:

- Étudier la Transformée de Fourier.
- Maîtriser l'analyse spectrale non–paramétrique.
- Calculer le périodogramme.

II.1. Le periodogrmme: la transformation de Fourier finie :

$$S\left(\frac{k\eta}{L}\right) = \frac{1}{L}\left|\sum_{l=0}^{L-1} x(l)e^{-2i\pi kl/L}\right|^2 = \frac{1}{L}|\hat{x}[k]|^2 \qquad (1)$$

Le spectre estime par le péridogrmme est souvent trés ‘’ irregulier’’. En fait, il souffre de deux défaut importants, que l’on peut quantifier en revenant un cadre aléatoire. En effet, si le signal x est modelisé comme un signal aléatoire, le périodogramme tel que definie en (1), est lui aussi aléatoire, ce qui permet d’introduire les quantités suivantes:

- Son biais.
- Aleatoire, le périodogramme tel que défini, est lui aussi aléatoire.

Le périodogrmme moyenne: le signal est découpé en segmets de longueur N, un spectre est estimé par calcul du périodogromme sur chacun des segments, puis on effectue les moyennes des estimées correspondantes du spectre.

$$S\left(\frac{k\eta}{N}\right) = \frac{1}{P}\sum_{p=0}^{P-1}\frac{1}{N}\left|\sum_{n=0}^{N-1} x[n+p\delta]e^{-2i\pi kn/N}\right|^2 \qquad (2$$

Le spectre estimé correspondant est generalement plus ’lisse’.

Le périodogrmme de Welch: il est similaire au périodogromme moyenne, a ceci presque des fenêtres sont utilisees pour la segmentation, au lieu de la fenêtre rectangulaire. En notant $\boldsymbol{h} \in \boldsymbol{R}^{\boldsymbol{N}}$ la fenêtre, on définit cette fois :

$$S\left(\frac{k\eta}{N}\right) = \frac{1}{P}\sum_{p=0}^{P-1}\frac{1}{N}\left|\sum_{n=0}^{N-1} h[n+p\delta]x[n+p\delta]e^{-2i\pi kn/N}\right|^2 \qquad (3$$

Interprétation.

1. Dans le cas du périodogramme classique, la fenêtre est une fenêtre rectangulaire, sa transformée de Fourier est une fonction qui présente plusieurs lobes, la convolution avec une

telle fonction crée des artefacts. On préfère donc des fenêtre plus "lisses", qui diminuent ces artefacts.

2. Plus la longueur N de la fenêtre est grande (donc plus P est petit), plus l'estimée du spectre est biaisée. Par contre plus N est grand plus le spectre estimé est régulier (lisse). On cherche donc généralement un compromis entre biais et régularité.

Le périodogramme est un estimateur de la DSP basé sur la Transformée de Fourier Discrète (TFD). La DSP estimée est le module de la TFD du signal à analyser, La TFD peut être calculée par la fft, qui est un algorithme rapide d'une structure dynamique. La Transformée de Fourier Rapide (TFR) d'un signal $x(n)$, peut être calculée par la fonction MATLAB selon la syntaxe suivante :

$$\boldsymbol{X} = \boldsymbol{fft}(\boldsymbol{x}, \boldsymbol{N}) \quad \textbf{(4}$$

Avec **x** est le signal à analyser, **N** est le nombre d'échantillons sur lequel est calculée la **fft** et **X** est le vecteur complexe contenant les coefficients sinusoïdaux. Le paramètre n est optionnel et est utilisé pour modifier la longueur des données à analyser.

La magnitude du spectre de puissance est obtenue en appliquant la fonction abs qui permet de calculer le module du vecteur complexe X :

$$\boldsymbol{Magnitude} = \boldsymbol{abs}\,(\boldsymbol{X}) \quad \textbf{(5}$$

La phase du spectre de puissance est calculée par la fonction angle :

$$\boldsymbol{Phase} = \boldsymbol{angle}\,(\boldsymbol{X}) \quad \textbf{(6}$$

La fonction angle permet de calculer l'arc–tangente de la partie imaginaire divisée par la partie réelle du vecteur X.

Calcul du Périodogramme: le programme MATLAB suivant permet de calculer le périodogramme d'un signal x .

Simulation : Script MATLAB N_1

```
clear all; close all; clc;
N = 1024; % nombre d'échantillons
fs = 1000; % la fréquence d'échantillonnage est de 1 kHz
Y = fft(x); %Calculer la TFD
PS = abs(Y).^2; % calculer PS comme magnitude au carré
```

freq = (1:N)/fs; % vecteur fréquence à utiliser avec le tracé
plot(freq,20*log10(PS)); tracer PS en échelle logarithmique title('spectre en puissance (symétrique de fs/2)'); xlabel('Fréquence (Hz)'); ylabel('spectre en puissance (dB)');

II.2. La methode de welch: permet d'estimer la DSP d'un signal donné selon les étapes suivantes:

- Diviser le signal en plusieurs segments avec ou sans chevauchement, Estimer les DSPs des différents segments du signal par la méthode du périodogramme.
- Calculer la moyenne des DSPs.

L'implantation de cet algorithme dans MATLAB est simple. La boîte à outils de traitement des signaux de Matlab comporte une fonction qui effectue ces opérations. Sous sa forme la plus générale, la fonction pwelch s'appelle comme :

$$[\boldsymbol{PS}, \boldsymbol{f}] = \boldsymbol{pwelch}(\boldsymbol{x}, \boldsymbol{window}, \boldsymbol{noverlap}, \boldsymbol{nfft}, \boldsymbol{fs}) \quad \textbf{(7}$$

Le programme MATLAB suivant permet d'estimer la DSP d'un signal x.

Simulation : **Script MATLAB N_2**
%Appliquer la méthode de Welch à un signal sinusoïdal contaminé par du bruit blanc :

```
clear all; close all; clc ;
N = 1024; % nombre d'échantillons
fs = 1000; % fréquence d'échantillonnage (1 kHz)
% Estimer le spectre de Welch en utilisant des segments à 128 points, une fenêtre triangulaire, et un chevauchement de 50 .
[PS,f] = pwelch(x, triang(N),[ ],N,fs);
plot(f,PS,'k'); % tracer le spectre en puissance title('Spectre en puissance (méthode de Welch)'); xlabel('Fréquence (Hz)'); ylabel('Spectre en puissance');
```

Le bruit de fond est considérablement plus lisse et réduit. L'onde sinusoïdale à 250 Hz est clairement visualisée, mais la crête est maintenant légèrement plus large indiquant une perte dans la résolution fréquentielle.

II.3. Fenêtres d'apodisation: L'environnement MATLAB incorpore un certain nombre de fenêtres d'apodisation prédéfinies. Une fonction appropriée de MATLAB produit d'une rangée de vecteur de n–points contenant la forme appropriée de la fenêtre. L'ensemble de ces fonctions ont la même syntaxe avec un nom de fonction correspondant au nom de la fenêtre d'apodisation:

$$\boldsymbol{w} = \boldsymbol{window\_name}(\boldsymbol{N}); \qquad \textbf{(8}$$

Cette commande permet de générer un vecteur **w** de longueur **N** contenant la fonction fenêtre d'un nom de fonction window_name. Où:

N est le nombre d'échantillons du vecteur de sortie.
window_name est le nom, ou une abréviation du nom, de la fenêtre désirée.

À cette écriture, treize fenêtres d'apodisation différentes sont disponibles en plus de la fenêtre rectangulaire **(rectwin)**, qui est incluse pour la perfection. La fenêtre d'aide de MATLAB fournira une liste de noms de fenêtres. Quelques–unes des fenêtres plus populaires sont : **bartlett, blackman, gausswin, hamming (une fenêtre commune par défaut de MATLAB), hann, kaiser**, **et triang**. Quelques unes des routines ont des arguments facultatifs additionnels. En particulier, **chebwin (fenêtre de Tchebychev)** niveau des lobes latéraux, a un deuxième argument pour spécifier l'amplitude du lobe latéral. Naturellement, plus ce niveau est placé petit, plus le lobe principal est large, et plus la résolution en fréquence est réduite. Des détails pour n'importe quelle fenêtre donnée peuvent être trouvés dans l'aide de MATLAB. En plus des différentes fonctions, toutes les fonctions de fenêtre peuvent être construites avec un appel du type :

$$\boldsymbol{w} = \boldsymbol{window}(\mathbf{@name}, \mathbf{N}, \mathbf{opt}) \qquad \textbf{(9}$$

C'est une fenêtre de N–échantillons du nom name qui représente le nom de la fonction spécifique de la fenêtre d'apodisation (précédée du symbole arrobase), N est le nombre d'échantillons désirés, opt représente des arguments optionnels facultatifs possibles exigés par quelques fenêtres spécifiques.

Pour appliquer une fenêtre à l'analyse de séries de Fourier, il suffit de multiplier point par point le signal digitalisé par la sortie de la fonction window_name de MATLAB avant d'appliquer la routine FFT.

Par exemple : **Simulation : Script MATLAB N_3**

```
w = triang (N); % Fenêtre triangulaire de N points
x = x .* w'; % Multiplier point–par–point les données par la fenêtre d'apodisation
X = fft(x); %Calculer la FFT, la fonction fenêtre produit un vecteur colonne.
```

Il est alors à noter que dans l'exemple précédent, il était nécessaire de transposer la fenêtre w afin qu'elle soit au même format que celui du signal analysé.

II.2. Les mêthodes AR, M, ARMA:

Le modèle **AR** a une fonction de transfert avec seulement une constante dans le numérateur et un polynôme dans le dénominateur. Ceci provoque une équation dans le domaine temporel semblable avec seulement un coefficient simple pour le numérateur, b(0), que l'on assume égal à 1:

$$\boldsymbol{y(n) = -\sum_{k=1}^{p} a(k)y(n-k) + u(n)} \quad \textbf{(10}$$

Avec; $u(n)$ est l'entrée qui est un bruit blanc dans ce cas, et p est l'ordre du modèle. Il est à noter que dans l'équation (1), la sortie est obtenue en convoluant la fonction de pondération du modèle, $a(k)$, avec les versions passées de la sortie qui sont $y(n-k)$. C'est semblable à un filtre à réponse impulsionnelle infinie (**IIR**: Infinite Impulse Response) avec un numérateur constant.

Le modèle à moyenne ajustée **MA** est utile pour l'évaluation des spectres avec des vallées sans aucune crête. La fonction de transfert de ce modèle a seulement un numérateur polynômial. L'équation temporelle pour un modèle **MA** est la même que pour un filtre à réponse impulsionnelle finie (**FIR**: Finite Impulse Response), avec le coefficient $a(0)$ du dénominateur mis à 1:

$$\boldsymbol{y(n) = -\sum_{k=1}^{p} b(k)u(n-k)} \quad \textbf{(11}$$

Avec; $u(n)$ est l'entrée et q est l'ordre du modèle. Si le spectre est susceptible de contenir les crêtes pointues et les vallées, alors un modèle qui combine les caractéristiques des modèles **ARMA** est à employer. La fonction de transfert d'un modèle **ARMA** contient alors des polynômes au numérateur ainsi qu'au dénominateur.

$$\boldsymbol{y(n) = -\sum_{k=1}^{p} a(k)y(n-k) - \sum_{k=1}^{q} b(k)u(n-k)} \quad \textbf{(12}$$

Cette partie consiste à étudier sous MATLAB, les méthodes d'analyse spectrale paramétrique, plus en particulier les méthodes basées sur les modèles Auto-Régressifs (AR).

Étant donné un signal de longueur L échantillons, on dispose de plusieurs approaches pour estimer un spectre. Pour identifier un model AR, on doit en premier lieu estimer les coefficients de corrélation. L function **xcorr** peut être utilisée cet effet. Ensuite, la function **levinson** peut être utilisée pour résoudre le probléme par l'algorithme de **Levinson-Durbin.**

Matériel nécessaire :

- MATLAB avec les boîtes à outils pour le traitement du signal (Signal Processing Toolbox).
- Des signaux simulés (combinaison de sinusoïdes et bruit) pour l'analyse.

Travail à faire

Étape 1 : Génération du signal simulé

Le signal simulé que nous allons utiliser est une combinaison de sinusoïdes et de bruit. Cela nous permet de tester la capacité des méthodes paramétriques à estimer correctement la densité spectrale.

- Générer le signal simulé.

Étape 2 : Estimation de la densité spectrale de puissance (PSD) par la méthode de Welch.

Avant de passer à l'estimation paramétrique, nous allons calculer la densité spectrale en utilisant une méthode **non paramétrique**, la méthode de **Welch**.

- Estimer la densité spectrale de puissance (PSD) par la méthode de welch.

Étape 3 : Estimation paramétrique de la PSD avec un modèle AR

Maintenant, nous allons estimer la PSD à l'aide d'un modèle **Auto-Régressif (AR)**. Le modèle AR est une représentation du signal comme une combinaison linéaire de ses propres valeurs passées. Pour cette estimation, nous utiliserons la méthode **Burg** qui est couramment utilisée pour ajuster un modèle AR.

- Estimer la densité spectrale paramétrique par la méthode de Burg.

Étape 4 : Estimation paramétrique de la PSD avec un modèle ARMA

Ensuite, nous allons étendre l'estimation paramétrique au modèle **ARMA** (Auto-Régressif - Moyenne Mobile). Le modèle ARMA combine les caractéristiques des modèles AR et MA (Moyenne Mobile).

- Estimer la densité spectrale paramétrique (DSP) par la méthode ARMA.

Note : Dans ce cas, nous utilisons une fonction arma_psd, qui est un substitut pour des estimations ARMA. MATLAB n'a pas de fonction native simple pour estimer directement la PSD via ARMA dans la boîte à outils standard, donc cette estimation nécessiterait l'utilisation d'une fonction personnalisée ou d'une méthode externe.

Étape 5 : Comparaison des différentes méthodes d'estimation de la PSD

- Comparer entre ces différents méthode.

- Conclure.