

TP 2 Estimation spectrale non-paramétrique

Traitement des signaux biomédicaux



Table des matières



I - Objectif	3
II - Étude théorique	4
III - Périodogramme	5
1. Définition	5
2. Calcul du Périodogramme	6
IV - Estimateur de Welch	7
V - Fenêtres d'apodisation	9
VI - Manipulations et questions	11
1. Manipulation 1	11
2. Manipulation 2	11
3. Manipulation 3	11
4. Manipulation 4	11
5. Comparaison	12
Abréviations	13
Références	14
Bibliographie	15

Objectif

Ce TP consiste à étudier sous MATLAB pour :

- Étudier la transformée de Fourier.
- Maîtriser l'analyse spectrale non-paramétrique.
- Calculer le periodogramme.



Étude théorique



L'analyse spectrale^{p.15} consiste à déterminer le contenu fréquentiel d'un signal donné. Plusieurs méthodes d'estimation spectrale existent ayant chacune ses avantages et inconvénients. Ces méthodes se regroupent dans deux grandes catégories, les méthodes classiques basées sur la transformation de Fourier et les méthodes modernes comme celles basées sur l'estimation des paramètres de modèles autorégressifs (AR^{p.13}), de moyenne ajustée (MA^{p.13}) ou ARMA.

La détermination précise du spectre d'un signal donné nécessite qu'il soit périodique, ou de longueur finie, et non-contaminé par du bruit. Dans plusieurs applications biologiques, le signal à analyser est infini ou de longueur suffisante mais seulement une partie du signal est disponible pour l'analyse. D'autre part, les signaux biomédicaux sont souvent corrompus par des quantités substantielles de bruit ou d'artefacts. Si seulement une partie du signal peut être analysée, et/ou si le signal contient du bruit, alors toutes les techniques d'analyse spectrale doivent nécessairement être approximatives. 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^{p.13}) d'un signal donné.

Périodogramme



Définition

5

Calcul du Périodogramme

6

1. Définition

Le périodogramme est un estimateur de la DSP^{p.13 AA} basé sur la Transformée de Fourier Discrète (TFD^{p.13 AA}) qui est donnée par l'équation (1). La DSP estimée est le module de la TFD du signal à analyser.

$$X(m) = \sum_{n=0}^{N-1} x(n)e^{-j2\pi \frac{mn}{N}} \quad (1)$$

La TFD peut être calculée par la `fft`^{p.13 AA} qui est un algorithme rapide d'une structure dyadique. La Transformée de Fourier Rapide (TFR) d'un signal $x(n)$ peut être calculée par la fonction MATLAB selon la syntaxe suivante :

$$X = \text{fft}(x, N)$$

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 sinusoidaux.

Le paramètre n est optionnel et est utilisé pour modifier la longueur des données à analyser: si $n < \text{length}(x)$, alors l'analyse est effectuée sur les n premiers échantillons; ou bien, si $n > \text{length}(x)$, alors x est complété par des zéros afin qu'il soit d'une longueur égale à n .

Le temps de calcul dépend de la longueur du signal et est meilleur si ce nombre d'échantillons est puissance de 2. Par exemple, sur une même machine, la FFT d'un signal de 4096 échantillons dure environ 2, 1 secondes, mais ça nécessite 7 secondes si le signal est de 4095 échantillons, et 58 secondes si la séquence est de 4097 points. Il est alors judicieux que la longueur de la séquence à traiter par un algorithme FFT soit une puissance de 2. Il est à noter que ces temps de calcul sont effectués sur une même machine afin d'assurer la cohérence de la comparaison. Ce temps de calcul change évidemment selon la machine utilisée pour effectuer le calcul.

La magnitude du spectre de puissance est obtenue en appliquant la fonction `abs`

qui permet de calculer le module du vecteur complexe X :

$$\text{magnitude} = \text{abs}(X)$$

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

$$\text{phase} = \text{angle}(X)$$

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

2. Calcul du Périodogramme

Le programme MATLAB suivant permet de calculer le périodogramme d'un signal x généré par la fonction sig_bruit ainsi développée.

Simulation : Script MATLAB N 1

```
clear all; close all;
```

```
N = 1024; % nombre d'échantillons
```

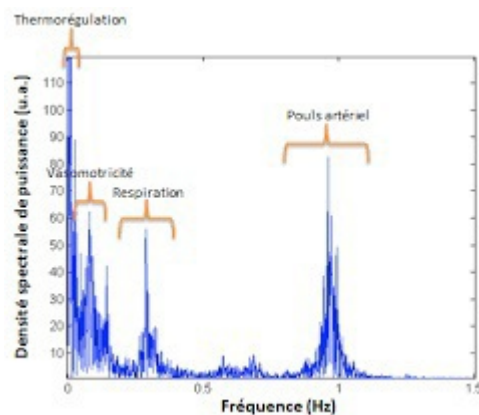
```
% Générer le signal à analyser en utilisant sig_bruit soit 250 Hz sin plus bruit blanc; N échantillons [x,t]
=sig_bruit(250,N);
```

```
fs = 1000; % la fréquence d'échantillonnage est de 1 kHz Y = fft(x); %Calculer la TFR
```

```
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)');
```



Périodogramme

Estimateur de Welch

IV

La méthode de Welch permet d'estimer la DSP d'un signal donné selon les étapes suivantes:

1. Diviser le signal en plusieurs segments avec ou sans chevauchement,
2. Estimer les DSPs des différents segments du signal par la méthode du périodogramme
3. 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 :

$$[PS, f] = pwelch(x, window, noverlap, nfft, fs)$$

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;

N = 1024; % nombre d'échantillons

fs = 1000; % fréquence d'échantillonnage (1 kHz)

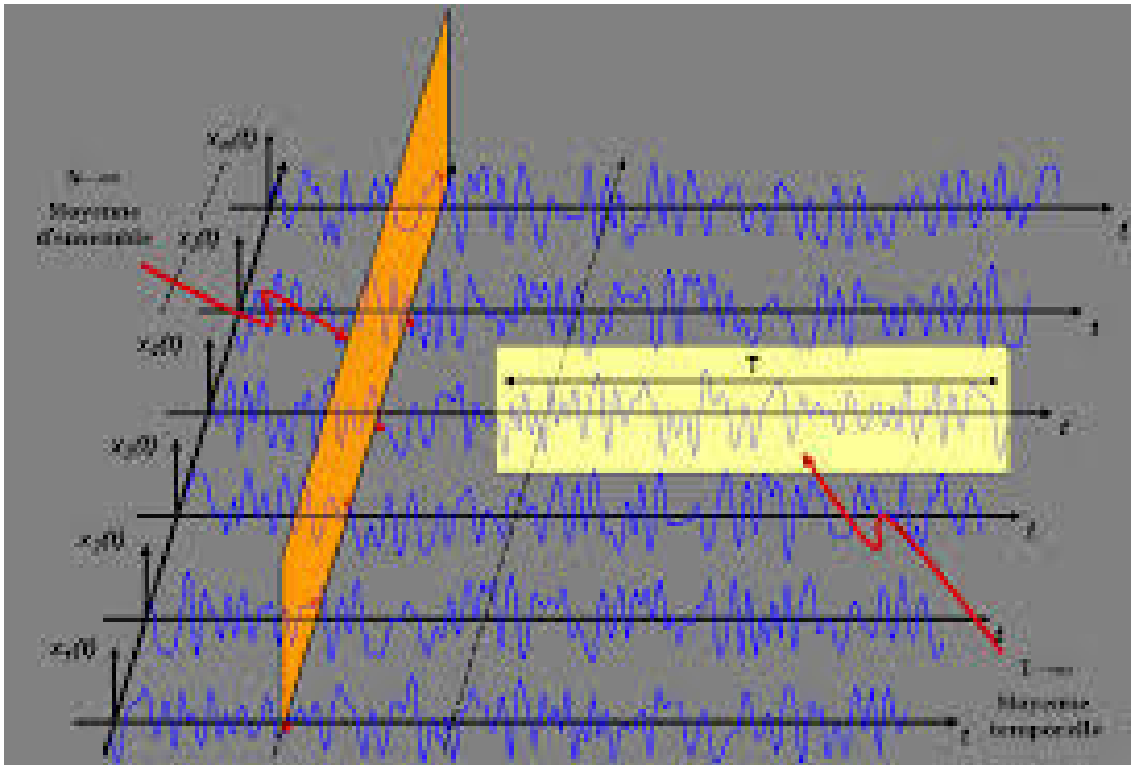
% Génération d'un signal sinusoïdal avec bruit blanc (sin:250 Hz) [x,t] = sig_bruit(250,N);

% 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(128),[ ],128,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.



Estimateur de Welch

Cf. "Power Spectrum Estimation Examples: Welch's Method"

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:

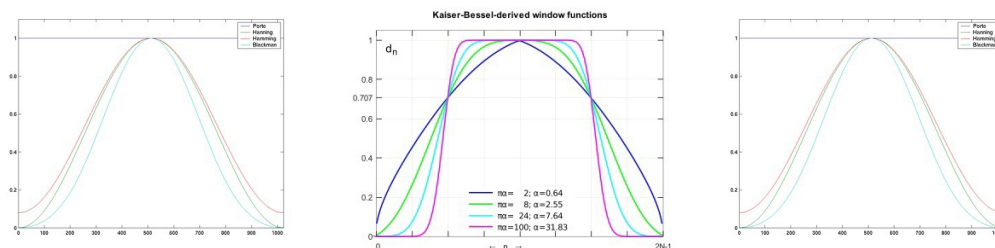
```
w = window_name(N);
```

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 et 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 :

```
w = window(@name,N,opt)
```

Fenetre



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, et 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

```
% Fenêtre triangulaire de N points
```

```
w = triang (N);
```

```
% Multiplier point-par-point les données par la fenêtre d'apodisation
```

```
x = x .* w';
```

```
%Calculer la
```

```
FFT X = fft(x);
```

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.

Manipulations et questions

VI

Manipulation 1	11
Manipulation 2	11
Manipulation 3	11
Manipulation 4	11
Comparaison	12

1. Manipulation 1

Construire deux vecteurs blancs: l'un à 128 points et l'autre à 1024.

1. Calculer les transformées de Fourier de ces deux vecteurs.
2. Est-ce que l'augmentation de la longueur du bruit blanc permet d'améliorer l'analyse.

Appliquer la méthode de Welch au vecteur bruit blanc le plus long en utilisant une fenêtre de Hanning avec un paramètre nfft de 128. L'analyse se fera sans chevauchement.

1. Cette approche améliore-t-elle l'estimation spectrale?
2. Maintenant, changer le chevauchement à 64 échantillons et observer le spectre ainsi calculé.

2. Manipulation 2

Employer la fonction sig_bruit pour générer un vecteur de 256 échantillons qui contient deux sinusoïdes étroitement aligné à 140 et 180 Hz.

3. Manipulation 3

La fonction sig_bruit assume une fréquence d'échantillonnage de 1 kHz.

En utilisant la méthode de Welch, calculer le spectre du signal pour des segments de :

1. 256 échantillons sans chevauchement,
2. 64 échantillons avec un chevauchement de 0% et 50%.

4. Manipulation 4

Employer la fonction sig_bruit pour produire un vecteur de 512 échantillons qui contient une sinusoïde à 200 Hz contaminé par du bruit blanc.

1. Estimer la DSP en calculant la FFT de la fonction d'autocorrélation.
2. Comparer ce spectre de puissance avec celui obtenu par périodogramme. Tracer les deux spectres côte à côte pour comparaison.

5. Comparaison

Pour mieux synthétiser la partie expérimentale, il est préférable de tirer les avantages et les inconvénients de chaque méthode en complétant le tableau suivant :

	Périodogramme	Estimateur de Welch
Avantage		
Inconvénient		
Avantage		
Inconvénient		

Comparaison

Abréviations

AR : Modèles AutoRégressifs

DSP : Densité Spectrale de Puissance

ECG : Électrocardiogramme

fft : fast fourier transform

MA : Moyenne Ajustée

TFD : Transformée de Fourier Discrète

Bibliographie

VI

Analyse et traitement du signal : signaux déterministes et aléatoires, filtrage, estimation avec exercices et problèmes corrigés, Bibliothèque de la faculté de Technologie, ISBN : 978-2-7298-3194-3.

Le traitement du signal sous Matlab : pratique et applications, Bibliothèque de la faculté de Technologie, ISBN : 978-2-7462-1645-7.