

2 Analyse Spectrale

2.1 Problématique de l'Analyse spectrale

La représentation du signal sous la forme $x(t)$ est une démarche est naturelle mais elle ne correspond pas forcément à la meilleure représentation physique des signaux rencontrés. En effet, l'individu ou les systèmes électroniques sont souvent plus sensibles à la puissance et à la fréquence des signaux et la représentation du signal sous la forme de sa répartition de puissance en fonction de la fréquence permet, dans bien des cas, d'extraire de manière plus immédiate l'information qui réside dans ce dernier. Le signal est alors représenté par une fonction $P(f)$ appelée densité spectrale de puissance. Le passage de $x(t)$ à $P(f)$ constitue l'Analyse Spectrale.

Il existe deux grandes classes de méthodes pour estimer la densité spectrale de puissance d'un signal $x(t)$. La première, l'estimation spectrale non paramétrique, n'utilise aucune connaissance a priori sur le signal et part uniquement de l'observation de ce dernier. La deuxième, l'estimation spectrale paramétrique, utilise un modèle paramétrique décrivant le signal, modèle à partir duquel il est aisé d'obtenir la densité spectrale de puissance. Les paramètres du modèle sont adaptés en fonction du signal observé. Entre ces deux méthodes il existe une troisième classe d'approches qui suppose que le signal est composé d'un certain nombre de raies spectrales dont il convient de trouver les fréquences et les puissances. Ce type de méthodes sera classé dans ce cours sous l'appellation d'estimation spectrale par décomposition harmonique.

2.2 Estimation Spectrale non paramétrique

2.2.1 Périodogramme, Corrélogramme

La Transformée de Fourier de la fonction d'autocorrélation $r_{xx}(\tau)$ du processus aléatoire $x(t)$ stationnaire à l'ordre 2 s'écrit:

$$P(f) = \int_{-\infty}^{+\infty} r_{xx}(\tau) e^{-j2\pi f\tau} d\tau \quad (10.)$$

elle est égale à la densité spectrale de puissance $P(f)$ du processus $x(t)$.

En effet, pour $\tau = 0$, il vient:

$$r(0) = E[x(t)^2] = \int_{-\infty}^{+\infty} P(f) e^{j2\pi f \cdot 0} df = \int_{-\infty}^{+\infty} P(f) df \quad (11.)$$

En supposant l'hypothèse ergodique vérifiée, l'espérance mathématique s'écrit:

$$E[x(t)^2] = \lim_{T \rightarrow \infty} \frac{1}{2T} \int_{-T}^{+T} x(t)^2 dt = \int_{-\infty}^{+\infty} P(f) df \quad (12.)$$

$P(f)$ représente donc bien le densité spectrale de puissance du processus $x(t)$. Il s'agit là du théorème de Wiener Kintchine.

L'hypothèse ergodique permettant de confondre l'espérance mathématique avec une moyenne temporelle, le développement de $r(\tau)$ donne :

$$P(f) = \int_{-\infty}^{+\infty} \left[\lim_{T \rightarrow \infty} \frac{1}{2T} \int_{-T}^{+T} x(t)x^*(t-\tau) \right] e^{-j2\pi f\tau} d\tau \quad (13.)$$

On peut aussi introduire le spectre complexe de la réalisation tronquée du processus $x(t)$:

$$X_{2T}(f) = \int_{-T}^{+T} x(t)e^{-j2\pi ft} dt \quad (14.)$$

En prenant le carré de cette expression, il vient :

$$\frac{1}{2T} |X_{2T}(f)|^2 = \frac{1}{2T} \int_{-T}^{+T} \int_{-T}^{+T} x(u)x^*(v) e^{-j2\pi fu} e^{+j2\pi fv} dudv \quad (15.)$$

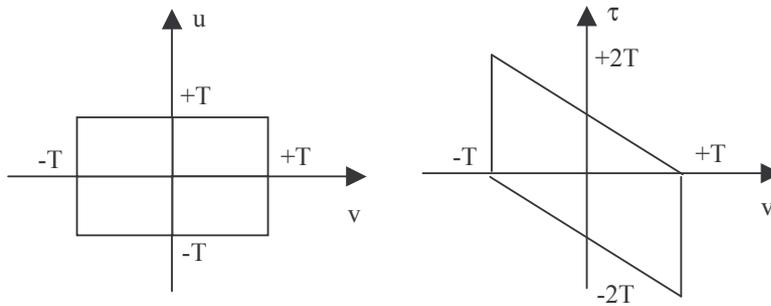
et en prenant l'espérance de cette expression, on obtient :

$$E \left[\frac{1}{2T} |X_{2T}(f)|^2 \right] = \frac{1}{2T} \int_{-T}^{+T} \int_{-T}^{+T} r_{xx}(u-v) e^{-j2\pi f(u-v)} dudv \quad (16.)$$

En effectuant alors le changement de variable suivant

$$\begin{cases} \tau = u - v \\ v' = v \end{cases}$$

et en prenant garde aux intervalles de variation des nouvelles variables, il vient :



Pour $\tau \geq 0$, v' varie de $-T$ à $T - \tau$ et pour $\tau \leq 0$, v' varie de T à $-T - \tau$

$$E \left[\frac{1}{2T} |X_{2T}(f)|^2 \right] = \frac{1}{2T} \int_{\tau=0}^{2T} \int_{v'=-T}^{T-\tau} r_{xx}(\tau) e^{-j2\pi f\tau} d\tau dv' + \int_{\tau=0}^{-2T} \int_{v'=T}^{-T-\tau} r_{xx}(\tau) e^{-j2\pi f\tau} d\tau dv' \quad (17.)$$

$$E \left[\frac{1}{2T} |X_{2T}(f)|^2 \right] = \frac{1}{2T} \int_{\tau=0}^{2T} r_{xx}(\tau)(2T - |\tau|) e^{-j2\pi f\tau} d\tau + \int_{\tau=0}^{-2T} r_{xx}(\tau)(-2T + |\tau|) e^{-j2\pi f\tau} d\tau \quad (18.)$$

d'où finalement :

$$E\left[\frac{1}{2T}|X_{2T}(f)|^2\right] = \int_{\tau=-2T}^{2T} r_{xx}(\tau) \left(1 - \frac{|\tau|}{2T}\right) e^{-j2\pi f\tau} d\tau \quad (19.)$$

En introduisant alors la fonction I_{4T} indicatrice de l'intervalle $[-2T, +2T]$, il vient :

$$E\left[\frac{1}{2T}|X_{2T}(f)|^2\right] = TF \left[r_{xx}(\tau) \left(1 - \frac{|\tau|}{2T}\right) I_{4T} \right] \quad (20.)$$

On obtient donc :

$$E\left[\frac{1}{2T}|X_{2T}(f)|^2\right] = P(f) * TF \left[\left(1 - \frac{|\tau|}{2T}\right) I_{4T} \right] \quad (21.)$$

Lorsque T tend vers l'infini, le deuxième terme du produit de convolution tend vers $\delta(f)$, d'où :

$$\lim_{T \rightarrow \infty} E\left[\frac{1}{2T}|X_{2T}(f)|^2\right] = P(f) \quad (22.)$$

En considérant alors un cas numérique, où l'observation du signal $x(t)$ se résume à N valeurs échantillonnées à la période T_e , la densité spectrale peut être **estimée** en limitant l'intégrale précédente et en oubliant l'espérance mathématique.

d'où :

$$P_{per}(f) = \frac{1}{NT_e} \left| \sum_{n=0}^{N-1} x(nT_e) e^{-j2\pi nT_e f} \right|^2 \quad (23.)$$

Cet estimateur de la densité spectrale de la densité spectrale de puissance du signal $x(nT_e)$ est appelé *périodogramme*.

Il demande, pour être calculé, la mise au carré de la Transformée de Fourier du signal numérique $x(nT_e)$ sur N points. Il est, depuis la mise au point de l'algorithme de Transformation de Fourier Rapide (TFR ou FFT en anglais) par J. Cooley et J. Tuckey en 1965, l'estimateur le plus employé.

Jusqu'à la mise au point de l'algorithme de la FFT, la méthode la plus utilisée consistait à estimer N' valeurs de la fonction d'autocorrélation $r(p)$ avec $N' < N$ et à calculer la Transformée de Fourier Discrète (TFD ou DFT en anglais) sur les N' points obtenus. Cet estimateur de la densité spectrale de puissance, du aux travaux de Blackman et Tuckey, porte le nom de *corrélogramme*.

$$P_{cor}(f) = \sum_{p=-(N'-1)}^{N'-1} r_{xx}(p) e^{-j2\pi pf} \quad (24.)$$

avec :

$$r(p) = \frac{1}{N} \sum_{n=p}^{N-1} x(n)x(n-p)^* \quad (25.)$$

pour $p = 0 \rightarrow N-1$

Pour $N' = N$ les estimateurs du *périodogramme* et du *corrélogramme* sont identiques.

En effet,

$$P_{cor}(f) = \frac{1}{N} \sum_{p=-(N-1)}^{N-1} \sum_{n=p}^{N-1} x(n)x(n-p)^* e^{-j2\pi pf} \quad (26.)$$

d'où:

$$P_{cor}(f) = \frac{1}{N} \sum_{p=-(N-1)}^{N-1} \sum_{n=p}^{N-1} x(n)e^{-j2\pi nf} x(n-p)^* e^{j2\pi(n-p)f} \quad (27.)$$

expression qui peut encore s'écrire:

$$P_{cor}(f) = \frac{1}{N} \left| \sum_{n=0}^{N-1} x(n)e^{-j2\pi nf} \right|^2 = P_{per}(f) \quad (28.)$$

Les deux estimateurs sont donc bien identiques pour $N' = N$.

Propriétés du périodogramme:

Le périodogramme constituant un estimateur de la densité spectrale de puissance du processus $x(nT_e)$, il est nécessaire d'étudier son biais et sa variance.

- Le calcul de l'espérance de $P_{per}(f)$ donne:

$$E[P_{per}(f)] = E \left[\sum_{p=-(N-1)}^{N-1} r_{xx}(p) e^{-j2\pi pf} \right] \quad (29.)$$

avec : $r(p) = \frac{1}{N} \sum_{n=p}^{N-1} x(n)x(n-p)^*$

d'où:

$$E[P_{per}(f)] = \sum_{p=-(N-1)}^{N-1} r_{xx}(p) \frac{N-|p|}{N} e^{-j2\pi pf} \quad (30.)$$

encore grâce aux propriétés de la Transformée de Fourier:

$$E[P_{per}(f)] = P(f) * N \left(\frac{\sin \pi f N}{N \sin \pi f} \right)^2 \quad (31.)$$

L'estimateur du périodogramme est donc une estimation biaisé de la densité spectrale de puissance $P(f)$ du processus $x(nT_e)$. L'estimateur est en fait le résultat du filtrage, dans le domaine fréquentiel de $P(f)$ par le filtre $N\left(\frac{\sin \pi f/N}{N \sin \pi f}\right)^2$. Lorsque N tend vers l'infini ce filtre tend vers $\delta(f)$, le périodogramme est donc asymptotiquement sans biais.

- Le calcul de la variance est délicat et conduit, pour des signaux gaussiens à :

$$Var(P_{per}(f)) = P(f) \left[1 + N \left(\frac{\sin(2\pi f/N)}{2\pi f/N} \right)^2 \right] \quad (32.)$$

Cette variance peut être diminuée en séparant l'ensemble des N valeurs de $x(nT_e)$ en $\frac{N}{K}$ sous ensembles de $K < N$ valeurs. Il est alors possible de calculer K estimateurs et de faire la moyenne des K estimateurs obtenus. La variance est alors divisée par K^2 . Cette amélioration de la variance de l'estimateur se paye par une diminution de résolution de ce dernier. En effet la résolution spectrale est en $\frac{1}{NT_e}$ dans le cas de N échantillons et en $\frac{1}{KT_e}$ dans le cas de K échantillons.

Enfin, les lobes secondaires de $N\left(\frac{\sin \pi f/n}{\pi f/N}\right)^2$ peuvent être atténués en introduisant des fenêtres de pondérations qui vont être appliquées directement sur le signal observé $x(nT_e)$.

En conclusion, le périodogramme est un estimateur de la densité spectrale de puissance qui est d'autant meilleur que le signal est observé sur une longue plage de stationnarité. L'algorithme de la Transformée de Fourier Rapide est bien connu et la plupart des processeurs de signaux sont vendus avec des routines de TFR optimisées. Cet estimateur est donc aisé à utiliser et c'est la raison pour laquelle c'est l'estimateur le plus employé aujourd'hui.

Pour calculer un périodogramme il faut donc:

- observer N valeurs du signal $x(nT_e)$,
- calculer une TFR sur N points avec mise au carré.

2.2.2 Méthode du minimum de variance: méthode de Capon

Pour chaque fréquence f , cette méthode cherche un *filtre adapté* dont la réponse vaut 1 pour cette fréquence f et 0 partout ailleurs. Une fois ce filtre obtenu l'estimateur $P_{Capon}(f)$ n'est autre que la puissance de sortie du filtre obtenu pour la fréquence f .

La sortie $y(n)$ de ce filtre s'écrit :

$$y(n) = \sum_{i=0}^N h_i^* x(n-i) \quad (33.)$$

ou plus simplement sous forme vectorielle :

$$y(n) = H^T X(n) \quad (34.)$$

$$\text{avec } H = \begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_N \end{bmatrix} \text{ et } X(n) = \begin{bmatrix} x(n) \\ x(n-1) \\ \vdots \\ x(n-N+1) \end{bmatrix}$$

Le filtre H doit donc minimiser $E[y(n)^2]$ avec la contrainte $H^T F = 1$

avec :

$$F = \begin{bmatrix} 1 \\ e^{j2\pi f} \\ e^{j4\pi f} \\ \vdots \\ e^{j2N\pi f} \end{bmatrix} \quad (35.)$$

Ce qui s'écrit, en utilisant un multiplicateur de Lagrange:

H optimal est tel que $E[H^T X(n)]^2 + \alpha(l - H^T F)$ est minimal

soit, en annulant la dérivé par rapport à H :

$$H = \frac{\alpha}{2} R_N^{-1} F \quad (36.)$$

avec $R_N = E[X(n)X(n)^T]$ matrice d'autocorrélation du signal $x(n)$.

en introduisant α dans l'expression de la contrainte $H^T F = 1$, il vient :

$$\alpha = \frac{2}{F^T R_N^{-1} F} \quad (37.)$$

d'où finalement :

$$H = \frac{R_N^{-1} F}{F^T R_N^{-1} F} \quad (38.)$$

Pour estimer $P_{Cap}(f)$ il ne reste plus qu'à calculer:

$$P_{Cap}(f) = E[y(n)^2] = E[H^T X(n)X(n)^T H] \quad (39.)$$

$$P_{Cap}(f) = \frac{F^T R_N^{-1} R_N R_{N+1}^{-1} F}{F^T R_N^{-1} F F^T R_N^{-1} F} \quad (40.)$$

$$P_{Cap}(f) = \frac{I}{F^T R_N^{-1} F} \quad (41.)$$

or :

$$F^T R_N^{-1} F = [1, e^{-j2\pi f}, e^{-j4\pi f}, \dots, e^{-j2N\pi f}] \begin{matrix} \rho_{kl} \\ \left[\begin{array}{c} I \\ e^{j2\pi f} \\ e^{j2N\pi f} \end{array} \right] \end{matrix} \quad (42.)$$

d'où:

$$P_{Cap}(f) = \frac{I}{\sum_{k=0}^N \sum_{l=0}^N \rho_{kl} e^{-j2\pi(k-l)f}} \quad (43.)$$

En conclusion pour obtenir l'estimateur de Capon, il faut:

- observer N échantillons du signal $x(nT_e)$
- estimer la matrice d'autocorrélation et l'inverser pour obtenir les termes ρ_{kl}
- calculer $P_{Cap}(f)$ pour chaque fréquence f .

Cette méthode souffre donc d'un coût de calcul supérieur aux estimateurs précédents. Il est possible de démontrer que cet estimateur à une variance minimale et c'est la raison pour laquelle il est souvent appelé estimateur du minimum de variance.

2.3 Estimation spectrale par décomposition harmonique

2.3.1 Méthode de Pisarenko:

Dans cette méthode, le signal $x(n)$ est supposé être constitué d'une somme de N sinusoides $s(n)$ et d'un bruit blanc additif $b(n)$.

$$x(n) = s(n) + b(n) \quad (44.)$$

Sachant que toute sinusoides pure $\sin(n\omega)$ peut s'écrire:

$$\sin(n\omega) = 2 \cos \omega \sin(n-1)\omega - \sin(n-2)\omega$$

il est possible d'écrire:

$$s(n) = - \sum_{m=1}^{2N} a_m s(n-m)$$

d'où:

$$x(n) = - \sum_{m=1}^{2N} a_m s(n-m) + b(n)$$

en remplaçant $s(n-m)$ par $x(n-m) - b(n-m)$, il vient:

$$\sum_{m=0}^{2N} a_m x(n-m) = \sum_{m=0}^{2N} a_m b(n-m)$$

ce qui peut s'écrire matriciellement:

$$X(n)^T A = B(n)^T A$$

avec:

$$X(n)^T = [x(n), x(n-1), \dots, x(n-2N)]^T$$

$$B(n)^T = [b(n), b(n-1), \dots, b(n-2N)]^T$$

et :

$$A = \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{2N} \end{bmatrix}$$

d'où:

$$E[X(n)X(n)^T]A = E[X(n)B(n)^T]A = E[(S(n) + B(n))B(n)^T]A$$

$$R_{2N+1}A = E[(S(n) + B(n))B(n)^T]A$$

Or, le bruit est supposé blanc, de variance σ^2 et décorrélé du mélange de sinusoides, cette équation devient donc:

$$R_{2N+1}A = \sigma^2 I A = \sigma^2 A$$

Le vecteur A est donc le vecteur propre associé à la valeur propre σ^2 avec la contrainte $a_0 = 1$.

Ayant le vecteur A , on peut écrire la transformée en Z de l'équation:

$$s(n) = - \sum_{m=1}^{2N} a_m s(n-m)$$

$$S(Z) \left(1 - \sum_{m=1}^{2N} a_m Z^{-m} \right) = 0$$

les valeurs de Z pour lesquelles cette équation est vérifiée donnent les valeurs des fréquences présentes dans le

mélange. Il faut donc extraire les racines du polynôme $1 - \sum_{m=1}^{2N} a_m Z^{-m}$ pour obtenir les N valeurs de fréquences.

Si N et σ^2 ne sont pas connus a priori, il suffit de surdimensionner la matrice R et d'analyser ses valeurs propres. Théoriquement celles ci doivent, au bout de N ordres, atteindre une valeur constante égale à σ^2 . Il existe de nombreux critères qui permettent de détecter ce blocage à σ^2 des valeurs propres.

Enfin, il est possible d'obtenir les puissances des sinusoïdes détectées. En effet, il est facile de vérifier que, dans le cas d'un mélange de N sinusoïdes de pulsation ω_i et de puissance P_i , on a :

$$\begin{aligned} r(1) &= P_1 \cos \omega_1 + P_2 \cos \omega_2 + \dots + P_N \cos \omega_N \\ r(2) &= P_1 \cos 2\omega_1 + P_2 \cos 2\omega_2 + \dots + P_N \cos 2\omega_N \\ &\dots \\ r(N) &= P_1 \cos N\omega_1 + P_2 \cos N\omega_2 + \dots + P_N \cos N\omega_N \end{aligned}$$

disposant des pulsations ω_i et des coefficients d'autocorrélation $r(i)$, il "suffit" de résoudre ce système pour trouver les puissances respectives P_i des raies spectrales identifiées.

En conclusion pour analyser un signal selon la méthode de Pisarenko, il faut:

- observer N valeurs du signal $x(nT_e)$,
- calculer la matrice d'autocorrélation et en faire la décomposition en éléments propres,
- détecter σ^2 et en déduire le nombre de sinusoïdes,
- extraire les racines d'un polynôme complexe de degré $2N$,
- enfin si l'on veut les puissances, résoudre un système réel de N équations à N inconnues.

La décomposition en éléments propres reste pour l'instant l'étape la plus délicate à réaliser de manière rapide et c'est le frein principal à l'emploi de cette méthode.

2.3.2 Méthode de Prony:

Dans cette méthode le signal est, comme dans la méthode de Pisarenko, supposé être constitué d'une somme de N sinusoïdes mais le bruit est remplacé par un amortissement sur ces dernières. L'hypothèse de départ s'écrit donc :

$$x(n) = \sum_{m=1}^N b_m Z_m^n$$

avec :

$$Z_m = e^{\alpha_m} e^{j2\pi f_m}$$

On peut alors fabriquer le polynôme :

$$\Psi(Z) = \prod_{k=1}^N (Z - Z_k) = \sum_{i=0}^N a_i Z^{N-i} \quad \text{avec } a_0 = 1$$

D'après l'hypothèse de départ sur $x(n)$ on a :

$$x(n-m) = \sum_{l=1}^N b_l Z_l^{n-m}$$

en multipliant cette équation par a_m et en effectuant une sommation sur m , il vient :

$$\sum_{m=0}^N a_m x(n-m) = \sum_{m=0}^N a_m \sum_{l=1}^N b_l Z_l^{n-m}$$

ou encore :

$$\sum_{m=0}^N a_m x(n-m) = \sum_{l=1}^N b_l Z_l^{n-N} \sum_{m=0}^N a_m Z_l^{N-m}$$

or $\sum_{m=0}^N a_m Z_l^{N-m} = 0$ par définition des coefficients a_m du polynôme $\psi(Z)$ dont les racines sont les Z_l , d'où :

$$\sum_{m=0}^N a_m x(n-m) = 0$$

et donc :

$$x(n) = - \sum_{m=1}^N a_m x(n-m)$$

Les coefficients a_m peuvent donc être obtenus par la résolution du système linéaire de dimension N suivant :

$$\begin{bmatrix} x(N+1) \\ \cdot \\ \cdot \\ x(2N) \end{bmatrix} = \begin{bmatrix} x(N) & \cdot & \cdot & x(1) \\ x(N+1) & \cdot & \cdot & x(2) \\ \cdot & \cdot & \cdot & \cdot \\ x(2N) & \cdot & \cdot & x(N) \end{bmatrix} \begin{bmatrix} a_1 \\ \cdot \\ \cdot \\ a_N \end{bmatrix}$$

A partir des N coefficients a_m il est possible de former le polynôme :

$$\psi(Z) = \sum_{m=0}^N a_m Z^{m-i} \text{ avec } a_0 = 1$$

et d'en extraire les N racines complexes. Les modules de ces racines donnent alors les affaiblissements α_m tandis que les phases donnent les fréquences f_m .

Les amplitudes respectives b_m des différentes sinusoides peuvent enfin être obtenues en résolvant le système linéaire suivant :

$$\begin{bmatrix} 1 & 1 & \cdot & 1 \\ Z_1 & Z_2 & \cdot & Z_N \\ \cdot & \cdot & \cdot & \cdot \\ Z_1^{N-1} & \cdot & \cdot & Z_N^{N-1} \end{bmatrix} \begin{bmatrix} b_1 \\ \cdot \\ \cdot \\ b_N \end{bmatrix} = \begin{bmatrix} x_0 \\ \cdot \\ \cdot \\ x_{N-1} \end{bmatrix}$$

Pour cette méthode il faut donc :

- observer $2N$ valeurs du signal $x(nT_e)$
- résoudre un système linéaire complexe de dimension N

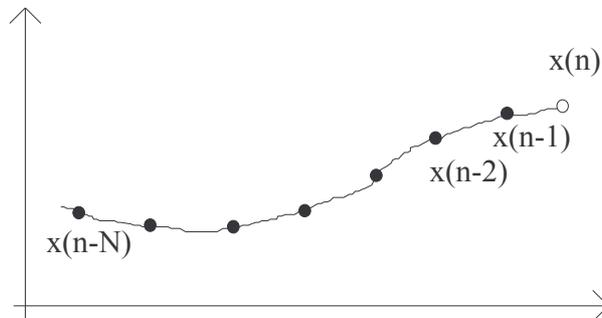
- extraire les racines d'un polynôme complexe de degré N
- et si l'on veut les amplitudes résoudre un système linéaire complexe de dimension N .

2.4 Estimation spectrale paramétrique

A la différence de l'estimation spectrale non paramétrique, qui ne faisait aucune hypothèse sur le signal observé $x(n)$, si ce n'est des propriétés de stationnarité à l'ordre 2, l'estimation spectrale paramétrique suppose que ce signal suit un modèle donné. Ce modèle comporte un certain nombre de paramètres qui sont adaptables en fonction du signal observé. Il existe principalement trois grandes classes de modèles:

Le modèle auto régressif - modèle AR -

Le signal $x(n)$ est supposé être prédictible en fonction d'un certain nombre de ses valeurs antérieures.



Il peut donc s'écrire:

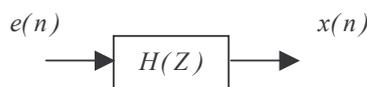
$$x(n) = \sum_{i=1}^N a_i x(n-i) + e(n)$$

équation où les coefficients $(a_i)_{i=1,N}$ constituent les paramètres du modèle et où $e(n)$ est un bruit blanc décorrélé de $x(n)$ de variance σ^2 et qui représente l'erreur de prédiction.

La transformée en Z de cette équation donne alors:

$$X(Z) \left(1 - \sum_{i=1}^N a_i Z^{-i} \right) = E(Z)$$

Le signal $x(n)$ peut donc être vu comme le résultat du passage d'un bruit blanc $e(n)$ de variance σ^2 à travers un filtre de fonction de transfert $H(Z)$.



avec :
$$H(Z) = \frac{1}{1 - \sum_{i=1}^N a_i Z^{-i}}$$

La connaissance des paramètres a_i permet donc de calculer la densité spectrale de puissance $P(f)$:

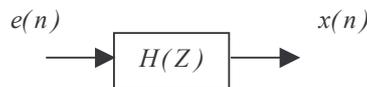
$$P_{ar}(f) = \frac{\sigma^2}{\left| 1 - \sum_{i=1}^N a_i e^{-j2\pi fi} \right|^2}$$

Le modèle à moyenne ajustée - modèle MA -

Le signal $x(n)$ est supposé pouvoir s'écrire comme une combinaison linéaire d'échantillons décorrélés entre eux, ce qui peut se formaliser comme une combinaison linéaire d'échantillons d'un bruit blanc $e(n)$:

$$x(n) = \sum_{i=0}^M b_i e(n-i)$$

Le signal $x(n)$ peut donc être vu comme le résultat du passage d'un bruit blanc $e(n)$ à travers un filtre de fonction de transfert $H(Z)$.



avec :
$$H(Z) = \sum_{i=0}^M b_i Z^{-i}$$

La densité spectrale de puissance du signal $x(n)$ s'écrit alors :

$$P_{ma}(f) = \left| \sum_{i=0}^M b_i e^{-j2\pi fi} \right|^2 \sigma^2$$

Le modèle auto régressif à moyenne ajustée - modèle ARMA-

Combinaison des modèles AR et MA où le signal $x(n)$ est supposé pouvoir s'écrire en fonction de N valeurs passées et de M échantillons d'un bruit blanc décorrélé.

$$x(n) = \sum_{i=1}^N a_i x(n-i) + \sum_{i=0}^M b_i e(n-i)$$

d'où:

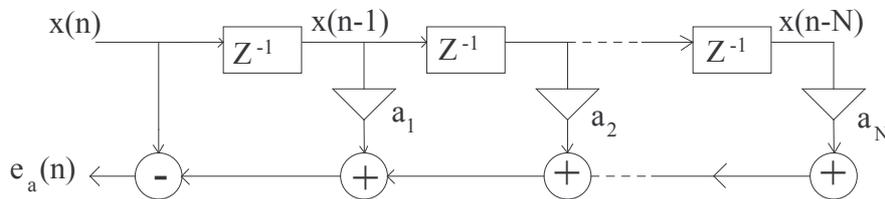
$$P_{arma}(f) = \left| \frac{\sum_{i=0}^M b_i e^{-j2\pi f i}}{1 - \sum_{i=1}^N a_i e^{-j2\pi f i}} \right|^2 \sigma^2$$

2.4.1 Modélisation AR

Pour effectuer cette modélisation il faut donc trouver les paramètres a_i du modèle:

$$x(n) = \sum_{i=1}^N a_i x(n-i) + e_a(n)$$

modèle qui correspond à la structure RIF suivante:



(l'erreur $e(n)$ est ici notée $e_a(n)$ pour traduire qu'il s'agit d'une erreur avant correspondant à une prédiction de $x(n)$ à partir de $\{x(n-1) \ x(n-2) \ \dots \ x(n-N)\}$)

Il est alors possible d'optimiser un critère de minimisation d'erreur quadratique, c'est à dire, de chercher le jeu de paramètres a_i qui minimisent $E[e_a(n)^2]$. Ce qui peut s'écrire matriciellement de la manière suivante:

$$A_N \text{ optimal est tel qu'il minimise: } E[e_a(n)^2] = E\left[\left(x(n) - X_N^T(n-1)A_N\right)^2\right]$$

avec :

$$X_N(n-1) = [x(n-1), x(n-2), \dots, x(n-N)]^T$$

et :

$$A_N = [a_1, a_2, \dots, a_N]^T$$

En développant $E[e_a(n)^2]$ il vient:

$$E[e_a(n)^2] = E\left[\left(x(n) - X_N^T(n-1)A_N\right)^T \left(x(n) - X_N^T(n-1)A_N\right)\right]$$

$$E[e_a(n)^2] = E\left[x(n)^2 - A_N^T X_N(n-1)x(n) - x(n)X_N^T(n-1)A_N + A_N^T X_N(n-1)X_N^T(n-1)A_N\right]$$

En annulant la dérivée de cette expression par rapport à A_N il vient :

$$A_N = E[X_N(n-1)X_N^T(n-1)]^{-1} E[X_N(n-1)x^*(n)]$$

ou encore:

$$A_N = R_N^{-1} r_N^a \quad (1)$$

avec:

$$R_N = \begin{bmatrix} r(0) & r(1) & \dots & r(N-1) \\ r^*(1) & r(0) & \dots & \cdot \\ \cdot & \cdot & \cdot & r(1) \\ r^*(N-1) & \cdot & r^*(1) & r(0) \end{bmatrix} \text{ matrice d'autocorrélation du signal et } r_N^a = \begin{bmatrix} r(1) \\ r(2) \\ \cdot \\ r(N) \end{bmatrix}$$

En posant $E_{aN} = E[e_a(n)^2]$ et en utilisant (1), il vient:

$$E_{aN} = E[x(n)^2] - 2E\left[\left(X_N^T(n-1)x(n)\right)A_N\right] + A_N^T E\left[X_N(n-1)X_N^T(n-1)\right]A_N$$

$$E_{aN} = E\left[e_a(n)^2\right] = E\left[\left(x(n) - X_N^T(n-1)A_N\right)^2\right]$$

$$E_{aN} = r(0) - 2r_N^{aT} A_N + r_N^{aT} A_N$$

$$E_{aN} = r(0) - r_N^{aT} A_N \quad (2)$$

En écrivant (1) et (2) sous une forme matricielle unique, il vient:

$$\begin{bmatrix} r(0) & r_N^{aT} \\ r_N^a & R_N \end{bmatrix} \begin{bmatrix} I \\ -A_N \end{bmatrix} = \begin{bmatrix} E_{aN} \\ 0 \end{bmatrix} \quad (3)$$

équation de Yule Walker avant (ou directe ou forward) dans le cas stationnaire

Il est donc possible de trouver le vecteur A_N en inversant la matrice R_N puis de calculer l'énergie d'erreur de prédiction avant. Cependant, le coût de calcul d'une telle approche est alors en $o(N^3)$ ce qui peut s'avérer gênant

lorsque N est important. Il est possible de résoudre cette équation avec un coût de calcul proportionnel à $o(N^2)$ en utilisant l'algorithme de Levinson.

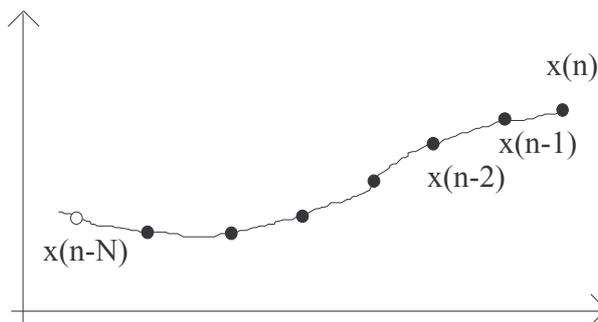
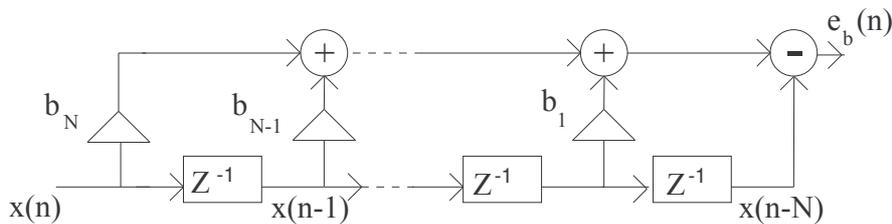
Algorithme de Levinson :

Pour faire passer le coût de calcul de la résolution de l'équation de Yule Walker de $o(N^2)$ à $o(N^3)$, cet algorithme va utiliser une récurrence sur l'ordre N du modèle prédicteur AR. Pour cela, il est nécessaire d'introduire l'erreur de prédiction arrière.

Le modèle de prédiction avant consistait à estimer $x(n)$ à partir de $\{x(n-1) \ x(n-2) \ \dots \ x(n-N)\}$, en inversant l'axe des temps, on peut construire un modèle de prédiction arrière qui va estimer $x(n-N)$ à partir de $\{x(n-N+1) \ \dots \ x(n)\}$. Cela correspond à l'équation de prédiction arrière suivante :

$$x(n-N) = \sum_{i=1}^N b_{N+1-i} x(n+1-i) + e_b(n)$$

modèle, dont la structure RIF est la suivante:



L'équation de prédiction arrière s'écrit de manière matricielle:

$$e_b(n) = x(n-N) - X_N^T(n)B_N$$

avec $X_N(n) = [x(n), x(n-1), \dots, x(n-N+1)]^T$

et $B_N = [b_N, b_{N-1}, \dots, b_1]^T$

Le vecteur B_N optimal est tel qu'il minimise $E_{b_N} = E[e_b(n)^2]$

ce qui conduit à: $B_N = E[X_N(n)X_N^T(n)]^{-1} E[X_N(n)x(n-N)]$

d'où:

$$B_N = R_N^{-1} r_N^b \quad (4)$$

avec $r_N^b = [r(N), r(N-1), \dots, r(1)]$

en reportant B_N dans l'expression de $E_{b_N} = E[e_b(n)^2]$ il vient:

$$E[e_b(n)^2] = E[x(n-N)^2] - 2E[x(n-N)X_N(n)]B_N + B_N^T E[X_N(n)X_N^T(n)]B_N$$

d'où:

$$E_{b_N} = r(0) - r_N^{bT} B_N \quad (5)$$

En réunissant (4) et (5) au sein d'une même équation matricielle, il vient:

$$\begin{bmatrix} R_N & r_N^b \\ r_N^{bT} & r(0) \end{bmatrix} \begin{bmatrix} -B_N \\ I \end{bmatrix} = \begin{bmatrix} 0 \\ E_{b_N} \end{bmatrix} \quad (6)$$

Equation de Yule Walker arrière (ou rétrograde ou backward) dans le cas stationnaire

L'algorithme de Levinson est obtenu en réunissant les équations avant et arrière.

- Il est aisé de vérifier $E_{a_N} = E_{b_N}$

En effet, l'équation (6) peut s'écrire:

$$R_{N+1} \begin{bmatrix} -B_N \\ I \end{bmatrix} = \begin{bmatrix} 0 \\ E_{b_N} \end{bmatrix}$$

en multipliant à gauche par la coidentité :

$$J_N = \begin{bmatrix} 0 & 0 & I \\ 0 & I & 0 \\ I & 0 & 0 \end{bmatrix}$$

et en remarquant que, du fait de la symétrie de R_N :

$$J_N R_N = R_N J_N$$

il vient:

$$R_{N+1} \begin{bmatrix} I \\ -JB_N \end{bmatrix} = \begin{bmatrix} E_{bN} \\ 0 \end{bmatrix}$$

en identifiant avec l'équation avant (3):

$$R_{N+1} \begin{bmatrix} I \\ -A_N \end{bmatrix} = \begin{bmatrix} E_{aN} \\ 0 \end{bmatrix}$$

il vient: $A_N = JB_N$ et $E_{bN} = E_{aN} = E_N$

Ce qui revient à dire que prédire $x(n-N)$ à partir de $x(n-N+1), \dots, x(n)$ est identique à prédire $x(n)$ à partir de $x(n-1), \dots, x(n-N)$. La seule différence entre les deux prédictions est le sens de parcours sur la trajectoire de $x(n)$.

- L'algorithme se base sur une récursion sur l'ordre N . Si on connaît A_{N-1} et B_{N-1} à l'ordre $N-1$, il vient pour la prédiction avant à l'ordre N :

$$\begin{bmatrix} R_N & r_N^b \\ r_N^{bT} & r(0) \end{bmatrix} \begin{bmatrix} I \\ -A_{N-1} \\ 0 \end{bmatrix} = \begin{bmatrix} E_{N-1} \\ 0 \\ K_N \end{bmatrix}$$

avec $K_N = r(N) - \sum_{i=1}^{N-1} a_{i,N-1} r(N-i)$

et pour la prédiction arrière:

$$\begin{bmatrix} r(0) & r_N^a \\ r_N^a & R_N \end{bmatrix} \begin{bmatrix} 0 \\ -B_{N-1} \\ I \end{bmatrix} = \begin{bmatrix} K_N \\ 0 \\ E_{N-1} \end{bmatrix}$$

d'où:

$$R_{N+1} \begin{bmatrix} I \\ -A_{N-1} \\ 0 \end{bmatrix} = \begin{bmatrix} E_{N-1} \\ 0 \\ K_N \end{bmatrix} \quad \text{et} \quad R_{N+1} \begin{bmatrix} 0 \\ -B_{N-1} \\ I \end{bmatrix} = \begin{bmatrix} K_N \\ 0 \\ E_{N-1} \end{bmatrix}$$

$$\text{d'où: } R_{N+1} \left[\begin{bmatrix} I \\ -A_{N-1} \\ 0 \end{bmatrix} - \frac{K_N}{E_{N-1}} \begin{bmatrix} 0 \\ -B_{N-1} \\ I \end{bmatrix} \right] = \begin{bmatrix} E_{N-1} - \frac{K_N^2}{E_{N-1}} \\ 0 \end{bmatrix}$$

d'où:

$$[A_N] = \begin{bmatrix} A_{N-1} \\ 0 \end{bmatrix} - k_N \begin{bmatrix} B_{N-1} \\ -I \end{bmatrix}$$

et $E_N = E_{N-1}(1 - k_N^2)$ avec $k_N = \frac{K_N}{E_{N-1}}$

- Algorithme de Levinson -

Ce qui correspond aux relations suivantes:

$$E(0) = r(0)$$

$$j : 1 \rightarrow N \begin{cases} k_j = \frac{1}{E_{j-1}} \left[r(j) - \sum_{i=1}^{j-1} a_{i,j-1} r(j-i) \right] \\ a_{j,j} = k_j \\ i : 1 \rightarrow j-1 \left\{ a_{i,j} = a_{i,j-1} - k_j a_{j-i,j-1} \right. \\ E_j = E_{j-1} (1 - k_j^2) \end{cases}$$

Remarque:

il est possible d'écrire: $\begin{pmatrix} I \\ -A_N \end{pmatrix} = \begin{pmatrix} I \\ -A_{N-1} \\ 0 \end{pmatrix} - k_N \begin{pmatrix} 0 \\ -B_{N-1} \\ I \end{pmatrix}$

et: $\begin{pmatrix} -B_N \\ I \end{pmatrix} = \begin{pmatrix} 0 \\ -B_{N-1} \\ I \end{pmatrix} - k_N \begin{pmatrix} I \\ -A_{N-1} \\ 0 \end{pmatrix}$

d'où:

$$A_N = \begin{pmatrix} A_{N-1} \\ 0 \end{pmatrix} - k_N \begin{pmatrix} B_{N-1} \\ -I \end{pmatrix}$$

et:

$$B_N = \begin{pmatrix} 0 \\ B_{N-1} \end{pmatrix} - k_N \begin{pmatrix} -I \\ A_{N-1} \end{pmatrix}$$

d'où en reprenant les expressions des erreurs avants et arrières:

$$e_{a_N}(n) = x(n) - X_N^T(n-1)A_N$$

$$e_{a_N}(n) = x(n) - X_N^T(n-1) \left[\begin{bmatrix} A_{N-1} \\ 0 \end{bmatrix} - k_N \begin{bmatrix} B_{N-1} \\ -I \end{bmatrix} \right]$$

$$e_{aN}(n) = e_{aN-1}(n) - k_N X_N^T(n-1) \begin{bmatrix} -B_{n-1} \\ I \end{bmatrix}$$

or: $e_{bN}(n) = x(n-N) - X_N^T(n)B_N$

$$e_{bN-1}(n) = x(n-N+1) - X_{N-1}^T(n)B_{N-1}$$

$$e_{bN-1}(n-1) = x(n-N) - X_{N-1}^T(n-1)B_{N-1}$$

$$\Rightarrow e_{aN}(n) = e_{aN-1}(n) - k_N e_{bN-1}(n-1)$$

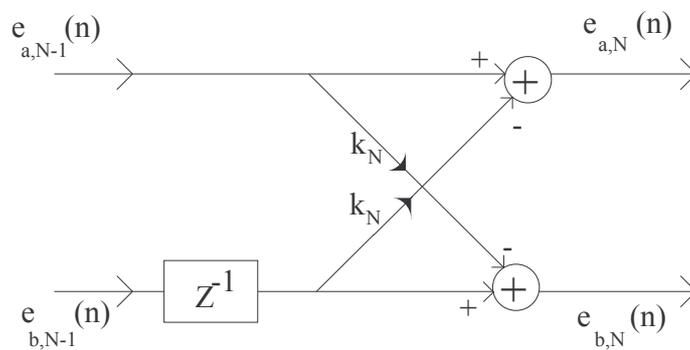
de la même manière :

$$e_{bN}(n) = x(n-N) - X_N^T(n)B_N$$

conduit à :

$$e_{bN}(n) = e_{bN-1}(n-1) - k_N e_{aN-1}(n)$$

Les coefficients k_i correspondent donc à la structure en treillis suivante:



Enfin, on peut démontrer:

$$k_N = \frac{E[e_{a,N-1}(n)e_{b,N-1}(n)]}{E_{N-1}}$$

d'où le nom de coefficients de corrélation partielle (PARCOR) donné souvent aux coefficients k_i .

En conclusion, la modélisation AR demande la mise à jour de N coefficients a_i pour cela il faut:

- observer au minimum N valeurs du signal,
- calculer les coefficients d'autocorrélation,
- appliquer l'algorithme de Levinson (coût de calcul proportionnel à $O(N^2)$).

2.4.2 Modélisation MA

Dans le cas du modèle AR, le vecteur A_N s'obtient en optimisant un critère d'erreur quadratique. Cette optimisation conduit de manière explicite à la résolution d'un système linéaire:

$$A_N = R_N^{-1} r_N^a$$

Par contre, pour un modèle MA:

$$x(n) = \sum_{i=0}^M b_i e(n-i)$$

l'erreur de prédiction s'écrit:

$$e(n) = x(n) - E_N^T(n-1)B_N$$

avec $E_N(n-1) = [e(n-1), e(n-2), \dots, e(n-M)]^T$

Le vecteur E_N dépend de B_N et le système à résoudre devient non linéaire. L'optimisation exacte du critère est alors très délicate. Cependant, l'algorithme de Durbin permet d'approcher la solution optimale avec de bons résultats.

Algorithme de Durbin:

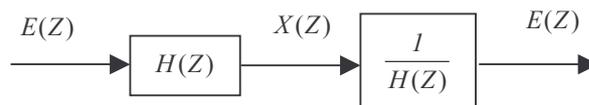
Le principe de cet algorithme consiste à identifier le modèle MA d'ordre M avec un modèle AR d'ordre $N \gg M$. En effet, tout modèle MA peut être identifié à un modèle AR d'ordre infini:

$$\sum_{i=0}^M b_i Z^{-i} = \frac{1}{\sum_{i=0}^{\infty} a_i Z^{-i}}$$

en remplaçant la borne infinie par une valeur $N \gg M$, il est possible de faire l'approximation suivante:

$$H(Z) = \sum_{i=0}^M b_i Z^{-i} \approx \frac{1}{\sum_{i=0}^N a_i Z^{-i}}$$

d'où :



$$\Rightarrow \sum_{i=0}^M b_i Z^{-i} \frac{1}{1 - \sum_{i=1}^N a_i Z^{-i}} = 1$$

ou encore dans le domaine temporel :

$$b_i * a_i = \delta_i$$

ce qui donne $\delta_j = \sum_{i=0}^M b_i a_{j-i}$, en posant $b_0 = 1$ il vient:

$$\delta_j = a_j + \sum_{i=1}^M b_i a_{j-i}$$

et donc:

$$\text{pour } j = 1 \text{ à } N : a_j + \sum_{i=1}^M b_i a_{j-i} = 0$$

Les coefficients b_i du modèle MA d'ordre M du signal $x(n)$ sont donc aussi les coefficients du modèle AR d'ordre M du signal a_n .

Conclusion, l'obtention des M valeurs b_i se fait en résolvant deux systèmes AR, il faut:

- modéliser le signal $x(n)$ sous la forme d'un processus AR d'ordre $N \gg M$ et trouver le vecteur des paramètres, grâce à l'algorithme de Levinson par exemple,
- modéliser le signal constitué par les paramètres précédents sous une forme AR d'ordre M et trouver le nouveau vecteur des paramètres.

2.4.3 Modélisation ARMA

La modélisation ARMA peut se décomposer en une modélisation AR suivie d'une modélisation MA.

En effet, il faut identifier le filtre:

$$H(Z) = \frac{\sum_{i=0}^M b_i Z^{-i}}{1 - \sum_{i=1}^N a_i Z^{-i}} = \left(\frac{1}{1 - \sum_{i=1}^N a_i Z^{-i}} \right) \left(\sum_{i=0}^M b_i Z^{-i} \right) = H_1(Z) H_2(Z)$$

$H_1(Z)$ est identifiable de manière exacte au moyen de l'algorithme de Levinson tandis que $H_2(Z)$ est identifiable de manière approchée au moyen de l'algorithme de Durbin.

En conclusion, pour identifier les paramètres d'un modèle MA il faut:

- identifier un modèle AR d'ordre N,

- filtrer le signal $x(n)$ par le filtre mis à jour,
- identifier un modèle MA d'ordre M à partir du signal filtré.

Les modélisations AR, MA et ARMA permettent finalement d'obtenir la densité spectrale de puissance du signal $x(n)$ grâce à l'identification des paramètres du modèle considéré.

- Le modèle AR ne comporte que des pôles (filtre tout pôles) et ne passe jamais par zéro:

$$P_{ar}(f) = \frac{\sigma^2}{\left| 1 - \sum_{i=1}^N a_i e^{-j2\pi fi} \right|^2}$$

Ce modèle est bien adapté aux signaux composés de raies pures dans du bruit blanc.

- Le modèle MA ne comporte que des zéros (filtre tout zéros):

$$P_{ma}(f) = \left| \sum_{i=0}^M b_i e^{-j2\pi fi} \right|^2 \sigma^2$$

Ce modèle est bien adapté aux signaux dont la puissance est nulle dans certaines bandes de fréquences.

- Le modèle ARMA possédant à la fois des pôles et des zéros:

$$P_{arma}(f) = \frac{\left| \sum_{i=0}^M b_i e^{-j2\pi fi} \right|^2}{\left| 1 - \sum_{i=1}^N a_i e^{-j2\pi fi} \right|^2} \sigma^2$$

il est adapté à n'importe quel type de signaux.

2.5 Conclusion

Il existe un grand nombre de méthodes d'analyse spectrale et le choix est souvent dicté par l'application considérée. En général, les méthodes non paramétriques trouvent leur intérêt en présence de signaux longs et stationnaires tandis que les méthodes paramétriques sont plutôt utilisées pour les signaux brefs et non stationnaires.

Enfin, les méthodes de modélisation AR peuvent être rendues adaptatives, ce qui permet une analyse temps réel de la densité spectrale de puissance du signal observé.