

TRAITEMENT DU SIGNAL

TRAITEMENT DU SIGNAL

Opérations - Algorithmes

Plan du cours

1. Signaux déterministes. Signaux aléatoires. Systèmes stochastiques

SIGNAUX DETERMINISTES

*Convolution - Reponse Impulsionnelle. Fonction de Transfert. Déconvolution.
Corrélation : Fonction d'intercorrélacion - Fonction d'autocorrélacion*

SIGNAUX ALEATOIRES

L'aléatoire en théorie du signal

Variables Aléatoires

*VA discrètes et continues. Moment d'ordre n . Loi de probabilité conjointe de 2 VA. Indépendance de 2 VA.
Densité de probabilité. Fonction de répartition. Moyenne d'une VA. Variance d'une VA. Covariance de 2 VA.
Autocovariance (\equiv variance) d'une VA. Corrélation de 2 VA. Autocorrélacion d'une VA.
Estimation de la densité de probabilité*

Lois de probabilité de VA usuelles

*Loi de Bernouilli. Loi binomiale. Loi de Poisson. Loi uniforme. Loi normale.
Lois de Rayleigh, de Laplace, de Cauchy*

Théorème central limite

Processus aléatoires - Processus aléatoires à Temps Continu et à Temps Discret

*Trajectoire d'un Processus Aléatoire. Moyenne d'un Processus Aléatoire.
Fonction d'Autocovariance d'un Processus Aléatoire. Fonction de Covariance de 2 Processus Aléatoires*

Stationnarité du 2nd ordre au sens large des processus aléatoires à TC et à TD

*Fonction d'Autocovariance d'un Processus Aléatoire SSL. Fonction de Covariance de 2 Processus Aléatoires SSL.
Densité Spectrale de Puissance (DSP) d'un Processus Aléatoire. Bruit blanc. Ergodicité. Stationnarité et ergodicité.
Formule du filtrage*

Modèles de processus aléatoires

*Le processus de Wiener. Le processus AR. Le processus MA. Le processus ARMA. Le processus ARMAX.
Le processus de Box et Jenkins (BJ). Le processus Markovien et modèle d'Etat*

ANNEXE

SYSTEMES STOCHASTIQUES

Transmission d'un signal aléatoire dans un système linéaire (utilisation des transformées)

Valeur moyenne de la sortie. Intercorrélacion. Autocorrélacion et DSP (Densité Spectrale de Puissance)

Processus générateur d'un signal aléatoire : filtres formeurs du 1er ordre

Processus générateur à Temps Discret (TD) du 1er ordre. Processus générateur à Temps Continu (TC) du 1er ordre

Processus générateurs : MA, AR, ARMA

Signal MA. Signal AR. Equations de Yule-Walker. Signal ARMA

Conclusion

ESTIMATION DE LA COVARIANCE

Méthode par ergodicité. Méthode des corrélacions. Méthode des covariances

RESOLUTION DU SYSTEME D'EQUATIONS DE YULE-WALKER PAR INVERSION MATRICIELLE

ALGORITHME DE LEVINSON

Résolution du système de Yule-Walker

METHODE DE BURG

Résolution du système de Yule-Walker

2. Synthèse du signal. Identification

Synthèse du signal

SIGNAUX DETERMINISTES

Expression analytique. Modèle. Table de look-up. Transformation mathématique inverse

SIGNAUX ALEATOIRES

Fonction modulo. Génération d'un bruit blanc de loi uniforme, de Rayleigh, de loi normale. Fonction OU exclusif

Identification

MODELES AR, MA ARMA

Processus MA d'ordre M

Relations entre coefficients du modèle et covariances

Processus AR d'ordre N

Relations entre coefficients du modèle et covariances (cas $N = 1$)

Relations entre coefficients du modèle et covariances (cas N quelconque)

Equations de Yule-Walker

Processus ARMA d'ordre $N - M$

Remarques

Application : Estimation du spectre d'un processus AR (Analyse spectrale - Estimation)

Relations entre coefficients du modèle et covariances estimées (cas $N = 1$)

Relations entre coefficients du modèle et covariances estimées (cas N quelconque)

Estimation du spectre d'un processus AR méthode haute résolution

3. Caractérisation (Analyse - Transformations fréquentielles). Estimation

Caractérisation

TRANSFORMATIONS FREQUENTIELLES

Transformation de Fourier Discrète (TFD)

Spectrogramme - Transformation de Fourier à Court Terme (TFCT)

Transformation de Wigner-Ville (ondelettes)

Transformation de Hilbert

ANALYSE CEPSTRALE

Cepstre de la réponse d'un filtre linéaire

Application : Traitement de la parole

Coefficients cepstraux

ANALYSE LPC (Codage Prédiction Linéaire)

Méthode d'autocorrélation

Méthode de covariance

Estimation

ANALYSE SPECTRALE

ELEMENTS DE LA THEORIE CLASSIQUE DE L'ESTIMATION

Biais et variance d'un estimateur

Cas monovariante

Consistance d'un estimateur

THEORIE CLASSIQUE DE L'ESTIMATION SPECTRALE

Estimateurs de corrélation

Estimation de l'autocorrélation

Estimation de l'intercorrélation

Estimateur de DSP : méthode du périodogramme

Périodogramme

Périodogramme moyenné

Estimateur de DSP : méthode du corrélogramme

Estimation de la DSP à partir du modèle AR du signal (Théorie moderne de l'estimation spectrale basée sur les processus générateurs de signal)

Equations de Yule-Walker

4. Conditionnement - Filtrage. Détection

CONDITIONNEMENT - FILTRAGE

Prétraitements (Mise en forme, amplification, fenêtrage, filtrage ...)

Préaccentuation et désaccentuation (*Traitement de la parole*)

DETECTION

Filtrage adapté

Position du problème. Remarques. Normalisation.

Solution. Cas particulier important. Remarque

5. Transmission (Codage. Egalisation. Extraction de signal immergé dans du bruit. Filtrage adaptatif)

INTRODUCTION

Objectif. Moyens. Modélisation mathématique. Données M – aires. Constellation. Diagramme en œil ...

CODAGE - DECODAGE (MODULATION - DEMODULATION)

Modulations analogiques

Modulation par multiplication. Modulation d'Amplitude (AM). Modulation Bande Latérale Unique (BLU). Modulation de Fréquence (FM). Modulation de Phase (PM)

Modulations numériques

Les modulations d'impulsions (amplitude, durée, position, densité). La modulation d'impulsions codées (MIC). La Modulation Δ . Transmission du signal numérique : Bande de base, modulation (ASK, FSK, PSK, DPSK) Codes détecteurs et correcteurs d'erreurs (bit de parité, codes à parités entrelacées, codes de Hamming, codes cycliques, codes continus)

Filtrage optimal au sens des moindres carrés

EGALISATION

Egalisation

Equations de Wiener-Hopf

EXTRACTION DE SIGNAL IMMERGE DANS DU BRUIT

Principe d'orthogonalité

Equations de Wiener-Hopf

FILTRAGE ADAPTATIF

Filtre adaptatif de Widrow

6. Prédiction

Prédiction

Lissage - Filtrage - Prédiction

PREDICTION LINEAIRE

Variance minimale

Equations de Yule-Walker

Remarque

7. Filtrage optimal

MOINDRES CARRES

Moindres carrés direct

Exemple

Moindres carrés récursif

Problème : Identification d'un canal de communication

Moindres carrés moyens

Algorithme LS

(Least Squares)

Algorithme RLS

(Recursive Least Squares)

Algorithme LMS

(Least Mean Squares)

7. ANNEXE

FILTRAGE DE WIENER

Algorithme du gradient

Moindres carrés moyens

Algorithme LMS

(Least Mean Squares)

FILTRAGE DE KALMAN

Exemple

Filtre de Kalman récursif

ANNEXE

8 Annexe. Décision (Classification, Reconnaissance des Formes)

CAS BI-CLASSE

Cas où les densités de probabilité du vecteur d'observation conditionnellement à chaque classe sont connues

Fonction de coût R

Test du rapport de vraisemblance

Cas où les densités de probabilité du vecteur d'observation conditionnellement à chaque classe ne sont pas connues

Règle du minimax

Courbe COR

CAS BI-CLASSE AVEC NON-DECISION

Cas où les densités de probabilité du vecteur d'observation conditionnellement à chaque classe sont connues

Règle de décision

Test du rapport de vraisemblance

Cas où les densités de probabilité du vecteur d'observation conditionnellement à chaque classe ne sont pas connues

Bibliographie

- | | | | |
|------|-----------------------------------|---|--------------------|
| [1] | F. Auger | « Introduction à la théorie du signal et de l'information » | <i>Technip</i> |
| [2] | G. Blanchet / M. Charbit | « Traitement numérique du signal : simulation sous Matlab » | <i>Hermès</i> |
| [3] | M. Bouvet | « Traitement des signaux pour les systèmes sonar » | <i>Masson</i> |
| [4] | J.M. Brossier | « Signal et communication numérique » | <i>Hermès</i> |
| [5] | M. Charbit | « Eléments de théorie du signal : aspects aléatoires » | <i>Ellipses</i> |
| [6] | P. Duvaut | « Traitement numérique du signal » | <i>Hermès</i> |
| [7] | M. Gevers / L. Vandendorpe | « Processus stochastiques : estimation et prédiction » | <i>UCL</i> |
| [8] | M. Kunt | « Traitement Numérique des Signaux / Atelier de TNS » | <i>Dunod / PPR</i> |
| [9] | Y. Thomas | « Signaux & systèmes linéaires : cours / exercices » | <i>Masson</i> |
| [10] | F. Truchetet | « Traitement linéaire du signal numérique » | <i>Hermès</i> |

*UCL : Université Catholique de Louvain
PPR : Presses Polytechniques Romandes*

1. Signaux déterministes. Signaux aléatoires. Systèmes stochastiques

I. SIGNAUX DETERMINISTES

CONVOLUTION - REPONSE IMPULSIONNELLE (RI)

Un système linéaire stationnaire (SLS) est caractérisé par sa RI: TC: $h(t)$ TD: $h(n)$



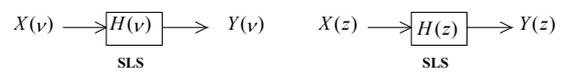
La réponse y d'un SLS d'entrée x s'écrit :
avec la relation de convolution * :

TC: $y(t) = h(t) * x(t)$ TD: $y(n) = h(n) * x(n)$

$$\text{TC: } y(t) = \int_{-\infty}^{\infty} h(\tau)x(t - \tau)d\tau \quad \text{TD: } y(n) = \sum_{k=-\infty}^{\infty} h(k)x(n - k)$$

FONCTION DE TRANSFERT (FT) (\equiv Gain complexe)

La FT d'un système caractérise, de façon duale à la RI, un SLS: TC: $H(\nu)$ TD: $H(z)$



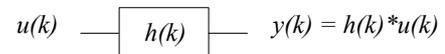
La réponse Y d'un SLS d'entrée X s'écrit :

TC: $Y(\nu) = H(\nu)X(\nu)$ TD: $Y(z) = H(z)X(z)$

avec la relation où u désigne indifféremment un signal x , y ou h :

$$\text{TC: } U(\nu) = TF[u(t)] = \int_{-\infty}^{\infty} u(t)e^{-i2\pi\nu t} dt \quad \text{TD: } U(z) = TZ[u(n)] = \sum_{n=-\infty}^{\infty} u(n)z^{-n}$$

DECONVOLUTION : Détermination de la Réponse Impulsionnelle



$$h(k) = y(k) \otimes u(k)$$

\otimes : déconvolution :

$$h(k) = \frac{1}{u(0)} \left[y(k) - \sum_{i=1}^k h(k-i).u(i) \right]$$

avec : $u(k)$ et $h(k)$ causales et $u(0) \neq 0$.

CORRELATION

Fonction d'intercorrélation $\varphi_{xy}(t)$ (TC) ou $\varphi_{xy}(n)$ (TD) de 2 signaux $x(t)$, $y(t)$ (TC) ou $x(n)$, $y(n)$ (TD) :

$$\text{(TC) } \varphi_{xy}(t) = \int_{-\infty}^{\infty} x(\tau)y(t + \tau)d\tau$$

$$\text{(TD) } \varphi_{xy}(n) = \sum_{k=-\infty}^{\infty} x(k)y(n + k)$$

$$\text{(TC) } \boxed{\varphi_{xy}(t) = \varphi_{yx}(-t)} \quad \text{anticommutativité}$$

$$\text{(TD) } \boxed{\varphi_{xy}(n) = \varphi_{yx}(-n)}$$

$$\text{(TC) } \boxed{\varphi_{xy}(t) = x(-t) * y(t)}$$

$$\text{(TD) } \boxed{\varphi_{xy}(n) = x(-n) * y(n)}$$

Fonction d'autocorrélation $\varphi_{xx}(t)$ (TC) ou $\varphi_{xx}(n)$ (TD) d'un signal $x(t)$ (TC) ou $x(n)$ (TD) :

$$\text{(TC) } \varphi_{xx}(t) = \int_{-\infty}^{\infty} x(\tau)x(t + \tau)d\tau$$

$$\text{(TD) } \varphi_{xx}(n) = \sum_{k=-\infty}^{\infty} x(k)x(n + k)$$

(TC) $\varphi_{xx}(t)$ maximale pour $t = 0$

(TD) $\varphi_{xx}(n)$ maximale pour $n = 0$

(TC) $\varphi_{xx}(-t) = \varphi_{xx}(t)$ ($\varphi_{xx}(t)$ paire)

(TD) $\varphi_{xx}(-n) = \varphi_{xx}(n)$ ($\varphi_{xx}(n)$ paire)

(TC) $\varphi_{xx}(t) = x(-t) * x(t)$

(TD) $\varphi_{xx}(n) = x(-n) * x(n)$

ENERGIE - PUISSANCE - RAPPORT SIGNAL SUR BRUIT

- *Energie d'un signal (cas complexe)*: (TC) $E = \int_{-\infty}^{\infty} |x(t)|^2 dt$ (TD) $E = \sum_{n=-\infty}^{\infty} |x(n)|^2$ ($E \geq 0$)

[unité: le Joule] (cas scalaire): (TC) $E = \int_{-\infty}^{\infty} x^2(t) dt$ (TD) $E = \sum_{n=-\infty}^{\infty} x^2(n)$

Pour un signal à énergie infinie (signal périodique par exemple) on préférera utiliser la *puissance* (moyenne) pour caractériser le signal :

- *Puissance (moyenne) d'un signal* : (TC) $P = \lim_{T \rightarrow \infty} \frac{1}{2T} \int_{-T}^{+T} |x(t)|^2 dt$ (TD) $P = \lim_{k \rightarrow \infty} \frac{1}{2k+1} \sum_{n=-k}^k |x(n)|^2$

[unité: le Watt] (cas scalaire): (TC) $P = \lim_{T \rightarrow \infty} \frac{1}{2T} \int_{-T}^{+T} x^2(t) dt$ (TD) $P = \lim_{k \rightarrow \infty} \frac{1}{2k+1} \sum_{n=-k}^k x^2(n)$

[unité: le déciBel (dB)] $P_{dB} = 10 \log(|P|) = 10 \log(P)$ ($P \geq 0$)

Ex.: $x(n) = \Gamma(n) \rightarrow$ l'énergie diverge ($E = \infty$) mais la puissance moyenne du signal reste finie :

$$P = \lim_{k \rightarrow \infty} \frac{1}{2k+1} \sum_{n=-k}^k |x(n)|^2 = \lim_{k \rightarrow \infty} \frac{1}{2k+1} \sum_{n=0}^k 1 = \lim_{k \rightarrow \infty} \frac{1}{2k+1} (k+1) = \frac{1}{2}$$

- *Rapport Signal/Bruit (SNR)* : $s(t)$ signal utile et $b(t)$ bruit, sont 2 signaux (TC) causaux, scalaires, de durée T
 s_n signal utile et b_n bruit, sont 2 signaux (TD) causaux, scalaires, de N échantillons

[unité: le déciBel (dB)] (TC) $SNR_{dB} = 10 \log \left(\frac{\int_0^T x^2(t) dt}{\int_0^T b^2(t) dt} \right) = 10 \log \left(\frac{P_x}{P_b} \right) = P_{x,dB} - P_{b,dB}$ (TD) $SNR_{dB} = 10 \log \left(\frac{\sum_{n=0}^{N-1} x_n^2}{\sum_{n=0}^{N-1} b_n^2} \right) = 10 \log \left(\frac{P_x}{P_b} \right) = P_{x,dB} - P_{b,dB}$

Dans le cas de bruit aléatoire non ergodique, il faut prendre sa moyenne statistique (espérance) plutôt que sa moyenne temporelle (1 seule réalisation du bruit ne suffit pas) :

Rapport des puissances instantanées à l'instant $t = T$ (TC) ou $n = N$ (TD) :

Cas aléatoire (TC) $SNR_{dB} = 10 \log \left(\frac{x^2(T)}{E[b^2(T)]} \right) = 10 \log \left(\frac{P_{x_i}}{P_{b_i}} \right) = P_{x_i,dB} - P_{b_i,dB}$ (TD) $SNR_{dB} = 10 \log \left(\frac{x_N^2}{E[b_N^2]} \right) = 10 \log \left(\frac{P_{x_i}}{P_{b_i}} \right) = P_{x_i,dB} - P_{b_i,dB}$

Remarque :
 . un SNR de 100 dB traduit un excellent Rapport Signal/Bruit (le bruit est rejeté)
 . un SNR de 10 dB traduit un Rapport Signal/Bruit minimal admissible (le bruit est couvert, il est « négligeable »)
 . un SNR de 0 dB traduit une même puissance entre Signal et Bruit.

II. SIGNAUX ALEATOIRES

Dans de nombreux systèmes réels, les signaux à traiter ne peuvent être modélisés de façon déterministe.

C'est le cas par exemple des signaux de **télécommunications** qui, par nature, **ne sont pas connus de façon certaine, sinon ils ne véhiculeraient aucune information** (quasiment).

C'est aussi le cas de nombreux signaux qui proviennent de mesures ou d'observations bruitées, qui font que le signal réellement observé est une forme corrompue du signal voulu. De ce fait, de nombreuses opérations de traitement du signal pensées pour des signaux déterministes doivent être repensées et optimisées en fonction de l'information dont on dispose à propos du signal aléatoire à traiter.

1. L'aléatoire en théorie du signal

Dans beaucoup d'expériences pratiques, les phénomènes observés dans des conditions apparemment identiques semblent présenter des variations imprévisibles. On dit que ces phénomènes sont aléatoires. Par exemple, le bruit de fond d'un récepteur radiophonique, ou encore le signal sonore produit par un compteur de particules sont des exemples de signaux aléatoires.

Dans le cas particulier d'un système de communication, non seulement le bruit de fond, mais aussi le message émis par la source, présentent un caractère aléatoire pour le destinataire. Il en est de même pour un signal de parole qui ne peut être décrit de façon exacte par une expression analytique, même en l'absence de bruit.

Pour étudier l'évolution des phénomènes aléatoires, on a recours à des modèles probabilistes. Ceux-ci tentent de représenter le signal observé par une famille de Variables Aléatoires (VA) indexées par le temps. Chacune de ces VA décrit l'aspect incertain du phénomène à un instant donné. Une telle famille porte le nom de processus aléatoire.

Un processus aléatoire peut être à TD (Temps Discret) ou à TC (Temps Continu).

2. Variables Aléatoires

Une VA X est une application qui associe à chaque épreuve, réalisation x possible d'une expérience, une valeur particulière (x): $X = X(\omega)$ (ω : épreuve)

Une image familière nous est donnée par l'apparition des valeurs 1 à 6 dans une expérience de lancé d'un dé à 6 faces.

VA discrètes et continues

Une VA est *discrète* ou *continue* suivant que l'ensemble des valeurs possibles de l'expérience aléatoire est respectivement *discret* ou *continu*.

VA discrète : un exemple du cas discret est donné par l'observation du nombre de personnes présentes dans une file d'attente : les seules valeurs possibles sont les entiers positifs ou nul.

VA continue : Par contre, le relevé de la vitesse des véhicules sur une autoroute fournit un exemple de VA continue (les valeurs possibles étant les nombres réels compris entre 0 et 130 km/h).

VA discrète

La loi de probabilité d'une VA X est complètement caractérisée par l'ensemble *dénombrable* $\{a_1, a_2, \dots, a_k, \dots\}$ des valeurs possibles et par la suite des valeurs : ($k \in \mathbf{Z}$)

$$p_X(k) = P(X = a_k)$$

qui représente la *probabilité* pour que X soit égal à l'élément a_k . Ces valeurs vérifient :

$$0 \leq p_X(k) \leq 1 \quad \text{et} \quad \sum_k p_X(k) = 1$$

Moment d'ordre n de la VA X : c'est le nombre (non aléatoire) défini par : ($n \in \mathbf{N}$)

$$E[X^n] = \sum_k a_k^n p_X(k)$$

cas : $n = 1$: on a l'*espérance mathématique* ou *moyenne statistique* de la VA X : $E[X] = \sum_k a_k p_X(k)$

cas : $n = 2$: on a le moment d'ordre 2 de la VA X : $E[X^2] = \sum_k a_k^2 p_X(k)$

Les moments d'ordre 1 et 2 suffisent généralement à caractériser une VA. Cependant, on peut parfois avoir recours aux *moments d'ordre supérieur* : ex. $n = 4$: moment d'ordre 4 : *kurtosis*.

Loi de probabilité conjointe de 2 VA X et Y :

Elle est caractérisée par les ensembles *dénombrables* $\{a_1, a_2, \dots, a_k, \dots\}$ et $\{b_1, b_2, \dots, b_m, \dots\}$ des valeurs possibles de X et Y et par la suite des probabilités : $p_{XY}(k, m) = P(X = a_k, Y = b_m)$ qui vérifient :

$$0 \leq p_{XY}(k, m) \leq 1 \quad \text{et} \quad \sum_k \sum_m p_{XY}(k, m) = 1$$

Indépendance de 2 VA X et Y :

2 VA X et Y sont indépendantes si et seulement si : $P(X = a_k, Y = b_m) = P(X = a_k)P(Y = b_m)$

VA continue

Densité de probabilité

La VA X prend ses valeurs dans le corps de réels \mathbf{R} , et sa loi est caractérisée par une fonction appelée *densité de probabilité* que l'on note $p_X(x)$ avec x réel, et qui permet le calcul de la probabilité pour X d'appartenir à un intervalle (a, b) par la relation :

$$P(X \in (a, b)) = \int_a^b p_X(x) dx$$

La fonction $p_X(x)$ est ≥ 0 (pas nécessairement ≤ 1) et vérifie la relation de normalisation : $\int_{-\infty}^{\infty} p_X(x) dx = 1$

Fonction de répartition

La fonction de répartition $F_X(x)$ d'une VA X est définie par : $F_X(x) = P(X \leq x)$

On a la relation entre densité de probabilité $p_X(x)$ et fonction de répartition $F_X(x)$: $F_X(x) = \int_{-\infty}^x p_X(u) du$

Moment d'ordre n de la VA X : c'est le nombre (non aléatoire) défini par : $(n \in \mathbf{N})$

$$E[X^n] = \int_{-\infty}^{\infty} x^n p_X(x) dx$$

cas : $n = 1$: on a l'*espérance mathématique* ou *moyenne statistique* de la VA X : $E[X] = \int_{-\infty}^{\infty} x p_X(x) dx$

cas : $n = 2$: on a le moment d'ordre 2 de la VA X : $E[X^2] = \int_{-\infty}^{\infty} x^2 p_X(x) dx$

Les moments d'ordre 1 et 2 suffisent généralement à caractériser une VA. Cependant, on peut parfois avoir recours aux *moments d'ordre supérieur* : ex. : $n = 3$: moment d'ordre 3 : *skewness*; $n = 4$: moment d'ordre 4 : *kurtosis*.

Loi de probabilité conjointe de 2 VA X et Y

De façon analogue, elle est caractérisée par sa densité de probabilité $p_{XY}(x, y)$, qui, par la formule suivante, autorise le calcul de la probabilité pour que le couple de VA (X, Y) appartienne à un domaine Δ du plan :

$$P((X, Y) \in \Delta) = \iint_{\Delta} p_{XY}(x, y) dx dy$$

La fonction $p_{XY}(x, y)$ est ≥ 0 (pas nécessairement ≤ 1) et vérifie la normalisation : $\iint_{\mathbf{R}^2} p_{XY}(x, y) dx dy = 1$

Indépendance de 2 VA X et Y

2 VA X et Y sont indépendantes si et seulement si : $p_{XY}(x, y) = p_X(x) p_Y(y)$

VA discrètes et continues

. Moyenne m d'une VA X

Cette grandeur est déterministe

$$m_X = E[X] \quad \text{moyenne statistique, espérance mathématique ou encore moment d'ordre 1 de } X$$

La moyenne d'une VA traduit le centre (le centre de gravité) des épreuves.

. **Variance σ^2 d'une VA X** Cette grandeur est déterministe

$$\sigma_x^2 = \text{var}[X] = E[(X - E[X])^2] = E[X^2] - E^2[X]$$

En effet : $E[(X - E[X])^2] = E[X^2 - 2XE[X] + E^2[X]] = E[X^2] - 2E^2[X] + E^2[X] = E[X^2] - E^2[X]$

La variance est une quantité toujours ≥ 0 . Elle peut s'interpréter comme la mesure des fluctuations de la VA autour de sa moyenne : plus elle est grande, plus les valeurs de X sont dispersées autour de $E[X]$.

En théorie du signal, la variance a une signification supplémentaire : elle s'interprète comme la *puissance* des fluctuations (quadratiques) de X autour de sa moyenne.

La variance d'une VA traduit sa puissance, sa dispersion. Elle est liée à l'amplitude des variations de la VA.

Moyenne et variance suffisent à décrire de nombreux phénomènes aléatoires de façon suffisante.

On a les propriétés :
$$\begin{cases} E[aX + b] = aE[X] + b \\ \text{var}[aX + b] = a^2 \text{var}[X] \end{cases} \quad \text{où : } a \text{ et } b \text{ sont des complexes non aléatoires.}$$

. **Ecart-type σ d'une VA X** Cette grandeur est déterministe

$$\sigma_x = \sqrt{\text{var}[X]} \quad \text{écart moyen}$$

. **Covariance φ_{xy} de 2 VA X et Y** Cette grandeur est déterministe Soit $X^0 \equiv X - E[X]$ et $Y^0 \equiv Y - E[Y]$:

$$\varphi_{xy} = \text{cov}[X, Y] = E[(X - E[X])(Y - E[Y])] = E[X^0 Y^0] = E[XY] - E[X]E[Y]$$

. **Autocovariance φ_{xx} d'une VA X (\equiv variance)** Cette grandeur est déterministe

$$\varphi_{xx} = \text{cov}[X, X] = E[(X - E[X])^2] = E[X^2] - E^2[X] = \sigma_x^2$$

. **Corrélation ρ_{xy} de 2 VA X et Y** Cette grandeur est déterministe

$$\rho_{xy} = \frac{\text{cov}[X, Y]}{\sqrt{\text{var}[X]} \sqrt{\text{var}[Y]}} = \frac{\varphi_{xy}}{\sigma_x \sigma_y}$$

On a la propriété, issue de l'inégalité de Schwarz : $-1 \leq \rho_{xy} \leq 1$

Décorrélations de 2 VA X et Y : si $\varphi_{xy} = \text{cov}[X, Y] = 0$ on dit alors que les VA X et Y sont décorréliées ($\rho_{xy} = 0$)

On montre que si 2 VA X et Y sont indépendantes alors elles sont décorréliées. Mais en général, la réciproque est fautive.
 X, Y indépendantes $\rightarrow X, Y$ décorréliées mais X, Y décorréliées $\nrightarrow X, Y$ indépendantes

. **Autocorrélation ρ_{xx} d'une VA X** Cette grandeur est déterministe

$$\rho_{xx} = \frac{\text{cov}[X, X]}{\text{var}[X]} = \frac{\varphi_{xx}}{\sigma_x^2} = 1$$

. **Estimation de la densité de probabilité**

Loi des grands nombres

$$P(X \in \Delta) \text{ peut être approché de façon efficace par le rapport } \frac{n_\Delta}{N} : \quad P(X \in \Delta) = \frac{n_\Delta}{N}$$

- avec :
- . $P(X \in \Delta)$: probabilité d'appartenance de la VA X à un intervalle Δ
 - . n_Δ : nombre de points, de résultats d'expérience observés dans l'intervalle Δ
 - . N : nombre total de points, de réalisations possibles.
 - . $\frac{n_\Delta}{N}$: fréquence relative

On peut alors estimer la densité de probabilité $p_X(x)$ de X au point x_1 , soit $p_X(x_1)$ par :
$$p_X(x_1) = \frac{n_\Delta}{N \cdot l_\Delta}$$

- avec :
- . $p_X(x_1)$: densité de probabilité de X au point x_1
 - . n_Δ : nombre de points, de résultats d'expérience observés dans l'intervalle Δ
 - . N : nombre total de points, de réalisations possibles.
 - . l_Δ : longueur de l'intervalle Δ

En effet, on a :
$$P(X \in \Delta) = \int_{\Delta} p_X(x) dx$$
 et si on suppose que Δ est un intervalle très petit autour du point x_1 , le terme :

$$\int_{\Delta} p_X(x) dx \text{ est approximativement égal à : } p_X(x_1)l_\Delta \text{ d'où : } p_X(x_1)l_\Delta = \frac{n_\Delta}{N}$$

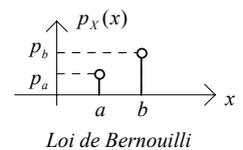
La procédure pour estimer la densité de probabilité au point x_0 peut alors suivre l'algorithme :

1. On effectue une expérience de longueur N
2. On partitionne l'intervalle des valeurs obtenues en P sous-intervalles I_1, I_2, \dots, I_P de longueur respective l_1, l_2, \dots, l_P autour des points x_1, x_2, \dots, x_P . En général, on choisit des sous-intervalles de même longueur et x_p est pris au milieu du sous-intervalle I_p . En pratique P ne dépasse pas $N/10$.
3. Enfin, on estime la densité de probabilité au point x_p par
$$p_X(x_p) = \frac{n_{I_p}}{N \cdot l_{I_p}}$$

3. Lois de probabilité de VA usuelles

Loi de Bernoulli (≡ loi à 2 valeurs)

La VA X peut prendre 2 valeurs a et b avec les probabilités respectives p_a et p_b avec $p_a + p_b = 1$.



La densité de probabilité est donnée par $p_X(x) = p_a \delta(x-a) + p_b \delta(x-b)$:

Moyenne m : $m = E[X] = ap_a + bp_b$ Moment d'ordre n : $E[X^n] = a^n p_a + b^n p_b$

Variance σ^2 : $\sigma^2 = E[X^2] - E^2[X] = a^2 p_a(1-p_a) + b^2 p_b(1-p_b) - 2abp_a p_b$

Exemple de VA de loi de Bernoulli : Résultat pile ou face du lancer d'une pièce.

Loi binomiale

Soit un évènement e associé à une expérience. Cet évènement a une probabilité p de survenir. On reproduit n fois l'expérience et on s'intéresse à la VA X qui donne le nombre de succès de l'évènement e au cours des n essais.

La densité de probabilité est donnée par $p_X(x) = \sum_{k=0}^n p_k \delta(x-k)$ avec $p_k = C_n^k p^k (1-p)^{n-k}$ et $C_n^k = \frac{n!}{k!(n-k)!}$

Moyenne m : $m = E[X] = np$

Variance σ^2 : $\sigma^2 = E[X^2] - E^2[X] = np(1-p)$

Exemple de VA de loi binomiale : On effectue n tirages avec remise d'une boule dans une urne contenant n_B boules blanches et n_N boules noires. L'évènement est le tirage d'une boule blanche.

Loi de Poisson

La densité de probabilité est du type : $p_X(x) = \sum_{k=0}^{\infty} p_k \delta(x - k)$ avec $p_k = \frac{\lambda^k e^{-\lambda}}{k!}$

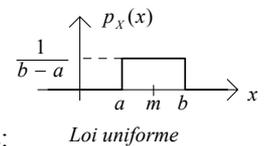
Moyenne m : $m = E[X] = \lambda$

Variance σ^2 : $\sigma^2 = E[X^2] - E^2[X] = \lambda$

Exemple de VA de loi binomiale : On effectue n tirages avec remise d'une boule dans une urne contenant n_B boules blanches et n_N boules noires. L'évènement est le tirage d'une boule blanche.

Loi uniforme sur $[a, b]$

L'ensemble des valeurs possibles de la VA X se réduit à l'intervalle $[a, b]$ et la probabilité de X d'appartenir à un intervalle $[c, d] \subset [a, b]$ est égale à $\frac{d - c}{b - a}$.



On en déduit que sa densité de probabilité est constante et vaut $\frac{1}{b - a}$ sur $[a, b]$ et 0 ailleurs: *Loi uniforme*

Moyenne m : $m = E[X] = \int_a^b \frac{x}{b - a} dx = \frac{a + b}{2}$

Moment d'ordre 2 : $E[X^2] = \int_a^b \frac{x^2}{b - a} dx = \frac{a^2 + ab + b^2}{3}$

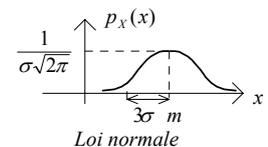
Variance σ^2 : $\sigma^2 = E[X^2] - E^2[X] = \frac{(b - a)^2}{12}$

Exemple de VA de loi uniforme : Erreur d'arrondi dans les calculs (bruit de quantification uniforme).

Génération d'une VA Y uniforme sur $[a, b]$ à partir d'une VA X uniforme sur $[0, 1]$: $Y = (b - a)X + a$

Loi normale (\equiv gaussienne)

Numériquement, plus de 99% des valeurs de la densité de probabilité $p_X(x)$ de la VA X tombent dans l'intervalle $[m - 3\sigma, m + 3\sigma]$, dit intervalle de confiance ($[m - 2\sigma, m + 2\sigma]$ donne 95% des valeurs de $p_X(x)$, $[m - 4\sigma, m + 4\sigma]$ pour 99.9%).



La densité de probabilité est définie par $p_X(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-m)^2}{2\sigma^2}}$:

Moyenne m : $E[X] = m$

Variance σ^2 : $E[X^2] - E^2[X] = \sigma^2$

Exemple de VA de loi normale : Bruit de fond dans les récepteurs.

Génération d'une VA Y normale (m, σ^2) à partir d'une VA X normale $(m_0 = 0, \sigma_0^2 = 1)$: $Y = \sigma X + m$

La distribution normale est linéaire, et elle est la seule loi pour laquelle il y a équivalence entre indépendance et non corrélation.

Lois de Rayleigh, de Laplace, de Cauchy

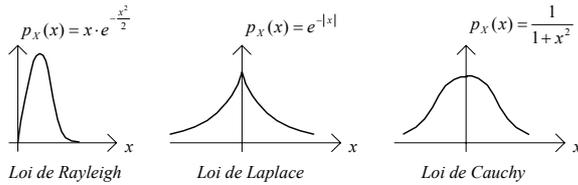


Tableau récapitulatif

VA discrètes					
Loi de distribution de la VA X	Probabilité $p_k = p(X = k)$	Densité de probabilité $p_X(x) = \sum_k p_k \delta(x - k)$	Fonction de répartition $F_X(x) = p(X \leq x) = \sum_k p_k \Gamma(x - k)$	Moyenne $m = \sum_k x_k p_k$	Variance $\sigma^2 = \sum_k (x_k - m)^2 p_k$
Bernouilli (a, b)	(p_a, p_b)	$p_a \delta(x - a) + p_b \delta(x - b)$	$p_a \Gamma(x - a) + p_b \Gamma(x - b)$	$ap_a + bp_b$	$a^2 p_a(1 - p_a) + b^2 p_b(1 - p_b) - 2abp_a p_b$
binomiale (n, p)	$C_n^k p^k (1 - p)^{n-k}$	$\sum_{k=0}^n C_n^k p^k (1 - p)^{n-k} \delta(x - k)$	$\sum_{k=0}^n C_n^k p^k (1 - p)^{n-k} \Gamma(x - k)$	np	$np(1 - p)$
Poisson (λ)	$\frac{\lambda^k e^{-\lambda}}{k!}$	$\sum_{k=0}^{\infty} \frac{\lambda^k e^{-\lambda}}{k!} \delta(x - k)$	$\sum_{k=0}^{\infty} \frac{\lambda^k e^{-\lambda}}{k!} \Gamma(x - k)$	λ	λ

VA continues					
Loi de distribution de la VA X	Probabilité $p(x) = p(X = x)$	Densité de probabilité $p_X(x) = \frac{dF_X(x)}{dx}$	Fonction de répartition $F_X(x) = p(X \leq x) = \int_{-\infty}^x p_X(u) du$	Moyenne $m = \int_{-\infty}^{\infty} x p_X(x) dx$	Variance $\sigma^2 = \int_{-\infty}^{\infty} (x - m)^2 p_X(x) dx$
uniforme sur $[a, b]$		$\frac{1}{b-a} \Pi_{b-a}(x - m) = \begin{cases} \frac{1}{b-a} & \text{si } a \leq X \leq b \\ 0 & \text{ailleurs} \end{cases}$	$\begin{cases} 0 & \text{si } X < a \\ \frac{x-a}{b-a} & \text{si } a \leq X \leq b \\ 1 & \text{si } X > b \end{cases}$	$\frac{a+b}{2}$	$\frac{(b-a)^2}{12}$
normale (m, σ^2)		$\frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-m)^2}{2\sigma^2}}$	$1 - \frac{1}{2} \operatorname{erfc}\left(\frac{x-m}{\sigma\sqrt{2}}\right)$ avec $\operatorname{erfc}(x) = \frac{2}{\sqrt{\pi}} \int_x^{\infty} e^{-t^2} dt$	m	σ^2

4. Théorème central limite

Chaque fois qu'un phénomène peut être considéré comme la résultante d'un grand nombre de causes aléatoires indépendantes, on peut présumer que ce phénomène suit une loi de distribution *normale*: c'est le *théorème central limite*.

5. Processus aléatoires

En théorie du signal, on étudie le plus souvent des grandeurs dépendant du temps et dont l'évolution semble imprévisible. L'écoute du son émis par des véhicules qui passent sur une route nous donne une image concrète et familière d'une telle grandeur (alors que le lancer d'un dé est une VA statique, indépendante du temps).

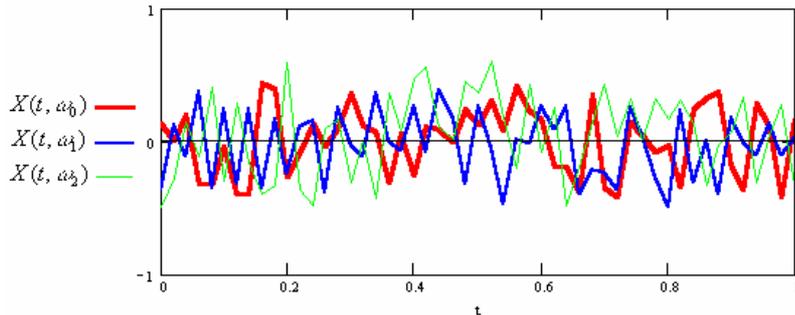
Pour les modéliser, on utilise la notion de processus aléatoire qui associe à chaque épreuve une **réalisation** qui n'est plus une *valeur* comme dans le cas des *Variables Aléatoires*, mais **une fonction du temps**.

On définit un Processus Aléatoire (PA) comme une application qui, à chaque épreuve ω , fait correspondre une fonction du temps t . On utilise la notation $X(t, \omega)$, ou plus usuellement, $X(t)$, dans laquelle on omet la dépendance statistique (\equiv vis-à-vis de l'épreuve ω), à l'exemple de ce que l'on fait pour les Variables Aléatoires que l'on note simplement X .

Trajectoire d'un Processus Aléatoire

Un processus aléatoire peut alors être vu :

- . soit, pour une épreuve fixée ω_0 , comme une fonction du temps que l'on appelle *trajectoire*.
- . soit, pour un instant fixé t_0 , comme une *Variable Aléatoire*.



Réalisations d'un processus aléatoire (exemple : réalisations d'un signal parole)

La démarche qui consiste à modéliser une expérience par plusieurs fonctions, ce qui suppose que le temps puisse se dérouler une infinité de fois à l'identique, peut choquer, puisqu'en pratique, on n'observe souvent qu'une réalisation, et encore uniquement sur un intervalle de temps fini !

Cette façon de faire se trouve justifiée par la pertinence des résultats qu'elle permet d'obtenir.

Processus aléatoires à Temps Continu et à Temps Discret

Certains processus aléatoires ont des trajectoires qui, tout en étant imprévisibles, semblent posséder des caractéristiques communes.

Processus Aléatoire $X(t)$ à TD et à TC

- Un Processus Aléatoire $X(t)$ est à TC si le temps $t \in \mathbf{R}$ (t réel).
- Un Processus Aléatoire $X(t)$ est à TD si le temps $t \in \mathbf{Z}$ (t entier).

Moyenne $m_X(t)$ d'un Processus Aléatoire $X(t)$

Cette grandeur est déterministe et dépend de t

$$m_X(t) = E[X(t)] \quad \text{moyenne statistique, espérance mathématique ou encore moment d'ordre 1 de } X(t)$$

Fonction d'Autocovariance $R_{XX}(t_1, t_2)$ d'un Processus Aléatoire $X(t)$

Cette grandeur est déterministe

(appelée aussi **Fonction d'Autocorrélation** lorsque le processus est stationnaire)

$$R_{XX}(t_1, t_2) = E[X^0(t_1)X^0(t_2)] = E[X(t_1)X(t_2)] - m_X(t_1)m_X(t_2) \quad (= E[X(t_1)X(t_2)]) \text{ si } X(t) \text{ centré}$$

$$R_{XX}(t_1, t_2) = E[X^0(t_1)X^{0*}(t_2)] = E[X(t_1)X^*(t_2)] - m_X(t_1)m_X^*(t_2) \quad \text{dans le cas de PA complexes} \quad (X^*(t) : \text{conjugué de } X(t))$$

où : $X^0(t) = X(t) - E[X(t)]$ désigne le processus centré

Fonction de Covariance $R_{XY}(t_1, t_2)$ de 2 Processus Aléatoires $X(t)$ et $Y(t)$

Cette grandeur est déterministe

(appelée aussi **Fonction de Corrélation** lorsque le(s) processus est (sont) stationnaire(s))

$$R_{XY}(t_1, t_2) = E[X^0(t_1)Y^0(t_2)] = E[X(t_1)Y(t_2)] - m_X(t_1)m_Y(t_2) \quad (= E[X(t_1)Y(t_2)]) \text{ si } X(t) \text{ ou } Y(t) \text{ centré}$$

$$R_{XY}(t_1, t_2) = E[X^0(t_1)Y^{0*}(t_2)] = E[X(t_1)Y^*(t_2)] - m_X(t_1)m_Y^*(t_2) \quad \text{dans le cas de PA complexes} \quad (X^*(t) : \text{conjugué de } X(t))$$

où : $X^0(t) = X(t) - E[X(t)]$ désigne le processus centré

On se limite aux processus stationnaires au 2nd ordre au sens large (SSL) (cas le plus courant)

Stationnarité du 2nd ordre au sens large des processus aléatoires à TC et à TD

Notations : si Processus à TC : $t, \tau, t_1, t_2 \in \mathbf{R}$

si Processus à TD : $t, \tau, t_1, t_2 \in \mathbf{Z}$

Un Processus Aléatoire stationnaire au 2nd ordre au sens large (SSL) est tel que sa moyenne et sa variance sont indépendantes de l'origine des temps

Un Processus Aléatoire $X(t)$ est SSL s'il vérifie les 2 propriétés 1 et 2 :

1. la moyenne du Processus Aléatoire SSL est indépendante de t :

$$m_X(t) = m_X$$

$$m_X = E[X(t)]$$

Fonction d'Autocovariance (=Autocorrélation dans le cas SSL) $R_{XX}(\tau)$ d'un Processus Aléatoire SSL $X(t)$

2. la fonction d'autocovariance d'un PA SSL ne dépend que de $\tau = t_2 - t_1$:

$$R_{XX}(t_1, t_2) = R_{XX}(\tau)$$

$$R_{XX}(\tau) = E[X^0(t-\tau)X^0(t)] = E[X(t-\tau)X(t)] - m_X^2$$

$$\text{si } X(t) \text{ centré: } R_{XX}(\tau) = E[X(t-\tau)X(t)]$$

$$R_{XX}(\tau) = E[X^0(t-\tau)X^{0*}(t)] = E[X(t-\tau)X^*(t)] - |m_X|^2 \quad \text{dans le cas de PA complexes } (X^*(t) : \text{conjugué de } X(t))$$

on a aussi :

$$R_{XX}(\tau) = E[X^0(t)X^0(t+\tau)] = E[X(t)X(t+\tau)] - m_X^2$$

$$\text{si } X(t) \text{ centré: } R_{XX}(\tau) = E[X(t)X(t+\tau)]$$

$$R_{XX}(-\tau) = E[X^0(t+\tau)X^0(t)] = R_{XX}(\tau)$$

$R_{XX}(\tau)$ est paire

on a enfin :

$$R_{XX}(\tau) = X^0(\tau) * X^0(-\tau)$$

$$\text{si } X(t) \text{ centré: } R_{XX}(\tau) = X(\tau) * X(-\tau)$$

Symétrie hermitienne : la fonction d'autocovariance possède la propriété de *symétrie hermitique*

$$R_{XX}(\tau) = R_{XX}^*(-\tau) \quad \text{parité de l'autocovariance}$$

$$R_{XX}(\tau) = R_{XX}^*(-\tau) \quad \text{dans le cas de PA complexes } (R_{XX}^*(t) : \text{conjugué de } R_{XX}(t))$$

Positivité : la fonction d'autocovariance possède la propriété de *positivité*

$$\sum_{k=0}^{N-1} \sum_{m=0}^{N-1} a_k a_m R_{XX}(k-m) \geq 0 \quad \text{pour tout } N, \text{ pour toute suite de valeurs réelles } a_0, a_1, \dots, a_{N-1}$$

$$\sum_{k=0}^{N-1} \sum_{m=0}^{N-1} a_k a_m^* R_{XX}(k-m) \geq 0 \quad \text{pour tout } N, \text{ pour toute suite de valeurs complexes } a_0, a_1, \dots, a_{N-1}$$

que l'on peut écrire matriciellement : (les exposants T, H indiquent resp. la transposition et la transposition-conjuguaison)

$$\mathbf{A}^T \mathbf{R} \mathbf{A} \geq 0 \quad \mathbf{A}^H \mathbf{R} \mathbf{A} \geq 0 \quad \text{dans le cas complexe}$$

La matrice \mathbf{R} est appelée *matrice de covariance*.

On a la propriété $\mathbf{R} = \mathbf{R}^T$ ($\mathbf{R} = \mathbf{R}^H$ dans le cas complexe) du fait de la symétrie hermitienne.

$$\text{Exemple pour } N = 3 : \quad \mathbf{A}^T \mathbf{R} \mathbf{A} = \begin{bmatrix} a_0 & a_1 & a_2 \end{bmatrix} \begin{bmatrix} R_{XX}(0) & R_{XX}(1) & R_{XX}(2) \\ R_{XX}(1) & R_{XX}(0) & R_{XX}(1) \\ R_{XX}(2) & R_{XX}(1) & R_{XX}(0) \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} \geq 0$$

dans le cas de PA complexe :

$$\mathbf{A}^H \mathbf{R} \mathbf{A} = \begin{bmatrix} a_0^* & a_1^* & a_2^* \end{bmatrix} \begin{bmatrix} R_{XX}(0) & R_{XX}(1) & R_{XX}(2) \\ R_{XX}(-1) & R_{XX}(0) & R_{XX}(1) \\ R_{XX}(-2) & R_{XX}(-1) & R_{XX}(0) \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} \geq 0$$

En raison de la stationnarité du PA, la matrice \mathbf{R} est telle que les parallèles à la diagonale principale sont constituées de termes égaux. Cette forme de matrice porte le nom de *matrice de Toeplitz*.

Fonction de Covariance (= Corrélation dans le cas SSL) $R_{XY}(\tau)$ de 2 Processus Aléatoires SSL $X(t)$ et $Y(t)$

. la fonction de Covariance de 2 PA SSL ne dépend que de $\tau = t_2 - t_1$:

$$R_{XY}(\tau) = E[X^0(t-\tau)Y^0(t)] \quad \text{si } X(t) \text{ et } Y(t) \text{ centrés : } \boxed{R_{XY}(\tau) = E[X(t-\tau)Y(t)]}$$

$$R_{XX}(\tau) = E[X^0(t-\tau)Y^0(t)] \quad \text{dans le cas de PA complexes } (X^*(t) : \text{conjugué de } X(t))$$

on a aussi :

$$\boxed{R_{XY}(\tau) = E[X^0(t)Y^0(t+\tau)]} \quad \text{si } X(t) \text{ et } Y(t) \text{ centrés : } \boxed{R_{XY}(\tau) = E[X(t)Y(t+\tau)]}$$

$$\boxed{R_{YX}(\tau) = E[Y^0(t-\tau)X^0(t)] = R_{XY}(-\tau)} \quad \text{si } X(t) \text{ et } Y(t) \text{ centrés : } \boxed{R_{YX}(\tau) = E[Y(t-\tau)X(t)] = R_{XY}(-\tau)}$$

on a enfin :

$$R_{XY}(\tau) = X^0(\tau) * Y^0(-\tau) \quad \text{si } X(t) \text{ et } Y(t) \text{ centrés : } R_{XY}(\tau) = X(\tau) * Y(-\tau)$$

Densité Spectrale de Puissance (DSP) d'un Processus Aléatoire $X(t)$

Quand un Processus Aléatoire est *stationnaire au 2nd ordre*, on définit sa *puissance* et sa *DSP* de la façon suivante :

. **Puissance P**

$$P_X \stackrel{\Delta}{=} E[X^2(t)] = R_{XX}(0) + m_X^2 \quad \text{moyenne statistique de la puissance instantanée}$$

$$P_X \stackrel{\Delta}{=} E[|X(t)|^2] = R_{XX}(0) + |m_X|^2 \quad \text{dans le cas de PA complexes}$$

$$\boxed{P_X = R_{XX}(0)} \quad \text{dans le cas de PA centrés}$$

. **Valeur efficace X_{eff}**

$$X_{eff} \stackrel{\Delta}{=} \sqrt{P_X}$$

. **DSP S**

$$\boxed{S_X(f) \stackrel{\Delta}{=} TF[R_{XX}(\tau)]} \quad f \text{ est la fréquence}$$

Théorème de Wiener – Kintchine

La DSP de $X(t)$ est la TF de la fonction d'autocovariance de $X(t)$ (et non la TF de $X(t)$!)

Note : $S_X(f)$ est encore noté $S_{XX}(f)$. $R_{XX}(\tau)$ est encore noté $R_X(\tau)$

TC : $S_X(f) = \int_{-\infty}^{\infty} R_{XX}(\tau) e^{-i2\pi f\tau} d\tau \quad \tau \text{ est réel } (\tau \in \mathbf{R})$

TD : $S_X(f) = \sum_{\tau=-\infty}^{\infty} R_{XX}(\tau) e^{-i2\pi f\tau} \quad \tau \text{ est discret } (\tau \in \mathbf{Z})$

Comme dans le cas déterministe, le spectre représente la *répartition*, la *localisation* de la puissance le long de l'axe des fréquences. Par conséquent, la puissance est donnée par :

TC : $P_X = R_{XX}(0) + m_X^2 = \int_{-\infty}^{\infty} S_X(f) df + m_X^2$

$$P_X = R_{XX}(0) + |m_X|^2 = \int_{-\infty}^{\infty} S_X(f) df + |m_X|^2 \quad \text{dans le cas de PA complexes}$$

TD : $P_X = R_{XX}(0) + m_X^2 = \int_{-1/2}^{1/2} S_X(f) df + m_X^2$

$$P_X = R_{XX}(0) + |m_X|^2 = \int_{-1/2}^{1/2} S_X(f) df + |m_X|^2 \quad \text{dans le cas de PA complexes}$$

Bruit blanc

Un bruit blanc $B(t)$ est un Processus Aléatoire stationnaire (SSL) à TC ou à TD, (généralement centré), dont la DSP $S_B(f)$ est *Constante* sur tout l'axe des fréquences (le nom « blanc » fait donc référence à la lumière blanche dont la puissance est répartie uniformément sur l'ensemble des fréquences optiques). Du fait de la définition de la DSP, un bruit blanc est donc caractérisé aussi par une *fonction d'autocovariance* $R_{BB}(\tau)$ *impulsionnelle* :

Bruit blanc de variance σ^2 :

Autocovariance : (TC) $R_{BB}(\tau) = \sigma^2 \delta(\tau)$ (TD) $R_{BB}(k) = \sigma^2 \delta_{k,0}$

DSP (TC) (Variance (TD)) : DSP (TC) $S_B(f) = TF [R_{BB}(\tau)] = \sigma^2$ Variance (TD)
 $S_B(f) = TF [R_{BB}(k)] = \sigma^2$

Autre définition : (sens faible) Un bruit blanc est une suite de VA *non corrélées* (réalisations *non corrélées*)
 (sens fort) Un bruit blanc est une suite de VA *indépendantes* (réalisations *indépendantes*).

Le terme « blancheur » vient de l'analogie avec la lumière blanche et traduit le fait que toutes les fréquences sont présentes dans un bruit blanc avec la même puissance.

Un bruit blanc à TD est réalisable en pratique alors qu'un bruit blanc à TC ne l'est pas car sa puissance (qui est égale à sa fonction d'autocovariance en 0) est infinie (Dirac) !

Loi de distribution : un processus blanc peut avoir une loi de distribution *quelconque* : normale, uniforme, ...

Ergodicité (≡ ergodisme) (relativement à la moyenne et à la fonction de covariance) d'un Processus Aléatoire

Un Processus Aléatoire peut être vu comme une multitude de trajectoires correspondant à autant de réalisations de l'expérience à l'identique.

Cependant, dans un grand nombre de cas pratiques, *une seule réalisation* du processus est accessible à la mesure.

Un processus stationnaire $X(t)$ est *ergodique* si sa *moyenne* et sa *fonction d'autocovariance* peuvent être obtenues en effectuant une *moyenne temporelle* sur *une seule trajectoire* (une seule réalisation) de *durée infinie*.

Ainsi, les formules de moyenne et de fonction d'autocovariance peuvent s'écrire à l'aide d'une moyenne temporelle :

$$\left\{ \begin{array}{l} m_X = \lim_{T \rightarrow +\infty} \frac{1}{2T+1} \sum_{t=-T}^T X(t) \\ R_{XX} = \lim_{T \rightarrow +\infty} \frac{1}{2T+1} \sum_{t=-T}^T X(t+\tau)X(t) \end{array} \right. \quad \left\{ \begin{array}{l} m_X = \lim_{T \rightarrow +\infty} \frac{1}{2T+1} \sum_{t=-T}^T X(t) \\ R_{XX} = \lim_{T \rightarrow +\infty} \frac{1}{2T+1} \sum_{t=-T}^T X(t+\tau)X^*(t) \end{array} \right. \quad \text{dans le cas de PA complexes}$$

Un processus $X(t)$ ergodique est donc tel que sa moyenne statistique (dont une estimation est : $\hat{m}_X(t) = \frac{1}{K} \sum_{k=1}^K x_k(t)$)

est égale à sa moyenne temporelle (dont une estimation est : $\hat{m}_X = \frac{1}{T} \sum_{t=0}^{T-1} X(t)$).

Dès lors : . la moyenne d'un Processus Aléatoire (PA) ergodique est vue comme un **offset (décalage) vertical au cours du temps**
 . la variance, la puissance d'un PA est liée à l'**amplitude du PA (l'amplitude de ses réalisations)**.

Stationnarité et ergodicité d'un Processus Aléatoire

La *quasi totalité des Processus Aléatoires* rencontrés en pratique satisfait aux conditions de *stationnarité et d'ergodicité*.

Cette hypothèse conduit aux formules suivantes d'estimation de la moyenne et de la fonction d'autocovariance :

$$\left\{ \begin{array}{l} \hat{m}_X = \frac{1}{T} \sum_{t=0}^{T-1} X(t) \\ R_{XX} = \frac{1}{T-\tau} \sum_{t=0}^{T-\tau-1} X^0(t+\tau)X^0(t) \end{array} \right. \quad \left\{ \begin{array}{l} \hat{m}_X = \frac{1}{T} \sum_{t=0}^{T-1} X(t) \\ R_{XX} = \frac{1}{T-\tau} \sum_{t=0}^{T-\tau-1} X^0(t+\tau)X^{0*}(t) \end{array} \right. \quad \text{dans le cas de PA complexes}$$

où : $X^0(t) = X(t) - \hat{m}_X$

Fonction de covariance et fonction de corrélation d'un Processus Aléatoire

La Fonction de covariance d'un Processus Aléatoire devient fonction de corrélation lorsque le processus est stationnaire :

Fonction de Covariance $R_{XY}(t_1, t_2)$ de 2 Processus Aléatoires $X(t)$ et $Y(t)$ SSL

$$R_{XY}(t_1, t_2) = E[X^0(t-t_1)Y^0(t-t_2)]$$

Fonction de Corrélation $R_{XY}(t_1 - t_2)$ de 2 Processus Aléatoires $X(t)$ et $Y(t)$ SSL

$$R_{XY}(t_1 - t_2) = E[X^0(t-t_1)Y^0(t-t_2)] = R_{XY}(t_1, t_2)$$

Formule du filtrage

Soit $X(t)$ un Processus Aléatoire SSL, de fonction d'autocovariance $R_{XX}(\tau)$ et de DSP $S_X(f)$, entrée d'un filtre linéaire de RI $h(t)$ et de gain complexe $H(f)$: (si $X(t)$ est à TC, $t \in \mathbf{R}$ et si $X(t)$ est à TD, $t \in \mathbf{Z}$, aussi noté $X_k, k \in \mathbf{Z}$)

$$TC : X(t) \longrightarrow \boxed{h(t)} \longrightarrow Y(t) \qquad TD : X(t) \longrightarrow \boxed{h(t)} \longrightarrow Y(t)$$

On montre que le processus $Y(t)$ en sortie du filtre linéaire est lui-même un Processus Aléatoire SSL.

TC :

. Moyenne de $Y(t)$	$m_Y = H(0)m_X$	$Y(t)$ est donc centré si $X(t)$ l'est.
. Filtrage de la DSP	$S_Y(f) = H(f) ^2 S_X(f)$	Autocorrélation de $Y(t)$ $R_{YY}(\tau) = h(\tau) * h(-\tau) * R_{XX}(\tau)$
. Interspectre	$S_{XY}(f) = H(f) S_X(f)$	Intercorrélation $R_{XY}(\tau) = h(\tau) * R_{XX}(\tau)$

De même à TD :

. Moyenne de Y_k	$m_Y = H(0)m_X$	Y_k est donc centré si X_k l'est.
. Filtrage de la DSP	$S_Y(f) = H(f) ^2 S_X(f)$	Autocorrélation de Y_k $R_{YY_k} = h_k * h_{-k} * R_{XX_k}$
. Interspectre	$S_{XY}(f) = H(f) S_X(f)$	Intercorrélation $R_{XY_k} = h_k * R_{XX_k}$

Modèles de processus aléatoires (PAs)

Le processus de Wiener (processus aléatoire non stationnaire)

Il modélise le mouvement Brownien, mouvement irrégulier de particules microscopiques soumises à l'agitation thermique dans un fluide. Soit $X(t)$ une des coordonnées de la particule, et choisissons l'origine telle que $X(0) = 0$. Le mouvement de la particule sur un intervalle de temps suffisamment long est le résultat d'une série d'impulsions provenant d'un nombre important de collisions. Il est donc raisonnable de penser que :

- . le théorème central limite s'applique : le mouvement Brownien est *gaussien* : le déplacement obéit à une loi normale
- . les propriétés statistiques du déplacement sur un intervalle de temps $(t, t + \tau)$ sont indépendantes de celles sur un intervalle de même longueur, pour autant que ces intervalles ne se chevauchent pas. De manière axiomatique, le processus de Wiener est donc défini par les conditions suivantes :

- . $X(0) = 0$
- . $X(k)$ est *gaussien* de moyenne nulle $\forall k$
- . $X(k)$ est un processus à *accroissements indépendants* :

soit $k_1 < k_2 < \dots < k_n$, les VA $X(k_n) - X(k_{n-1}), X(k_{n-1}) - X(k_{n-2}), \dots, X(k_2) - X(k_1), X(k_1)$ sont mutuellement indépendantes (\equiv indépendantes les unes des autres) (si ces VA sont seulement non corrélées entre elles, le processus est dit à accroissements orthogonaux).

Ces propriétés permettent de se donner un modèle pour le processus de Wiener : $X(k) = X(k-1) + W(k)$

avec : $X(0) = 0$

$W(k)$: bruit blanc, gaussien, centré (\equiv moyenne nulle), et de variance σ_W^2

Il résulte ce modèle que :

. $E[X(k)] = 0 \quad \forall k \geq 0$

la moyenne d'un processus de Wiener est nulle: $m_X = 0$

. $X(k) = \sum_{i=1}^k W(i)$

. $C_{XX}(k, l) = E[X(k)X(l)] = \sigma_W^2 \min(k, l)$ qui implique que la variance de $X(k)$ croît linéairement avec k :

$\sigma_X^2(k) = C_{XX}(k, k) = k\sigma_W^2$ la variance d'un process de Wiener croît linéairement:

$\sigma_X^2(k) = k\sigma_W^2$

cela signifie que, plus on s'éloigne de l'origine des temps, plus l'amplitude du déplacement peut être importante.

La Fonction de Transfert du filtre de Wiener est :

$$W(k) \xrightarrow{\boxed{H(z)}} X(k)$$

Bruit blanc Filtre de Wiener

$$H(z) = \frac{X(z)}{W(z)} = \frac{1}{1-z^{-1}} \quad X(z)[1-z^{-1}] = W(z) \xrightarrow{TZ^{-1}} X(k) = X(k-1) + W(k)$$

$H(z)$ a un pôle juste instable en $z = 1$. Il ne s'agit donc pas d'un filtre stationnaire et sa DSP n'est pas définie.

Strictement parlant, le processus de Wiener a été défini comme un processus continu : $X(t) = \int_0^t W(u) du$

où $W(t)$ est un bruit blanc gaussien *continu* centré.

La version discrète du processus de Wiener est appelée « random walk ».

Le processus auto-régressif (AR ou encore RII) (AutoRegressive) (processus aléatoire stationnaire)

$$W_n \xrightarrow{\boxed{H(z)}} Y_n$$

Bruit blanc centré Filtre AR

$$H(z) = \frac{Y(z)}{W(z)} = \frac{1}{1 + \sum_{i=1}^N a_i z^{-i}} \quad (\text{cf. } \S \text{ Systèmes stochastiques})$$

On a immédiatement : $Y(z) \left[1 + \sum_{i=1}^N a_i z^{-i} \right] = W(z) \xrightarrow{TZ^{-1}} Y_n = W_n - \sum_{i=1}^N a_i Y_{n-i}$

Le processus à moyenne mobile (MA ou encore RIF) (Moving Average) (processus aléatoire stationnaire)

$$W_n \xrightarrow{\boxed{H(z)}} Y_n$$

Bruit blanc centré Filtre MA

$$H(z) = \sum_{i=0}^N b_i z^{-i} \quad (\text{cf. } \S \text{ Systèmes stochastiques})$$

On a immédiatement : $Y(z) = H(z)W(z) = \sum_{i=0}^N b_i z^{-i} W(z) \xrightarrow{TZ^{-1}} Y_n = \sum_{i=0}^N b_i W_{n-i}$

Le processus auto-régressif à moyenne mobile (ARMA) (AutoRegressive with Moving Average) (PA stationnaire)

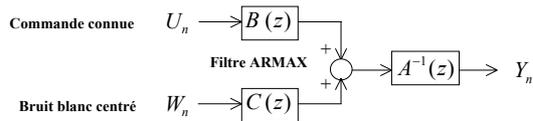
$$W_n \xrightarrow{\boxed{H(z)}} Y_n$$

Bruit blanc centré Filtre ARMA

$$H(z) = \frac{\sum_{i=0}^N b_i z^{-i}}{1 + \sum_{i=1}^N a_i z^{-i}} \quad (\text{cf. } \S \text{ Systèmes stochastiques})$$

$$H(z) = \frac{Y(z)}{W(z)} = \frac{\sum_{i=0}^N b_i z^{-i}}{1 + \sum_{i=1}^N a_i z^{-i}} \rightarrow Y(z) \left(1 + \sum_{i=1}^N a_i z^{-i} \right) = W(z) \left(\sum_{i=0}^N b_i z^{-i} \right) \xrightarrow{TZ^{-1}} Y_n = \sum_{i=0}^N b_i W_{n-i} - \sum_{i=1}^N a_i Y_{n-i}$$

Le processus ARMAX (Auto Regressive Moving Average with eXogeneous inputs) (processus aléatoire stationnaire)

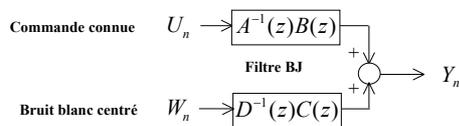


$$A(z)Y(z) = B(z)U(z) + C(z)W(z) \quad \text{avec} \quad \begin{cases} A(z) = 1 + \sum_{i=1}^N a_i z^{-i} & (\text{polynôme stable}) \\ B(z) = \sum_{i=0}^M b_i z^{-i} \\ C(z) = 1 + \sum_{i=1}^P c_i z^{-i} & (\text{polynôme stable}) \end{cases}$$

Contrairement aux précédents, ce modèle n'est pas purement stochastique (réponse à un bruit blanc) mais la sortie combine une composante purement aléatoire (réponse à un bruit blanc) et une réponse à un signal déterministe et connu (dans des applications de contrôle, ce signal est l'entrée de commande).

Les polynômes $A(z)$, $C(z)$ sont stables et moniques (\equiv de coeff. de plus haut degré égal à 1) et le polynôme $B(z)$ quelconque.

Le processus de Box et Jenkins (BJ) (processus aléatoire stationnaire)

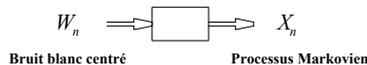


$$Y(z) = \frac{B(z)}{A(z)}U(z) + \frac{C(z)}{D(z)}W(z) \quad \text{avec} \quad \begin{cases} A(z) = 1 + \sum_{i=1}^N a_i z^{-i} & (\text{polynôme monique stable}) \\ B(z) = \sum_{i=0}^M b_i z^{-i} \\ C(z) = 1 + \sum_{i=1}^P c_i z^{-i} & (\text{polynôme monique stable}) \\ D(z) = 1 + \sum_{i=1}^K d_i z^{-i} & (\text{polynôme monique stable}) \end{cases}$$

Généralisation du processus ARMAX avec les polynômes $A(z)$, $C(z)$, $D(z)$ stables et moniques (\equiv de coeff. de plus haut degré égal à 1) et le polynôme $B(z)$ quelconque.

Le processus Markovien et modèle d'Etat (cas des processus de Markov linéaires stationnaires à TD)

Processus Markovien



Un processus stochastique $X(t)$ est **Markovien** si pour tout instant $t_1 < t_2 < \dots < t_k < t$, la densité de probabilité conditionnelle a la propriété suivante : $p_X[x(t) | x(t_1), x(t_2), \dots, x(t_k)] = p_X[x(t) | x(t_k)]$

Donc, même si plusieurs valeurs antérieures de la réalisation du processus sont connues, seule la plus récente apporte de l'information sur la densité de probabilité de $X(t)$.

(Le processus de Wiener décrit plus haut est un cas particulier de processus de Markov).

Si de plus, $X(t)$ est une séquence aléatoire qui ne peut prendre qu'un nombre fini de valeurs possibles, la séquence est une **chaîne de Markov**.

Un processus de Markov linéaire stationnaire est décrit par le modèle d'Etat : $X(k+1) = AX(k) + W(k)$ où :

- . $X(k)$ est un processus stochastique vectoriel : $X(k) \in \mathbf{R}^N$
- . A est une matrice carrée constante (du fait de la stationnarité) : $A \in \mathbf{R}^{N \times N}$
- . $W(k)$ est un bruit blanc (variance Q) vectoriel centré indépendant de $X(k)$: $W(k) \in \mathbf{R}^N$

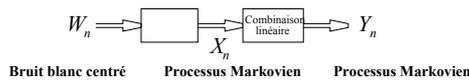
. Moyenne $m_X(k)$ de $X(k)$: $m_X(k) = A^k m_X(0)$

. Covariance $C_{XX}(k+j, k)$ de $X(k)$ ($j \geq 0$) : $C_{XX}(k+j, k) = A^j C_{XX}(k, k)$

. Covariance $C_{XX}(k, k+j)$ de $X(k)$ ($j \geq 0$) : $C_{XX}(k, k+j) = C_{XX}(k, k)(A^j)^T$

avec : $C_{XX}(k, k) = E[(X(k) - m_X(k))(X(k) - m_X(k))^T]$

Processus à représentation Markovienne



Le processus auquel on s'intéresse est parfois une combinaison linéaire stationnaire bruitée d'un processus linéaire Markovien.

Un tel processus est décrit par le modèle d'Etat :

$$\begin{cases} X(k+1) = AX(k) + W(k) \\ Y(k) = CX(k) + V(k) \end{cases} \quad \text{où :}$$

. $X(k)$ est un processus stochastique vectoriel :

$$X(k) \in \mathbf{R}^N$$

. $Y(k)$ est un processus stochastique vectoriel :

$$Y(k) \in \mathbf{R}^M$$

. A est une matrice carrée constante (du fait de la stationnarité) :

$$A \in \mathbf{R}^{N \times N}$$

. C est une matrice constante (du fait de la stationnarité) :

$$C \in \mathbf{R}^{M \times N}$$

. $W(k)$ est un bruit blanc (variance Q) vectoriel centré indépendant de $X(k)$ et non corrélé à $V(k)$: $W(k) \in \mathbf{R}^N$

. $V(k)$ est un bruit blanc (variance R) vectoriel centré indépendant de $X(k)$ et non corrélé à $W(k)$: $V(k) \in \mathbf{R}^M$

. Moyenne $m_Y(k)$ de $Y(k)$:

$$m_Y(k) = C m_X(k)$$

. Covariance $C_{YY}(k+j, k)$ de $Y(k)$ ($j \geq 0$) :

$$C_{YY}(k+j, k) = C A^j C_{XX}(k, k) C^T + R \delta_{j,0}$$

. Covariance $C_{YY}(k, k+j)$ de $Y(k)$ ($j \geq 0$) :

$$C_{YY}(k, k+j) = C C_{XX}(k, k) (A^j)^T C^T + R \delta_{j,0}$$

1 ANNEXE. Signaux déterministes. Signaux aléatoires. Systèmes stochastiques

. III. SYSTEMES STOCHASTIQUES

. ESTIMATION DE LA COVARIANCE

. RESOLUTION DU SYSTEME D'EQUATIONS DE YULE-WALKER PAR INVERSION MATRICIELLE

. ALGORITHME DE LEVINSON *Résolution itérative du système de Yule-Walker*

. METHODE DE BURG *Résolution itérative du système de Yule-Walker avec contrainte de stabilité*

III. SYSTEMES STOCHASTIQUES

Un système stochastique (ainsi appelé abusivement) est un système déterministe excité par un signal aléatoire.

1. Transmission d'un signal aléatoire dans un système linéaire (utilisation des transformées)

Soit un système linéaire à Temps Continu $H(s)$ (ou à Temps Discret $H(z)$) et de Réponse Impulsionnelle $h(t)$ (resp. h_n) excité par un signal aléatoire X , stationnaire et ergodique, de valeur moyenne \bar{X} et de fonction d'autocorrélation : $\varphi_{xx}(\tau) = E[X^0(t)X^0(t+\tau)]$ où : $X^0(t) = X(t) - \bar{X}$ (resp. à Temps Discret : $\varphi_{xxk} = E[X_n^0 X_{n+k}^0]$).

$$TC : X(t) \xrightarrow{H(s)} Y(t)$$

$$TD : X_n \xrightarrow{H(z)} Y_n$$

1.1. Valeur moyenne de la sortie

TC :

On a : $Y(t) = X(t) * h(t)$ d'où : $E[Y(t)] = E[X(t) * h(t)] = E[X(t)] * h(t)$ (le système est déterministe)

$$\boxed{\bar{Y} = \bar{X} * h(t)}$$
 et : $Y^0(t) = X^0(t) * h(t)$

$$\text{car : } Y^0(t) = Y(t) - \bar{Y} = X(t) * h(t) - \bar{X} * h(t) = [X(t) - \bar{X}] * h(t) = X^0(t) * h(t)$$

TD :

$$\text{De même : } \boxed{E[Y_n] = \bar{X} * h_n}$$

De façon à alléger les écritures, on supposera dans la suite que $X(t)$ (resp. X_n) et $Y(t)$ (resp. Y_n) sont des signaux aléatoires centrés.

1.2. Intercorrélation

TC :

$$\varphi_{xy}(\tau) = E[X(t-\tau)Y(t)] = E[X(t-\tau)\{X(t) * h(t)\}] = E\left[X(t-\tau) \int_{-\infty}^{\infty} h(u)X(t-u)du\right]$$

$$\varphi_{xy}(\tau) = \int_{-\infty}^{\infty} h(u)E[X(t-\tau)X(t-u)]du = \int_{-\infty}^{\infty} h(u)\varphi_{xx}(\tau-u)du = \varphi_{xx}(\tau) * h(\tau)$$

$$\boxed{\varphi_{xy}(\tau) = h(\tau) * \varphi_{xx}(\tau)} \xrightarrow{TL} \boxed{\Phi_{XY}(s) = H(s) \cdot \Phi_{XX}(s)}$$

TD :

$$\text{De même : } \boxed{\varphi_{xyk} = h_k * \varphi_{xxk}} \xrightarrow{TZ} \boxed{\Phi_{XY}(z) = H(z) \cdot \Phi_{XX}(z)}$$

1.3. Autocorrélation et DSP (Densité Spectrale de Puissance)

TC :

$$\varphi_{yy}(\tau) = E[Y(t)Y(t+\tau)] = E\left[\int_{-\infty}^{\infty} h(u)X(t+\tau-u)du \cdot \int_{-\infty}^{\infty} h(v)X(t-v)dv\right] = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h(u)h(v)E[X(t+\tau-u)X(t-v)]dudv$$

$$\varphi_{yy}(\tau) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h(u)h(v)\varphi_{xx}(\tau-u+v)dudv \xrightarrow{TL} \Phi_{YY}(s) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h(u)h(v)\varphi_{xx}(\tau-u+v)e^{-s\tau}dudvd\tau$$

soit, en posant : $\tau - u + v = \lambda$: $\Phi_{YY}(s) = \int_{-\infty}^{\infty} h(u)e^{-su}du \int_{-\infty}^{\infty} h(v)e^{sv}dv \int_{-\infty}^{\infty} \varphi_{xx}(\lambda)e^{-s\lambda}d\lambda$

d'où : $\Phi_{YY}(s) = H(s) \cdot H(-s) \cdot \Phi_{XX}(s)$ et : $\xrightarrow{TL^{-1}} \varphi_{yy}(\tau) = h(\tau) * h(-\tau) * \varphi_{xx}(\tau)$

En prenant : $s = j\omega$ (régime harmonique), on obtient la relation entre les Densités Spectrales de Puissance (DSP) :

$$\Phi_{YY}(\omega) = |H(\omega)|^2 \cdot \Phi_{XX}(\omega) \text{ encore noté : } S_Y(f) = |H(f)|^2 \cdot S_X(f) \quad f : \text{fréquence}$$

TD :

De même :

$$\varphi_{yyk} = \sum_{u=-\infty}^{\infty} \sum_{v=-\infty}^{\infty} \int_{-\infty}^{\infty} h_u h_v \varphi_{xx k-u+v} \xrightarrow{TZ} \Phi_{YY}(z) = H(z) \cdot H(z^{-1}) \cdot \Phi_{XX}(z)$$

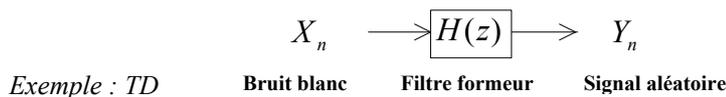
et : $\Phi_{YY}(z) = H(z) \cdot H(z^{-1}) \cdot \Phi_{XX}(z) \xrightarrow{TZ^{-1}} \varphi_{yyk} = h_k * h_{-k} * \varphi_{xxk}$

En prenant : $z = e^{j\omega}$ (régime harmonique), on obtient la relation entre les Densités Spectrales de Puissance (DSP) :

$$\Phi_{YY}(\omega) = |H(\omega)|^2 \cdot \Phi_{XX}(\omega) \text{ encore noté : } S_Y(f) = |H(f)|^2 \cdot S_X(f) \quad f : \text{fréquence}$$

2. Processus générateur d'un signal aléatoire : filtres formeurs du 1er ordre

Un signal aléatoire peut être synthétisé (modélisé) comme la réponse d'un filtre linéaire excité par un bruit blanc :



Le filtre linéaire en question est appelé filtre formeur du signal ou encore processus générateur.

La description du signal peut alors consister en les paramètres du modèle linéaire (≡ du filtre linéaire) et en la variance du bruit blanc (sa DSP dans le cas continu).

2.1. Processus générateur à Temps Discret (TD) du 1er ordre

Soit l'équation récurrente stochastique du 1er ordre (correspondant à un filtre linéaire causal de FT : $H(z) = \frac{1}{z-a}$) :

$$Y_{n+1} = aY_n + X_n \text{ avec } |a| < 1 \text{ (filtre stable) où : } \{X_n\} \text{ est une séquence indépendante (≡ bruit blanc)}$$

$$\{X_n\} \text{ bruit blanc centré de variance } V : \begin{cases} E[X_n] = 0 \\ E[X_n^2] = V \\ E[X_n \cdot X_{n+k}] = 0 \quad \forall k \neq 0 \end{cases} \equiv \begin{cases} E[X_n] = 0 \\ \varphi_{xxk} = E[X_n \cdot X_{n+k}] = V\delta_{k,0} \end{cases}$$

D'autre part, le filtre étant récursif il faut préciser les CI :
$$\begin{cases} E[Y_0] \stackrel{\Delta}{=} \bar{y}_0 \\ E[Y_0^2] \stackrel{\Delta}{=} E[Y_0 - \bar{y}_0]^2 \stackrel{\Delta}{=} P_0 \end{cases} \quad \text{où : } Y_n^0 \stackrel{\Delta}{=} Y_n - \bar{y}_n$$

On note h_n la RI du filtre. On a : $H(z) = \frac{Y(z)}{X(z)} = \frac{z^{-1}}{1 - az^{-1}} \xrightarrow{TZ^{-1}} h_n = a^{n-1} \Gamma_{n-1}$. En particulier $h_0 = 0$.

Analyse : Examen des caractéristiques statistiques de la séquence $\{Y_n\}$ en réponse à un bruit blanc $\{X_n\}$

. Moyenne : $E[\{Y_n\}] = ?$ (Rappel : $E[\cdot]$ est un opérateur linéaire)

$$E[Y_{n+1}] = aE[Y_n] \quad (\{X_n\} \text{ est centrée}) \rightarrow \boxed{E[Y_n] = \bar{y}_n = a^n \bar{y}_0} \quad \text{Comme } |a| < 1 : \boxed{\bar{y}_n \rightarrow 0} \text{ quand } n \rightarrow \infty$$

Une fois le régime transitoire terminé, le filtre formeur fournit donc un signal stationnaire (du 1er ordre) centré.

Un signal stationnaire du 1er ordre est tel que sa moyenne est indépendante de l'origine des temps.

Un signal stationnaire du 2nd ordre, dit stationnaire au sens large est tel que sa moyenne et sa variance sont indépendantes de l'origine des temps.

. Variance : $Var[\{Y_n\}] = P_n = ?$

$$[Y_{n+1} - \bar{y}_{n+1}]^2 = [aY_n + X_n - a\bar{y}_n]^2 = [a(Y_n - \bar{y}_n) + X_n]^2 = [aY_n^0 + X_n]^2 = a^2 Y_n^{0^2} + 2aY_n^0 X_n + X_n^2$$

$$E[Y_{n+1} - \bar{y}_{n+1}]^2 = E[a^2 Y_n^{0^2} + 2aY_n^0 X_n + X_n^2] = a^2 E[Y_n^{0^2}] + E[X_n^2] + 2aE[(Y_n - \bar{y}_n)X_n]$$

$$E[Y_{n+1} - \bar{y}_{n+1}]^2 = a^2 E[Y_n^{0^2}] + E[X_n^2] + 2aE[X_n Y_n] - 2a\bar{y}_n E[X_n] = a^2 E[Y_n^{0^2}] + E[X_n^2] + 2aE[X_n Y_n]$$

Or: $\varphi_{xy_k} = h_k * \varphi_{xx_k} = h_k * (V\delta_{k,0}) = V(h_k * \delta_{k,0}) = V \cdot h_k$ et $E[X_n Y_n] = \varphi_{xy_0} = V \cdot h_0 = 0$ (car $h_0 = 0$)

$$\rightarrow E[Y_{n+1} - \bar{y}_{n+1}]^2 = a^2 E[Y_n^{0^2}] + E[X_n^2] \quad \text{soit : } \boxed{P_{n+1} = a^2 P_n + V}$$

Cette équation de récurrence donnant la variance de $\{Y_n\}$ est initialisée par P_0 . Sa solution est : (à l'aide de la TZ par ex.)

$$P_n = a^{2n} P_0 + V(1 + a^2 + \dots + a^{2n}) \quad \text{d'où : } \boxed{P_\infty = \frac{V}{1-a^2}} \quad (\text{car } |a| < 1).$$

Une fois le régime transitoire terminé, le filtre formeur fournit donc un signal stationnaire centré, de variance $\frac{V}{1-a^2}$ (régime stationnaire ou encore permanent).

. Autocorrélation de $\{Y_n\}$: ? $\varphi_{yy_k} = E[Y_n Y_{n+k}] = ?$

$$\begin{aligned} Y_{n+1} &= aY_n + X_n & Y_{n+2} &= a^2 Y_n + aX_n + X_{n+1} & Y_{n+3} &= a^3 Y_n + a^2 X_n + aX_{n+1} + X_{n+2} & \dots \\ Y_{n+k} &= a^k Y_n + a^{k-1} X_n + a^{k-2} X_{n+1} + \dots + X_{n+k-1} \end{aligned}$$

Donc, pour $k > 0$: $\varphi_{yy_k} = E[Y_n Y_{n+k}] = E[Y_n (a^k Y_n + a^{k-1} X_n + \dots + X_{n+k-1})] = a^k E[Y_n^2] = a^k P_\infty = a^k \frac{V}{1-a^2}$

L'autocorrélation étant une séquence paire, on a :
$$\boxed{\varphi_{yy_k} = a^{|k|} \frac{V}{1-a^2}}$$

Synthèse : Dédution du filtre formeur en fonction des caractéristiques statistiques de la séquence $\{Y_n\}$ donnée

Inversement, si l'on veut générer un signal aléatoire $\{Y_n\}$ stationnaire centré, d'autocorrélation $\varphi_{yy_k} = P a^{|k|}$, il suffit d'utiliser un filtre du 1er ordre, d'équation : $Y_{n+1} = aY_n + X_n$ excité par une séquence blanche $\{X_n\}$ centrée, et de variance $V = (1-a^2)P$.

2.2. Processus générateur à Temps Continu (TC) du 1er ordre

La démarche est identique qu'à TD.

Soit l'équation différentielle stochastique du 1er ordre (correspondant à un filtre linéaire causal de FT : $H(s) = \frac{1}{s-a}$):

$$\dot{Y}(t) = \alpha Y(t) + X(t) \quad \text{avec } \alpha < 0 \text{ (filtre stable)} \quad \text{où : } X(t) \text{ est un bruit blanc}$$

$$X(t) \text{ bruit blanc centré de DSP } V : \begin{cases} E[X(t)] = 0 \\ \varphi_{xx}(\tau) = E[X(t) \cdot X(t+\tau)] = Q\delta(\tau) \\ E[X(t) \cdot Y(\theta)] = 0 \quad \forall \theta < t \end{cases}$$

On peut écrire que $X(t)$ est la dérivée d'un processus de Wiener $W(t)$: $X(t) = \frac{dW(t)}{dt}$

et l'équation $\dot{Y}(t) = \alpha Y(t) + X(t)$ peut se mettre sous la forme : $dY(t) = \alpha Y(t)dt + dW(t)$

Si on discrétise cette dernière relation (au pas d'échantillonnage T), on obtient : $Y_{n+1} - Y_n = \alpha Y_n T + W_{n+1} - W_n$

Les accroissements $W_{k+1} - W_k$, étant indépendants, constituent une séquence indépendante (\equiv blanche) $\{B_n\}$ centrée et de variance $E[B_n^2] = QT$. La relation précédente : $Y_{n+1} - Y_n = \alpha Y_n T + W_{n+1} - W_n$ s'écrit donc :

$$Y_{n+1} = (1 + \alpha T)Y_n + B_n \quad \text{relation de la même forme que celle du cas discret avec : } \begin{cases} a = 1 + \alpha T \\ V = QT \end{cases}$$

Tous les résultats du cas discret traité avant s'appliquent, et ils sont ensuite ramenés à TC (en faisant $T \rightarrow 0$) :

Analyse : Examen des caractéristiques statistiques de la réponse $Y(t)$ à un bruit blanc $X(t)$

. **Moyenne :** $E[Y(t)] = ?$

$$\bar{y}_{n+1} = (1 + \alpha T)\bar{y}_n \rightarrow \text{quand } T \rightarrow 0 \quad \boxed{\bar{y}'(t) = \alpha \bar{y}(t)} \quad \text{Comme } \alpha < 0 : \quad \boxed{\bar{y}(t) \rightarrow 0} \quad \text{quand } t \rightarrow \infty$$

En régime permanent, le filtre formeur fournit donc un signal centré.

. **Variance :** Var de $Y(t)$?

$$\text{D'après les résultats du cas discret : } \quad \boxed{P_{n+1} = (1 + \alpha T)^2 P_n + QT}$$

Quand $T \rightarrow 0$ (en négligeant les termes en T^2) : $\dot{P}(t) = 2\alpha P(t) + Q$

$$\text{En régime stationnaire (} \dot{P} \equiv 0 \text{ : équilibre atteint : } P_{n+1} = P_n \text{): } \quad \boxed{P(\infty) = -\frac{Q}{2\alpha}}$$

. **Autocorrélation de $Y(t)$:** $\varphi_{yy}(\tau) = E[Y(t)Y(t+\tau)] = ?$

$$\text{Le résultat du cas discret donne ici : } \quad \varphi_{yyk} = (1 + \alpha T)^{|k|} \frac{QT}{1 - (1 + \alpha T)^2}$$

$$\text{Quand } T \rightarrow 0 : (1 + \alpha T)^{|k|} \approx e^{\alpha|k|T} \text{ donc : } \varphi_{yyk} = -\frac{Q}{2\alpha} e^{\alpha|k|T} \text{ et : } \boxed{\varphi_{yy}(\tau) = -\frac{Q}{2\alpha} e^{\alpha|\tau|}} \quad (\text{TD} \rightarrow \text{TC: } \tau = kT)$$

Synthèse : Dédution du filtre formeur en fonction des caractéristiques statistiques du signal $Y(t)$ donné

Inversement, si l'on veut générer un signal aléatoire $Y(t)$ stationnaire centré, d'autocorrélation $\varphi_{yy}(\tau) = P e^{\alpha|\tau|}$,

il suffit d'utiliser un filtre du 1er ordre, d'équation : $\dot{Y}(t) = \alpha Y(t) + X(t)$ excité par un bruit blanc $X(t)$ centré, et de DSP $Q = -2\alpha P$.

3. Processus générateurs : MA, AR, ARMA

Le concept de processus générateur de signal a été particulièrement développé et appliqué avec des filtres numériques ayant une Fonction de Transfert en z de la forme :

$$TD : \quad H(z) = \frac{\sum_{i=0}^N b_i z^{-i}}{1 + \sum_{i=1}^N a_i z^{-i}} \quad \begin{array}{ccc} X_n & \xrightarrow{H(z)} & Y_n \\ \text{Bruit blanc} & \text{Filtre MA, AR, ARMA} & \end{array}$$

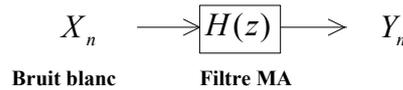
Lorsque le processus $H(z)$ est excité par une séquence $\{X_n\}$ indépendante (\equiv blanche), la sortie $\{Y_n\}$ est un signal dit ARMA (Auto Regressive and Moving Average).

Si $H(z) = \sum_{i=0}^N b_i z^{-i}$, le signal $\{Y_n\}$ est MA. (Le filtre $H(z)$ est RIF (\equiv à Réponse Impulsionnelle Finie)

Si $H(z) = \frac{1}{1 + \sum_{i=1}^N a_i z^{-i}}$, le signal $\{Y_n\}$ est AR. (Le filtre $H(z)$ est RII (\equiv à Réponse Impulsionnelle Infinie)

Nous allons étudier chacun de ces 3 processus générateurs. Etant donné les caractéristiques statistiques de $\{Y_n\}$ (c'est-à-dire φ_{yy_k}), il s'agit de déterminer les paramètres a_i et b_i du filtre $H(z)$ qui génère $\{Y_n\}$ à partir de $\{X_n\}$, dont il faut aussi connaître la variance.

3.1. Signal MA



Les hypothèses sont :

$$\begin{cases} E[X_n] = 0 \\ E[X_n^2] = V \\ E[X_n \cdot X_{n+k}] = 0 \quad \forall k \neq 0 \end{cases} \quad \equiv \quad \begin{cases} E[X_n] = 0 \\ \varphi_{xxk} = E[X_n \cdot X_{n+k}] = V\delta_{k,0} \end{cases}$$

($\{X_n\}$ bruit blanc centré de variance V)

et : $H(z) = \sum_{i=0}^N b_i z^{-i} = \sum_{i=0}^N h_i z^{-i}$

En effet, comme $H(z)$ est la TZ de la RI h_n : $H(z) = \sum_{i=0}^N h_i z^{-i}$, on a immédiatement : $h_i = b_i \quad i \in [0, N]$.

On a immédiatement : $Y(z) = H(z)X(z) = \sum_{i=0}^N h_i z^{-i} X(z) \xrightarrow{TZ^{-1}} Y_n = \sum_{i=0}^N h_i X_{n-i}$

. **Moyenne de $\{Y_n\}$:** ? $m = E[Y_n] = 0$ puisque le bruit blanc d'entrée est centré.

. **Autocorrélation de $\{Y_n\}$:** ? $\varphi_{yy_k} = E[Y_{n-k} Y_n] = E[Y_n Y_{n+k}]$?

$$\varphi_{yy_k} = E \left[\sum_{i=0}^N h_i X_{n-i} \sum_{j=0}^N h_j X_{n+k-j} \right] = \sum_{i=0}^N \sum_{j=0}^N h_i h_j E[X_{n-i} X_{n+k-j}] = V \sum_{i=0}^{N-k} h_i h_{k+i} \quad (\varphi_{yy_k} = 0 \quad \forall k > N)$$

car par hypothèse $E[X_n \cdot X_{n+k}] = V\delta_{k,0}$ d'où $E[X_{n-i} X_{n+k-j}]$ est toujours nulle sauf pour $j = k + i$.

Les équations $\varphi_{yy_k} = V \sum_{i=0}^{N-k} h_i h_{k+i}$ sont non linéaires en les paramètres $\{h_i\}$ du filtre. Il faudrait utiliser un algorithme de Programmation Non Linéaire pour obtenir la séquence $\{h_i\}$ du filtre à partir de la séquence $\{\varphi_{yy_k}\}$.

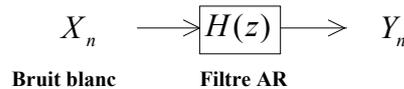
• Puissance P de $\{Y_n\}$: ? $P = \varphi_{yy_0} = V \sum_{i=0}^N h_i^2$

• DSP $\Phi_{YY}(\omega)$ de $\{Y_n\}$: ? $\Phi_{YY}(\omega) = V |H(e^{-j\omega})|^2 = V \left| \sum_{m=0}^N h_m e^{-jm\omega} \right|^2$

ou encore en prenant la TF (Discrète) de la séquence de corrélation : $\Phi_{YY}(\omega) = \sum_{k=-\infty}^{\infty} \varphi_{yy_k} e^{-jk\omega}$

et comme $\Phi_{YY}(\omega)$ est pair quand le signal Y est réel : $\Phi_{YY}(\omega) = \varphi_{yy_0} + 2 \sum_{k=1}^{\infty} \varphi_{yy_k} \cos(k\omega)$

3.2. Signal AR



Hypothèses : $\begin{cases} E[X_n] = 0 \\ E[X_n^2] = V \\ E[X_n \cdot X_{n+k}] = 0 \quad \forall k \neq 0 \end{cases} \equiv \begin{cases} E[X_n] = 0 \\ \varphi_{xxk} = E[X_n \cdot X_{n+k}] = V \delta_{k,0} \end{cases}$ et $H(z) = \frac{1}{1 + \sum_{i=1}^N a_i z^{-i}} = \frac{Y(z)}{X(z)}$

($\{X_n\}$ bruit blanc centré de variance V)

On a immédiatement : $Y(z) \left[1 + \sum_{i=1}^N a_i z^{-i} \right] = X(z) \xrightarrow{TZ^{-1}} Y_n = X_n - \sum_{i=1}^N a_i Y_{n-i}$

Cette fois, la RI $\{h_n\}$ n'apparaît pas directement, mais on sait que : $\varphi_{xy_k} = h_k * \varphi_{xxk}$

Ici $\varphi_{xxk} = V \delta_{k,0}$ donc $\varphi_{xy_k} = h_k * (V \delta_{k,0}) = V (h_k * \delta_{k,0}) = V \cdot h_k$ $h_k = \frac{\varphi_{xy_k}}{V}$

$h_k = \frac{\varphi_{xy_k}}{V} = \frac{E[X_{n-k} Y_n]}{V} = \frac{E \left[\left(Y_{n-k} + \sum_{i=1}^N a_i Y_{n-k-i} \right) Y_n \right]}{V}$ car $X_n = Y_n + \sum_{i=1}^N a_i Y_{n-i}$

\rightarrow $h_k = \frac{\varphi_{yy_k} + \sum_{i=1}^N a_i \varphi_{yy_{k+i}}}{V}$ $\rightarrow \varphi_{yy_k} = V \cdot h_k - \sum_{i=1}^N a_i \varphi_{yy_{k+i}}$

• Moyenne de $\{Y_n\}$: ? $m = E[Y_n] = 0$ (car bruit blanc d'entrée centré) si les CI sont nulles ou de moyenne nulle.

• Autocorrélation de $\{Y_n\}$: ? $\varphi_{yy_k} = E[Y_{n-k} Y_n] = ?$

$\varphi_{yy_k} = E[Y_{n-k} Y_n] = E \left[Y_{n-k} \left(X_n - \sum_{i=1}^N a_i Y_{n-i} \right) \right] = E[X_n Y_{n-k}] - E \left[\sum_{i=1}^N a_i Y_{n-i} Y_{n-k} \right] = E[X_{n+k} Y_n] - \sum_{i=1}^N a_i E[Y_{n-i} Y_{n-k}]$

$\varphi_{yy_k} = V \cdot h_{-k} - \sum_{i=1}^N a_i \varphi_{yy_{k-i}}$ car : $h_k = \frac{\varphi_{xy_k}}{V} = \frac{E[X_{n-k} Y_n]}{V} \rightarrow h_{-k} = \frac{\varphi_{xy_{-k}}}{V} = \frac{E[X_{n+k} Y_n]}{V} = \frac{\varphi_{yx_k}}{V}$

$$.k = 0 : \varphi_{yy_0} = V \cdot h_0 - \sum_{i=1}^N a_i \varphi_{yy_{-i}} = V \cdot h_0 - \sum_{i=1}^N a_i \varphi_{yy_i} \quad \text{car une autocorrélation est paire.}$$

Comme $h_0 = 1$ (en effet : $Y_n = X_n - \sum_{i=1}^N a_i Y_{n-i} \rightarrow h_n = \delta_n - \sum_{i=1}^N a_i h_{n-i} \rightarrow h_0 = 1$ (filtre au repos initialement))

on a :
$$\varphi_{yy_0} = V - \sum_{i=1}^N a_i \varphi_{yy_i}$$

$$.k > 0 : \varphi_{yy_k} = -\sum_{i=1}^N a_i \varphi_{yy_{k-i}} \quad \text{car } h_{-k} \equiv 0 \text{ pour } k > 0 : \text{ le filtre } h_n \text{ est causal : } Y_n = X_n - \sum_{i=1}^N a_i Y_{n-i}$$

on a aussi :
$$\varphi_{yy_k} = -\sum_{i=1}^N a_i \varphi_{yy_{i-k}}$$
 car une autocorrélation est paire.

Ces équations peuvent s'écrire :

$$\begin{cases} \varphi_{yy_0} + \sum_{i=1}^N a_i \varphi_{yy_i} = V \\ \varphi_{yy_1} + \sum_{i=1}^N a_i \varphi_{yy_{i-1}} = 0 \\ \dots \\ \varphi_{yy_N} + \sum_{i=1}^N a_i \varphi_{yy_{N-i}} = 0 \end{cases} \quad \Leftrightarrow \text{pour } 0 \leq k \leq N : \sum_{n=0}^N \varphi_{yy_{k-n}} a_n = \begin{cases} V & \text{si } k = 0 \\ 0 & \text{si } 1 \leq k \leq N \end{cases} \quad (a_0 = 1) (\varphi_{yy_k} \text{ paire})$$

soit, sous forme matricielle : ce sont les équations de **Yule-Walker** :

$$\begin{bmatrix} \varphi_{yy_0} & \varphi_{yy_1} & \dots & \varphi_{yy_N} \\ \varphi_{yy_1} & \varphi_{yy_0} & \dots & \varphi_{yy_{N-1}} \\ \vdots & \vdots & \ddots & \vdots \\ \varphi_{yy_N} & \varphi_{yy_{N-1}} & \dots & \varphi_{yy_0} \end{bmatrix} \begin{bmatrix} 1 \\ a_1 \\ \vdots \\ a_N \end{bmatrix} = \begin{bmatrix} V \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad \text{ou encore : } \boxed{\mathbf{R} \underline{a} = \underline{v}}$$

La matrice d'autocorrélation **R** de terme général $r_{i,j}$ ne dépendant que de $i - j$ (φ_{yy_k} paire) est dite matrice de Toeplitz. (Dans une telle matrice, les parallèles à la diagonale principale sont constituées de termes égaux). On montre que $\mathbf{R} \geq 0$.

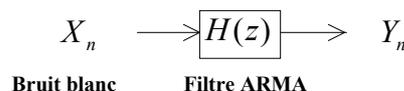
Exemple de matrice de Toeplitz :

$$\mathbf{R} = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 2 & 1 & 2 & 3 \\ 3 & 2 & 1 & 2 \\ 4 & 3 & 2 & 1 \end{bmatrix}$$

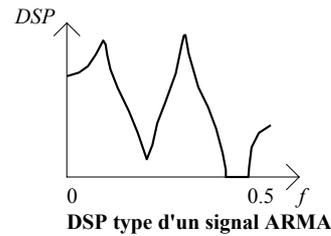
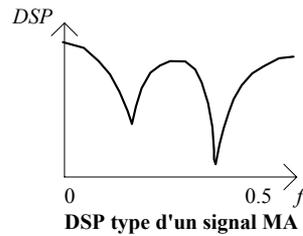
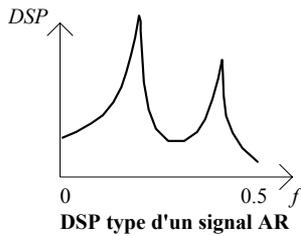
Ainsi, à partir de la connaissance de la séquence d'autocorrélation de $\{Y_n\}$, la relation matricielle permet de déterminer les $N + 1$ inconnues : V et les paramètres a_i du filtre formeur.

Pour réaliser ce calcul, une inversion matricielle est possible (cf annexe) mais on utilise généralement l'algorithme efficace itératif de **Levinson** ou encore la méthode de **Burg** (cf annexe) qui évitent une inversion matricielle en mettant à profit le fait que **R** a le caractère Toeplitz.

3.3. Signal ARMA



Les types classiques de DSP de signaux AR et MA étant les suivants, pour les satisfaire, il pourra être nécessaire de modéliser certains signaux par un processus ARMA :



Les hypothèses sont :

$$\begin{cases} E[X_n] = 0 \\ E[X_n^2] = V \\ E[X_n \cdot X_{n+k}] = 0 \quad \forall k \neq 0 \end{cases} \equiv \begin{cases} E[X_n] = 0 \\ \varphi_{xxk} = E[X_n \cdot X_{n+k}] = V\delta_{k,0} \end{cases}$$

($\{X_n\}$ bruit blanc centré de variance V)

et la FT d'un processus ARMA :

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{i=0}^N b_i z^{-i}}{1 + \sum_{i=1}^N a_i z^{-i}}$$

Cette notation suppose que si l'ordre du numérateur est inférieur à N , un certain nombre de coeffs b_i sont nuls. L'équation de récurrence s'obtient à partir de la FT du processus :

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{i=0}^N b_i z^{-i}}{1 + \sum_{i=1}^N a_i z^{-i}} \rightarrow Y(z) \left(1 + \sum_{i=1}^N a_i z^{-i} \right) = X(z) \left(\sum_{i=0}^N b_i z^{-i} \right) \xrightarrow{TZ^{-1}} Y_n = \sum_{i=0}^N b_i X_{n-i} - \sum_{i=1}^N a_i Y_{n-i}$$

. **Moyenne de $\{Y_n\}$:** ? $m = E[Y_n] = 0$ (car bruit blanc d'entrée centré) si les CIs sont nulles ou de moyenne nulle.

. **Autocorrélation de $\{Y_n\}$:** ? $\varphi_{yyk} = E[Y_{n-k} Y_n] = ?$

$$\varphi_{yyk} = E[Y_{n-k} Y_n] = E \left[Y_{n-k} \left(\sum_{i=0}^N b_i X_{n-i} - \sum_{i=1}^N a_i Y_{n-i} \right) \right] = E \left[\sum_{i=0}^N b_i X_{n-i} Y_{n-k} \right] - E \left[\sum_{i=1}^N a_i Y_{n-i} Y_{n-k} \right] = -\sum_{i=1}^N a_i E[Y_{n-i} Y_{n-k}] + \sum_{i=0}^N b_i E[X_{n-i} Y_{n-k}]$$

$$\rightarrow \boxed{\varphi_{yyk} = -\sum_{i=1}^N a_i \varphi_{yyk-i} + \sum_{i=0}^N b_i E[X_{n-i} Y_{n-k}]}$$

Le 2nd terme du membre de droite étant non linéaire en a_i, b_i , il n'est pas simple d'extraire les paramètres du modèle à partir de la matrice d'autocorrélation.

On peut cependant le faire en 2 étapes en s'appuyant sur les résultats précédents (MA et AR).

On peut d'abord remarquer que pour $k > N$ l'équation : $\varphi_{yyk} = -\sum_{i=1}^N a_i \varphi_{yyk-i} + \sum_{i=0}^N b_i E[X_{n-i} Y_{n-k}]$ se réduit à :

$$\varphi_{yyk} = -\sum_{i=1}^N a_i \varphi_{yyk-i} \quad \text{si } k > N \quad \text{qui a la même forme que } \varphi_{yyk} \text{ du cas AR : } \varphi_{yyk} = -\sum_{i=1}^N a_i \varphi_{yyk-i}$$

Si donc on écrit : $\varphi_{yyk} = -\sum_{i=1}^N a_i \varphi_{yyk-i} + \sum_{i=0}^N b_i E[X_{n-i} Y_{n-k}]$ pour $k = N, N+1, \dots, 2N$ on pourra extraire les coeffs a_i

(et le terme $b_0 b_N V$), soit :

$$\begin{bmatrix} \varphi_{yyN} & \varphi_{yyN-1} & \cdots & \varphi_{yy0} \\ \varphi_{yyN+1} & \varphi_{yyN} & \cdots & \varphi_{yy1} \\ \vdots & \vdots & \vdots & \vdots \\ \varphi_{yy2N} & \varphi_{yy2N-1} & \cdots & \varphi_{yyN} \end{bmatrix} \begin{bmatrix} 1 \\ a_1 \\ \vdots \\ a_N \end{bmatrix} = \begin{bmatrix} b_0 b_N V \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Pour la 2^{de} étape (identification des coeffs b_i), introduisons un signal intermédiaire $\{U_n\}$ pour réécrire l'équation de

réurrence du filtre : $Y_n = \sum_{i=0}^N b_i X_{n-i} - \sum_{i=1}^N a_i Y_{n-i}$:

Soit : $U_n = Y_n + \sum_{i=1}^N a_i Y_{n-i}$

comme : $Y_n = \sum_{i=0}^N b_i X_{n-i} - \sum_{i=1}^N a_i Y_{n-i}$, on a aussi : $U_n = X_n + \sum_{i=1}^N b_i X_{n-i}$ (en supposant $b_0 = 1$)

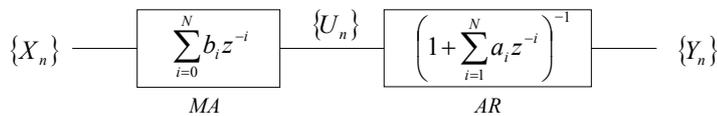
Cette dernière relation fait apparaître $\{U_n\}$ comme un signal MA, donc d'autocorrélation (établie dans le cas MA) :

$$\varphi_{uu_k} = V \sum_{i=0}^{N-k} b_i b_{k+i} \quad \text{si } k \leq N \quad \text{et :} \quad \varphi_{uu_k} = 0 \quad \text{si } k > N$$

A partir du signal $\{Y_n\}$ filtré par le filtre : $U_n = Y_n + \sum_{i=1}^N a_i Y_{n-i}$, on peut obtenir φ_{uu_k} et, par Programmation Non

Linéaire sur la relation : $\varphi_{uu_k} = V \sum_{i=0}^{N-k} b_i b_{k+i}$, on peut obtenir les coeffs b_i .

Ayant maintenant le processus générateur ARMA de $\{Y_n\}$:



la DSP de $Y(\omega)$ s'écrit :
$$\Phi_{YY}(\omega) = \left| \frac{\sum_{m=0}^N b_m e^{-j\omega m}}{1 + \sum_{m=1}^N a_m e^{-j\omega m}} \right|^2$$

ou, en tenant compte du cas déjà évoqué où Y est réel, pour lequel :
$$\Phi_{YY}(\omega) = \varphi_{yy0} + 2 \sum_{k=1}^{\infty} \varphi_{yyk} \cos(k\omega)$$

on a encore :
$$\Phi_{YY}(\omega) = \frac{\varphi_{uu0} + 2 \sum_{k=0}^N \varphi_{uu_k} \cos(k\omega)}{\left| 1 + \sum_{m=1}^N a_m e^{-j\omega m} \right|^2}$$

4. Conclusion

La représentation paramétrique d'un signal sous forme de processus générateur constitue une alternative aux méthodes classiques de calcul de spectre d'un signal. Une application importante des processus générateurs est l'identification (modélisation) de signal, et aussi la compression de signal.

ESTIMATION DE LA COVARIANCE

L'objectif est de déterminer la fonction d'autocovariance φ_{XX_k} , notée aussi souvent R_{XX_k} , d'un signal aléatoire (ou pas) x_k à partir de la séquence $\{x_k\}$ des échantillons du signal.

Ergodicité à court terme *Méthode d'estimation de la moyenne et de la covariance la plus courante*

Soit X_n un processus aléatoire à TD.

L'hypothèse d'ergodicité (*moyennes statistiques \equiv moyennes temporelles*) conduit aux formules suivantes d'estimation de la moyenne m et de la fonction d'autocovariance R_{XX_k} :

$$\left\{ \begin{array}{l} \hat{m} = \frac{1}{K} \sum_{n=0}^{K-1} X_n \\ \hat{R}_{XX_k} = \frac{1}{K} \sum_{n=0}^{K-k-1} X_{n+k}^0 X_n^{0*} \end{array} \right. \quad \text{soit, dans le cas d'un processus } X_n \text{ réel :}$$

$$\left\{ \begin{array}{l} \hat{m} = \langle X \rangle = \frac{1}{K} \sum_{n=0}^{K-1} X_n \\ \hat{R}_{XX_k} = \frac{1}{K} \sum_{n=0}^{K-k-1} X_{n+k}^0 X_n^0 \end{array} \right.$$

K est le nombre de points d'observation de X_n et doit être le plus élevé possible

* représente la conjugaison complexe

X_n^0 est le processus centré (cas réel) : $X_n^0 = X_n - \hat{m}$

Ergodicité à long terme

Lorsque le nombre de points du signal est élevé, l'estimation par ergodicité de l'autocovariance peut tenir compte de la moyenne pour effectuer un recentrage du signal, et corriger la normalisation du support de calcul :

$$\left\{ \begin{array}{l} \hat{m} = \frac{1}{K} \sum_{n=0}^{K-1} X_n \\ \hat{R}_{XX_k} = \frac{1}{K-k} \sum_{n=0}^{K-k-1} X_{n+k}^0 X_n^{0*} \end{array} \right. \quad \text{soit, dans le cas d'un processus } X_n \text{ réel :}$$

$$\left\{ \begin{array}{l} \hat{m} = \langle X \rangle = \frac{1}{K} \sum_{n=0}^{K-1} X_n \\ \hat{R}_{XX_k} = \frac{1}{K-k} \sum_{n=0}^{K-k-1} X_{n+k}^0 X_n^0 \end{array} \right.$$

K est le nombre de points d'observation de X_n et doit être le plus élevé possible

* représente la conjugaison complexe

X_n^0 est le processus centré (cas réel) : $X_n^0 = X_n - \hat{m}$

Calcul direct *Cas X_n réel*

Dans l'établissement des différents résultats, on a supposé les covariances connues. En pratique, ces dernières sont estimées à partir des échantillons dont on dispose. De plus, on ne dispose souvent que d'une seule réalisation x_n du processus X_n . Ainsi, pour obtenir les $N + 1$ premières valeurs $\{R_{XX_0}, \dots, R_{XX_N}\}$ de l'autocovariance, on peut utiliser les méthodes suivantes :

Méthode des corrélations *on doit prendre $N \ll K$, mais typiquement on peut aller jusqu'à $N = K - 1$ ($N < K$)*

$$\begin{bmatrix} R_{XX_0} \\ \vdots \\ R_{XX_N} \end{bmatrix} = \frac{1}{K} \mathbf{D} \begin{bmatrix} x_0 \\ \vdots \\ x_{K-1} \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad \text{avec :} \quad \mathbf{D} = \begin{bmatrix} x_0 & x_1 & \dots & x_{K-1} & 0 & \dots & 0 \\ 0 & x_0 & x_1 & \dots & x_{K-1} & \dots & 0 \\ \vdots & \vdots & \vdots & \dots & \vdots & \dots & 0 \\ 0 & \dots & 0 & x_0 & x_1 & \dots & x_{K-1} \end{bmatrix}$$

$(N + 1) \times 1$ $(K + N) \times 1$ $(N + 1) \times (K + N)$

Cette façon de procéder garantit que la **matrice de covariance** estimée, définie par $\frac{1}{K} \mathbf{D}\mathbf{D}^T$, est une matrice de Toeplitz définie positive.

(le caractère Toeplitz d'une matrice est intéressant car il autorise l'implémentation rapide, temps réel, de l'inversion matricielle)

(Pour tout vecteur \mathbf{a} , on peut écrire que $\mathbf{a}\mathbf{D}\mathbf{D}^T\mathbf{a} = (\mathbf{D}\mathbf{a})(\mathbf{D}\mathbf{a})^T$. On en conclue, en notant que le scalaire $(\mathbf{D}\mathbf{a})(\mathbf{D}\mathbf{a})^T$ est la somme des modules au carré des composantes de $\mathbf{D}\mathbf{a}$, et est donc positif.)

Cette méthode revient à border la séquence observée, à droite et à gauche, par N zéros. Son inconvénient majeur est donc d'introduire des données fausses, en l'occurrence des 0, de part et d'autre des données observées.

Quand la taille K de l'observation est faible, on lui préfère la *méthode des covariances*.

Méthode des covariances on doit prendre $N \ll K$, mais typiquement on peut prendre $N = K / 10$ ($N < K$)

$$\begin{bmatrix} R_{XX_0} \\ \vdots \\ R_{XX_N} \end{bmatrix} = \frac{1}{K} \mathbf{D} \begin{bmatrix} x_0 \\ \vdots \\ x_{K-N-1} \end{bmatrix} \quad \text{avec :} \quad \mathbf{D} = \begin{bmatrix} x_0 & x_1 & \cdots & x_{K-N-1} \\ x_1 & x_2 & \cdots & x_{K-N} \\ \vdots & \vdots & \vdots & \vdots \\ x_N & x_{N+1} & \cdots & x_{K-1} \end{bmatrix}$$

$(N+1) \times 1$ $(K-N) \times 1$ $(N+1) \times (K-N)$

La **matrice de covariance** estimée, définie par $\frac{1}{K-N-1} \mathbf{D}\mathbf{D}^T$, reste bien entendu positive mais elle perd alors son caractère Toeplitz nécessaire à certains algorithmes rapides d'inversion matricielle.

Il existe 2 autres variantes de construction de \mathbf{D} , suivant que l'on borde à droite ou à gauche par des zéros.

Méthode des covariances (variante bordage à droite) $N \ll K$ Typiquement $N = K / 10$ ($N < K$)

Par exemple :

$$\mathbf{D} = \begin{bmatrix} x_0 & x_1 & \cdots & \cdots & \cdots & x_{K-1} \\ x_1 & x_2 & \cdots & \cdots & x_{K-1} & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_N & \cdots & x_{K-1} & 0 & \cdots & 0 \end{bmatrix}$$

$$\begin{bmatrix} R_{XX_0} \\ \vdots \\ R_{XX_N} \end{bmatrix} = \frac{1}{K} \mathbf{D} \begin{bmatrix} x_0 \\ \vdots \\ x_{K-1} \end{bmatrix} \quad \text{avec :} \quad \mathbf{D} = \begin{bmatrix} x_0 & x_1 & \cdots & \cdots & \cdots & x_{K-1} \\ x_1 & x_2 & \cdots & \cdots & x_{K-1} & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_N & \cdots & x_{K-1} & 0 & \cdots & 0 \end{bmatrix}$$

$(N+1) \times 1$ $(K) \times 1$ $(N+1) \times (K)$

Il est évident que toutes ces méthodes donnent sensiblement les mêmes résultats si $K \gg N$.

RESOLUTION DU SYSTEME D'EQUATIONS DE YULE-WALKER PAR INVERSION MATRICIELLE

$$\begin{bmatrix} \varphi_{yy_0} & \varphi_{yy_1} & \cdots & \varphi_{yy_N} \\ \varphi_{yy_1} & \varphi_{yy_0} & \cdots & \varphi_{yy_{N-1}} \\ \vdots & \vdots & \ddots & \vdots \\ \varphi_{yy_N} & \varphi_{yy_{N-1}} & \cdots & \varphi_{yy_0} \end{bmatrix} \begin{bmatrix} 1 \\ a_1 \\ \vdots \\ a_N \end{bmatrix} = \begin{bmatrix} V \\ 0 \\ \vdots \\ 0 \end{bmatrix} \equiv \mathbf{R} \underline{a} = \underline{v}$$

avec :

$$\mathbf{R} = \begin{bmatrix} \varphi_{yy_0} & \varphi_{yy_1} & \cdots & \varphi_{yy_N} \\ \varphi_{yy_1} & \varphi_{yy_0} & \cdots & \varphi_{yy_{N-1}} \\ \vdots & \vdots & \ddots & \vdots \\ \varphi_{yy_N} & \varphi_{yy_{N-1}} & \cdots & \varphi_{yy_0} \end{bmatrix} : (N+1) \times (N+1) \quad \underline{a} = \begin{bmatrix} 1 \\ a_1 \\ \vdots \\ a_N \end{bmatrix} : (N+1) \times (1) \quad \underline{v} = \begin{bmatrix} V \\ 0 \\ \vdots \\ 0 \end{bmatrix} : (N+1) \times (1)$$

$$\rightarrow \begin{cases} \varphi_{yy_0} + \varphi_{yy_1} a_1 + \cdots + \varphi_{yy_N} a_N = V \\ \varphi_{yy_1} + \varphi_{yy_0} a_1 + \cdots + \varphi_{yy_{N-1}} a_N = 0 \\ \dots \\ \varphi_{yy_N} + \varphi_{yy_{N-1}} a_1 + \cdots + \varphi_{yy_0} a_N = 0 \end{cases} \rightarrow \begin{cases} \varphi_{yy_0} \left(1 - \frac{V}{\varphi_{yy_0}} \right) + \varphi_{yy_1} a_1 + \cdots + \varphi_{yy_N} a_N = 0 \\ \varphi_{yy_1} \left(1 - \frac{V}{\varphi_{yy_0}} \right) + \varphi_{yy_0} a_1 + \cdots + \varphi_{yy_{N-1}} a_N = -V \frac{\varphi_{yy_1}}{\varphi_{yy_0}} \\ \dots \\ \varphi_{yy_N} \left(1 - \frac{V}{\varphi_{yy_0}} \right) + \varphi_{yy_{N-1}} a_1 + \cdots + \varphi_{yy_0} a_N = -V \frac{\varphi_{yy_N}}{\varphi_{yy_0}} \end{cases}$$

$$\rightarrow \mathbf{R} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_N \end{bmatrix} = -\frac{V}{\varphi_{yy_0}} \begin{bmatrix} 0 \\ \varphi_{yy_1} \\ \vdots \\ \varphi_{yy_N} \end{bmatrix} \quad \text{avec : } a_0 = 1 - \frac{V}{\varphi_{yy_0}}$$

$$\rightarrow \frac{1}{V} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_N \end{bmatrix} = -\frac{1}{\varphi_{yy_0}} \mathbf{R}^{-1} \begin{bmatrix} 0 \\ \varphi_{yy_1} \\ \vdots \\ \varphi_{yy_N} \end{bmatrix} \rightarrow \boxed{\underline{\alpha} = -\frac{1}{\varphi_{yy_0}} \mathbf{R}^{-1} \underline{b}} \quad \text{avec : } \underline{\alpha} \stackrel{\Delta}{=} \frac{1}{V} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_N \end{bmatrix} \stackrel{\Delta}{=} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_N \end{bmatrix} \quad \text{et } \underline{b} \stackrel{\Delta}{=} \begin{bmatrix} 0 \\ \varphi_{yy_1} \\ \vdots \\ \varphi_{yy_N} \end{bmatrix}$$

Comme : $\alpha_0 = \frac{a_0}{V} = \frac{1}{V} \left(1 - \frac{V}{\varphi_{yy_0}} \right)$, on en déduit :

$$\boxed{V = \frac{\varphi_{yy_0}}{\varphi_{yy_0} \alpha_0 + 1}}$$

Puis on obtient les paramètres a_i ($i = 1, 2, \dots, N$) :

$$\boxed{\begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_N \end{bmatrix} = V \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_N \end{bmatrix}}$$

ALGORITHME DE LEVINSON (I) *Résolution itérative du système de Yule-Walker*

La méthode de résolution des équations de Yule-Walker par inversion matricielle est coûteuse en temps calcul et peut engendrer des erreurs numériques dues à l'inversion matricielle. L'algorithme de Levinson procure la même solution que la méthode directe par inversion matricielle mais avec un algorithme itératif temps réel.

$$\begin{bmatrix} \varphi_{yy_0} & \varphi_{yy_1} & \cdots & \varphi_{yy_N} \\ \varphi_{yy_1} & \varphi_{yy_0} & \cdots & \varphi_{yy_{N-1}} \\ \vdots & \vdots & \ddots & \vdots \\ \varphi_{yy_N} & \varphi_{yy_{N-1}} & \cdots & \varphi_{yy_0} \end{bmatrix} \begin{bmatrix} 1 \\ a_1 \\ \vdots \\ a_N \end{bmatrix} = \begin{bmatrix} V \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad \text{qui peut aussi s'écrire :} \quad \mathbf{R} \underline{a} = \underline{v}$$

ou encore :

$$\text{pour } 0 \leq k \leq N : \quad \sum_{i=0}^N \varphi_{yy_{k-i}} a_i = \begin{cases} V & \text{si } k = 0 \\ 0 & \text{si } 1 \leq k \leq N \end{cases} \quad (a_0 = 1) \quad (\varphi_{yy_k} \text{ paire})$$

Le principe de l'algorithme de **Levinson**, qui permet de résoudre efficacement les équations précédentes, repose sur une itération sur l'ordre n du modèle : $n = 0, 1 \dots N$

On écrit donc les équations précédentes en plaçant un indice $[n]$ sur a_i et V , signifiant que l'on utilise le modèle à n paramètres et l'on va itérer sur $n = 0, 1 \dots N$:

$$\text{pour } 0 \leq k \leq n : \quad \sum_{i=0}^n \varphi_{yy_{k-i}} a_i^{[n]} = \begin{cases} V_{[n]} & \text{si } k = 0 \\ 0 & \text{si } 1 \leq k \leq n \end{cases} \quad (a_0^{[n]} = 1) \quad (\varphi_{yy_k} \text{ paire})$$

Les solutions à l'ordre n : $a_0^{[n]}, a_1^{[n]}, \dots, a_n^{[n]}, V_{[n]}$ sont utilisées pour résoudre l'équation du modèle d'ordre $n + 1$:

$$\text{pour } 0 \leq k \leq n + 1 : \quad \sum_{i=0}^{n+1} \varphi_{yy_{k-i}} a_i^{[n+1]} = \begin{cases} V_{[n+1]} & \text{si } k = 0 \\ 0 & \text{si } 1 \leq k \leq n + 1 \end{cases} \quad (a_0^{[n+1]} = 1) \quad (\varphi_{yy_k} \text{ paire})$$

La procédure de Levinson est la suivante : on itère de $n = 0, 1 \dots N - 1$:

1. $n = 0, \quad a_0^{[n]} = 1, \quad V_{[n]} = \varphi_{yy_0} \quad (\text{Initialisation})$
2. $\rho_{[n+1]} = -V_{[n]}^{-1} \sum_{k=0}^{[n]} \varphi_{yy_{n+1-k}} a_k^{[n]}$
3. $V_{[n+1]} = (1 - \rho_{[n+1]}^2) V_{[n]}$
4. pour $0 \leq k \leq n + 1$: $a_k^{[n+1]} = \begin{cases} 1 & \text{si } k = 0 \\ a_k^{[n]} + \rho_{[n+1]} a_{n+1-k}^{[n]} & \text{si } 1 \leq k \leq n \\ \rho_{[n+1]} & \text{si } k = n + 1 \end{cases}$
5. $n = n + 1$
6. Revenir à 2. tant que $n < N$

Le vecteur résultat $\begin{bmatrix} 1 \\ a_1 \\ \vdots \\ a_N \end{bmatrix}$ et le scalaire V cherchés sont obtenus *en fin d'itérations* ($n = N$) :

$$\begin{bmatrix} 1 \\ a_1 \\ \vdots \\ a_N \end{bmatrix} = \begin{bmatrix} 1 \\ a_1^{[N]} \\ \vdots \\ a_N^{[N]} \end{bmatrix} \quad (\text{soit : } a_i = a_i^{[N]}, n = 1, 2 \dots N) \quad \text{et :} \quad V = V_{[N]}.$$

L'algorithme de Levinson peut être écrit ainsi : (itération de $n = 0, 1 \dots N - 1$) :

$$n = 0, \quad a_0 = 1, \quad V = \varphi_{yy_0} \quad (\text{Initialisation})$$

Pour $0 \leq n \leq N - 1$:

{

$$\rho = -V^{-1} \sum_{k=0}^n \varphi_{yy_{n+1-k}} a_k$$

$$V = (1 - \rho^2)V$$

$$\text{Pour } 0 \leq k \leq n + 1 : \quad a_k = \begin{cases} 1 & \text{si } k = 0 \\ a_k + \rho a_{n+1-k} & \text{si } 1 \leq k \leq n \\ \rho & \text{si } k = n + 1 \end{cases}$$

}

Le vecteur résultat $\begin{bmatrix} 1 \\ a_1 \\ \vdots \\ a_N \end{bmatrix}$ et le scalaire V cherchés sont obtenus *en fin d'itération*.

ALGORITHME DE LEVINSON (II)

Résolution itérative du système de Yule-Walker

$$\begin{bmatrix} R_0 & R_1 & \cdots & R_N \\ R_1 & R_0 & \cdots & R_{N-1} \\ \vdots & \vdots & \ddots & \vdots \\ R_N & R_{N-1} & \cdots & R_0 \end{bmatrix} \begin{bmatrix} 1 \\ a_1 \\ \vdots \\ a_N \end{bmatrix} = \begin{bmatrix} P \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

qui peut aussi s'écrire : $\mathbf{R} \mathbf{a} = \mathbf{v}$

La matrice de covariance \mathbf{R} est une matrice de Toeplitz. Son terme général R_i désigne l'autocovariance du signal x_n .

Le principe de l'algorithme de Levinson repose sur le calcul de la solution au rang n à partir de la solution au rang $n - 1$. Pour cela, on adopte la notation suivante où l'ordre n , noté en indice $[n]$, doit être itéré de 0 à N :

($P_{[n]}$ est la puissance du bruit qui peut être vu comme l'erreur quadratique d'estimation au rang $n - 1$)

$$\begin{bmatrix} R_0 & R_1 & \cdots & R_n \\ R_1 & R_0 & \cdots & R_{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ R_n & R_{n-1} & \cdots & R_0 \end{bmatrix} \begin{bmatrix} a_0^{[n]} \\ a_1^{[n]} \\ \vdots \\ a_n^{[n]} \end{bmatrix} = \begin{bmatrix} P_{[n]} \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

avec $a_0^{[n]} = 1$

En posant : $\mathbf{r}_n^B = [R_n \ \cdots \ R_0]^T$ et $\mathbf{r}_n^F = [R_0 \ \cdots \ R_n]^T$ (B et F signifiant *Backward* et *Forward*)

on peut écrire le système de Yule-Walker de façon plus concise : $\mathbf{R}_n \mathbf{a}_n^F = \begin{bmatrix} P_{[n]} \\ \mathbf{0} \end{bmatrix}$

soit, au rang $n - 1$: $\mathbf{R}_{n-1} \mathbf{a}_{n-1}^F = \begin{bmatrix} P_{[n-1]} \\ \mathbf{0} \end{bmatrix}$

En passant au rang n , la matrice de covariance s'écrit : $\mathbf{R}_n = \begin{bmatrix} \mathbf{R}_{n-1} & \mathbf{r}_n^B \\ \mathbf{r}_n^{B^T} & R_0 \end{bmatrix} = \begin{bmatrix} R_0 & \mathbf{r}_n^{F^T} \\ \mathbf{r}_n^F & \mathbf{R}_{n-1} \end{bmatrix}$

On a : $\mathbf{R}_n \begin{bmatrix} \mathbf{a}_{n-1}^F \\ 0 \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{n-1} & \mathbf{r}_n^B \\ \mathbf{r}_n^{B^T} & R_0 \end{bmatrix} \begin{bmatrix} \mathbf{a}_{n-1}^F \\ 0 \end{bmatrix} = \begin{bmatrix} P_{[n-1]} \\ \mathbf{0} \\ \mathbf{r}_n^{B^T} \mathbf{a}_{n-1}^F \end{bmatrix}$ avec : $\mathbf{a}_{n-1}^F = [1 \ a_1^{[n-1]} \ \cdots \ a_{n-1}^{[n-1]}]^T$

On a aussi : $\mathbf{R}_n \begin{bmatrix} 0 \\ \mathbf{a}_{n-1}^B \end{bmatrix} = \begin{bmatrix} R_0 & \mathbf{r}_n^{F^T} \\ \mathbf{r}_n^F & \mathbf{R}_{n-1} \end{bmatrix} \begin{bmatrix} 0 \\ \mathbf{a}_{n-1}^B \end{bmatrix} = \begin{bmatrix} \mathbf{r}_n^{F^T} \mathbf{a}_{n-1}^B \\ \mathbf{0} \\ P_{[n-1]} \end{bmatrix}$ avec : $\mathbf{a}_{n-1}^B = [a_{n-1}^{[n-1]} \ \cdots \ a_1^{[n-1]} \ 1]^T$

Par combinaison linéaire : $\mathbf{R}_n \begin{bmatrix} \begin{bmatrix} \mathbf{a}_{n-1}^F \\ 0 \end{bmatrix} + K_n \mathbf{a}_{n-1}^F \begin{bmatrix} 0 \\ \mathbf{a}_{n-1}^B \end{bmatrix} \end{bmatrix} = \begin{bmatrix} P_{[n-1]} + K_n \mathbf{r}_n^{F^T} \mathbf{a}_{n-1}^B \\ \mathbf{0} \\ \mathbf{r}_n^{B^T} \mathbf{a}_{n-1}^F + K_n P_{[n-1]} \end{bmatrix}$

En choisissant le coefficient K_n tel que : $\mathbf{r}_n^{BT} \mathbf{a}_{n-1}^F + K_n P_{[n-1]} = 0$, on obtient la solution au rang n qui s'écrit en fonction de la solution au rang $n-1$:

$$\begin{cases} K_n = -\frac{\mathbf{r}_n^{BT} \mathbf{a}_{n-1}^F}{P_{[n-1]}} \\ \mathbf{a}_n^F = \begin{bmatrix} \mathbf{a}_{n-1}^F \\ 0 \end{bmatrix} + K_n \mathbf{a}_{n-1}^F \begin{bmatrix} 0 \\ \mathbf{a}_{n-1}^B \end{bmatrix} \\ P_{[n]} = P_{[n-1]}(1 - K_n^2) \end{cases}$$

$P_{[n]}$ étant positif (il représente une erreur quadratique), la dernière équation montre que les coefficients K_n sont de module inférieur à 1. Les coefficients K_n portent le nom de *coefficients de réflexion*.

Algorithme de Levinson

Valeurs initiales : $P_0 = R_0 = \frac{1}{K} \sum_{n=0}^K |x_n|^2$ $a_0^{[0]} = 1$

Pour $n = 1, \dots, N-1$:

$$\left\{ \begin{array}{l} K_n = -\frac{[R_n a_0^{[n-1]} + \dots + R_0 a_{n-1}^{[n-1]}]}{P_{n-1}} \end{array} \right.$$

Pour $k = 1, \dots, n-1$:

$$\left\{ \begin{array}{l} a_0^{[n]} = 1 \\ a_k^{[n]} = a_k^{[n-1]} + K_n a_{n-k}^{[n-1]} \\ a_n^{[n]} = K_n \end{array} \right.$$

$$P_{[n]} = P_{[n-1]}(1 - K_n^2)$$

}

Vérification à l'étape $n = 1$

$$\begin{cases} K_1 = -\frac{R_1}{R_0} \\ a_0^{[1]} = 1 \\ a_1^{[1]} = K_1 \\ P_{[1]} = P_{[0]}(1 - K_1^2) \end{cases}$$

METHODE DE BURG

Résolution itérative du système de Yule-Walker avec contrainte de stabilité

La méthode de résolution des équations de Yule-Walker par inversion matricielle ou par l'algorithme de Levinson peut conduire à une solution d'un filtre non stable. La méthode itérative de Burg pour la résolution des équations de Yule-Walker garantit un filtre stable comme solution, même si cette solution n'est pas exactement identique à la solution directe ou de Levinson des équations de Yule-Walker (solution approchée - du fait de la contrainte de stabilité ajoutée au système de Yule-Walker - mais toujours stable).

$$\begin{bmatrix} R_0 & R_1 & \cdots & R_N \\ R_1 & R_0 & \cdots & R_{N-1} \\ \vdots & \vdots & \ddots & \vdots \\ R_N & R_{N-1} & \cdots & R_0 \end{bmatrix} \begin{bmatrix} 1 \\ a_1 \\ \vdots \\ a_N \end{bmatrix} = \begin{bmatrix} P \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad \text{qui peut aussi s'écrire :} \quad \mathbf{R}\mathbf{a} = \mathbf{v}$$

La matrice de covariance \mathbf{R} est une matrice de Toeplitz. Son terme général R_i désigne l'autocovariance du signal x_n .

L'idée de Burg fut de calculer directement à partir des données une estimée des coefficients de réflexion K_n et ce, sans

passer par le calcul des covariances. On en déduit, si nécessaire, par la relation de Levinson :

$$\begin{cases} K_n = -\frac{\mathbf{r}_n^{BT} \mathbf{a}_{n-1}^F}{P_{[n-1]}} \\ \mathbf{a}_n^F = \begin{bmatrix} \mathbf{a}_{n-1}^F \\ 0 \end{bmatrix} + K_n \mathbf{a}_{n-1}^B \begin{bmatrix} 0 \\ \mathbf{a}_{n-1}^B \end{bmatrix} \\ P_{[n]} = P_{[n-1]}(1 - K_n^2) \end{cases}$$

les coefficients du modèle. Pour cela, on repart de l'expression de l'erreur de prédiction au rang N : (t est discret)

$$\begin{aligned} \varepsilon_n^F(t) &= x(t) + a_1^{[n]}x(t-1) + \cdots + a_n^{[n]}x(t-n) = \mathbf{a}_n^{FT} \mathbf{x} \quad (\text{erreur directe}) \quad F \equiv \text{Forward} \\ \varepsilon_n^B(t) &= a_n^{[n]}x(t) + a_{n-1}^{[n]}x(t-1) + \cdots + x(t-n) = \mathbf{a}_n^{BT} \mathbf{x} \quad (\text{erreur rétrograde}) \quad B \equiv \text{Backward} \end{aligned}$$

En utilisant les coefficients de réflexion, on a :

$$\varepsilon_n^F(t) = \begin{bmatrix} \mathbf{a}_{n-1}^F \\ 0 \end{bmatrix} + K_n \begin{bmatrix} 0 \\ \mathbf{a}_{n-1}^B \end{bmatrix} \mathbf{x}(t) = \varepsilon_{n-1}^F(t) + K_n \varepsilon_{n-1}^B(t-1)$$

et :

$$\varepsilon_n^B(t) = \begin{bmatrix} 0 \\ \mathbf{a}_{n-1}^B \end{bmatrix} + K_n \begin{bmatrix} \mathbf{a}_{n-1}^F \\ 0 \end{bmatrix} \mathbf{x}(t) = \varepsilon_{n-1}^B(t-1) + K_n \varepsilon_{n-1}^F(t)$$

Pour trouver un estimateur de K_n , Burg propose de minimiser par rapport à K_n , la somme des erreurs :

$$E \left[\left| \varepsilon_n^F(t) \right|^2 + \left| \varepsilon_n^B(t) \right|^2 \right]. \text{ En partant des 2 équations précédentes, on aboutit à une équation du 2nd degré en } K_n \text{ qui,}$$

par annulation de sa dérivée, conduit à :

$$K_n = -\frac{2E \left[\varepsilon_{n-1}^B(t-1) \varepsilon_{n-1}^F(t) \right]}{E \left[\left| \varepsilon_{n-1}^F(t) \right|^2 + \left| \varepsilon_{n-1}^B(t-1) \right|^2 \right]}$$

Dans l'algorithme de calcul des coefficients K_n , les espérances (moyennes statistiques) sont remplacées par des moyennes temporelles (*ergodicité*). Pour un processus AR d'ordre $N - 1$, on a :

Méthode de Burg

Valeurs initiales : $P_0 = R_0 = \frac{1}{K} \sum_{n=0}^K |x_n|^2$ $\varepsilon_1^F(t) = \varepsilon_1^B(t) = x(t) \quad t = 0, 1, \dots, K-1$

Pour $n = 1, \dots, N-1$:

{

$$K_n = - \frac{2 \sum_{t=n}^{K-1} \varepsilon_{n-1}^B(t-1) \varepsilon_{n-1}^F(t)}{\sum_{t=n}^{K-1} |\varepsilon_{n-1}^F(t)|^2 + |\varepsilon_{n-1}^B(t-1)|^2}$$

Pour $k = 2, \dots, n-1$:

{

$$\begin{cases} a_0^{[n]} = 1 \\ a_k^{[n]} = a_k^{[n-1]} + K_n a_{n-k}^{[n-1]} \\ a_n^{[n]} = K_n \end{cases}$$

}

$$P_{[n]} = P_{[n-1]} (1 - K_n^2)$$

$$\varepsilon_n^F(t) = \varepsilon_{n-1}^F(t) + K_n \varepsilon_{n-1}^B(t-1) \quad t = 0, 1, \dots, K-1$$

$$\varepsilon_n^B(t) = \varepsilon_{n-1}^B(t-1) + K_n \varepsilon_{n-1}^F(t) \quad t = 0, 1, \dots, K-1$$

}

Une propriété de l'algorithme de Burg est de garantir que les estimées de coefficients de réflexion sont de module inférieur à 1.

TP 1. Signaux déterministes. Signaux aléatoires. Systèmes stochastiques

II. SIGNAUX ALEATOIRES

ESTIMATION DE LA COVARIANCE

Faire varier les paramètres K et N pour voir leur influence pour chaque méthode.

K : Nombre d'échantillons du signal observé (ici un bruit blanc)

N : Fenêtre coulissante à l'intérieur de l'intervalle K ou bordant K à gauche ou à droite: $N < K$

ORIGIN := 0

K := 128 k := 0..K - 1 x := whiten(K) a := -0.5 b := 0.5

Note : la fonction MathCad "whiten()" produit un bruit blanc centré de distribution uniforme :

les réalisations sont réparties uniformément entre $a = -0.5$ et $b = +0.5$

0. MOYENNE ET COVARIANCE THEORIQUES

La théorie dit qu'une loi uniforme sur $[a,b]$ a pour moyenne et variance :

$$m := \int_a^b x \cdot \frac{1}{b-a} dx$$

$$m = \frac{a+b}{2}$$

$$\sigma^2 := \int_a^b (x-m)^2 \cdot \frac{1}{b-a} dx$$

$$\sigma^2 = \frac{(b-a)^2}{12}$$

L'amplitude (crête) c du signal peut être déduite de σ^2 :

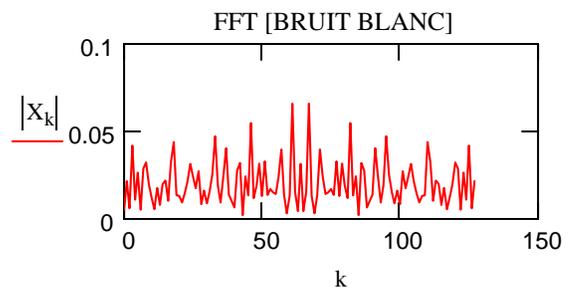
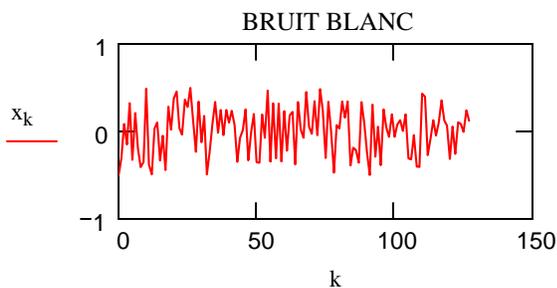
$$c := \frac{b-a}{2}$$

$$c = 0.5$$

$$c := \frac{\sqrt{12 \cdot \sigma^2}}{2}$$

$$c = \sigma \cdot \sqrt{3}$$

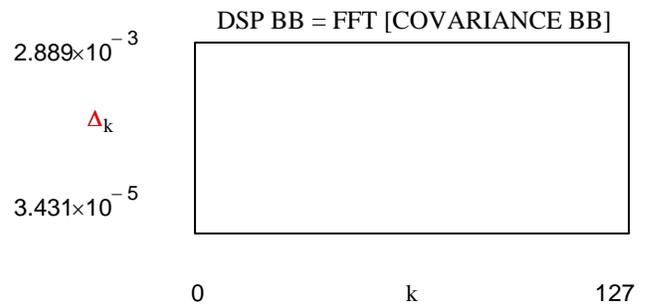
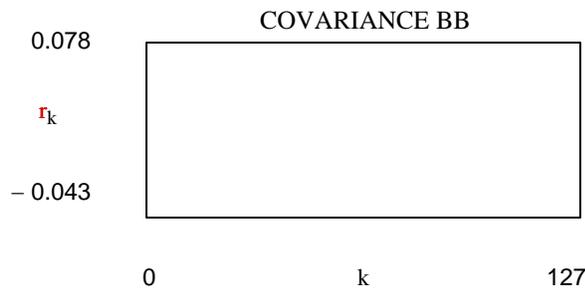
X := CFFT(x)



1. ESTIMATION DE LA MOYENNE ET DE LA COVARIANCE PAR ERGODICITE

. Moyenne : $m :=$ $m =$

. Covariance : $r_k :=$ $\Lambda := \text{CFFT}(r)$ $\Delta_k := |\Lambda_k|$



Conclusion : Variance du bruit blanc : $\sigma^2 :=$ $\sigma^2 =$

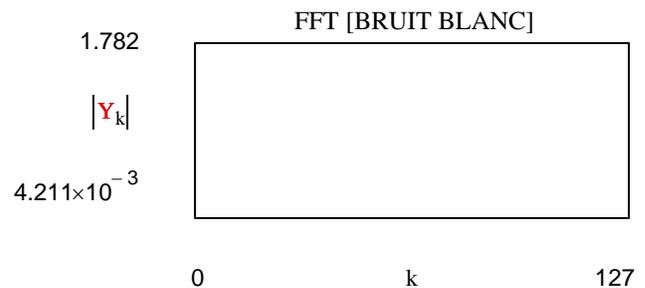
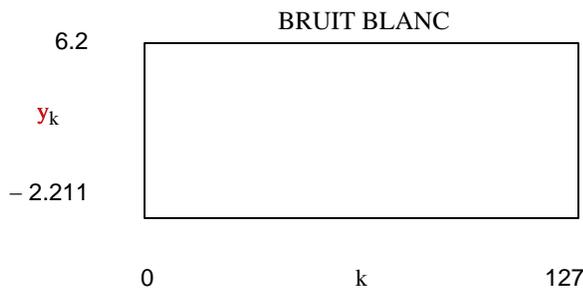
2. BRUIT BLANC DE MOYENNE m ET DE VARIANCE σ^2

A partir de la fonction MathCad "whiten()", fabriquer un bruit blanc (de loi de distribution uniforme) de moyenne m donnée et de variance σ^2 donnée. Vérifier à l'aide du calcul de la moyenne et de la covariance par ergodicité.

. Fonction BruitBlanc(K,m,sigma2) : $\text{BruitBlanc}(K, m, \text{sigma}2) :=$

RAZ : $r := \text{submatrix}(r, 0, 0, 0, 0)$

$k := 0..K - 1$ $m := 2$ $\text{sigma}2 := 6$ $y := \text{BruitBlanc}(K, m, \text{sigma}2)$ $Y := \text{CFFT}(y)$



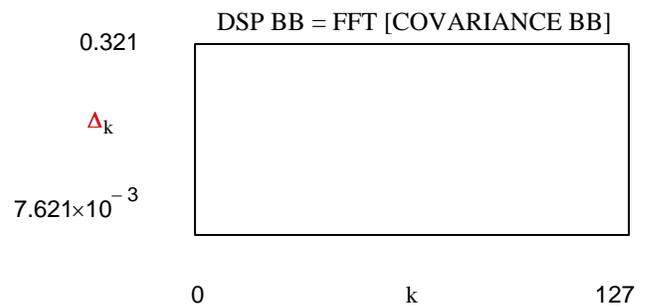
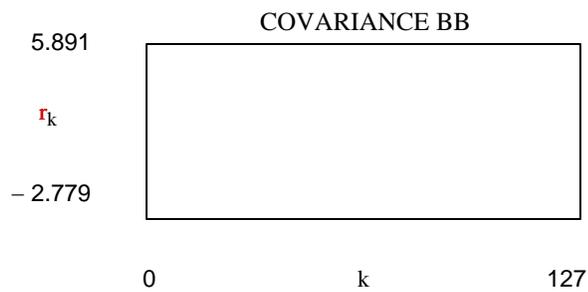
L'amplitude (crête) c du signal peut être déduite de σ^2 : $c := \frac{\sqrt{12 \cdot \text{sigma}2}}{2}$ $c = 4.243$

VERIFICATION. Estimation de la covariance par ergodicité :

. Moyenne : $m :=$ $m =$

. Covariance : $r_k :=$ $\Lambda := \text{CFFT}(r)$ $\Delta_k := |\Lambda_k|$

amplitude := max(y) - m amplitude =



Conclusion :

Variance du bruit blanc: $\sigma^2 :=$ $\sigma^2 =$

TP 1 ANNEXE. Signaux déterministes. Signaux aléatoires. Systèmes stochastiques

I. SIGNAUX DETERMINISTES

Un signal à TC sera programmé comme une fonction $s(t)$, t réel

Un signal à TD sera programmé comme une fonction $s(n)$, n entier, ou mieux comme un vecteur s_n

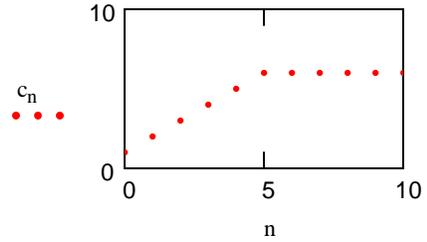
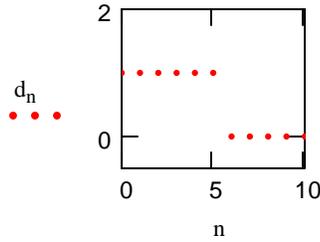
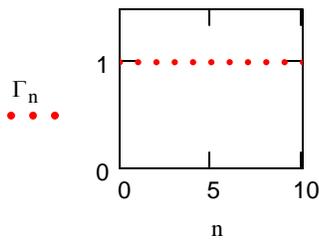
0. Convolution discrète c_n de signaux causaux

Convolution d'une transition Γ_n avec une porte Π_n

$$C := 10 \quad n := 0..C \quad \Gamma_n := \Phi(n) \quad \Pi_n := \begin{cases} 1 & \text{if } 0 \leq n \leq \frac{C}{2} \\ 0 & \text{if } n > \frac{C}{2} \end{cases}$$

$$s_n := \Gamma_n \quad d_n := \Pi_n$$

$$c_n := \sum_{k=0}^n s_k \cdot d_{n-k}$$



Convolution causale Fonction convcausal(x, h) de convolution causale de 2 signaux x et h

$$\text{convcausal}(x, h) \equiv \begin{cases} M \leftarrow \text{length}(x) \\ N \leftarrow \text{length}(h) \\ \text{for } n \in 0..M-1 \\ \quad \left| \begin{cases} y_n \leftarrow \sum_{k=0}^n x_k \cdot h_{n-k} & \text{if } n < N \\ y_n \leftarrow 0 & \text{otherwise} \end{cases} \end{cases} \right. y$$

II. SIGNAUX ALEATOIRES

ESTIMATION DE LA COVARIANCE

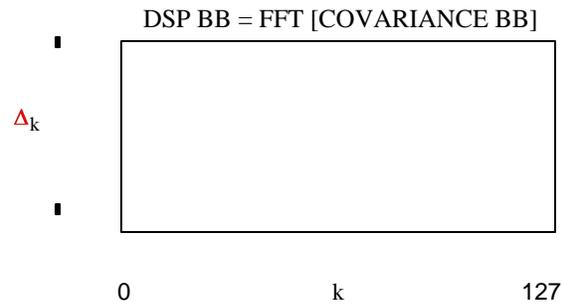
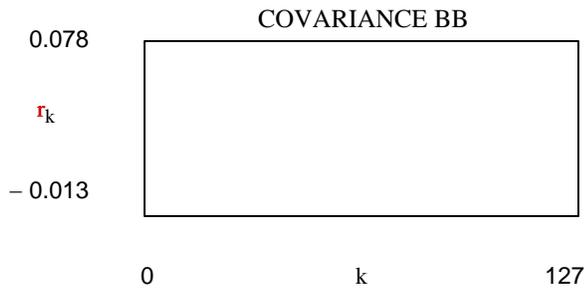
2. ESTIMATION DE LA COVARIANCE PAR LA METHODE DES CORRELATIONS

$N := K - 1$ $n := 0..N - 1$ $Nul_n := 0$ $X := \text{stack}(x, Nul)$ **RAZ** : $r := \text{submatrix}(r, 0, 0, 0, 0)$

$F(i, j) := \begin{cases} \text{if } j \geq i \\ \quad \begin{cases} x_{j-i} & \text{if } (j - i) < K \\ 0 & \text{otherwise} \end{cases} \\ 0 & \text{otherwise} \end{cases}$ $D := \text{matrix}(N + 1, K + N, F)$

. Covariance : **$r :=$** $\Lambda := \text{CFFT}(r)$ $\Delta_k := |\Lambda_k|$

. Matrice de Covariance : **$R :=$** La matrice R ainsi construite a le caractère Toeplitz



Conclusion :

Variance du bruit blanc: **$\sigma^2 :=$** **$\sigma^2 =$**

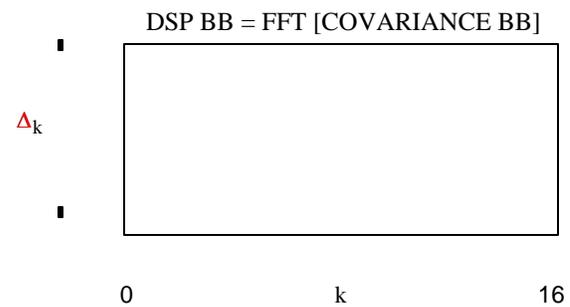
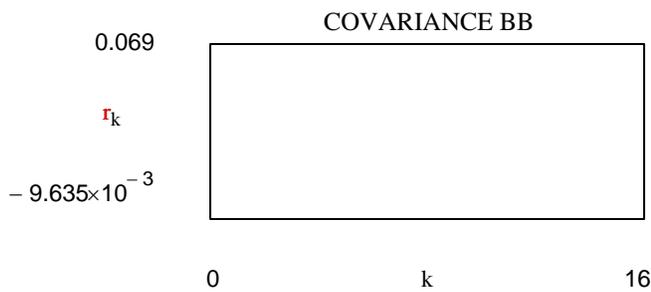
3. ESTIMATION DE LA COVARIANCE PAR LA METHODE DES COVARIANCES

$$N := \frac{K}{8} \quad k := 0..N-1 \quad X := \text{submatrix}(x, 0, K - N - 1, 0, 0) \quad \text{RAZ : } r := \text{submatrix}(r, 0, 0, 0, 0)$$

$$F(i, j) := \quad D :=$$

. Covariance : $r :=$ r $\Lambda := \text{CFFT}(r)$ $\Delta_k := |\Lambda_k|$

. Matrice de Covariance : $R :=$ R



Conclusion :

Variance du bruit blanc: $\sigma_2 :=$ σ_2 $\sigma_2 =$ σ_2

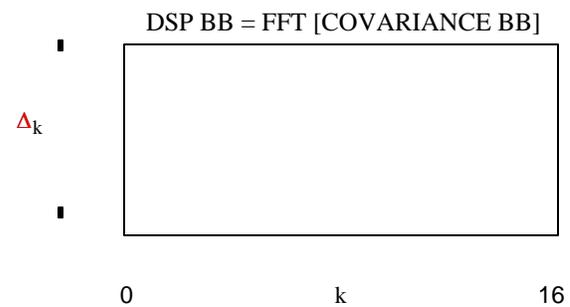
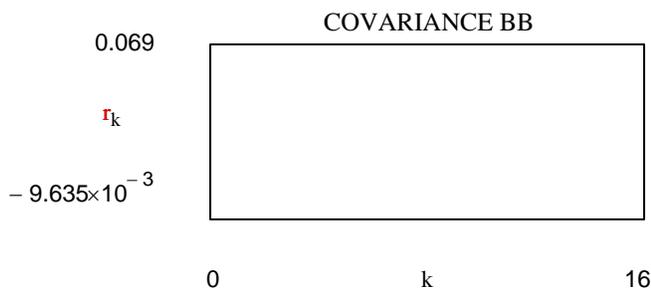
[3-VBD]. ESTIMATION DE LA COVARIANCE PAR LA METHODE DES COVARIANCES : Variante Bordage à droite

$$N := \frac{K}{8} \quad k := 0..N-1 \quad X := \text{submatrix}(x, 0, K - 1, 0, 0) \quad \text{RAZ : } r := \text{submatrix}(r, 0, 0, 0, 0)$$

$$F(i, j) := \quad D := \text{matrix}(N + 1, K, F)$$

. Covariance : $r :=$ r $\Lambda := \text{CFFT}(r)$ $\Delta_k := |\Lambda_k|$

. Matrice de Covariance : $R :=$ R



Conclusion :

Variance du bruit blanc: $\sigma_2 :=$ σ_2 $\sigma_2 =$ σ_2

III. SYSTEMES STOCHASTIQUES

PROCESSUS GENERATEUR DE SIGNAL ALEATOIRE : AR

Soit le filtre AR d'ordre $N = 2$ excité par un bruit blanc :

$$y_n := -a_1 \cdot y_{n-1} - a_2 \cdot y_{n-2} + x_n \quad H(z) := \frac{1}{1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2}}$$

On désire que la fonction d'autocorrélation ϕ_{yy} de la sortie y soit telle que :

$$\phi_{yy0} := 6$$

$$\phi_{yy1} := 3$$

$$\phi_{yy2} := 0$$

Déterminer les coefficients a_1 et a_2 du filtre formeur AR ainsi que la variance V du bruit blanc x à l'entrée du filtre permettant d'avoir ϕ_{yy} désiré.

On résoudra le système de Yule-Walker par inversion matricielle.

On vérifiera le fait que y possède bien la fonction d'autocovariance escomptée en la calculant par la méthode par ergodicité, après avoir déterminé la sortie y sur $K = 100$ points.

1. RESOLUTION DU SYSTEME DE YULE-WALKER $Ra = v$ PAR INVERSION MATRICIELLE

$$N := 2 \quad i := 0..N \quad j := 0..N \quad k := 1..N \quad \text{RAZ : } \phi_{yy} := \text{submatrix}(\phi_{yy}, 0, 0, 0, 0)$$

$$R := \text{submatrix}(R, 0, 0, 0, 0)$$

$$r := \text{submatrix}(r, 0, 0, 0, 0)$$

Saisie du vecteur autocorrélation ϕ_{yy_i} :

$$\phi_{yy0} := 6$$

$$\phi_{yy1} := 3$$

$$\phi_{yy2} := 0$$

$$\phi_{yy} = \begin{pmatrix} 6 \\ 3 \\ 0 \end{pmatrix}$$

Création de la matrice **R** :

$$R_{i,j} := \phi_{yy} |i-j| \quad R = \begin{pmatrix} 6 & 3 & 0 \\ 3 & 6 & 3 \\ 0 & 3 & 6 \end{pmatrix} \quad \text{Exemple, pour } N = 2, R = \begin{pmatrix} \phi_{yy0} & \phi_{yy1} & \phi_{yy2} \\ \phi_{yy1} & \phi_{yy0} & \phi_{yy1} \\ \phi_{yy2} & \phi_{yy1} & \phi_{yy0} \end{pmatrix}$$

Réservation du vecteur a (ce n'est pas une initialisation) et Initialisation du vecteur b :

$$a_i := 0 \quad b_i := \phi_{yyi} \quad b_0 := 0 \quad b = \begin{pmatrix} 0 \\ 3 \\ 0 \end{pmatrix}$$

Calcul du vecteur a et de V :

$$a := \begin{pmatrix} -1 \\ \phi_{yy0} \end{pmatrix} \cdot R^{-1} \cdot b \quad V := \frac{\phi_{yy0}}{\phi_{yy0} \cdot a_0 + 1} \quad V = 4$$

$$i := 1 .. N \quad a_i := V \cdot a_i$$

Résultat :

du vecteur a, il ne faut retenir que les N paramètres cherchés : $a_1, a_2 \dots a_N$ et ignorer a_0 :

$$a = \begin{pmatrix} 0.083 \\ -0.667 \\ 0.333 \end{pmatrix} \quad a_k = \begin{matrix} -0.667 \\ 0.333 \end{matrix}$$

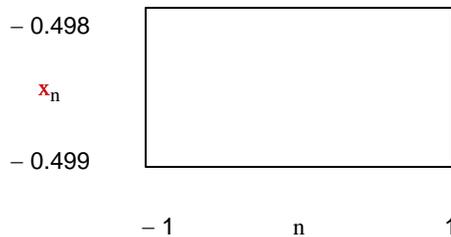
2. CALCUL du BRUIT BLANC de VARIANCE V sur K = 100 points

BruitBlanc(K, m, V) := ■

RAZ : $x := \text{submatrix}(x, 0, 0, 0, 0)$

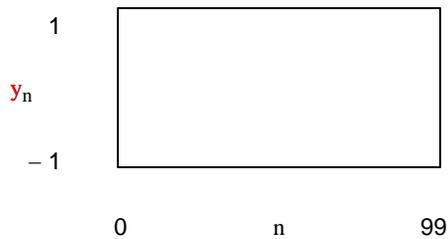
K := 100 m := 0 x := **BruitBlanc**(K, m, V)

n := 0 .. K - 1



3. CALCUL de la REPONSE y du FILTRE AR au BRUIT BLANC CENTRE de VARIANCE V

RAZ : $y := \text{submatrix}(y, 0, 0, 0, 0)$ $n := 0..K - 1$ $y_n := 0$ $y_n := \blacksquare$



4. VERIFICATION : CALCUL de la FONCTION d'AUTOCOVARIANCE par ERGODICITE

. Signal d'entrée x :

RAZ : $r := \text{submatrix}(r, 0, 0, 0, 0)$

. Moyenne : $m := \frac{1}{K} \cdot \sum_{n=0}^{K-1} x_n$

$m = \blacksquare$

. Covariance de x : $k := 0..N$

$r_k := \frac{1}{K} \cdot \sum_{n=0}^{K-k-1} x_{n+k} \cdot x_n$

$r_0 = \blacksquare$

à rapprocher de :

$V = 4$

$r = \blacksquare$

. Signal de sortie y :

RAZ : $r := \text{submatrix}(r, 0, 0, 0, 0)$

. Moyenne : $m := \frac{1}{K} \cdot \sum_{n=0}^{K-1} y_n$

$m = \blacksquare$

. Covariance de y : $k := 0..N$

$r_k := \frac{1}{K} \cdot \sum_{n=0}^{K-k-1} y_{n+k} \cdot y_n$

$r = \blacksquare$

à comparer avec :

$\phi_{yy} = \begin{pmatrix} 6 \\ 3 \\ 0 \end{pmatrix}$

Conclusion : avec un ordre N plus important pour le filtre AR, les résultats seraient encore plus proches des résultats escomptés.

2. Synthèse du signal. Identification

Synthèse du signal

I. SIGNAUX DETERMINISTES

La génération d'un signal déterministe peut être réalisée à partir :

- . de l'expression analytique (formule) du signal
- . d'un modèle (exemple : modèle AutoRégressif (AR))
- . d'une table de look-up (fichier mémoire)
- . d'une Transformation mathématique inverse (Fourier, Laplace ...)

Exemple : Génération d'un signal sinusoïdal (sans phase initiale)

. Expression analytique : Génération d'un signal sinusoïdal de fréquence f et d'amplitude A

L'expression à Temps Continu s'écrit : $x(t) = A \sin(2\pi ft)$

Le signal à Temps Discret x est généré en se donnant $N = \frac{F}{f}$ échantillons par période ($\frac{1}{f}$) du signal à générer.

$F = \frac{1}{T}$ est la fréquence d'échantillonnage.

Elle doit vérifier la condition de Shannon pour créer le signal avec une bonne résolution : $F \geq 2f$ car f est la plus grande fréquence (et ici la seule !) contenue dans le spectre de x .

→ $x(n) = A \sin(2\pi fnT)$ $n \in \mathbf{Z}$ est l'indice, généralement temporel, du signal.

. Modèle : Génération d'un signal sinusoïdal de fréquence f et d'amplitude 1

Un modèle AR d'ordre 2 peut être utilisé pour générer un sinus (et toute suite trigonométrique) : (démonstration par récurrence, le signal étant récursif) (obtention du modèle par identification par exemple)

$x(n) = \alpha x(n-1) + \beta x(n-2)$ avec :

coefficients : $\begin{cases} \alpha = 2A \cos(2\pi fT) \\ \beta = -1 \end{cases}$

initialisation : $\begin{cases} x(0) = \varphi(\text{phase initiale}) \\ x(1) = A \sin(2\pi fT) \end{cases}$ si on veut une phase initiale non nulle

ou $\begin{cases} x(0) = 0(\text{phase initiale}) \\ x(1) = A \sin(2\pi fT) \end{cases}$ si on veut une phase initiale nulle

et où : $x(n)$ représente le signal sinusoïdal à l'instant nT .

. Table de look-up : Fichier mémoire

. Transformation mathématique inverse (Laplace) : Génération d'un signal sinusoïdal de fréquence f et d'amplitude 1

$$X(p) = \frac{2\pi f}{p^2 + (2\pi f)^2} \xrightarrow{\mathcal{TL}^{-1}} x(t) = \sin(2\pi ft) \quad \text{à l'aide de tables}$$

II. SIGNAUX ALEATOIRES

Signaux aléatoires par fonction modulo

Une méthode de génération de signaux aléatoires (en fait pseudo-aléatoires du fait de la périodicité des signaux engendrés) consiste à utiliser la fonction modulo.

Signal aléatoire à distribution uniforme. Bruit blanc à distribution uniforme

Le signal $x(n)$ aléatoire à distribution uniforme est décrit par (démonstration par récurrence, le signal étant récursif) :

$$x(n) = Ax(n-1) \text{ modulo } P \quad \text{L'initialisation } x(0) \text{ devant vérifier : } 0 < x(0) < P$$

(En indexant l'initialisation sur une horloge par exemple, on a un signal aléatoire générant à chaque essai une séquence différente. Sinon la séquence engendrée est déterministe dans le sens où l'équation décrivant le signal est analytique, paramétrique).

On a :
$$0 < x(n) < P$$

(Pour avoir $0 < x(n) < 1$, il suffit de diviser chaque terme de la séquence par P).

Le signal généré est pseudo-aléatoire car périodique de période P . Pour ne pas, lorsque n augmente, retomber vite sur la même série de coefficients de la séquence, il vaut mieux choisir P grand.

P doit être un nombre premier pour que le résultat du modulo ne soit pas nul.

On peut prendre par exemple :
$$x(0) = 100, A = 281, P = 31357.$$

Les échantillons peuvent être considérés comme indépendants les uns des autres, conduisant ainsi à fournir un *bruit blanc à distribution uniforme*.

Signal aléatoire à distribution de Rayleigh

Le signal $y(n)$ aléatoire à distribution de Rayleigh peut être obtenu à partir du signal aléatoire $x(n)$ à distribution uniforme :

$$y(n) = 2\sigma \text{Ln}\{1/x(n)\}$$

σ caractérise la dynamique du signal.

On peut prendre par exemple :
$$\sigma = 10.$$

Signal aléatoire à distribution gaussienne

Le signal $w(n)$ aléatoire à distribution gaussienne peut être obtenu à partir des signaux aléatoires $x(n)$ à distribution uniforme et $y(n)$ à distribution de Rayleigh :

$$w(n) = y(n) \cos\{2\pi x(n)\}$$

Signaux aléatoires par fonction OU exclusif

Une autre méthode de génération de signaux aléatoires consiste à utiliser la fonction OU exclusif.

Un échantillon de la séquence aléatoire est codé sur m bits. Le mot de l'échantillon $x(n+1)$ de la séquence est obtenu à partir du mot de l'échantillon $x(n)$ en effectuant un OU exclusif entre les 2 bits de plus faible poids de $x(n)$, et en décalant circulairement vers la droite les bits du mot : le LSB est perdu, et il rentre le résultat du OU exclusif pour nouveau MSB.

Un mot a conventionnellement son MSB à l'extrême gauche et son LSB à l'extrême droite. Le MSB est le bit de plus fort poids, le LSB celui de poids le plus faible.

Identification (modélisation)

I. IDENTIFICATION PAR FILTRES FORMEURS - MODELES AR, MA ARMA

1. Processus MA (Moving Average - Moyenne mobile) **d'ordre M**

Un signal à TD x_n peut être identifié comme sortie d'un filtre MA excité par un bruit blanc (car il y a **tout** dans un *bruit blanc*, un bruit blanc excite tous les modes du processus, il y a toutes les fréquences dans un bruit blanc) :

$$x_n = \sum_{k=0}^M b_k w_{n-k} = b_0 w_n + b_1 w_{n-1} + \dots + b_M w_{n-M} \quad \text{avec } b_0 = 1 \quad (\text{le fait de diviser tous les coefficients par } b_0 \text{ conduit à } b_0 = 1)$$

où :

w_n désigne un processus aléatoire réel blanc, centré, stationnaire, de variance σ^2

et : $\{b_m\}$ est une séquence de $M + 1$ coefficients réels.

Le processus ainsi construit a pour RI la séquence $\{b_m\}$: $\{h_n\} = \{b_n\}$

Ce filtre a donc pour FT :

$$H(z) = TZ[b_n] = \sum_{n=0}^M b_n z^{-n} = b_0 + b_1 z^{-1} + \dots + b_M z^{-M}$$

Cette FT ne possède pas de pôles. Elle conduit donc à un filtre causal stable, facilitant ainsi l'identification du signal x_n .

Relations entre coefficients du modèle et covariances

Malheureusement, les relations entre coefficients du modèle et covariances ne sont pas linéaires.

Il suffit pour s'en convaincre de traiter en exemple le processus MA d'ordre $M = 2$ à TD engendré par :

$$x_n = \sum_{k=0}^2 b_k w_{n-k} = w_n + b_1 w_{n-1} + b_2 w_{n-2}$$

Calculons l'autocovariance $R_{XX_k} = E[X_{n+k} X_n]$ où : X_n est la VA dont x_n est une réalisation

W_n est la VA dont w_n est une réalisation

(on rappelle qu'un bruit blanc de variance σ^2 est tel que : $E[W_{n+k} W_n] = \sigma^2 \delta_{n,k}$ et que R_{XX_k} est paire)

$$\begin{cases} R_{XX_0} = E[X_n^2] = E[W_n^2] + b_1^2 E[W_{n-1}^2] + b_2^2 E[W_{n-2}^2] = (1 + b_1^2 + b_2^2) \sigma^2 \\ R_{XX_1} = E[X_{n+1} X_n] = E[(W_{n+1} + b_1 W_n + b_2 W_{n-1})(W_n + b_1 W_{n-1} + b_2 W_{n-2})] = (b_1 + b_1 b_2) \sigma^2 \\ R_{XX_2} = E[X_{n+2} X_n] = E[(W_{n+2} + b_1 W_{n+1} + b_2 W_n)(W_n + b_1 W_{n-1} + b_2 W_{n-2})] = b_2 \sigma^2 \\ R_{XX_k} = E[X_{n+k} X_n] = 0 \quad \text{pour } k \geq M + 1 \end{cases}$$

soit :

$$\begin{cases} R_{XX_0} = (1 + b_1^2 + b_2^2) \sigma^2 \\ R_{XX_1} = (b_1 + b_1 b_2) \sigma^2 \\ R_{XX_2} = b_2 \sigma^2 \\ R_{XX_k} = 0 \quad \text{pour } k \geq M + 1 \end{cases}$$

Lorsque l'on veut identifier (\equiv modéliser) un signal x_n comme un processus MA d'ordre donné (ici 2) d'entrée w_n :

$x_n = w_n + b_1 w_{n-1} + b_2 w_{n-2}$ alors la covariance R_{XX_k} est connue (déduite de la séquence $\{x_k\}$ du signal par une méthode d'estimation de la *fonction d'autocovariance*, par exemple, la méthode par *ergodicité*) et b_1 , b_2 et σ^2 sont les inconnues.

Le système d'équations est non linéaire et pour le résoudre on fait appel à des algorithmes de Programmation Non Linéaire.

2. Processus AR (*Auto-Regressive* - Auto-Régressif) **d'ordre N**

Un signal à TD x_n peut être vu comme sortie d'un filtre modèle AR excité par un bruit blanc (car il y a *tout* dans un *bruit blanc*, un bruit blanc excite tous les modes du processus, il y a toutes les fréquences dans un bruit blanc) :

$$\sum_{k=0}^N a_k x_{n-k} = w_n \quad \text{avec} \quad a_0 = 1 \quad (\text{le fait de diviser tous les coefficients par } a_0 \text{ ne change rien et conduit à } a_0 = 1)$$

soit : $a_0 x_n + a_1 x_{n-1} + \dots + a_N x_{n-N} = w_n$

où : w_n désigne un processus aléatoire réel blanc, centré, stationnaire, de variance σ^2

et : $\{a_n\}$ est une séquence de $N + 1$ coefficients réels.

Le processus ainsi construit a pour RI la séquence $\{h_n\} = TZ^{-1}[H(z)]$ où $H(z)$ est la FT du filtre :

$$H(z) = \frac{1}{\sum_{n=0}^N a_n z^{-n}} = \frac{1}{a_0 + a_1 z^{-1} + \dots + a_N z^{-N}} \quad (\text{filtre } \textit{tout-pôle})$$

Cette FT possède des pôles. Ceux-ci doivent être de module < 1 pour assurer la stabilité du filtre causal.

Relations entre coefficients du modèle et covariances (cas $N = 1$)

Prenons pour exemple le processus AR d'ordre $N = 1$ à TD engendré par :

$$x_n + a_1 x_{n-1} = w_n \quad \text{avec} \quad |a_1| < 1 \quad \text{pour assurer la stabilité du filtre.}$$

En prenant l'espérance des 2 membres, et en utilisant l'hypothèse que w_n est centré, on en déduit que le processus x_n est lui-même centré.

Calculons sa fonction d'autocovariance $R_{XX_k} = E[X_{n+k} X_n]$ où : X_n est la VA dont x_n est une réalisation

W_n est la VA dont w_n est une réalisation

(on rappelle qu'un bruit blanc de variance σ^2 est tel que : $E[W_{n+k} W_n] = \sigma^2 \delta_{n,k}$ et que R_{XX_k} est paire)

Comme : $x_n = w_n - a_1 x_{n-1}$

on a : $x_n^2 = x_n w_n - a_1 x_n x_{n-1} = (w_n - a_1 x_{n-1}) w_n - a_1 x_n x_{n-1} = w_n^2 - a_1 x_{n-1} w_n - a_1 x_n x_{n-1}$

d'où, comme il n'y a pas de dépendance causale entre x_{n-1} et w_n

(x_n s'exprime causalement en fonction de w_n , ce qui signifie que x_{n-1} est fonction de w_{n-1}, w_{n-2}, \dots)

Comme w_{n-1}, w_{n-2}, \dots sont indépendants de w_n (bruit blanc), alors x_{n-1} et w_n sont indépendants donc non corrélés, ce qui s'écrit :

$$E[X_{n-1} W_n] = E[X_{n-1}] E[W_n] = 0 \quad \text{car le bruit blanc est centré}$$

on a : $E[X_n^2] = E[W_n^2] - a_1 E[X_n X_{n-1}]$ soit : $R_{XX_0} = \sigma^2 - a_1 R_{XX_1}$

On a aussi : $x_n x_{n-1} = x_{n-1} w_n - a_1 x_{n-1}^2$

d'où : $E[X_n X_{n-1}] = -a_1 E[X_{n-1}^2]$ soit : $R_{XX_1} = -a_1 R_{XX_0}$

On a donc le système d'équations :

$$\begin{cases} R_{XX_0} + a_1 R_{XX_1} = \sigma^2 \\ R_{XX_1} + a_1 R_{XX_0} = 0 \end{cases}$$

Ce système linéaire d'équations peut s'écrire sous forme matricielle :

$$\text{avec : } \mathbf{R} = \begin{bmatrix} R_{XX_0} & R_{XX_1} \\ R_{XX_1} & R_{XX_0} \end{bmatrix} \quad \mathbf{a} = \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} \quad \mathbf{R} \mathbf{a} = \mathbf{v} \quad \mathbf{v} = \begin{bmatrix} \sigma^2 \\ 0 \end{bmatrix} \quad a_0 = 1$$

La solution de ce système linéaire d'équations est immédiate :

$$\begin{cases} a_1 = -\frac{R_{XX_1}}{R_{XX_0}} \\ \sigma^2 = R_{XX_0} - \frac{R_{XX_1}^2}{R_{XX_0}} \end{cases}$$

Relations entre coefficients du modèle et covariances (cas N quelconque)

L'expression matricielle permet la généralisation à un modèle AR d'ordre N dont la fonction d'autocovariance vérifie le système linéaire d'équations : (ce sont les équations de **Yule-Walker**)

$$\begin{bmatrix} R_{XX_0} & R_{XX_1} & \cdots & R_{XX_N} \\ R_{XX_1} & R_{XX_0} & \cdots & R_{XX_{N-1}} \\ \vdots & \vdots & \ddots & \vdots \\ R_{XX_N} & R_{XX_{N-1}} & \cdots & R_{XX_0} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_N \end{bmatrix} = \begin{bmatrix} \sigma^2 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad \text{ou encore : } \mathbf{R} \mathbf{a} = \mathbf{v} \quad a_0 = 1$$

La matrice d'autocovariance \mathbf{R} de terme général $r_{i,j}$ ne dépendant que de $i-j$ (R_{XX_k} paire) est dite matrice de Toeplitz.

Pour réaliser ce calcul, une inversion matricielle est possible mais on utilise généralement l'algorithme efficace itératif de **Levinson** ou encore de **Burg**, qui évitent une inversion matricielle en mettant à profit le caractère Toeplitz de la matrice \mathbf{R} .

La séquence $\{a_n\}$, solution avec σ^2 , du système linéaire de **Yule-Walker** à partir de la connaissance de la matrice d'autocovariance \mathbf{R} , conduit à la relation de récurrence du filtre.

La détermination de la séquence $\{a'_n\}$ des pôles de $H(z) = \frac{1}{1 + a_1 z^{-1} + \cdots + a_N z^{-N}} = \frac{1}{(1 - a'_1 z^{-1})(1 - a'_2 z^{-1}) \cdots (1 - a'_N z^{-1})}$ **conduit, de façon équivalente à celle $\{a_n\}$, à l'obtention du filtre d'identification de la séquence x_n .**

L'avantage de la modélisation AR par rapport à l'identification par un modèle MA est qu'elle conduit à un système linéaire d'équations pour la détermination des coefficients du filtre.

Applications :

- . synthèse de signal
- . compression de signal (de parole par exemple) :

un signal x_k de K échantillons d'un signal de parole peut être codé non pas par mémorisation de la séquence brute $\{x_k\}$ mais compressée pour être ramenée aux coefficients a_i du filtre modèle (AR) d'ordre N et au coefficient σ^2 , variance du bruit blanc placé à l'entrée du filtre pour produire, synthétiser le signal x_k . La décompression consiste à générer, synthétiser x_k à partir du bruit blanc de variance σ^2 par application de l'algorithme AR du filtre formeur d'ordre N au bruit blanc.

3. Processus ARMA d'ordre $N - M$

Un processus ARMA s'obtient en mettant en série une structure AR et une structure MA.

Soit l'équation récurrente :

$$\sum_{k=0}^N a_k x_{n-k} = \sum_{i=0}^M b_i w_{n-i} \quad \text{avec} \quad a_0 = 1 \quad (\text{le fait de diviser tous les coefficients par } a_0 \text{ ne change rien et conduit à } a_0 = 1)$$

soit : $a_0 x_n + a_1 x_{n-1} + \dots + a_N x_{n-N} = b_0 w_n + b_1 w_{n-1} + \dots + b_M w_{n-M}$

où :

w_n désigne un processus aléatoire réel blanc, centré, stationnaire, de variance σ^2
 et : $\{a_n\}$ et $\{b_m\}$ sont des séquences de coefficients réels.

Le processus ainsi construit a pour RI la séquence $\{h_n\} = TZ^{-1}[H(z)]$ où $H(z)$ est la FT du filtre :

$$H(z) = \frac{\sum_{m=0}^M b_m z^{-m}}{\sum_{n=0}^N a_n z^{-n}} = \frac{b_0 + b_1 z^{-1} + \dots + b_M z^{-M}}{a_0 + a_1 z^{-1} + \dots + a_N z^{-N}}$$

Cette FT possède des pôles. Ceux-ci doivent être de module < 1 pour assurer la stabilité du filtre causal.

L'équation récurrente admet une solution unique x_n , stationnaire au 2nd ordre, unique, qui s'exprime causalement en fonction de w_n , si et seulement si les racines du dénominateur (\equiv pôles de $H(z)$) sont de module < 1 .

4. Remarques

. Il peut paraître réducteur de modéliser un processus par un nombre restreint de paramètres. Mais cette modélisation correspond à un processus SLTI (Système Linéaire à Temps Invariant), décrit par une équation aux différences linéaire stationnaire (\equiv à coefficients constants).

. On montre que tout processus ARMA ou MA peut être approché par un processus AR d'ordre suffisamment élevé (l'avantage d'une modélisation AR est de conduire à un système **linéaire** d'équations pour la détermination des paramètres du modèle, mais il faut en contre-partie veiller à la **stabilité** du filtre).

De même, tout processus ARMA ou AR peut être approché par un processus MA d'ordre suffisamment élevé (l'avantage d'une modélisation MA est d'assurer la **stabilité** du filtre modèle, mais en contre-partie, cette modélisation conduit à un système **non linéaire** d'équations rendant ainsi moins simple la détermination des paramètres du modèle).

Ce résultat est fondamental en pratique puisque, si l'on choisit parmi ces 3 modèles, le plus « mauvais » modèle, une précision raisonnable peut encore être obtenue en prenant un ordre assez élevé.

Un processus dont le spectre possède des « vallées profondes » nécessite en effet moins de paramètres si on utilise, pour le représenter, un modèle MA plutôt qu'un modèle AR.

De même, un processus dont le spectre possède des « pics élevés » nécessite moins de paramètres si on utilise, pour l'identifier, un modèle AR plutôt qu'un modèle MA.

. Application de l'identification : **Compression de signal**

Plutôt que de mémoriser le signal brut, on peut stocker les paramètres AR ou MA ou ARMA du modèle.

La compression consiste, après le choix d'un ordre d'autant plus élevé que la précision souhaitée, en la détermination des paramètres du modèle et de la variance du bruit blanc.

La décompression consiste alors en la génération du signal étant donné le modèle choisi.

. Autre procédé d'identification par les modèles MA, AR ou ARMA

Exemple : modélisation AR

Plutôt que d'utiliser un bruit blanc pour l'entrée du filtre, on prend une constante (b), et on détermine cette constante ainsi que les paramètres (a_n) du modèle en minimisant (par application du critère des moindres carrés) (cf. annexe)

l'énergie de l'erreur entre la séquence x_n à identifier et son estimation \hat{x}_n décrite par :

$$\hat{x}_n = b + \sum_{k=1}^N a_k \hat{x}_{n-k}$$

5. Application : Estimation du spectre d'un processus AR (Analyse spectrale - Estimation)

Dans le cas d'un processus AR d'ordre N , la relation entre les paramètres à estimer et les covariances ($\mathbf{R} \mathbf{a} = \mathbf{v}$) est linéaire. Elle permet simplement d'obtenir une estimation des paramètres $\{a_n\}$ du modèle et la variance σ^2 du bruit blanc d'entrée, en substituant à la matrice de covariance exacte une matrice de covariance estimée.

Relations entre coefficients du modèle et covariances estimées (cas $N = 1$)

On remplace dans l'équation obtenue avec les covariances exactes :

$$\begin{cases} R_{XX_0} + a_1 R_{XX_1} = \sigma^2 \\ R_{XX_1} + a_1 R_{XX_0} = 0 \end{cases}$$

R_{XX_0} et R_{XX_1} par leur estimée respective.

L'ergodicité toujours supposée nous donne : (X_n est supposé centré)

$$\begin{cases} R_{XX_0} = \frac{1}{K} \sum_{n=0}^{K-1} x_n^2 = \frac{1}{K} (x_0^2 + x_1^2 + \dots + x_{K-1}^2) \\ R_{XX_1} = \frac{1}{K-1} \sum_{n=0}^{K-2} x_{n+1} x_n = \frac{1}{K-1} (x_1 x_0 + x_2 x_1 + \dots + x_{K-1} x_{K-2}) \end{cases}$$

L'estimée respective de R_{XX_0} et de R_{XX_1} est donnée par (méthode des corrélations) :

$$\begin{cases} \hat{R}_{XX_0} = \frac{1}{K} \sum_{n=0}^K x_n^2 = \frac{1}{K} (x_0^2 + x_1^2 + \dots + x_K^2) \\ \hat{R}_{XX_1} = \frac{1}{K} \sum_{n=0}^{K-1} x_n x_{n+1} = \frac{1}{K} (x_0 x_1 + x_1 x_2 + \dots + x_{K-1} x_K) \end{cases}$$

On obtient pour matrice de covariance estimée :

$$\hat{\mathbf{R}} = \begin{bmatrix} \hat{R}_{XX_0} & \hat{R}_{XX_1} \\ \hat{R}_{XX_1} & \hat{R}_{XX_0} \end{bmatrix}$$

Et le coefficient a_1 du modèle AR ainsi que la variance σ^2 du bruit blanc d'entrée sont :

$$\begin{cases} a_1 = -\frac{\hat{R}_{XX_1}}{\hat{R}_{XX_0}} \\ \sigma^2 = \hat{R}_{XX_0} - \frac{\hat{R}_{XX_1}^2}{\hat{R}_{XX_0}} \end{cases}$$

Relations entre coefficients du modèle et covariances estimées (cas N quelconque)

Un résultat important est que si la matrice \mathbf{R} est de *Toeplitz*, le polynôme dénominateur de la FT $H(z)$:

$1 + a_1 z^{-1} + \dots + a_N z^{-N}$ a toutes ses racines à l'intérieur du cercle unité, assurant ainsi la *stabilité* du filtre modèle.

Estimation du spectre d'un processus AR

méthode haute résolution

Une fois les paramètres $\{a_n\}$ du modèle et la variance σ^2 du bruit blanc d'entrée estimés (par $\{\hat{a}_n\}$ et $\hat{\sigma}^2$), on détermine le spectre $S_X(f)$ de la sortie x_n du filtre AR d'entrée b_n (bruit blanc).

On a : $S_X(f) = |H(f)|^2 S_B(f) = |H(f)|^2 \hat{\sigma}^2$ car le spectre (DSP) d'un bruit blanc est σ^2 :

$$S_B(f) = \hat{\sigma}^2$$

d'où : $S_X(f) = |H(f)|^2 \hat{\sigma}^2 = |H(z = e^{i2\pi f})|^2 \hat{\sigma}^2$ avec : $H(z) = \frac{1}{1 + \hat{a}_1 z^{-1} + \dots + \hat{a}_N z^{-N}}$

donc :
$$S_X(f) = \frac{\hat{\sigma}^2}{|1 + \hat{a}_1 e^{-i2\pi f} + \dots + \hat{a}_N e^{-i2\pi N f}|^2}$$

Cette méthode d'estimation spectrale s'appelle méthode *haute résolution*.

II. IDENTIFICATION PAR PREDICTION LINEAIRE (Linear Prediction Coding)

[Voir Chapitre sur la Prédiction]

III. IDENTIFICATION PAR MOINDRES CARRES

[Voir Chapitre sur le filtrage optimal]

Remarque

Une identification par la méthode du *polynôme de Lagrange* est inenvisageable en traitement du signal car pour identifier un signal de n points, n coefficients (les n coefficients du polynôme de Lagrange d'ordre n) sont nécessaires, ce qui est énorme en comparaison des autres méthodes d'identification qui nécessitent un nombre m de paramètres bien moindre que la taille du signal à identifier : $m \ll n$.

TP 2. Synthèse du signal. Identification

I. SYNTHÈSE DE SIGNAL

1. Signaux déterministes

Générer par récurrence un signal sinusoïdal u d'amplitude 1 et de fréquence $f = 5$ Hz sur une "durée" $N = 50$ points (durée = $N \cdot T$). La fréquence d'échantillonnage $F = 1/T$ doit respecter la condition de Shannon : $F > 2 F_m$ avec ici $F_m = f$.

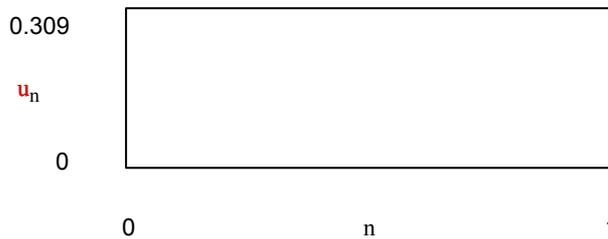
$$f := 5 \quad F := 100 \quad T := \frac{1}{F} \quad \alpha := 2 \cdot \cos(2 \cdot \pi \cdot f \cdot T) \quad \beta := -1 \quad N := 50 \quad n := 0..N - 1$$

Initialisation : $u_0 := \blacksquare$ $u_1 := \blacksquare$

Equation récurrente : $u_n := \blacksquare$

$T = 0.01$ seconde

$n := 0..N - 1$



Visuellement, on vérifie bien :

"période" : $\frac{F}{f} = 20$ points

période : $p := \frac{F}{f} \cdot T$ $p = 0.2$ sec.

fréquence : $\frac{1}{p} = 5$ Hertz

2. Signaux aléatoires

Génération d'un Bruit blanc centré de distribution uniforme

Générer un signal aléatoire de distribution uniforme, variant entre $a = -0.5$ et $b = +0.5$ sur une "durée" (en nombre de points) $N = 100$ points. Vérifier qu'il s'agit d'un bien d'un bruit blanc centré en calculant par la méthode par ergodicité sa moyenne, sa fonction d'autocovariance et sa variance.

Bruit blanc v compris entre 0 et P :

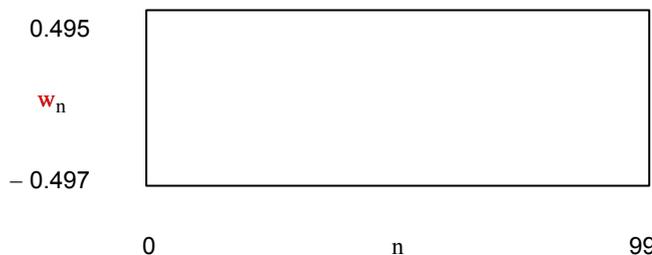
$$A := 281 \quad P := 31357 \quad N := 100 \quad n := 1..N - 1 \quad (\text{optionnel}) \quad \text{WhiteNoise}(N) := \blacksquare$$

Initialisation : $v_0 := 100$

Equation récurrente : $v_n := \blacksquare$

Bruit blanc w compris entre a et b :

$$a := -0.5 \quad b := 0.5 \quad n := 0..N - 1 \quad w_n := \blacksquare$$



Vérification :

. Moyenne et variance théoriques :

La théorie dit qu'une loi uniforme sur [a,b] a pour moyenne et variance :

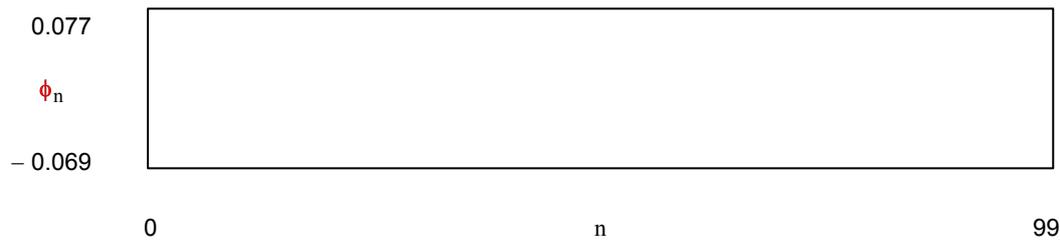
$$m := \frac{(a + b)}{2} \quad m = 0$$

$$\sigma^2 := \frac{(b - a)^2}{12} \quad \sigma^2 = 0.083$$

. Estimation de la covariance par ergodicité :

. Moyenne : $m := \frac{1}{N} \cdot \sum_{i=0}^{N-1} w_i \quad m = \bullet$

. Covariance : $\phi_n := \frac{1}{N} \cdot \sum_{i=0}^{N-n-1} (w_{i+n} - m) \cdot (w_i - m)$



Conclusion: la fonction de covariance a l'allure attendue par la théorie: $\phi_n = \sigma^2 \cdot \delta(n)$

Varance du bruit blanc: $\sigma^2 := \phi_0$

$$\sigma^2 = \bullet$$

II. IDENTIFICATION *Faire varier l'ordre N du filtre modèle AR puis la taille K du signal à identifier. Conclusion ? Essayer ensuite sur un second signal.*

Soit le filtre AR d'ordre N excité par un bruit blanc.

Equation de récurrence du filtre AR d'ordre N :

$$y_k := x_k - \sum_{i=1}^N a_i \cdot y_{k-i}$$

Fonction de Transfert correspondante :

$$H(z) := \frac{1}{1 + \sum_{i=1}^N a_i \cdot z^{-i}}$$

L'identification d'un signal s donné consiste à le modéliser comme réponse d'un filtre AR d'ordre N à un bruit blanc de variance V. Le problème est alors de déterminer les coefficients a_i du filtre AR modèle et la variance V du bruit blanc.

1. Fonction YuleWalker(φ_{xx},N): Résolution du système de Yule-Walker Ra = v par inversion matricielle

Paramètres d'entrée : . φ_{xx} : le vecteur fonction d'autocovariance (dim N) du signal x : φ_{xx0}, φ_{xx1} ... φ_{xxN-1}
 . N : dim de φ_{xx} et de l'ordre (max) du filtre formeur AR

Paramètre de sortie : . le vecteur a (dim N+1):
 a₀ = V : variance du bruit blanc; a₁, a₂, ... a_N : paramètres du filtre formeur AR d'ordre N

```

YuleWalker(φxx,N) :=
    for i ∈ 0..N
        for j ∈ 0..N
            Ri,j ← φxx[i-j]
        b0 ← 0
        for i ∈ 1..N
            bi ← φxxi
        α ← ( (-1) / φxx0 ) · R-1 · b
        a0 ← φxx0 / ( φxx0 · α0 + 1 )
        for i ∈ 1..N
            ai ← a0 · αi
    a
    
```

Exemple : K := 3 r₀ := 6

 N := 2 r₁ := 3

 r₂ := 0

a := YuleWalker(r,N)

V := a₀

$$a = \begin{pmatrix} 4 \\ -0.667 \\ 0.333 \end{pmatrix}$$

V = 4

i := 1..N

a_i =

-0.667
0.333

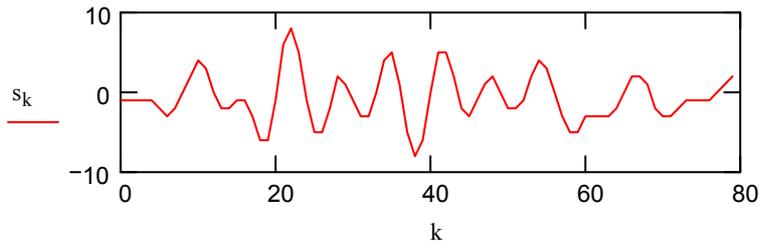
2. Extraction des paramètres a_1, a_2, \dots, a_N du filtre AR d'ordre N et de la Variance V du bruit blanc d'entrée

Signal de parole sur une fenêtre de K points :

. On considère 2 signaux de parole enregistrés dans les fichiers a1 et o1 issus de l'enregistrement des voyelles "a" et "o", échantillonnés à 8 kHz mono 8 bits et de durée 0.1s -> leur taille est de 800 échantillons.

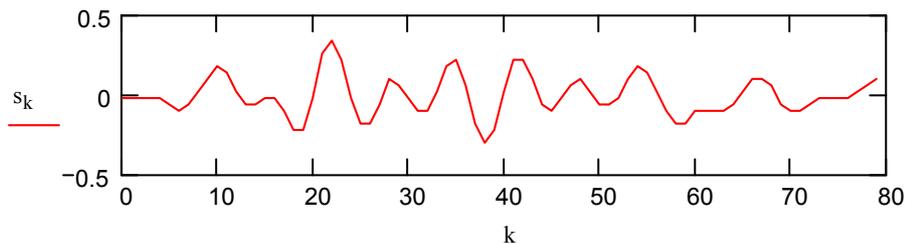
. Lecture du fichier de données ASCII (1 donnée (float) par ligne, 800 colonnes)

```
K := 80      k := 0..K - 1      s := READWAV("a1.wav")      s := s - 128
```



. Centrage du signal et recadrage d'amplitude dans la gamme [-0.5, +0.5] : (car on synthétisera à partir d'un bruit blanc centré)

```
moy := mean(s)      s_k := s_k - moy      maxi := max(s)      mini := min(s)      s_k := (2*s_k) / (maxi - mini)
```



. Calcul de la fonction d'autocovariance r du signal x par la méthode d'ergodicité :

. Moyenne : $m := \frac{1}{K} \cdot \sum_{n=0}^{K-1} s_n$ $m = -3.4 \times 10^{-3}$

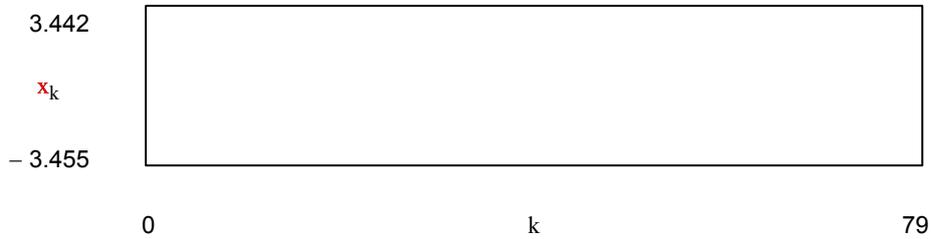
. Covariance : $r_k := \frac{1}{K} \cdot \sum_{n=0}^{K-k-1} s_{n+k} \cdot s_n$

. Extraction des N paramètres a_1, a_2, \dots, a_N du filtre AR d'ordre N et de la Variance V du bruit blanc d'entrée :

```
N := 2      a := ■      V := ■      V = ■      Attention : il faut V > 0
i := 1..N   a_i = ■
```

3. Génération du bruit blanc centré de variance V de K points :

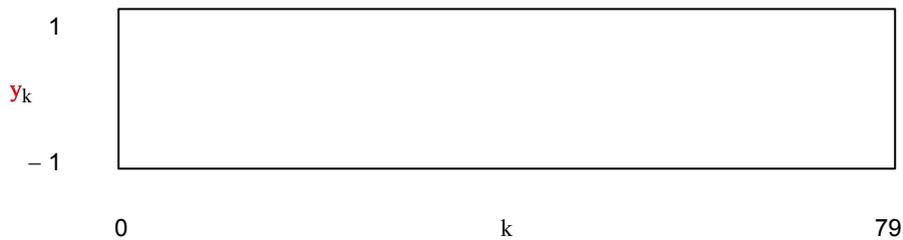
$\text{BruitBlanc}(K, m, V) := \sqrt{12 \cdot V} \cdot \text{WhiteNoise}(K) + m$ ou encore : $\text{BruitBlanc}(K, m, V) := \sqrt{12 \cdot V} \cdot \text{whiten}(K) + m$
 $m := 0$ $x := \text{BruitBlanc}(K, m, V)$



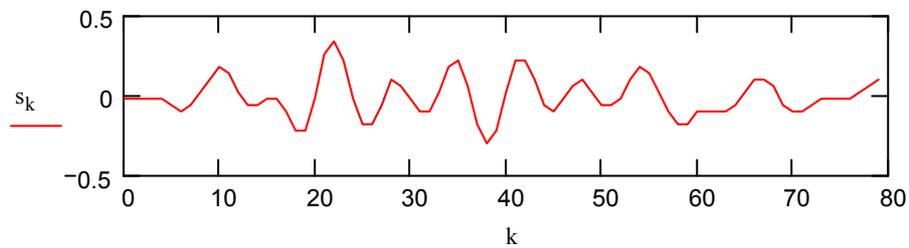
4. Calcul de la réponse y du filtre AR d'ordre N au bruit blanc x centré de variance V

RAZ : $y_k := 0$ $y_k :=$

signal synthétisé y : *Attention : il faut que le filtre soit stable*



à comparer au signal de parole s (il faudrait écouter pour comparer !) :



Rapport Signal sur Bruit :

RSB := ■

RSB = ■ dB

APPLICATION :

Synthèse, Compression du signal de parole dans sa totalité (K = 800 échantillons, durée 100 ms)

On remplace chaque bloc de P échantillons du signal de parole de K = 800 échantillons (durée 100 ms) par les N coefficients du modèle AR. Ainsi si P = 80 et N = 8 on obtient un taux de compression d'environ 10.

Pour mettre en oeuvre ce traitement, on centre d'abord le signal de K = 800 échantillons et on recadre son amplitude dans la gamme [-1, +1].

On fenêtré ensuite ce signal par tranches de P = 80 échantillons (durée 10 ms), en prenant 50% de chevauchement entre les fenêtres pour atténuer l'effet des variations éventuellement brutales des coefficients de compression des coefficients d'une fenêtre à la suivante.

Synthèse, Compression du signal image

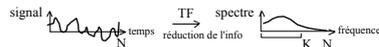
Selon le même principe, effectuer la compression d'une image en se ramenant à 1D et en traitant l'image colonne par colonne.

3. Caractérisation (Analyse - Transformations fréquentielles). Estimation

Caractérisation *(Caractérisation de signal pour la distinction, la reconnaissance de signal)*

Le but de la caractérisation d'un signal est de trouver une signature, une caractéristique pour un signal, autorisant sa reconnaissance, sa distinction, sa discrimination. La difficulté est de trouver, suivant le type de signal, le bon outil de caractérisation, permettant sa discrimination par rapport à des signaux différents tout en autorisant une relative variabilité de la signature, dans le cas de réalisations multiples d'un même type de signal (ex : reconnaissance de parole : plusieurs reproductions du même mot doivent être classifiées ensemble et discriminées par rapport à la prononciation d'autres mots).

Plutôt que d'utiliser le signal brut comme signature, on prend une transformée de ce signal, ayant l'avantage de compresser, de réduire la quantité d'informations utiles, pertinentes du signal.



Exemple 1 :

Il y a réduction de l'information car le spectre s'annule vite quand la fréquence augmente (un signal ne peut être infiniment rapide), l'information contenue dans les N points du signal temporel se retrouve, pour l'essentiel, dans les Kers points (les K 1ers points, avec $K < N$) de sa Représentation Fréquentielle.

Exemple 2 : Reconnaissance de parole, d'image

La reconnaissance est basée sur la corrélation, mais plutôt que de corréler les signaux bruts (le signal à reconnaître et les enregistrements de référence en Base De Données), ce qui est coûteux en temps calcul et pas très efficace pour la reconnaissance, on corréle des primitives, des caractéristiques extraites du signal à reconnaître avec les mêmes primitives issues des enregistrements de référence, ce qui accroît la rapidité et la qualité de la reconnaissance (exemple de primitive de caractérisation de la parole (reconnaissance de la parole, du locuteur) : le *cepstre*).

I. TRANSFORMATIONS FREQUENTIELLES

outils de caractérisation fréquentiels

TRANSFORMATION DE FOURIER DISCRETE (TFD)

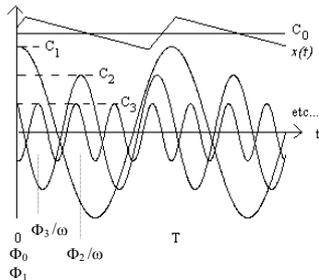
Elle permet de caractériser un signal stationnaire en donnant son occupation en fréquences.

Rappel : Transformation de Fourier

Un signal $x(t)$ périodique de période T , décomposé en série de Fourier s'écrit :

$$x(t) = \sum_{k=0}^{\infty} C_k \cos(k\omega t + \Phi_k) \quad \text{avec : } \omega = \frac{2\pi}{T} \quad T : \text{période de } x(t)$$

Exemple : Décomposition en série de Fourier d'un signal périodique « triangulaire »



TFD :

$$\left| \begin{array}{l} x(n) \\ M \text{ termes et causale} \end{array} \right. \xrightarrow{\text{TFD}} \left| \begin{array}{l} X(k) = \frac{1}{M} \sum_{n=0}^{M-1} x(n)e^{-i2\pi kn/M} = \text{TFD}[x(n)] \\ M \text{ termes} \end{array} \right.$$

n représente un indice temporel, k un indice fréquentiel.

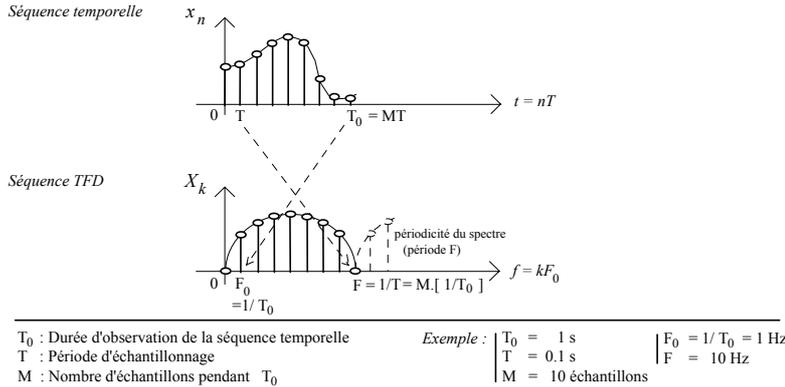
Restriction : la TFD ne donne pas de localisation (temporelle) des fréquences, mais seulement leur répartition globale.

TFD Inverse (TFDI)

Elle est donnée par la relation :

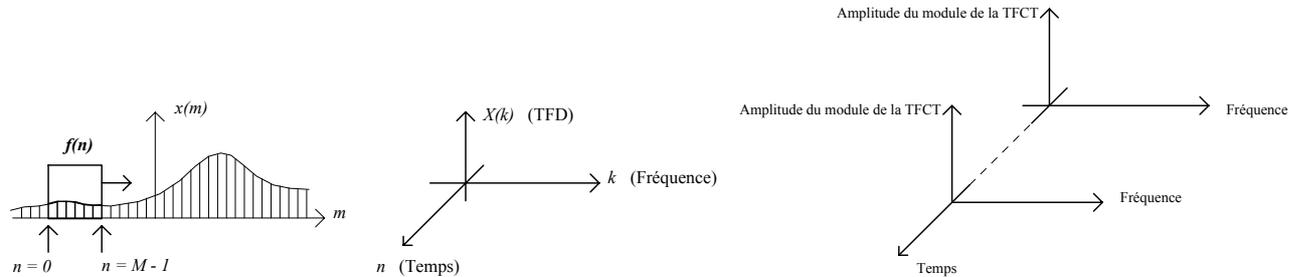
$$\left| \begin{array}{l} x(n) \\ M \text{ termes} \end{array} \right. = \left| \begin{array}{l} \sum_{k=0}^{M-1} X(k)e^{i2\pi kn/M} = \text{TFDI}[X(k)] \\ M \text{ termes} \end{array} \right.$$

Interprétation



SPECTROGRAMME - TRANSFORMATION DE FOURIER A COURT TERME (TFCT) (Analyse temps-fréquence, comme l'analyse en ondelettes)

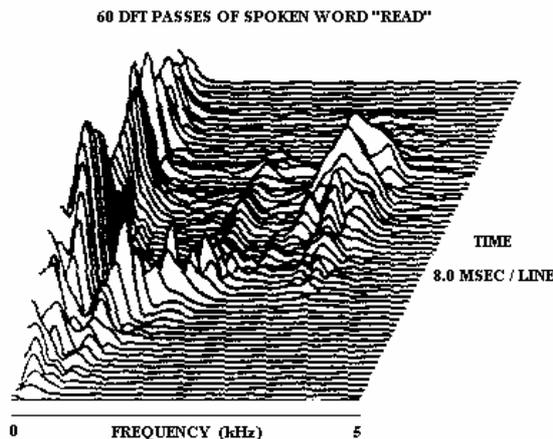
Le spectrogramme calcule la TFD sur une fenêtre $f(n)$ glissante au cours du temps sur le signal. Les TFDs sur chaque fenêtre glissante fournissent le spectrogramme, qui permet d'adapter la Transformation de Fourier à la caractérisation des signaux non stationnaires. Le spectrogramme permet de caractériser un signal non stationnaire en donnant son occupation en fréquences en fonction du temps : analyse Temps-Fréquence.



On obtient pour chaque indice n de fenêtre $f(n)$ une TFD :

$$X(k) = \frac{1}{M} \sum_{m=0}^{M-1} x(m) e^{-i2\pi km / M}$$

et au total , non plus un spectre (2D), mais un spectrogramme (3D) (Représentation Temps-Fréquence) :



Spectrogram (from 2D graphs) of the word "READ" computed from contiguous 8 msec. speech segments.

On a la relation de définition de la TFCT :

$$X(n, k) = \frac{1}{M} \sum_{m=-\infty}^{\infty} x(m) f(n - m) e^{\frac{i2\pi km}{M}}$$

$f(n)$ peut être une fenêtre de pondération (Blackman, Bartlett ...) plutôt qu'une simple fenêtre rectangulaire de troncature. Cette analyse temps-fréquence présente l'avantage par rapport à l'analyse de Fourier simple, de permettre la localisation temporelle des fréquences contenues dans le signal (voir « spectrogram of the word READ »).

TRANSFORMATION DE WIGNER-VILLE (Analyse temps-fréquence, comme l'analyse en ondelettes)

Elle permet de caractériser un signal non stationnaire en donnant son occupation en fréquences en fonction du temps : analyse Temps-Fréquence.

La TWV d'un signal $x(t)$ s'écrit : $TWV[x(t)] = \int_{-\infty}^{\infty} x\left(t + \frac{\tau}{2}\right) \cdot x^*\left(t - \frac{\tau}{2}\right) \cdot e^{-i2\pi v\tau} d\tau = X(v, t)$ (x^* : conjugué de x)

Le résultat, fonction de v et t , autorise une analyse dans l'espace Temps(t)-Fréquence(v).

TRANSFORMATION DE HILBERT

La Transformation de Hilbert (TH) $\hat{x}(t)$ d'un signal $x(t)$ consiste en la convolution de $x(t)$ par la fonction $\frac{1}{\pi t}$:

$$x(t) \xrightarrow{TH} \hat{x}(t) = x(t) * \frac{1}{\pi t}$$

ONDELETTES

La plupart des signaux ne sont pas stationnaires, et c'est justement dans l'évolution de leurs caractéristiques (statistiques, fréquentielles, temporelles, spatiales) que réside l'essentiel de l'information qu'ils contiennent (ex. : signaux de parole, d'image). Avec l'analyse de Fourier (sauf la TFCT, Transformée de Fourier à Court Terme, qui consiste en des TF sur des fenêtres glissantes du signal), l'approche du signal est globale et toute notion de localisation temporelle (ou spatiale pour des images) disparaît dans l'espace de Fourier. La décomposition d'un signal en une somme d'ondelettes (et non plus en une somme d'ondes comme avec l'analyse de Fourier) apporte un contenu fréquentiel tout en préservant la localisation afin d'obtenir une représentation temps/fréquence ou espace/échelle du signal (ex. en musique : décomposition en ondelettes d'un signal non stationnaire, ayant un début et une fin, comme la frappe d'une touche au piano, contrairement à la décomposition en ondes d'un signal stationnaire, comme la vibration maintenue, prolongée d'un diapason, émise indéfiniment depuis l'origine des temps). La notion d'ondelette (ondelette de Gabor) intègre (au moins) 3 informations : un commencement, une fin et entre les deux, une fréquence précise.

Transformée de Fourier continue

Approche de Fourier : signal = somme d'ondes (= de fonctions sinus) de fréquence multiple et translattées

La transformation (décomposition) de Fourier continue $F(v)$ d'une fonction $f(t)$ s'écrit : $F(v) = \int_{-\infty}^{\infty} f(t)e^{-i2\pi vt} dt$

notée : $F(v) = \langle f, \psi \rangle$ où $\psi(t) = e^{-i2\pi vt}$ onde, et : $\langle f, g \rangle = \int_{-\infty}^{\infty} f(\tau)g(\tau)d\tau$ produit scalaire (pour fonctions réelles)

$$\langle f, g \rangle = \int_{-\infty}^{\infty} f(\tau)g^*(\tau)d\tau \text{ produit scalaire (pour fonctions complexes) } (g^* : \text{conjugué de } g)$$

La transformation (décomposition) de Fourier inverse (*reconstruction*) s'écrit : $f(t) = \int_{-\infty}^{\infty} F(v)e^{i2\pi vt} dv$

La Transformation de Fourier constitue un outil adapté à l'analyse de signaux stationnaires, qui peuvent se décrire globalement et par une combinaison linéaires d'ondes. Elle ne peut caractériser efficacement, localement, un signal non stationnaire, qui lui s'exprime comme combinaison linéaire d'ondelettes.

Transformée en ondelettes continue

Approche ondelettes : signal = somme d'ondelettes (= ondes évanescents, ondes localisées, ondes amorties) réglables en fréquence et translation

La transformation (décomposition) en ondelettes continue $F(a, b)$ d'une fonction $f(t)$ suivant l'ondelette $\psi(t)$ s'écrit :

$$F(a, b) = \int_{-\infty}^{+\infty} f(t)\psi_{a,b}(t) dt = \frac{1}{\sqrt{a}} \int_{-\infty}^{+\infty} f(t)\psi\left(\frac{t-b}{a}\right) dt \quad \text{notée} \quad F(a, b) = \langle f, \psi_{a,b} \rangle \quad (a, b) \in \mathbf{R}$$

Les ondelettes filles $\psi_{a,b}(t)$ s'obtiennent par dilatation et translation de l'ondelette mère $\psi(t)$ (a est le facteur d'échelle (dilatation) et b le paramètre de translation).

La décomposition d'un signal en ondelettes est ainsi paramétrable : les ondelettes filles sont, à l'aide d'une fonction d'échelle φ , réglables en dilatation et en translation de l'ondelette mère.

La variable a joue le rôle de l'inverse de la fréquence : plus a est petit moins l'ondelette est étendue temporellement, donc plus la fréquence centrale de son spectre est élevée.

On peut interpréter cette expression comme une projection du signal sur une famille de fonctions analysantes $\psi_{a,b}$

construite à partir d'une fonction mère ψ conformément à l'équation suivante :
$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}} \psi\left(\frac{t-b}{a}\right)$$

Les premières ondelettes utilisées ont été l'ondelette de Morlet, une gaussienne modulée par une exponentielle complexe, le « chapeau mexicain », dérivée seconde d'une gaussienne, et celle de Haar, définie par morceaux.

Inversion (reconstruction) - Admissibilité

Une ondelette est une onde localisée, évanescence. Sa forme n'est pas unique (il suffit qu'elle autorise la reconstruction) :

Reconstruction
$$f(t) = C_\psi^{-1} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \frac{1}{a^2} F(a,b) \psi_{a,b}(t) da db$$
 avec
$$C_\psi = 2\pi \int_{-\infty}^{\infty} |\Psi(v)|^2 \frac{dv}{v}$$

 où $\Psi(v)$ représente la TF de $\psi(t)$

La condition d'existence de C_ψ est la condition d'admissibilité d'une fonction comme ondelette :

Condition d'existence
$$\int_0^{\infty} |\Psi(v)|^2 \frac{dv}{|v|} = \int_{-\infty}^0 |\Psi(v)|^2 \frac{dv}{|v|} < \infty$$

Cette équation se ramène le plus souvent à la condition suivante (utilisation du théorème de Parseval), peu contraignante qui indique seulement que la fonction ondelette soit à moyenne nulle :

$$\int_{-\infty}^{+\infty} \psi(t) dt = 0$$

Le choix de l'ondelette est donc en principe très ouvert. Cependant, la robustesse et la vitesse de convergence de l'algorithme de reconstruction (ainsi que la compacité de la décomposition, caractérisée par la non redondance de l'information) sont très dépendants du choix de l'ondelette.

Transformée en ondelettes discrète $(m,n) \in \mathbf{N}$

$$F(m,n) = a_0^{-\frac{m}{2}} \int_{-\infty}^{+\infty} f(t) \psi(a_0^{-m}t - nb_0) dt = \langle f, \psi_{m,n} \rangle$$

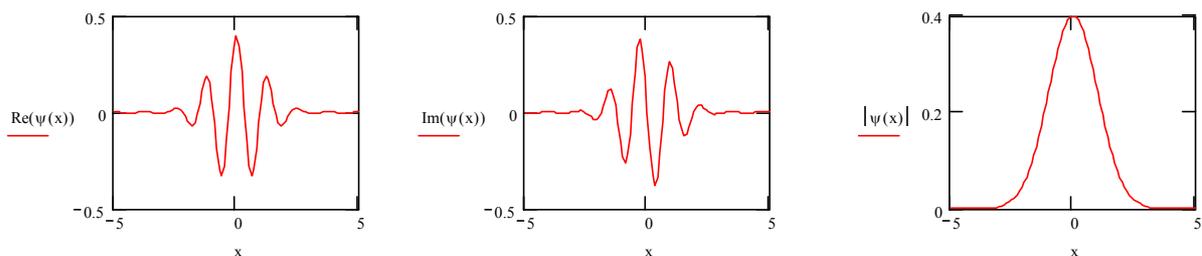
Si on choisit $a_0 = 2$ et $b_0 = 1$, on parle de **transformée dyadique** :
$$F(m,n) = 2^{-\frac{m}{2}} \int_{-\infty}^{+\infty} f(t) \psi(2^{-m}t - n) dt$$

Reconstruction
$$x(t) = \sum_m \sum_n F(m,n) \psi_{m,n}(t) = \sum_m x_m(t)$$
 avec
$$x_m(t) = \sum_n F(m,n) \psi_{m,n}(t)$$

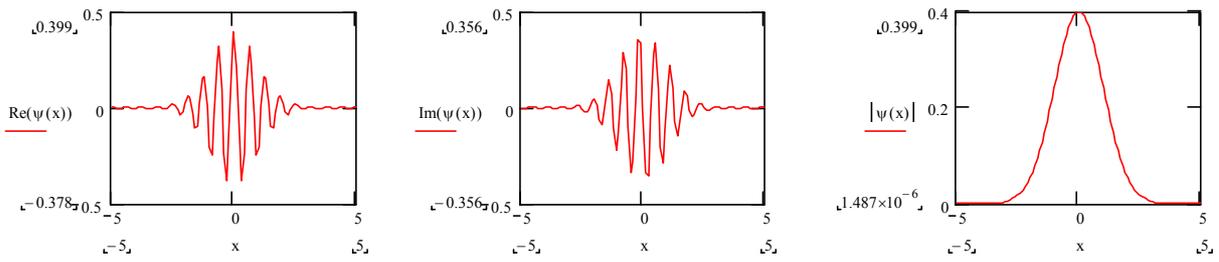
Types d'ondelettes : Ondelettes de Morlet

Ondelette mère
$$\psi(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} e^{-i\omega_0 x}$$

Facteur d'échelle : $\omega_0 = 5$



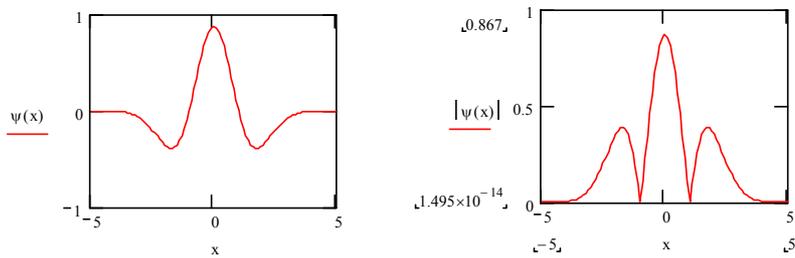
Facteur d'échelle : $\omega_0 = 10$



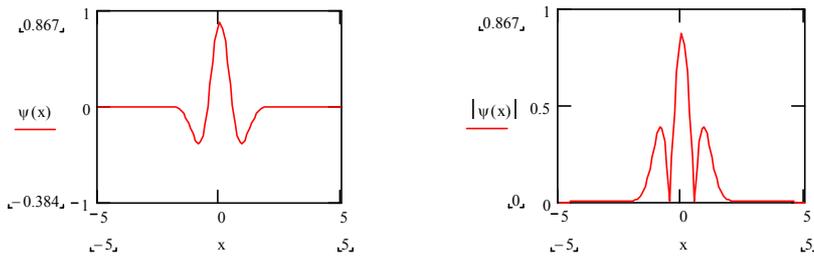
Types d'ondelettes : Chapeau mexicain

Ondelette mère $\psi(x) = \frac{2}{\sqrt{3}} \pi^{-1/4} (1-x^2) e^{-\frac{x^2}{2}}$ Avec le facteur d'échelle $a : \psi\left(\frac{x}{a}\right)$

Facteur d'échelle : $a = 1$



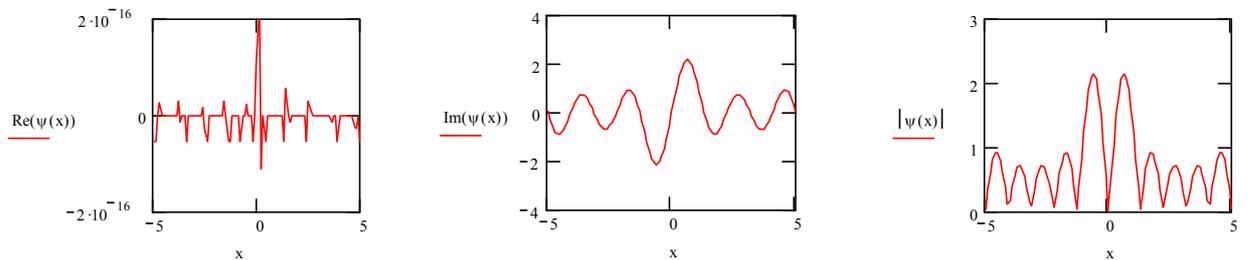
Facteur d'échelle : $a = 0.5$



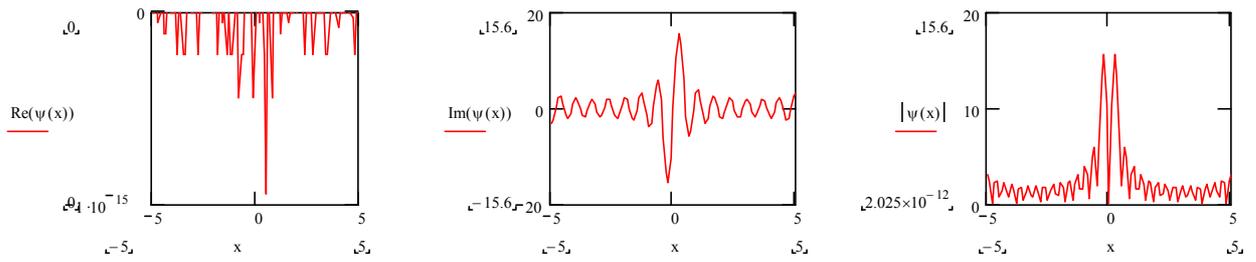
Types d'ondelettes : Ondelettes d'Ingrid Daubechies

Ondelette mère $\psi(x) = \sum_{k=-(2N-1)}^{2N-1} \frac{1}{5} k [\cos(kx) + i \sin(kx)]$

Facteur d'échelle : $N = 2$



Facteur d'échelle : $N = 5$

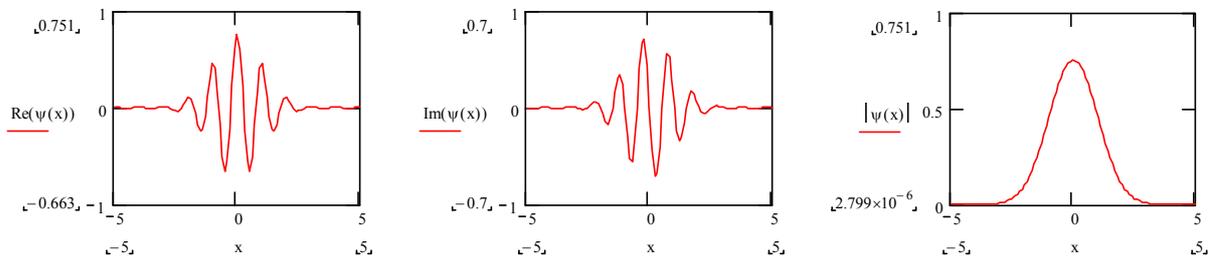


Types d'ondelettes : Ondelettes de Gabor

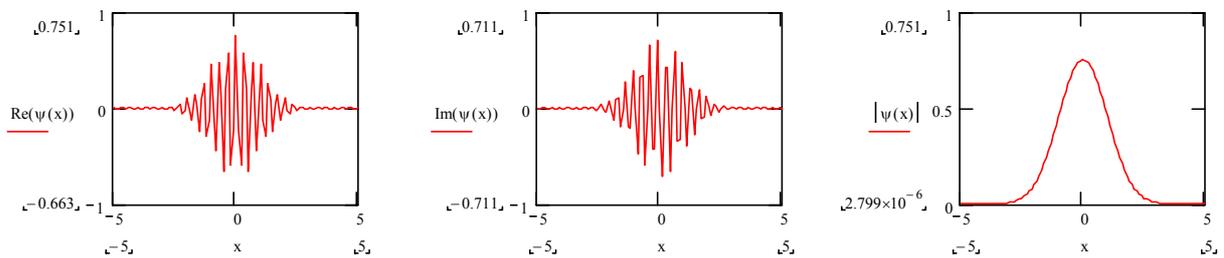
Transformée en ondelettes de Gabor
$$F(t, \nu) = \pi^{-\frac{1}{4}} \int_{-\infty}^{+\infty} f(\tau) e^{-\frac{(\tau-t)^2}{2}} e^{-i2\pi\nu\tau} d\tau$$

Ondelette mère (Gabor)
$$\psi(x) = \pi^{-\frac{1}{4}} e^{-\frac{x^2}{2}} e^{-i2\pi\nu x}$$

Facteur d'échelle : $\nu = 1$

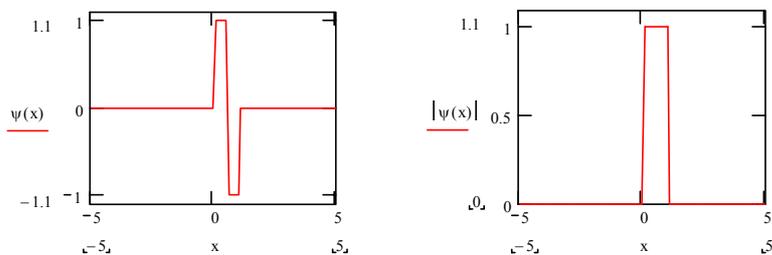


Facteur d'échelle : $\nu = 3$



Ondelettes de Haar

Ondelette mère
$$\psi(x) = \begin{cases} 0 & \text{si } x < 0 \\ 1 & \text{si } 0 \leq x < 1/2 \\ -1 & \text{si } 1/2 \leq x < 1 \\ 0 & \text{si } x \geq 1 \end{cases}$$



Algorithme de Mallat

L'algorithme pyramidal de Mallat réalise la décomposition d'un signal en une somme d'ondelettes ainsi que sa reconstruction, suivant un processus itératif rapide. Le signal doit avoir une longueur égale à une puissance de 2. La transformée en ondelettes a la même longueur que le signal.

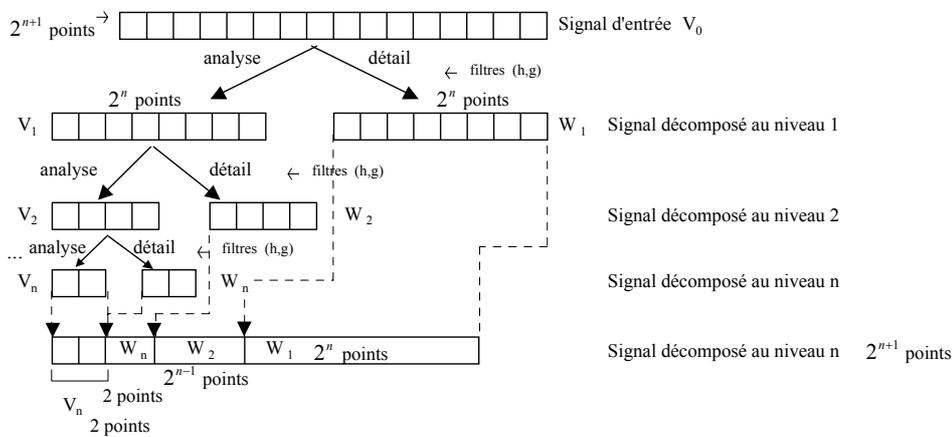
Le signal décomposé, est constitué [à chaque étape m de la décomposition (partitionnement) qui compte n niveaux] de 2 parties :

- . la partie **analyse** ou **approximation**, constituée des 2 lers coefficients du signal décomposé,
- . la partie **détail**, constituée du reste des coefficients du signal décomposé

La partie détail est elle-même formée de plusieurs séquences, représentant l'information à différentes échelles (analyse **mutirésolution**) : le résultat de la décomposition du signal est un partitionnement en une partie dite **analyse** (à 2 coefficients) et une partie dite **détail** (le reste des points) contenant la décomposition du signal en ondelettes à différentes résolutions.

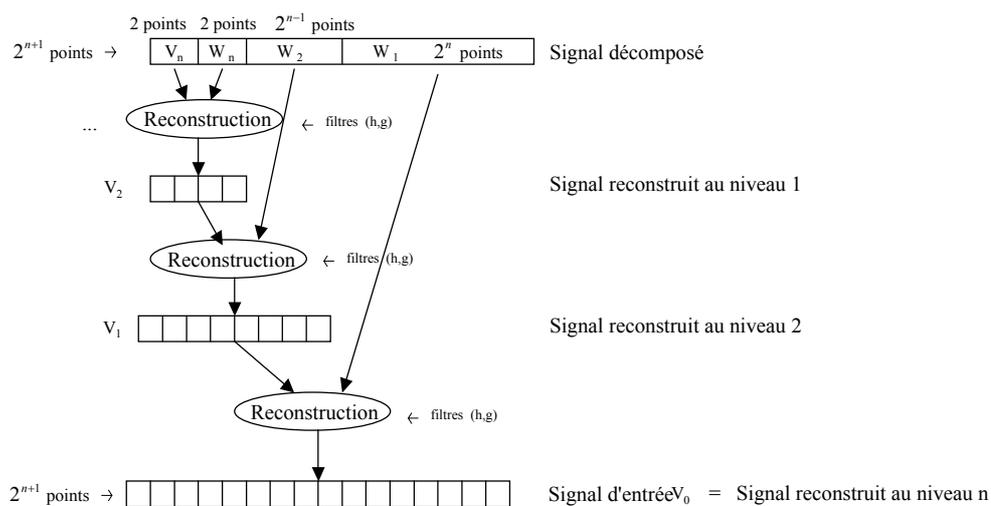
Analyse en ondelettes (wavelets)

Décomposition (analyse mutirésolution ondelettes)

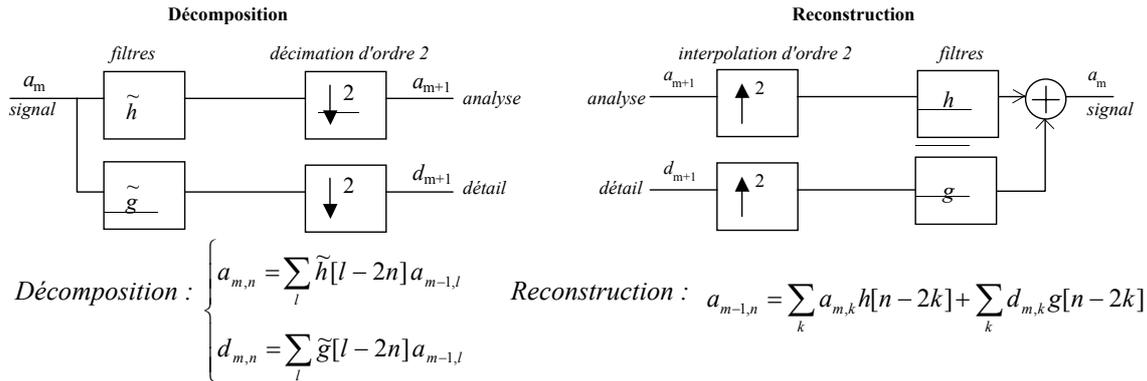


(n : nombre de niveaux de décomposition, de résolution)

Reconstruction (synthèse)



Les n étapes de décomposition et de reconstruction peuvent être vues comme un filtrage, de RI h et g :



m : indice du niveau de décomposition ; n, k, l : indices sur les signaux a, d, h, g . \tilde{h} représente h renversé dans le temps. La décimation d'ordre 2 consiste en la suppression (sous-échantillonnage) d'un coefficient sur deux. L'interpolation (ou sur-échantillonnage) d'ordre 2 s'effectue en intercalant un 0 entre chaque échantillon du signal.

RI des filtres de Décomposition/Reconstruction $\begin{cases} h[n-2k] = \langle \varphi_{m,k}, \varphi_{m-1,n} \rangle \text{ passe-bas} \\ g[n-2k] = \langle \psi_{m,k}, \varphi_{m-1,n} \rangle \text{ passe-haut} \end{cases}$. On a $g[n] = (-1)^n h[1-n]$

avec : ψ : ondelette mère

φ : fonction d'échelle, encore appelée ondelette père, et définie par :

$$\varphi_{m,n}(t) = 2^{-m/2} \varphi(2^{-m}t - n),$$

où $\varphi(t-n), n \in \mathbf{Z}$ est une base de V_0, V_j constituant une suite de sous-espaces emboîtés de $L^2(\mathbf{R})$,

ensemble des fonctions continues d'une variable réelle et de carré sommable.

Les filtres h et g dépendent donc des ondelettes utilisées.

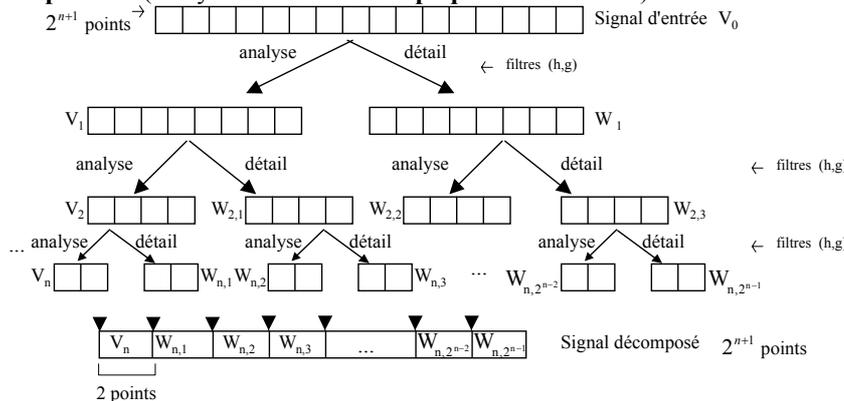
Exemple : • *Ondelettes de Haar* : $h[n] = \left\{ \dots, 0, \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}, 0, \dots \right\}$ où l'élément souligné correspond à l'indice $n = 0$

$$g[n] = \left\{ \dots, 0, \frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}, 0, \dots \right\}$$

• *Ondelettes de Daubechies* : $h[n]$ est tel que : $|h[n]|^2 + |h[n+1]|^2 = 1$

Analyse en paquet d'ondelettes (wavelets packets)

Décomposition (analyse mutirésolution paquet d'ondelettes)



(n : nombre de niveaux de décomposition, de résolution)

Domaines principaux d'application des ondelettes

- l'analyse vocale
- l'analyse de signaux radar
- la compression des signaux

Exemples de résultats d'une analyse en ondelettes

Transformée en ondelettes de Daubechies à 4 coefficients

La longueur du signal d'entrée doit être une puissance de 2, et >=4

ONDELETTES 1D

Définition de l'ondelette mère $G(x)$

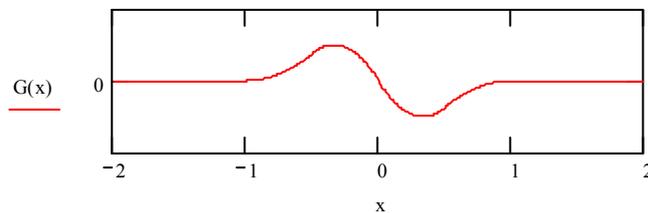
$$f1(x) := 3 \cdot (x + 1)^2$$

$$f2(x) := \frac{3}{4} + 3 \cdot \left(x + \frac{1}{2}\right) - 9 \cdot \left(x + \frac{1}{2}\right)^2$$

$$F(x) := \begin{cases} f1(x) & \text{if } x < -0.5 \\ f2(x) & \text{otherwise} \end{cases}$$

$$G(x) := \begin{cases} 0 & \text{if } |x| > 1 \\ \text{otherwise} \\ \begin{cases} F(x) & \text{if } x < 0 \\ -F(-x) & \text{otherwise} \end{cases} \end{cases}$$

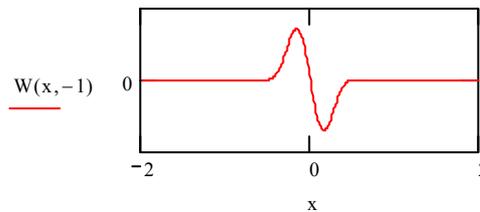
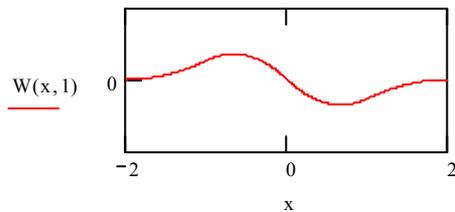
$$x := -2, -1.99..2$$



Définition des ondelettes filles $W(x,j)$ (dyadiques) obtenues à partir de l'ondelette mère $G(x)$

$$W(x, j) := \frac{1}{\sqrt{2^j}} \cdot G\left(\frac{x}{2^j}\right)$$

Quand j entier varie, les ondelettes $W(x,j)$ changent en échelle d'un facteur égal à une puissance de 2. Exemple :



Transformée en ondelettes $T(p,j,x)$ de la fonction p au point x à l'échelle 2^j

$$T(p, j, x) := \int_{x-2^j}^{x+2^j} p(t) \cdot W(t - x, j) dt$$

T est définie à travers tout l'axe réel, mais les ondelettes sont en temps limité.

La valeur de T pour un instant particulier x et une échelle j procure de l'information sur p sur un intervalle de temps de largeur 2^{j+1} centré en x .

Les transformées pour une échelle j large fournit de l'information pour des intervalles de temps larges.

Les transformées pour une échelle j négative analyse les caractères de p à une petite résolution

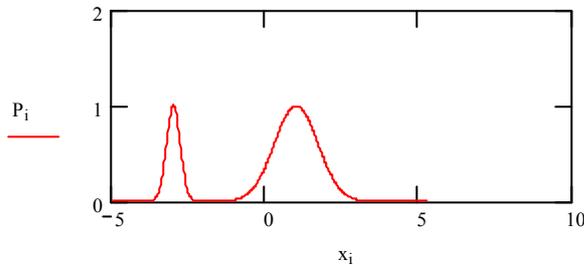
Exemple : le signal doit avoir une longueur égale à une puissance de 2 (algorithme de Mallat).

La transformée en ondelettes a la même longueur que le signal

Soit la fonction p :
$$p(x) := e^{-10 \cdot (x+3)^2} + e^{-(x-1)^2}$$

Echantillonnons p à 512 points autorisant la construction du vecteur signal P :

$$i := 0..511 \quad x_i := -5 + .02 \cdot i \quad P_i := p(x_i)$$



Transformée en ondelettes discrète : (par la fonction intégrée Mathcad :)

$$TP := \text{dwavelet}(P) \quad \text{length}(TP) = 512 \quad \text{length}(P) = 512$$

Les 2 1ers éléments de TP sont appelés les coefficients d'approximation.

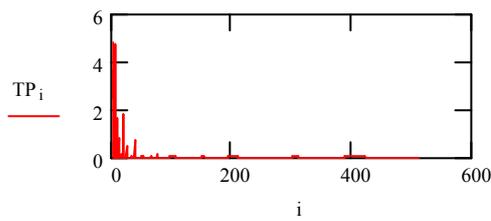
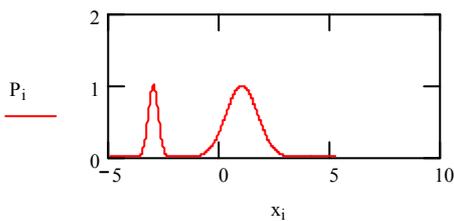
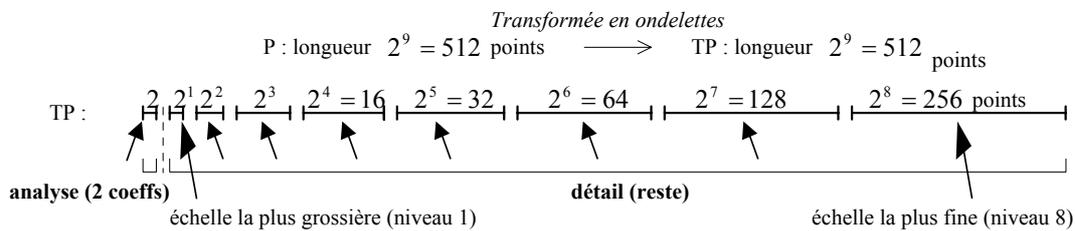
Les éléments restants sont les coefficients de détail.

La transformée TP contient 8 niveaux de détail :

Les 256 derniers points représentent les informations à la plus fine résolution, les 128

points précédents représentent une échelle 2 fois plus large, etc ...

Les éléments 2 and 3 donnent l'information à l'échelle la plus large.



Les coefficients de la transformée en ondelettes TP sont proches de 0 pour les résolutions les plus fines (fin du tableau de coeffs TP).

-> La compression du signal P par troncature de la transformée TP (annulation des derniers coefficients de la transformée, correspondant à la résolution la plus fine) peut être envisagée.

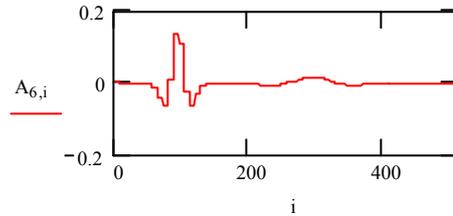
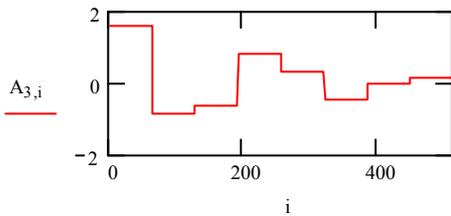
Pour chaque échelle, la position d'un élément à l'intérieur du niveau auquel il appartient coïncide avec la localisation dans le temps de la caractéristique à mesurer du signal.

Pour visualiser l'information, il est ainsi pratique d'étendre chaque niveau à la même longueur que le signal original P. Nous placerons les 8 niveaux étendus dans un tableau unique A :

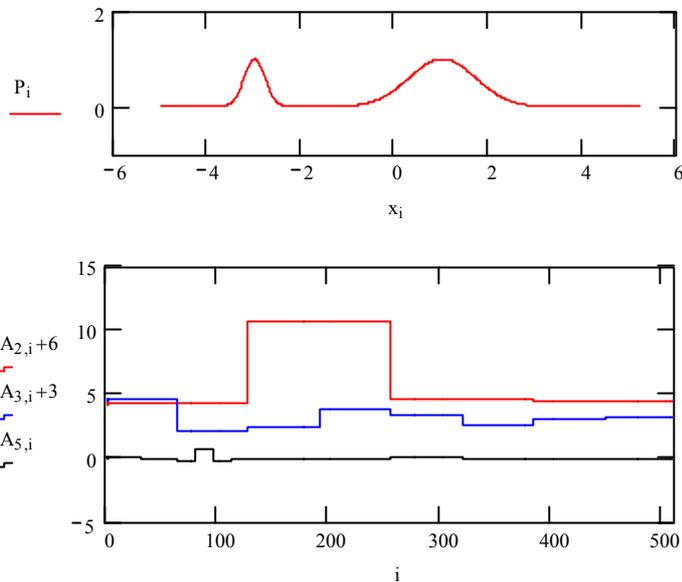
Niveau : $r := 1..8$ Information à l'échelle r : $A_{r,i} := TP \cdot 2^{r+\text{floor}\left(\frac{i}{2^{9-r}}\right)}$ (floor = partie entière)

Niveau 3, contenant 8 coefficients : $r := 3$

Niveau 6, contenant 64 coefficients : $r := 6$



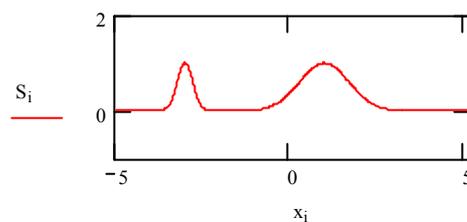
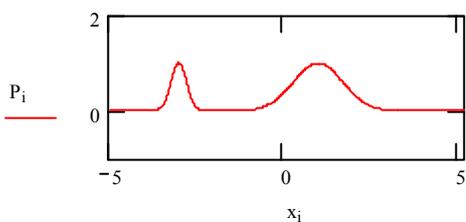
Pour voir la relation entre les niveaux et le signal original, affichons plusieurs niveaux de détail sur le même graphe, sous le tracé du signal original :



L'information dans TP autorise la reconstruction complète du signal P.

Pour reconstruire P, nous appliquons la transformation en ondelettes inverse :

$$p(t) := \sum_x \sum_j T(p, j, x) \cdot W(t - x, j) \quad \text{soit, par la fonction intégrée Mathcad : } S := \text{idwavelet}(TP)$$



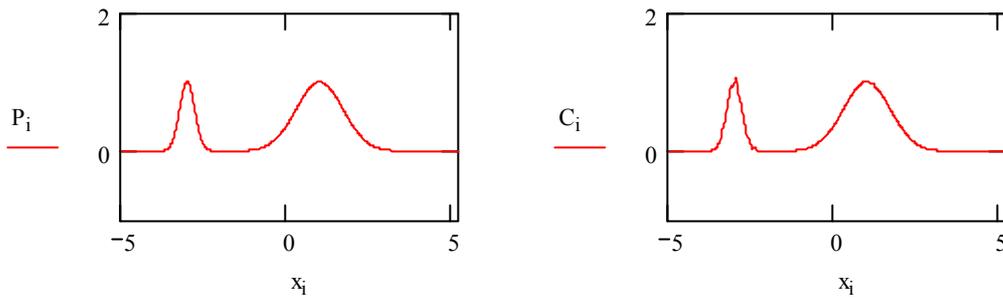
Le graphe montre que S coïncide avec P.

Remarquant que les coefficients à l'échelle la plus petite (fin du tableau de coeffs TP) sont bien inférieurs aux coefficients des résolutions supérieures, cela suggère que l'on puisse compresser le signal P par troncature de la transformée (= annulation des derniers coefficients de la transformée, correspondant à la résolution la plus fine).

Ignorons par exemple les 3 plus petits niveaux, et reconstruisons P à partir de la transformée tronquée, ne contenant que 64 éléments différents de 0.

Les 3 derniers niveaux sont constitués des derniers 256+128+64 = 448 éléments de TP.

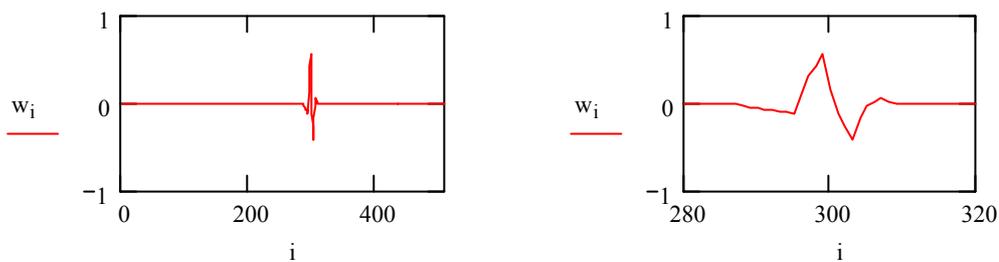
```
k := 64..511      TP_k := 0      C := idwavelet(TP)
```



Le graphe montre que C est une assez bonne réplique de P, bien que le fait d'avoir ignoré les détails les plus fins produise une image assez pauvre du pic étroit de P, qui est une caractéristique de détail assez fin de P (partie rapide de P occupant donc les Hautes Fréquences).

Pour visualiser les ondelettes elles-mêmes, on peut inverser une séquence transformée contenant seulement un coefficient non nul.

```
t := | for n ∈ 0..511      w := idwavelet(t)
      | t_n ← 0
      | t_100 ← 1
      | t
```



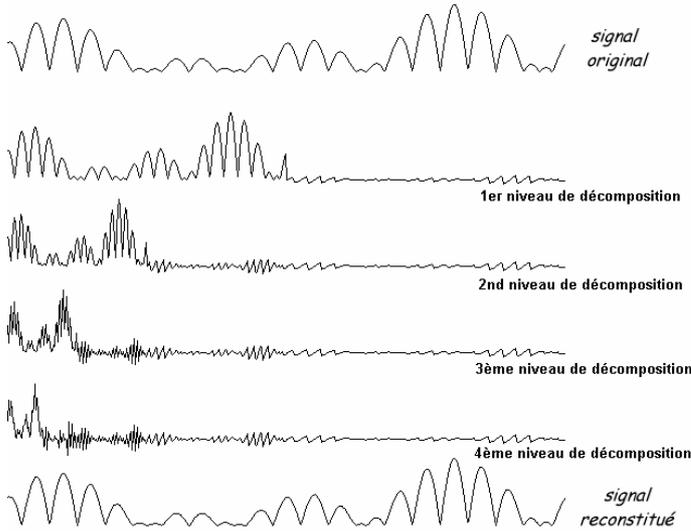
Le coefficient non nul de t se situe dans le niveau qui débute à l'élément 64, ainsi nous pouvons situer la localisation temporelle de l'ondelette démarrant à :

$$\frac{100 - 64}{64} \cdot 512 = 288$$

Application : compression de l'information (exemple avec les ondelettes de Haar)

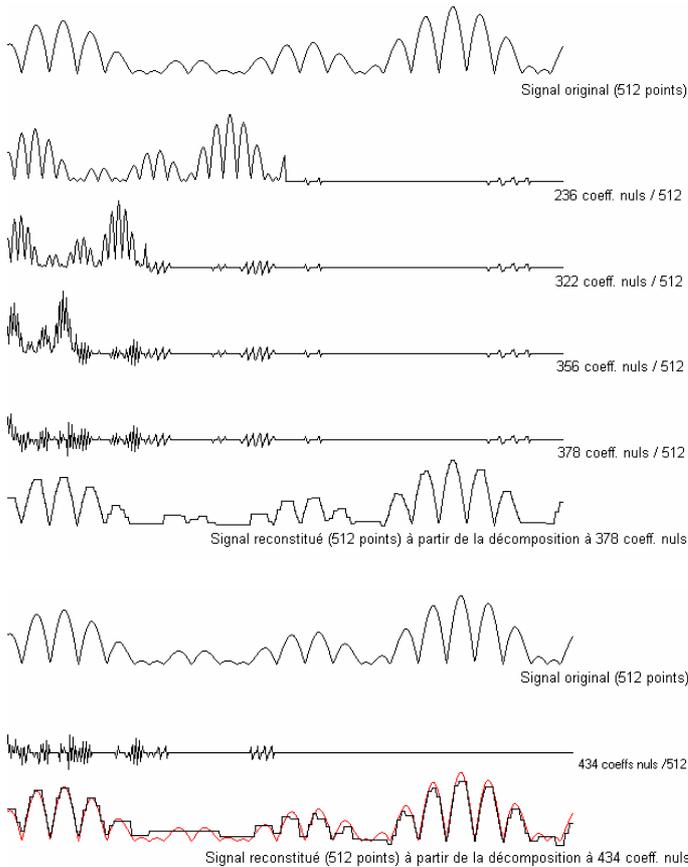
la transformation en ondelettes réalise une compression temporelle (spatiale dans le cas 2D) du signal

Décomposition du signal sans pertes



Décomposition du signal avec pertes (compression avec pertes)

Tout ou partie (coefficients les plus faibles en valeur absolue) du détail d'un ou plusieurs niveaux de décomposition est annulé



Compression & analyse multirésolution

- . à 1D : la compression consiste à supprimer les coefficients d'analyse les plus faibles en valeur absolue (filtre Donoho)
- . à 2D : la compression consiste à supprimer les niveaux de résolution les plus fins.

Ondelettes de Haar 1D & 2D - Compression

Transformée en ondelettes de Haar de niveau 1 (longueur du signal $\text{length}(v) \geq 4$)

```

analyse(v) :=
  h ←  $\begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix}$ 
  for i ∈ 0..rows(h) - 1
    hri ← h(rows(h)-i)-1
  for k ∈ 0..rows(v) - 1
    a0 ← v0·hr0 if k = 0
    ak ←  $\sum_{i=k-1}^k v_i \cdot hr_{k-i}$  otherwise
  for k ∈ 0.. $\frac{\text{rows}(a)}{2} - 1$ 
    ask ← a2·k+1
  as

  RI h (passe-bas) du filtre de Haar
  h =  $\begin{pmatrix} 0.707 \\ 0.707 \end{pmatrix}$ 
  h reverse      hr =  $\begin{pmatrix} 0.707 \\ 0.707 \end{pmatrix}$ 
  Coefficients d'analyse a :
  a = hr*v
  Décimation(sous-éch.) d'ordre 2 de a

detail(v) :=
  for i ∈ 0.. 1
    hi ←  $\frac{1}{\sqrt{2}}$ 
  for i ∈ 0.. rows(h) - 1
    gi ← h1-i·(-1)i
  for i ∈ 0.. rows(g) - 1
    gri ← g(rows(g)-i)-1
  for k ∈ 0.. rows(v) - 1
    d0 ← v0·gr0 if k = 0
    dk ←  $\sum_{i=k-1}^k v_i \cdot gr_{k-i}$  otherwise
  for k ∈ 0.. $\frac{\text{rows}(d)}{2} - 1$ 
    dsk ← d2·k+1
  ds

  RI h (passe-bas) du filtre de Haar
  h =  $\begin{pmatrix} 0.707 \\ 0.707 \end{pmatrix}$ 
  RI g (passe-haut) du filtre de Haar
  g =  $\begin{pmatrix} 0.707 \\ -0.707 \end{pmatrix}$ 
  g reverse      gr =  $\begin{pmatrix} -0.707 \\ 0.707 \end{pmatrix}$ 
  Coefficients de détail d :
  d = gr*v
  Décimation(sous-éch.) d'ordre 2 de d

waveHaar(v) :=
  a ← analyse(v)
  d ← detail(v)
  w ← stack(a,d)
  w
  analyse
  détail
  w = analyse | détail
  
```

Transformée en ondelettes inverse de Haar de niveau 1

```

reconstruction (a, d) :=
  h ←  $\begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix}$ 
  for i ∈ 0..rows(h) - 1
    gi ← h1-i · (-1)i
  for k ∈ 0..rows(a) - 1
    as2k ← ak
    as2·k+1 ← 0
  for k ∈ 0..rows(d) - 1
    ds2k ← dk
    ds2·k+1 ← 0
  for k ∈ 0..rows(as) - 1
    v10 ← as0 · h0 if k = 0
    v1k ←  $\sum_{i=k-1}^k as_i \cdot h_{k-i}$  otherwise
  for k ∈ 0..rows(ds) - 1
    v20 ← ds0 · g0 if k = 0
    v2k ←  $\sum_{i=k-1}^k ds_i \cdot g_{k-i}$  otherwise
  v ← v1 + v2
  v
  
```

Rl h (passe-bas)
 $h = \begin{pmatrix} 0.707 \\ 0.707 \end{pmatrix}$
Rl g (passe-haut)
 $g = \begin{pmatrix} 0.707 \\ -0.707 \end{pmatrix}$
Interpolation (sur-éch.)
d'ordre 2 de a & d pour
donner resp. as & ds

 $v1 = as \cdot h$

 $v2 = ds \cdot g$

 $v = as \cdot h + ds \cdot g$

```

iwaveHaar(w) :=
  a ← submatrix(w, 0,  $\frac{\text{rows}(w)}{2} - 1, 0, 0$ )
  d ← submatrix(w,  $\frac{\text{rows}(w)}{2}$ , rows(w) - 1, 0, 0)
  v ← reconstruction (a, d)
  v
  
```

Coefficients d'analyse a
Coefficients de détail d
Reconstruction

Transformée en ondelettes de Haar de niveau n ($length(v) = 2^m$; $1 \leq n \leq m$; $m \geq 1$)

```

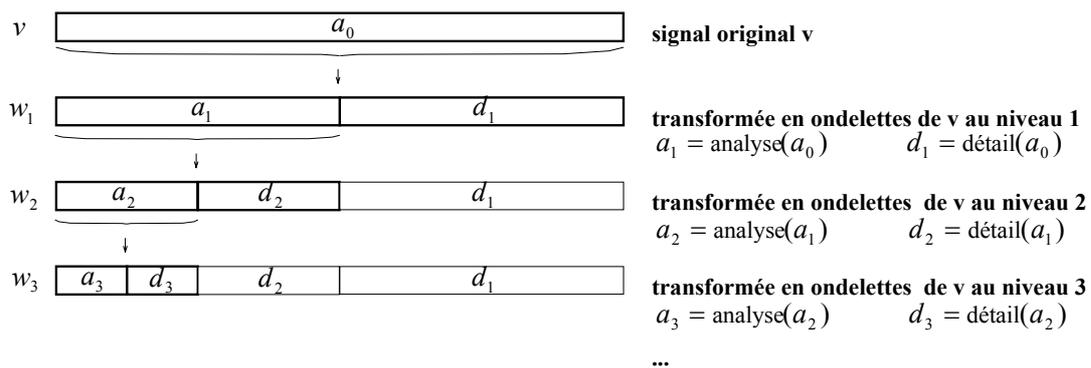
wavehaar(v,n) :=
  a0 ← v
  for i ∈ 1..n
    ai ← analyse(ai-1)
    di ← detail(ai-1)
  w ← dn
  for i ∈ n,n-1..2 if n > 1
    w ← stack(w,di-1)
  w ← stack(an,w)
  w
    
```

Transformée en ondelettes inverse de Haar de niveau n

```

iwavehaar(w,n) :=
  an ← submatrix(w,0,rows(w)/2n-1,0,0)
  for i ∈ n,n-1..1
    di ← submatrix(w,rows(w)/2i,rows(w)/2i-1-1,0,0)
  for i ∈ n-1,n-2..0
    ai ← reconstruction(ai+1,di+1)
  a0
    
```

Programmation de la décomposition en ondelettes de Haar 1D de niveau n



Test sur séquence

séquence

$$\begin{array}{ccc}
 \begin{array}{c} s := \\ \left(\begin{array}{c} 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array} \right) \end{array} & \begin{array}{c} \text{Décomposition au niveau 1} \\ w := \text{waveHaar}(s) \end{array} & \begin{array}{c} \left(\begin{array}{c} 0.707 \\ 3.536 \\ 6.364 \\ 9.192 \\ -0.707 \\ -0.707 \\ -0.707 \\ -0.707 \end{array} \right) \\ w = \end{array} & \begin{array}{c} \text{Reconstruction au niveau 1} \\ v := \text{iwaveHaar}(w) \end{array} & \begin{array}{c} \left(\begin{array}{c} 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array} \right) \\ v = \end{array}
 \end{array}$$

$$\begin{array}{ccc}
 & \begin{array}{c} \text{Décomposition au niveau 1} \\ w := \text{wavehaar}(s, 1) \end{array} & \begin{array}{c} \left(\begin{array}{c} 0.707 \\ 3.536 \\ 6.364 \\ 9.192 \\ -0.707 \\ -0.707 \\ -0.707 \\ -0.707 \end{array} \right) \\ w = \end{array} & \begin{array}{c} \text{Reconstruction au niveau 1} \\ v := \text{iwavehaar}(w, 1) \end{array} & \begin{array}{c} \left(\begin{array}{c} 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array} \right) \\ v = \end{array}
 \end{array}$$

$$\begin{array}{ccc}
 & \begin{array}{c} \text{Décomposition au niveau 2} \\ w := \text{wavehaar}(s, 2) \end{array} & \begin{array}{c} \left(\begin{array}{c} 3 \\ 11 \\ -2 \\ -2 \\ -0.707 \\ -0.707 \\ -0.707 \\ -0.707 \end{array} \right) \\ w = \end{array} & \begin{array}{c} \text{Reconstruction au niveau 2} \\ v := \text{iwavehaar}(w, 2) \end{array} & \begin{array}{c} \left(\begin{array}{c} 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array} \right) \\ v = \end{array}
 \end{array}$$

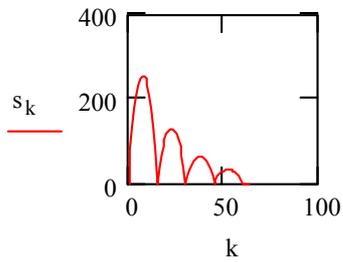
$$\begin{array}{ccc}
 & \begin{array}{c} \text{Décomposition au niveau 3} \\ w := \text{wavehaar}(s, 3) \end{array} & \begin{array}{c} \left(\begin{array}{c} 9.899 \\ -5.657 \\ -2 \\ -2 \\ -0.707 \\ -0.707 \\ -0.707 \\ -0.707 \end{array} \right) \\ w = \end{array} & \begin{array}{c} \text{Reconstruction au niveau 3} \\ v := \text{iwavehaar}(w, 3) \end{array} & \begin{array}{c} \left(\begin{array}{c} 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array} \right) \\ v = \end{array}
 \end{array}$$

Test sur signal 1D

on voit la compression temporelle du signal

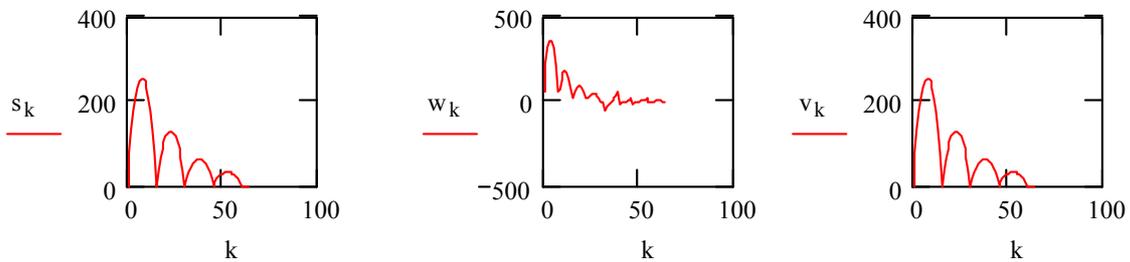
```
file := "s.txt"  s := READPRN(file)  k := 0..length(s) - 1  length(s) = 64
```

signal 1D



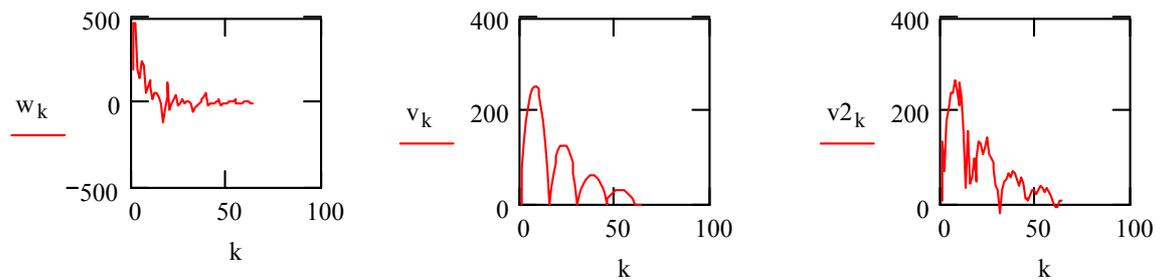
Décomposition au niveau 1 /Reconstruction au niveau 1

```
w := wavehaar(s, 1)  v := iwavehaar(w, 1)  length(w) = 64
```



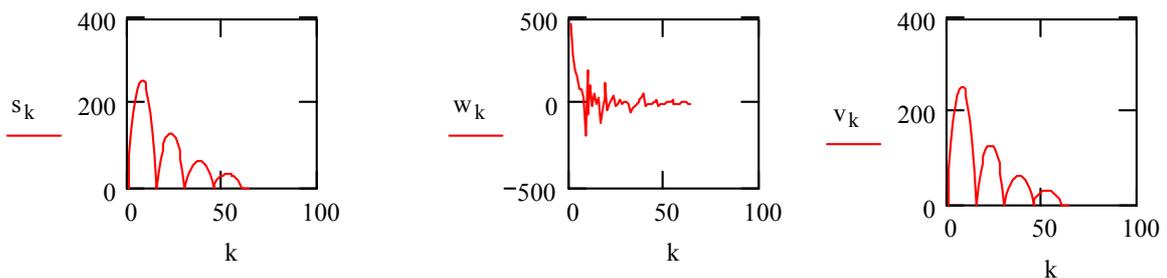
Décomposition au niveau 2 /Reconstruction au niveau 2

```
w := wavehaar(s, 2)  v := iwavehaar(w, 2)
v2 := iwavehaar(iwavehaar(w, 1), 1) faux mais rigolo car approximation du résultat !
```



Décomposition au niveau 3 /Reconstruction au niveau 3

```
w := wavehaar(s, 3)  v := iwavehaar(w, 3)
```



Test sur signal 2D

```
file := "bato2.bmp" I := READ_IMAGE(file)
```

signal 2D (image)



I

Transformée de Haar 1D en x de niveau n

$$\text{wavehaarx}(I, n) := \left| \begin{array}{l} \text{for } y \in 0.. \text{rows}(I) - 1 \\ \quad v \leftarrow (I^T)^{\langle y \rangle} \\ \quad w \leftarrow \text{wavehaar}(v, n) \\ \quad W^{\langle y \rangle} \leftarrow w \\ W^T \end{array} \right.$$

Transformée de Haar 1D en y de niveau n

$$\text{wavehaary}(I, n) := \left| \begin{array}{l} \text{for } x \in 0.. \text{cols}(I) - 1 \\ \quad v \leftarrow I^{\langle x \rangle} \\ \quad w \leftarrow \text{wavehaar}(v, n) \\ \quad W^{\langle x \rangle} \leftarrow w \\ W \end{array} \right.$$

Transformée Haar inverse 1D en x de niveau n

$$\text{iwavehaarx}(W, n) := \left| \begin{array}{l} \text{for } y \in 0.. \text{rows}(W) - 1 \\ \quad w \leftarrow (W^T)^{\langle y \rangle} \\ \quad v \leftarrow \text{iwavehaar}(w, n) \\ \quad I^{\langle y \rangle} \leftarrow v \\ I^T \end{array} \right.$$

Transformée Haar inverse 1D en y de niveau n

$$\text{iwavehaary}(W, n) := \left| \begin{array}{l} \text{for } x \in 0.. \text{cols}(W) - 1 \\ \quad w \leftarrow W^{\langle x \rangle} \\ \quad v \leftarrow \text{iwavehaar}(w, n) \\ \quad I^{\langle x \rangle} \leftarrow v \\ I \end{array} \right.$$

Transformée de Haar 2D de niveau n comme Transformées 1D en x et y (séparabilité)

$$\text{wavehaar2d}(I, n) := \left| \begin{array}{l} Wx \leftarrow \text{wavehaarx}(I, n) \\ W \leftarrow \text{wavehaary}(Wx, n) \\ W \end{array} \right. \equiv$$

$$\left| \begin{array}{l} Wy \leftarrow \text{wavehaary}(I, n) \\ W \leftarrow \text{wavehaarx}(Wy, n) \\ W \end{array} \right.$$

Transformée Haar inverse 2D de niveau n

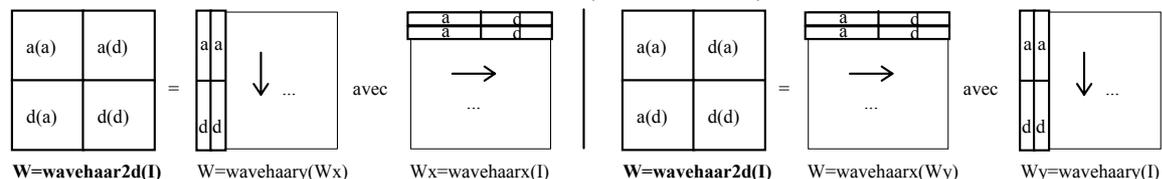
$$\text{iwavehaar2d}(W, n) := \left| \begin{array}{l} Ix \leftarrow \text{iwavehaarx}(W, n) \\ I \leftarrow \text{iwavehaary}(Ix, n) \\ I \end{array} \right. \equiv$$

ou encore (cela revient au même) :

$$\left| \begin{array}{l} Iy \leftarrow \text{iwavehaary}(W, n) \\ I \leftarrow \text{iwavehaarx}(Iy, n) \\ I \end{array} \right.$$

Décomposition en ondelettes de Haar 2D à partir de décompositions de Haar 1D en x wavehaarx() et en y wavehaary()

ex. niveau 1



Recadrage de dynamique (pour occuper exactement la plage 0 -> 255) d'une image

```

recdyn(I) :=
  min ← I0,0
  max ← I0,0
  for y ∈ 0..rows(I) - 1
    for x ∈ 0..cols(I) - 1
      max ← Iy,x if Iy,x > max
      min ← Iy,x if Iy,x < min
  for y ∈ 0..rows(I) - 1           if max ≠ min
    for x ∈ 0..cols(I) - 1
      Jy,x ← trunc( (Iy,x - min) / (max - min) * 255 )
  J ← I otherwise
  J
    
```

Transformée de Haar 1D en x et en y de niveau n

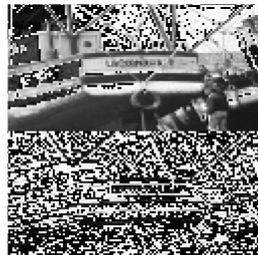
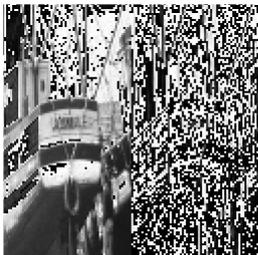
on voit la compression spatiale

Décomposition/Reconstruction au niveau 1 en x

Décomposition/Reconstruction au niveau 1 en y

Wx := wavehaarx(I, 1) Jx := iwavehaarx(Wx, 1)

Wy := wavehaary(I, 1) Jy := iwavehaary(Wy, 1)



Wx

Jx

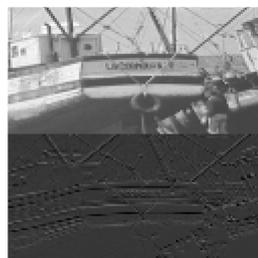
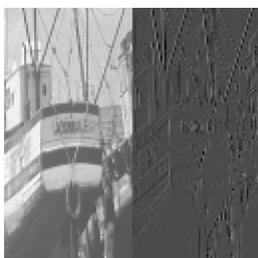
Wy

Jy

Recadrage de dynamique

Wx := recdyn(Wx)

Wy := recdyn(Wy)



Wx

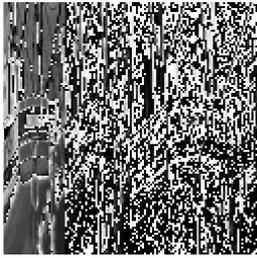
Jx

Wy

Jy

Décomposition/Reconstruction au niveau 2 en x

$W_x := \text{wavehaarx}(I, 2)$ $J_x := \text{iwavehaarx}(W_x, 2)$



W_x



J_x

Décomposition/Reconstruction au niveau 2 en y

$W_y := \text{wavehaary}(I, 2)$ $J_y := \text{iwavehaary}(W_y, 2)$



W_y

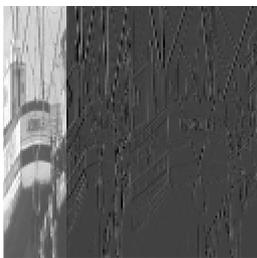


J_y

Recadrage de dynamique

$W_x := \text{recdyn}(W_x)$

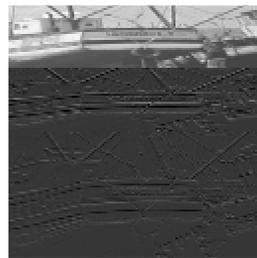
$W_y := \text{recdyn}(W_y)$



W_x



J_x



W_y



J_y

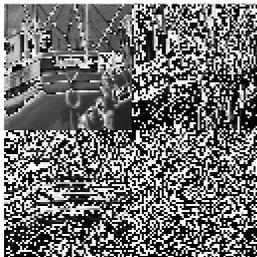
Transformée de Haar 2D de niveau n comme Transformées 1D en x et en y de niveau n

Décomposition/Reconstruction 2D au niveau 1

$W := \text{wavehaar2d}(I, 1)$ $J_x := \text{iwavehaar2d}(W, 1)$

Décomposition/Reconstruction 2D au niveau 2

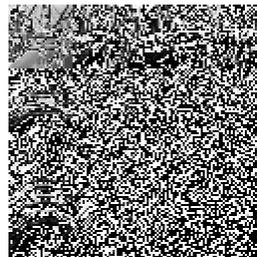
$W_2 := \text{wavehaar2d}(I, 2)$ $J_y := \text{iwavehaar2d}(W_2, 2)$



W



J_x



W_2

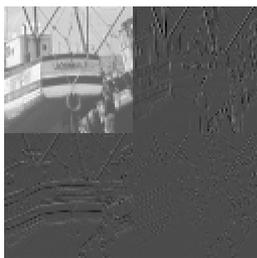


J_y

Recadrage de dynamique

$W := \text{recdyn}(W)$

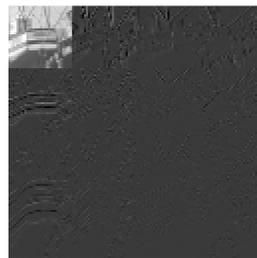
$W_2 := \text{recdyn}(W_2)$



W



J_x



W_2



J_y

Compression de signal (avec pertes)

effectuée ici sur le signal entier alors qu'en réalité on l'effectue à 1D par fenêtres (trames de 20 ms pour la parole) ou à 2D par blocs carrés de pixels (8x8 par exemple) (JPEG2000) ou par bandelettes (zones de pixels). (de plus les ondelettes de Haar ne soient pas très adaptées, contrairement à celles de Daubechies plus compactes)

Annulation des coefficients de détail de niveau n de la transformée de Haar 1D

$$\text{cutdetail}(w, n) := \left| \begin{array}{l} a_n \leftarrow \text{submatrix} \left(w, 0, \frac{\text{rows}(w)}{2^n} - 1, 0, 0 \right) \\ \text{for } i \in n, n - 1..1 \\ \quad d_i \leftarrow \text{submatrix} \left(w, \frac{\text{rows}(w)}{2^i}, \frac{\text{rows}(w)}{2^{i-1}} - 1, 0, 0 \right) \\ \quad \text{for } k \in 0.. \text{length}(d_i) - 1 \\ \quad \quad (d_i)_k \leftarrow 0 \\ \quad \text{for } i \in n - 1, n - 2..0 \\ \quad \quad a_i \leftarrow \text{stack}(a_{i+1}, d_{i+1}) \\ a_0 \end{array} \right.$$

Annulation des coefficients de détail de niveau n de la transformée de Haar 2D

$$\text{cutdetailx}(W, n) := \left| \begin{array}{l} \text{for } y \in 0.. \text{rows}(W) - 1 \\ \quad \left| \begin{array}{l} w \leftarrow (W^T)^{\langle y \rangle} \\ v \leftarrow \text{cutdetail}(w, n) \\ V^{\langle y \rangle} \leftarrow v \end{array} \right. \\ V^T \end{array} \right. \quad \text{cutdetaily}(W, n) := \left| \begin{array}{l} \text{for } x \in 0.. \text{cols}(W) - 1 \\ \quad \left| \begin{array}{l} w \leftarrow W^{\langle x \rangle} \\ v \leftarrow \text{cutdetail}(w, n) \\ V^{\langle x \rangle} \leftarrow v \end{array} \right. \\ V \end{array} \right.$$

$$\text{cutdetail2d}(W, n) := \left| \begin{array}{l} Wx \leftarrow \text{cutdetailx}(W, n) \\ W \leftarrow \text{cutdetaily}(Wx, n) \\ W \end{array} \right.$$

Rapport signal/bruit (overall et en dB) : indicateur de la qualité de la compression

$$\text{SNR}(x, u) := 10 \cdot \log \left[\frac{\sum_{n=0}^{\text{length}(x)-1} (x_n)^2}{\sum_{n=0}^{\text{length}(x)-1} (x_n - u_n)^2} \right]$$

Conversion 2D en 1D

$$\text{c2D1D}(I) := \left| \begin{array}{l} \text{for } y \in 0.. \text{rows}(I) - 1 \\ \quad \text{for } x \in 0.. \text{cols}(I) - 1 \\ \quad \quad \left| \begin{array}{l} n \leftarrow y \cdot \text{cols}(I) + x \\ i_n \leftarrow I_{y, x} \end{array} \right. \\ i \end{array} \right.$$

3.

Conversion 1D en 2D

$$\text{c1D2D}(i, Nx) := \left| \begin{array}{l} Ny \leftarrow \frac{\text{length}(i)}{Nx} \\ \text{for } y \in 0.. Ny - 1 \\ \quad \text{for } x \in 0.. Nx - 1 \\ \quad \quad \left| \begin{array}{l} n \leftarrow y \cdot Nx + x \\ I_{y, x} \leftarrow i_n \end{array} \right. \\ I \end{array} \right.$$

22

Compression de séquence

séquence

$$\begin{matrix}
 s := \begin{pmatrix} 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{pmatrix} & \begin{matrix} \text{Décomposition au niveau 1} \\ w := \text{wavehaar}(s, 1) \end{matrix} & w = \begin{pmatrix} 0.707 \\ 3.536 \\ 6.364 \\ 9.192 \\ -0.707 \\ -0.707 \\ -0.707 \\ -0.707 \end{pmatrix} & \begin{matrix} \text{Reconstruction au niveau 1} \\ v := \text{iwavehaar}(w, 1) \end{matrix} & v = \begin{pmatrix} 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{pmatrix}
 \end{matrix}$$

Annulation détail niveau 1 (taux compression élevé 50%)

Décomposition niveau 1 $w := \text{wavehaar}(s, 1)$

$$\begin{matrix}
 w = \begin{pmatrix} 0.707 \\ 3.536 \\ 6.364 \\ 9.192 \\ -0.707 \\ -0.707 \\ -0.707 \\ -0.707 \end{pmatrix} & \begin{matrix} \text{Compression niveau 1} \\ w := \text{cutdetail}(w, 1) \end{matrix} & w = \begin{pmatrix} 0.707 \\ 3.536 \\ 6.364 \\ 9.192 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} & \begin{matrix} \text{Reconstruction au niveau 1} \\ v := \text{iwavehaar}(w, 1) \\ \text{SNR}(s, v) = 18.451 \end{matrix} & v = \begin{pmatrix} 0.5 \\ 0.5 \\ 2.5 \\ 2.5 \\ 4.5 \\ 4.5 \\ 6.5 \\ 6.5 \end{pmatrix}
 \end{matrix}$$

Annulation détail niveau 2 (taux compression faible 25%)

Décomposition niveau 2 $w := \text{wavehaar}(s, 2)$

Avec un même taux de compression, on aurait une bien meilleure qualité en annulant la moitié des coefficients du détail de niveau 1 qui sont les plus faibles (en valeur absolue) !

$$\begin{matrix}
 w = \begin{pmatrix} 3 \\ 11 \\ -2 \\ -2 \\ -0.707 \\ -0.707 \\ -0.707 \\ -0.707 \end{pmatrix} & \begin{matrix} \text{Compression niveau 2} \\ w := \text{cutdetail}(w, 2) \end{matrix} & w = \begin{pmatrix} 3 \\ 11 \\ 0 \\ 0 \\ -0.707 \\ -0.707 \\ -0.707 \\ -0.707 \end{pmatrix} & \begin{matrix} \text{Reconstruction niveau 2} \\ v := \text{iwavehaar}(w, 2) \\ \text{SNR}(s, v) = 12.43 \end{matrix} & v = \begin{pmatrix} 1 \\ 2 \\ 1 \\ 2 \\ 5 \\ 6 \\ 5 \\ 6 \end{pmatrix}
 \end{matrix}$$

Annulation détail niveaux 1 & 2 (compression très élevé 75%)

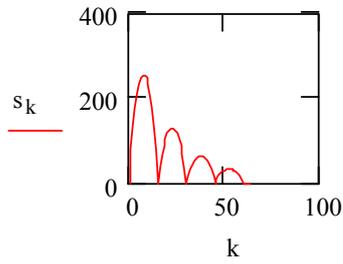
Décomposition niveau 2 $w := \text{wavehaar}(s, 2)$

$$\begin{matrix}
 w := \text{cutdetail}(w, 1) \quad w := \text{cutdetail}(w, 2) & \begin{matrix} \text{Compression niveaux 1 \& 2} \end{matrix} & w = \begin{pmatrix} 3 \\ 11 \\ -2 \\ -2 \\ -0.707 \\ -0.707 \\ -0.707 \\ -0.707 \end{pmatrix} & \begin{matrix} \text{Reconstruction au niveau 1} \\ v := \text{iwavehaar}(w, 2) \\ \text{SNR}(s, v) = 11.461 \end{matrix} & v = \begin{pmatrix} 1 \\ 2 \\ 1 \\ 2 \\ 5 \\ 6 \\ 5 \\ 6 \end{pmatrix}
 \end{matrix}$$

Compression signal 1D

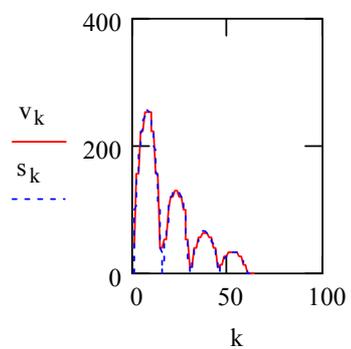
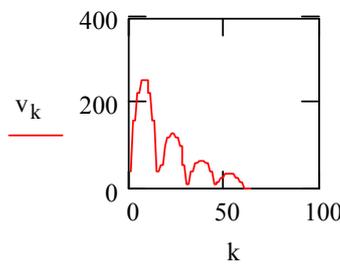
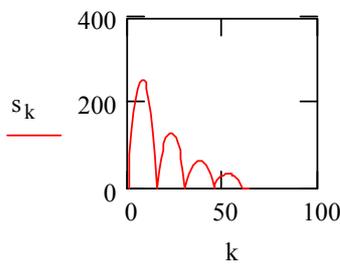
file := "s.txt" s := READPRN(file) k := 0..length(s) - 1 length(s) = 64

signal 1D



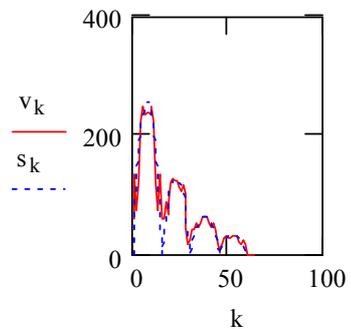
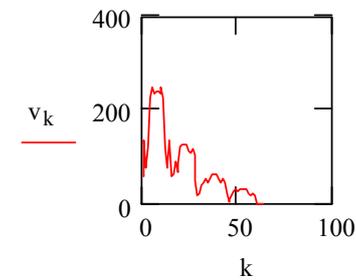
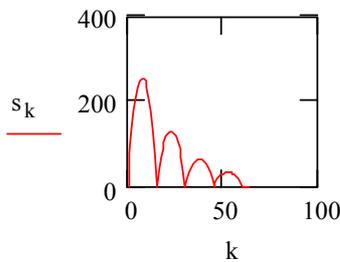
Décomposition niveau 1 / Compression niveau 1 / Reconstruction niveau 1

w := wavehaar(s, 1) w := cutdetail(w, 1) v := iwavehaar(w, 1) SNR(s, v) = 18.672



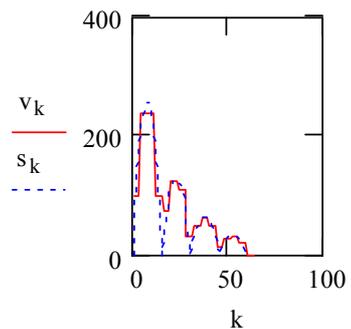
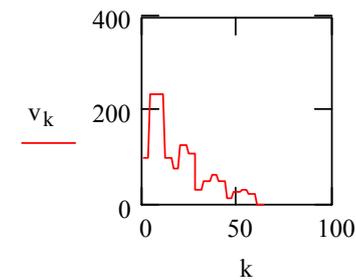
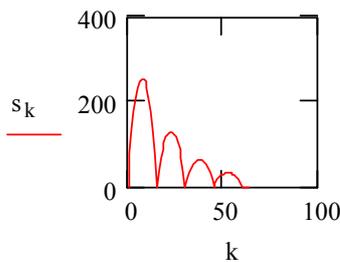
Décomposition niveau 2 / Compression niveau 2 / Reconstruction niveau 2

w := wavehaar(s, 2) w := cutdetail(w, 2) v := iwavehaar(w, 2) SNR(s, v) = 13.296



Décomposition niveau 2 / Compression niveaux 1 & 2 / Reconstruction niveau 2

w := wavehaar(s, 2) w := cutdetail(w, 1) w := cutdetail(w, 2) v := iwavehaar(w, 2) SNR(s, v) = 12.19



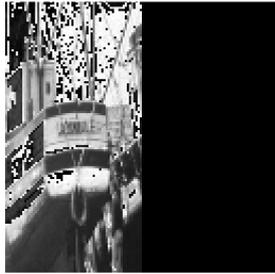
Compression signal 2D file := "bato2.bmp" I := READ_IMAGE(file)



Transformée Haar 1D en x et en y niveau n

Décomposition niveau 1 / Compression niveau 1 / Reconstruction niveau 1 (Taux 50%)

Wx := wavehaarx(I, 1) Wx := cutdetailx(Wx, 1) Jx := iwavehaarx(Wx, 1)
 Wy := wavehaary(I, 1) Wy := cutdetaily(Wy, 1) Jy := iwavehaary(Wy, 1)



Wx



Jx



Wy



Jy

Conversion 2D en 1D et Transformée 1D

Nx := cols(I)

2D->1D / Décomposition niveau 1 / Compression niveau 1 / Reconstruction niveau 1 / 1D->2D (Taux 50%) : c'est pareil !

i := c2D1D(I) w := wavehaar(i, 1) w := cutdetail(w, 1) v := iwavehaar(w, 1) J := c1D2D(v, Nx)



I



J

Transformée de Haar 2D de niveau n comme Transformées 1D en x et en y de niveau n

Décomp. niv. 1 / Compr. niv. 1 / Reconstr. niv. 1 (Taux 75%)

Décomp. niv. 2 / Compr. niv. 1 & 2 / Reconstr. niv. 2 (Taux 87.5%)

W := wavehaar2d(I, 1)

W2 := wavehaar2d(I, 2) W2 := cutdetail2d(W2, 1)

W := cutdetail2d(W, 1) J := iwavehaar2d(W, 1)

W2 := cutdetail2d(W2, 2) J2 := iwavehaar2d(W2, 2)



W



J



W2

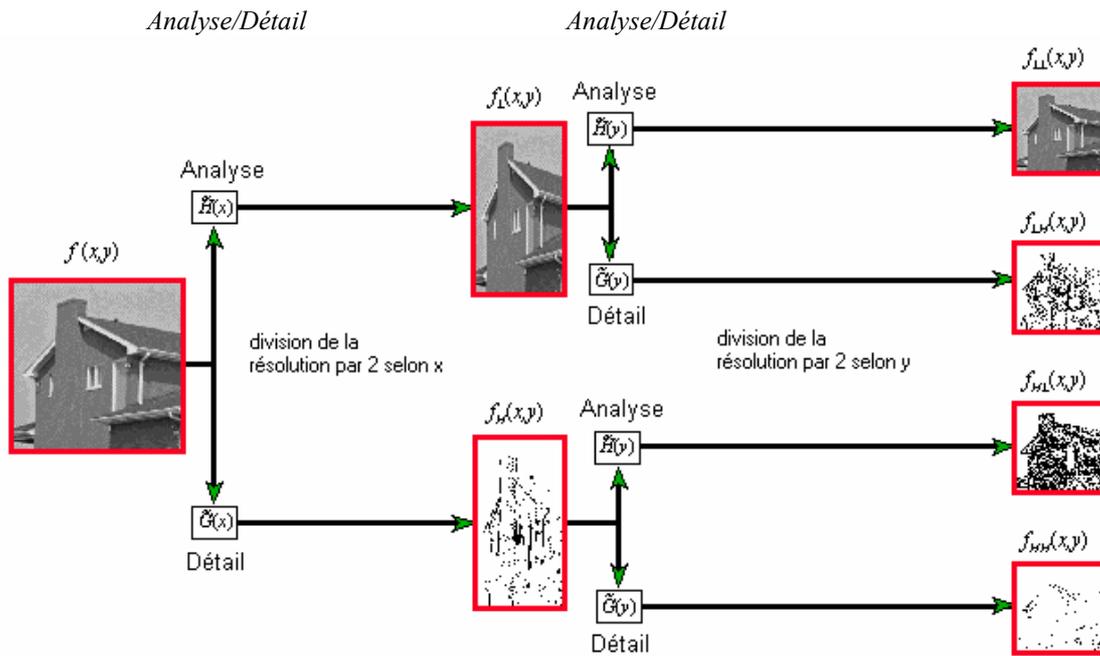


J2

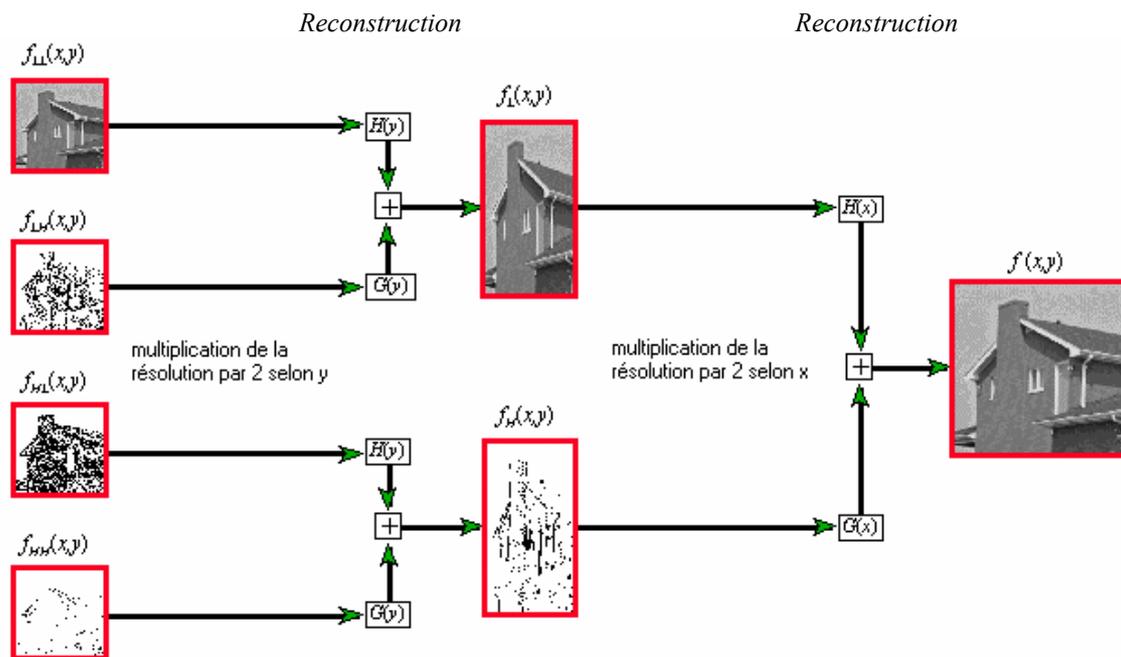
Décomposition en ondelettes 2D (images)



Décomposition (1^{er} niveau : obtention des 4 quadrants)



Reconstruction (1^{er} niveau : à partir des 4 quadrants)



II. ANALYSE CEPSTRALE

outil de caractérisation temporel

Le Cepstre complexe de la séquence $x(n)$ est la séquence $c_x(n)$ ainsi définie : $c_x(n) = TZ^{-1}[\ln(X(z))]$

avec : $X(z) = TZ[x(n)] = \sum_{n=-\infty}^{\infty} x(n)z^{-n}$

La séquence des coefficients cepstraux $c_x(n)$ d'un signal $x(n)$ fournit une analyse temporelle du signal $x(n)$.

La TZ $C_x(z)$ du cepstre $c_x(n)$ est donnée par (si $C_x(z)$ converge) : $C_x(z) = \ln(X(z)) = \sum_{n=-\infty}^{\infty} c_x(n)z^{-n}$

En régime harmonique ($z = e^{i\omega} = e^{i2\pi f}$), on a, en exprimant $X(f) = |X(f)| \cdot e^{i\varphi(f)}$ avec $\varphi(f) = Arg[X(f)]$:

$C_x(f) = \ln(X(f)) = \ln(|X(f)| \cdot e^{i\varphi(f)}) = \ln|X(f)| + i\varphi(f)$ **Cepstre fréquentiel** $TF[c_x(n)] = C_x(f)$

En transformant, à l'aide du logarithme, un produit en somme, la représentation de $C_x(f)$ des signaux à Temps Discret s'apparente au diagramme de Bode des signaux à Temps Continu.

Le Cepstre $c_x(n)$ peut donc également être obtenu par Transformation de Fourier inverse de $C_x(f)$:

$$c_x(n) = TF^{-1}[C_x(f)] = TF^{-1}[\ln|X(f)|] + i \cdot TF^{-1}[\varphi(f)] = \int_{-\pi}^{\pi} \ln|X(f)| e^{i2\pi nf} df + i \cdot \int_{-\pi}^{\pi} \varphi(f) e^{i2\pi nf} df$$

Pour un signal $x(n)$ réel, les propriétés de symétrie de la TF donnent le résultat simplifié :

$$c_x(n) = 2 \left\{ \int_0^{\pi} \ln|X(f)| \cos(2\pi nf) df - \int_0^{\pi} \varphi(f) \sin(2\pi nf) df \right\}$$

Remarque

Dans certaines applications comme le traitement de la parole, on ne calcule qu'une partie de Cepstre qui correspond au module de $X(f)$:

$$c_x(n) = 2 \int_0^{\pi} \ln|X(f)| \cos(2\pi nf) df$$

Cepstre de la réponse d'un filtre linéaire

Soit un filtre numérique linéaire de FT $H(z)$ et d'entrée/sortie respectives $x(n)$ et $y(n)$.

Le Cepstre $c_y(n)$ de la sortie s'obtient à partir de celui ($c_x(n)$) de l'entrée et de celui ($c_h(n)$) de la RI $h(n)$ du filtre :

$$c_y(n) = TZ^{-1}[\ln(Y(z))] = TZ^{-1}[\ln(H(z)X(z))] = TZ^{-1}[\ln(H(z)) + \ln(X(z))] = TZ^{-1}[\ln(H(z))] + TZ^{-1}[\ln(X(z))]$$

soit : $c_y(n) = c_h(n) + c_x(n)$

Application : Traitement de la parole

Si l'on prend l'exemple d'une application en traitement de la parole, un modèle de production de la parole consiste dans la relation suivante :

$$f(nT) = h(nT) * e(nT)$$

- où :
- . $f(nT)$ est le signal vocal échantillonné à la période T ,
 - . $h(nT)$, la Réponse Impulsionnelle du conduit vocal (caractérisant ce dernier dans un laps de temps) et
 - . $e(nT)$, le signal d'excitation permettant de synthétiser le signal vocal.

La méthode de calcul des coefficients cepstraux conduit à effectuer dans la fenêtre temporelle d'analyse une déconvolution de la relation précédente afin de séparer les contributions du conduit vocal et de l'excitation.

L'analyse cepstrale procède de la façon suivante pour transformer le produit de convolution en addition :

- Pour transformer le produit de convolution de la relation précédente en produit simple, elle opère une TFD

de $f(nT)$, soit :

$$F(\omega_k) = \sum_{n=0}^{N-1} f(nT) e^{-i\omega_k nT/N} = \sum_{n=0}^{N-1} f(nT) e^{-i\omega_n kT/N}$$

avec : N le nombre de points de la TFD, $\omega = \frac{2\pi}{N}$, $\omega_n = n\omega$, $\omega_k = k\omega$.

(On note indifféremment la TFD de $f(nT)$ par : $F(k)$, $F(kT)$, $F(\omega_k)$, $F(\omega_k T)$, $F(e^{\omega_k})$...)

On a, en notant $H(\omega_k)$ et $E(\omega_k)$, les TFs respectives de $h(nT)$ et $e(nT)$:

$$F(\omega_k) = H(\omega_k)E(\omega_k)$$

- Elle applique le logarithme à la relation précédente, pour transformer le produit en addition :

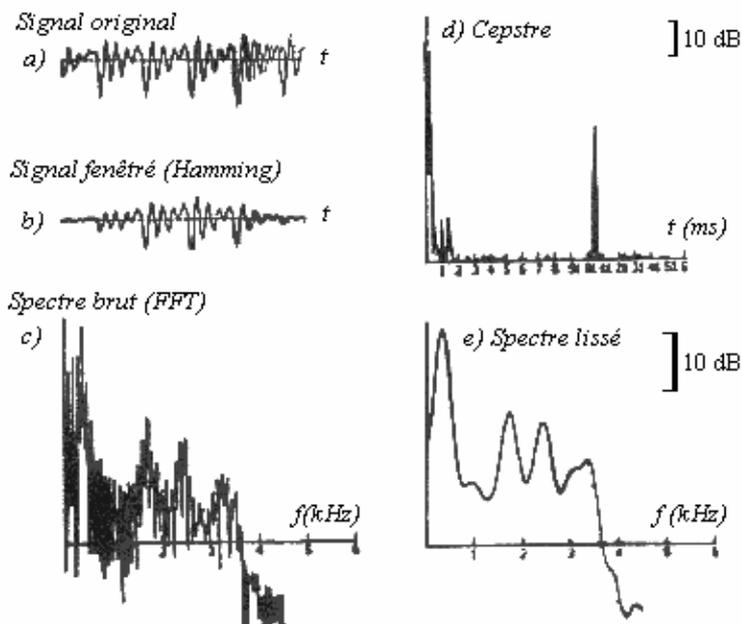
$$\log|F(\omega_k)| = \log|H(\omega_k)| + \log|E(\omega_k)|$$

- Elle revient au domaine temporel pour donner le cepstre caractérisant la signal $f(nT)$ défini par :

$$TF^{-1}[\log|F(\omega_k)|]$$

L'analyse cepstrale est une analyse temporelle (le cepstre bien qu'anagramme du mot « spectre », n'est pas une sorte de spectre). Elle donne les **coefficients cepstraux**.

Coefficients cepstraux



Les différentes étapes d'une analyse cepstrale

III. ANALYSE LPC (Codage Prédiction Linéaire)

outil de caractérisation temporel

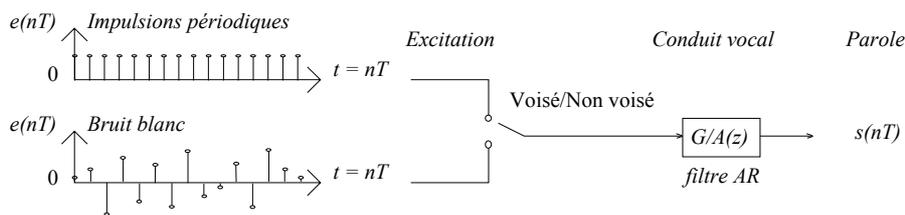
(Voir chapitre sur la prédiction pour compléments)

L'analyse LPC est une analyse temporelle, mais de nombreux paramètres peuvent être établis dans le domaine fréquentiel. On l'a déjà rencontrée avec les filtres formeurs AR.

L'analyse LPC s'appuie sur les coefficients a_i du filtre formeur AR d'ordre N qui, excité par un bruit blanc de variance V à déterminer, produit le signal considéré, à étudier. La séquence $\{a_i\}$ est donc le résultat du codage LPC et sert de base à l'analyse LPC.

Détermination des coefficients a_i du filtre formeur AR :

Dans une application de traitement de la parole par exemple, la méthode LPC calcule des coefficients sur un échantillon de parole par prédiction à partir d'une pondération linéaire d'un nombre fini d'échantillons le précédant. Cette méthode utilise le modèle suivant de production de la parole : (T : période d'échantillonnage)



Le conduit vocal, qui filtre le signal d'excitation $e(nT)$, peut être assimilé à un filtre récursif. Avec une bonne approximation, le signal $s(nT)$ qu'il émet à un instant peut être calculé à partir des valeurs qu'il a prises aux m instants précédents.

A l'instant nT , le signal $s(nT)$ peut donc s'exprimer comme issu d'une combinaison linéaire de son passé aux m instants précédents, et d'un terme d'erreur $e(nT)$ à cet instant (représenté par le signal d'excitation du modèle de production de parole) :

$$s(nT) = \sum_{i=1}^m a'_i s[(n-i)T] + Ge(nT)$$

G : gain de réglage

$Ge(nT)$: erreur de prédiction

$\sum_{i=1}^m a'_i s[(n-i)T]$: partie prédite

$a'_i \equiv -a_i \quad \forall i \in [1 .. m]$: coefficients de prédiction

Le modèle de production de la parole (*filtre formeur AR*) s'écrit :

$$S(z) = \frac{G}{A(z)} E(z)$$

G : gain de réglage

$G/A(z)$: FT du conduit vocal

$E(z)$: TZ de l'excitation

$S(z)$: TZ de la sortie du conduit vocal (signal de parole)

Modèle : $A(z) = \sum_{i=0}^m a_i z^{-i}$ d'où : $GE(z) = S(z)A(z) = S(z) \sum_{i=0}^m a_i z^{-i}$ soit dans le domaine temporel :

$$Ge(nT) = \sum_{i=0}^m a_i s[(n-i)T] \quad \text{soit, en fixant } a_0 = 1 : \quad Ge(nT) = s(nT) - \left(- \sum_{i=1}^m a_i s[(n-i)T] \right)$$

Le terme $Ge(nT)$ est l'erreur de prédiction, erreur entre l'échantillon de parole réel et l'échantillon prédit.

La prédiction linéaire de $s(nT)$ est donnée par :

$$s(nT) = - \sum_{i=1}^m a_i s[(n-i)T]$$

Pour déterminer les coefficients a_i , on peut minimiser l'énergie W de l'erreur sur l'intervalle de temps $[pT, rT]$:

$$W = \sum_{n=p}^r [Ge(nT)]^2$$

On a :
$$W = \sum_{n=p}^r \left[\sum_{i=0}^m a_i s[(n-i)T] \right]^2 = \sum_{n=p}^r \left[\sum_{i=0}^m a_i s[(n-i)T] \sum_{j=0}^m a_j s[(n-j)T] \right]$$

soit :
$$W = \sum_{n=p}^r \sum_{i=0}^m \sum_{j=0}^m a_i a_j s[(n-i)T] s[(n-j)T] = \sum_{i=0}^m \sum_{j=0}^m a_i a_j \sum_{n=p}^r s[(n-i)T] s[(n-j)T]$$

et :
$$W = \sum_{i=0}^m \sum_{j=0}^m a_i a_j C_{ij} \quad \text{avec :} \quad C_{ij} = \sum_{n=p}^r s[(n-i)T] s[(n-j)T]$$

Pour minimiser l'énergie W de l'erreur, on peut utiliser la méthode des moindres carrés :

En écrivant que la dérivée partielle de W par rapport à a_k est nulle, on a l'équation suivante : $\sum_{i=0}^m a_i C_{ik} + \sum_{j=0}^m a_j C_{kj} = 0$

Pour la résolution, l'intervalle de sommation $[pT, rT]$ peut être défini de 2 façons selon que l'on utilise l'une ou l'autre des 2 méthodes suivantes :

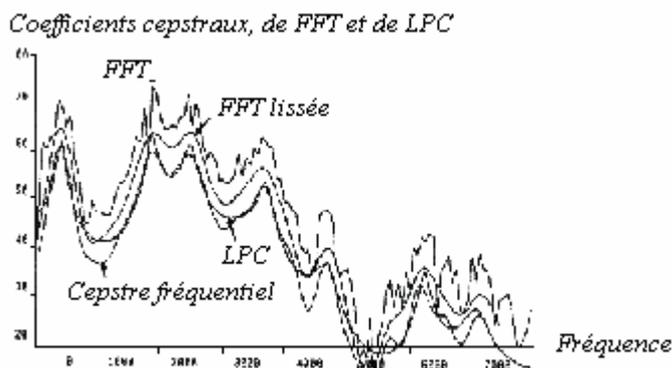
- *Méthode d'autocorrélation* : l'intervalle de temps $[pT, rT]$ va de $-\infty$ à $+\infty$ et on considère le signal nul hors de l'intervalle d'échantillons $[0, N]$ où N est le nombre de points considérés pour le signal de parole.
- *Méthode de covariance* : l'intervalle de temps $[pT, rT]$ est limité à $[0, (N-1)T]$.

Les coefficients de prédiction LPC a_k conduisent à une estimation du spectre de parole une fois l'influence de l'excitation éliminée. Ce sont les coefficients du filtre formeur AR dont l'entrée est un bruit blanc de variance V .

(cf. § 2.2.3. « Estimation de la DSP à partir du modèle AR du signal » du présent chapitre : $\text{spectre} = \frac{V}{\left| 1 + \sum_{n=1}^N a_n e^{-jn\omega} \right|^2} = \phi_{ss}(\omega)$)

L'analyse LPC présente principalement 2 avantages :

- Elle est adaptée à l'étude de phénomènes évoluant rapidement au contraire des méthodes fréquentielles qui doivent être effectuées sur une durée suffisamment longue si l'on veut que le spectre soit connu avec une précision convenable.
- La prédiction permet d'éliminer une part importante de la redondance du signal. La redondance est caractérisée par l'arrivée d'échantillons ne fournissant pas d'informations nouvelles; prédire la valeur du signal en fonction de ses valeurs passées permet de débarrasser le signal de ses pseudo-informations. Cette redondance n'est qu'éphémère (locale) et les coefficients de prédiction linéaire doivent être réajustés régulièrement.



LPC \equiv FFT du signal généré par le filtre formeur AR de coefficients a_k

Exemple de filtrage LPC, FFT et Cepstre fréquentiel sur la voyelle « ei » du mot « soleil » dans la phrase parlée : « La bise et le soleil se disputaient ».

Estimation

La théorie de l'estimation est un domaine d'application important du Traitement du Signal. Il est en effet très fréquent qu'une information x que l'on cherche à connaître ne soit pas directement accessible, mais soit perceptible à travers une autre grandeur observable y . Les méthodes existantes d'estimation d'une information d'après la connaissance d'une mesure se répartissent en 2 groupes suivant la nature que l'on donne à cette information :

- . si elle est *aléatoire*, ces méthodes relèvent de la *théorie bayésienne de l'estimation*, tandis que
- . si elle est *déterministe*, on parlera alors de théorie de *Fisher*.

Le noyau théorique central de la 1ère est constitué des techniques d'estimation *bayésienne*. Ces dernières sont par la suite complétées du principe d'orthogonalité statistique, qui conduit notamment à 2 méthodes très classiques de filtrage statistique, que sont le *filtrage de Wiener* et le *filtrage de Kalman*.

Basée sur une approche très différente, la théorie de *Fisher* montre tout d'abord que l'estimation d'une information déterministe à partir de mesures bruitées est nécessairement entachée d'erreurs. Une classe particulièrement intéressante d'estimateurs constituent alors les estimateurs de maximum de vraisemblance.

IV. ANALYSE SPECTRALE

Le spectre d'un signal déterministe s'obtient par Transformation de Fourier.

Pour un signal aléatoire, on ne dispose que d'une réalisation nécessairement tronquée du signal et pour en obtenir le spectre ou plutôt la Densité Spectrale de Puissance, il est indispensable de qualifier le résultat obtenu (le spectre), en termes statistiques par référence au véritable spectre.

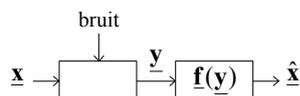
1. ELEMENTS DE LA THEORIE CLASSIQUE DE L'ESTIMATION

Dans les problèmes d'estimation, on a généralement affaire à 2 catégories de variables :

\underline{x} : un vecteur de variables inconnues que l'on cherche à déterminer.

\underline{y} : un vecteur de variables mesurées (observation), dont la valeur dépend d'une manière ou d'une autre de celle de \underline{x} .

Déterminer \underline{x} à partir de \underline{y} (et donc en fonction de \underline{y}) consiste à traiter de manière déterministe les mesures \underline{y} pour obtenir une grandeur $\hat{\underline{x}}$, proche de \underline{x} , soit : $\hat{\underline{x}} = \underline{f}(\underline{y})$



En pratique, il ne suffit pas d'effectuer une seule expérience (utilisation d'un seul \underline{y}) pour apprécier si la fonction \underline{f} est satisfaisante. Il faut envisager des résultats moyens portant sur un grand nombre d'expériences.

Le meilleur moyen d'effectuer cette analyse de validité ou de qualité de \underline{f} est de placer le problème dans un contexte statistique, en considérant que tout ou partie des variables \underline{x} , \underline{y} intervenant dans le problème, sont des réalisations de variables aléatoires.

Exemple : cas monovariabile

Soit à déterminer l'accélération de la pesanteur g à partir de l'observation de la vitesse de la chute d'un corps :

l'accélération g s'obtient à partir de la vitesse v du corps et du temps t par la relation linéaire : $v = gt$.

En pratique, on n'a pas accès à la vitesse vraie : v est une mesure bruitée de la vitesse.

On a la correspondance avec les notations précédentes :

$$\begin{aligned} g &\equiv x \\ \hat{g} &\equiv \hat{x} \\ v &\equiv y \\ \hat{g} = \frac{v}{t} = f(v) &\equiv \hat{x} = f(y) \end{aligned}$$

Si les vecteurs \underline{x} et \underline{y} sont considérés tous deux comme des réalisations de Variables Aléatoires \underline{X} et \underline{Y} , la théorie de l'estimation est dite *théorie générale de l'estimation* (dûe à Bayes).

Si \underline{x} est une grandeur inconnue déterminée (\equiv déterministe) et \underline{y} une réalisation de la VA \underline{Y} , on est dans le domaine de la *théorie classique de l'estimation (théorie de Fisher)*.

C'est cette seconde hypothèse qui va permettre de traiter de l'estimation spectrale.

- . $\hat{\underline{x}} = \underline{f}(\underline{y})$ sera une réalisation de $\hat{\underline{X}} = \underline{f}(\underline{Y})$
- . $\hat{\underline{X}}$, vecteur aléatoire, est appelé **estimateur**
- . $\hat{\underline{x}}$, réalisation de $\hat{\underline{X}}$, est appelé **estimation**

Biais et variance d'un estimateur

On définit le biais de l'estimateur $\hat{\underline{X}} = \underline{f}(\underline{Y})$ par : $\underline{b} = E[\hat{\underline{X}} - \underline{x}] = E[\underline{f}(\underline{Y}) - \underline{x}] = \underline{b}[\underline{x}]$

Selon que $\underline{b} = \underline{0}$ ou non, on dira que l'estimateur est *non biaisé* ou *biaisé*.

On définit la matrice de *variance-covariance* de l'estimateur $\hat{\underline{X}} = \underline{f}(\underline{Y})$ par: $P_{xx} = E[(\hat{\underline{X}} - E[\hat{\underline{X}}])(\hat{\underline{X}} - E[\hat{\underline{X}}])^T] = P_{xx}(\underline{x})$

Cas monovariable

Dans le cas monovariable, l'estimateur $\hat{X} = f(Y)$ a pour biais $b = E[\hat{X} - x]$ et pour variance: $\sigma^2 = E[\hat{X} - E[\hat{X}]]^2$

Evidemment, plus b et σ^2 sont faibles, meilleur est l'estimateur. Malheureusement, généralement, la diminution de l'un provoque l'augmentation de l'autre. Dans ces conditions, un estimateur biaisé pourrait être préférable, si le biais est faible, à un estimateur non biaisé mais de très grande variance.

Le compromis entre biais et variance peut être étudié et réalisé en examinant l'EQM de l'estimateur: $EQM = E[\hat{X} - x]^2$:
(EQM : Erreur Quadratique Moyenne)

$$E[\hat{X} - x]^2 = E[\hat{X}^2] - 2xE[\hat{X}] + x^2 \quad (E[\cdot] \text{ est un opérateur linéaire, et } x \text{ est une réalisation de } X \text{ donc } E[x] = x)$$

Comme : $E[\hat{X} - x] = E[\hat{X}] - x$ on a : $E^2[\hat{X} - x] = (E[\hat{X}] - x)^2 = E^2[\hat{X}] - 2xE[\hat{X}] + x^2$

D'autre part: $E[\hat{X} - E[\hat{X}]]^2 = E[\hat{X}^2 - 2\hat{X}E[\hat{X}] + E^2[\hat{X}]] = E[\hat{X}^2] - 2E^2[\hat{X}] + E^2[\hat{X}] = E[\hat{X}^2] - E^2[\hat{X}]$

donc : $E[\hat{X} - x]^2 = E[\hat{X}^2] - 2xE[\hat{X}] + x^2 = E[\hat{X} - E[\hat{X}]]^2 + E^2[\hat{X}] - 2xE[\hat{X}] + x^2$

c'est-à-dire : $E[\hat{X} - x]^2 = E[\hat{X} - E[\hat{X}]]^2 + E^2[\hat{X}] - 2xE[\hat{X}] + x^2 = \sigma^2 + b^2$ d'où: $b^2 = E[\hat{X} - x]^2 - \sigma^2$

Le compromis apparaît bien car la réduction de la variance σ^2 par exemple provoque un accroissement du biais b .

Consistance d'un estimateur

On peut également s'intéresser aux propriétés asymptotiques des estimateurs. Pour une inconnue x à estimer, on considère des vecteurs d'observation \underline{Y} de dimension croissante correspondant à une accumulation des mesures d'où une suite d'estimateurs $\hat{x}_N, \hat{x}_{N+1}, \dots$ et des propriétés asymptotiques des biais b_N et des variances σ_N .

Si $\lim_{N \rightarrow \infty} b_N = \lim_{N \rightarrow \infty} \sigma_N = 0$ l'estimateur est dit *consistant*.

Exemple 1

Si $\{X_n\}$ est une séquence indépendante gaussienne, de moyenne m et d'écart-type σ (\equiv de variance σ^2), stationnaire et ergodique, on peut en estimer la valeur moyenne à l'aide de l'estimation :

$$\hat{m} = \frac{1}{N} \sum_{i=0}^{N-1} x_i$$

On peut calculer le biais et la variance de l'estimateur $\hat{M} = \frac{1}{N} \sum_{i=0}^{N-1} X_i$:

biais b

$$E[\hat{M}] = \frac{1}{N} E\left[\sum_{i=0}^{N-1} X_i\right] = \frac{1}{N} \sum_{i=0}^{N-1} E[X_i] = \frac{1}{N} \sum_{i=0}^{N-1} m = m : \text{ le biais } b = E[\hat{M} - m] = E[\hat{M}] - m \text{ est nul} \quad \boxed{b=0}$$

variance σ_N^2

$$\sigma_N^2 = E[\hat{M} - E[\hat{M}]]^2 = E[\hat{M} - m]^2 = E\left[\frac{1}{N} \sum_{i=0}^{N-1} X_i - \frac{N}{N} m\right]^2 = \frac{1}{N^2} E\left[\sum_{i=0}^{N-1} X_i - Nm\right]^2 = \frac{1}{N^2} E\left[\sum_{i=0}^{N-1} (X_i - m)\right]^2$$

or la séquence $\{X_n\}$ est indépendante : $E[(X_i - m)(X_j - m)] = \sigma^2 \delta_{i,j}$ et l'opérateur $E[\cdot]$ linéaire,

$$\text{donc : } \sigma_N^2 = \frac{1}{N^2} E\left[\sum_{i=0}^{N-1} (X_i - m)\right]^2 = \frac{1}{N^2} E\left[\sum_{i=0}^{N-1} (X_i - m)^2\right] = \frac{1}{N^2} \sum_{i=0}^{N-1} E[X_i - m]^2 = \frac{1}{N^2} N \sigma^2 = \frac{1}{N} \sigma^2$$

$$\boxed{\sigma_N^2 = \frac{1}{N} \sigma^2}$$

Comme $b = 0$ et que $\lim_{N \rightarrow \infty} \sigma_N = 0$, l'estimateur est consistant.

Exemple 2

Sous les même hypothèses que l'exemple 1, on peut estimer cette fois la variance σ^2 de $\{X_n\}$ à l'aide de l'estimation $\hat{\sigma}^2$:

$$\text{estimateur 1 : } \hat{\sigma}^2 = \frac{1}{N} \sum_{i=0}^{N-1} (x_i - m)^2 \quad \text{si } m \text{ est connu}$$

$$\text{ou estimateur 2 : } \hat{\sigma}^2 = \frac{1}{N} \sum_{i=0}^{N-1} (x_i - \hat{m})^2 \quad \text{si } m \text{ doit être estimé simultanément.}$$

On peut calculer le biais, b_1 , de l'estimateur 1, $\hat{\Sigma}^2 = \frac{1}{N} \sum_{i=0}^{N-1} (X_i - m)^2$:

biais b_1

$$E[\hat{\Sigma}^2] = \frac{1}{N} E\left[\sum_{i=0}^{N-1} (X_i - m)^2\right] = \frac{1}{N} \sum_{i=0}^{N-1} E[X_i - m]^2 = \frac{1}{N} N \sigma^2 = \sigma^2 :$$

$$\text{le biais } b_1 = E[\hat{\Sigma}^2 - \sigma^2] = E[\hat{\Sigma}^2] - \sigma^2 \text{ est nul} \quad \boxed{b_1=0}$$

On peut calculer le biais, b_2 , de l'estimateur 2, $\hat{\Sigma}^2 = \frac{1}{N} \sum_{i=0}^{N-1} (X_i - \hat{M})^2$:

biais b_2

$$E[\hat{\Sigma}^2] = \frac{1}{N} E\left[\sum_{i=0}^{N-1} (X_i - \hat{M})^2\right] = \frac{1}{N} E\left[\sum_{i=0}^{N-1} \left(X_i - \frac{1}{N} \sum_{i=0}^{N-1} X_i\right)^2\right] = \frac{1}{N} \sum_{i=0}^{N-1} E\left[X_i - \frac{1}{N} \sum_{i=0}^{N-1} X_i\right]^2$$

$$E[\hat{\Sigma}^2] = \frac{1}{N} \sum_{i=0}^{N-1} E\left[(X_i - m) - \frac{1}{N} \sum_{i=0}^{N-1} (X_i - m)\right]^2 = \frac{1}{N} \sum_{i=0}^{N-1} E\left[(X_i - m)^2 - \frac{2}{N} (X_i - m)^2 + \frac{\sigma^2}{N}\right]$$

$$E[\hat{\Sigma}^2] = \frac{1}{N} \sum_{i=0}^{N-1} \left(E[X_i - m]^2 - \frac{2}{N} E[X_i - m]^2 + \frac{\sigma^2}{N}\right) = \frac{1}{N} \sum_{i=0}^{N-1} E[X_i - m]^2 - \frac{2}{N^2} \sum_{i=0}^{N-1} E[X_i - m]^2 + \frac{1}{N} \sum_{i=0}^{N-1} \frac{\sigma^2}{N}$$

$$E[\hat{\Sigma}^2] = \frac{1}{N} N\sigma^2 - \frac{2}{N^2} N\sigma^2 + \frac{1}{N} \sigma^2 = \sigma^2 - \frac{1}{N} \sigma^2 = \sigma^2 \left(1 - \frac{1}{N}\right) :$$

le biais $b_2 = E[\hat{\Sigma}^2 - \sigma^2] = E[\hat{\Sigma}^2] - \sigma^2$ est non nul $b_2 = -\frac{\sigma^2}{N}$

On peut cependant l'annuler en prenant pour estimation : $\hat{\sigma}^2 = \frac{1}{N-1} \sum_{i=0}^{N-1} (x_i - \hat{m})^2$

2. THEORIE CLASSIQUE DE L'ESTIMATION SPECTRALE

2.1. Estimateurs de corrélation

La séquence d'intercorrélation de 2 signaux $\{X_n\}$ et $\{Y_n\}$ s'écrit : $\phi_{xy_k} = E[X_n Y_{n+k}]$

Avec l'hypothèse ergodique (moyenne statistique \equiv moyenne temporelle), on a : $\phi_{xy_k} = \lim_{N \rightarrow \infty} \frac{1}{2N+1} \sum_{n=-N}^N x_n y_{n+k}$

De même le signal d'autocorrélation s'écrit : $\phi_{xx_k} = \lim_{N \rightarrow \infty} \frac{1}{2N+1} \sum_{n=-N}^N x_n x_{n+k}$

Estimation de l'autocorrélation

Cela revient pour un k donné, à une estimation de la moyenne sur n du signal $\{x_n x_{n+k}\}$.

Dans la pratique, on ne dispose évidemment pas d'une infinité d'échantillons temporels d'une séquence réalisation $\{x_n\}$ de $\{X_n\}$, mais seulement de N échantillons.

On ne peut alors effectuer cette moyenne que sur $N - |k|$ échantillons de $\{x_n\}$:

en effet : le calcul des termes $x_n x_{n+k}$ est soumis à : $\begin{cases} x_n \text{ défini pour } 0 \leq n \leq N-1 \\ \text{et} \\ x_{n+k} \text{ défini pour } 0 \leq n+k \leq N-1 \end{cases}$ et il ne peut donc se

faire que sur l'intervalle intersection de : $\begin{cases} 0 \leq n \leq N-1 \\ \text{et} \\ -k \leq n \leq N-k-1 \end{cases}$ soit : $\begin{cases} 0 \leq n \leq N-k-1 & \text{si } k \geq 0 \\ -k \leq n \leq N-1 & \text{si } k < 0 \end{cases}$

mais en remarquant qu'une autocorrélation est paire, on a : $\phi_{xx_{k<0}} = \phi_{xx_{k>0}} = \phi_{xx_{|k|}}$

d'où les expressions de l'estimation $\hat{\phi}_{xx_k}$ et de l'estimateur $\hat{\phi}_{xx_k}$ de l'autocorrélation:

$$\hat{\phi}_{xx_k} = \frac{1}{N - |k|} \sum_{n=0}^{N-|k|-1} x_n x_{n+k}$$
 $\hat{\phi}_{xx_k} = \frac{1}{N - |k|} \sum_{n=0}^{N-|k|-1} X_n X_{n+k}$

Comme $E[\hat{\phi}_{xx_k}] = \frac{1}{N-|k|} \sum_{n=0}^{N-|k|-1} E[X_n X_{n+k}] = \frac{1}{N-|k|} (N-|k|) \phi_{xx_k} = \phi_{xx_k}$, le biais de l'estimateur est nul :

$$b = E[\hat{\phi}_{xx_k} - \phi_{xx_k}] = E[\hat{\phi}_{xx_k}] - \phi_{xx_k} = 0$$

On montre que la variance de l'estimateur est proportionnelle à $\frac{1}{N-|k|}$. L'estimateur est donc consistant.

Comme on l'a déjà dit, le calcul direct de l'autocorrélation ϕ_{xx_k} n'est pas représentatif car on ne dispose que d'une réalisation $\{x_n\}$.

C'est l'estimation $\hat{\phi}_{xx_k}$ de l'autocorrélation qui en donne un résultat fiable.

L'algorithme le plus souvent utilisé pour le calcul de l'estimation $\hat{\phi}_{xx_k}$ fait appel à la Transformation de Fourier car on

a la relation : $\hat{\phi}_{xx_k} = \frac{1}{N-|k|} (x_k * x_{-k})$, et la propriété suivante, où :

$$\begin{cases} \{X_k\} = TFD(\{x_n\}) \\ X_k^* \text{ est le conjugué de } X_k \end{cases} : \quad \hat{\phi}_{xx_k} = \frac{1}{N-|k|} (x_k * x_{-k}) \xrightarrow{TFD} \hat{\Phi}_{xx}(\omega) = \frac{1}{N-|k|} (X_k \cdot X_k^*) = \frac{1}{N-|k|} |X_k|^2$$

L'algorithme du calcul de l'estimation $\hat{\phi}_{xx_k}$ peut donc être celui-ci :

1. Calcul de $\{X_k\} = TFD(\{x_n\})$ $k = 0, \dots, N-1; \quad n = 0, \dots, N-1$
2. Calcul de $\{|X_k|^2\}$ $k = 0, \dots, N-1$
3. Calcul de $\hat{\phi}_{xx_k} = \frac{1}{N-|k|} TFD^{-1}(\{|X_k|^2\})$ $k = 0, \dots, N-1$

Estimation de l'intercorrélation

L'intercorrélation n'est pas une séquence paire. On peut transposer le cas précédent pour des signaux X_n, Y_n stationnaires, ergodiques et à valeur moyenne nulle :

$$\boxed{\begin{cases} \hat{\phi}_{xy_k} = \frac{1}{N-|k|} \sum_{n=0}^{N-k-1} x_n y_{n+k} & \text{si } 0 \leq k \leq N \\ \hat{\phi}_{xy_k} = \frac{1}{N-|k|} \sum_{n=0}^{N+k-1} x_{n-k} y_n & \text{si } -N \leq k \leq 0 \end{cases}}$$

On peut montrer que cette estimation est sans biais et consistante.

2.2. Estimateurs de DSP

Rappels : . Fonction d'autocorrélation \equiv Fonction d'autocovariance (abus de langage)

. DSP de $x_n = TF[\text{fonction d'autocovariance de } x_n]$

2.2.1. Estimateur de DSP : méthode du périodogramme

Périodogramme

Soit $\{x_n\}$ le signal à caractériser $x(t)$ échantillonné au pas $T = 1$, et une fenêtre $\{f_n\}$ de largeur N , fenêtre d'observation de $\{x_n\}$.

Soit : $y_n = f_n x_n$ (signal observé sur une certaine durée).

$$\text{On a : } TFD(\{y_n\}) = \sum_{n=0}^{N-1} f_n x_n e^{-j\omega n} = Y(\omega)$$

On appelle **périodogramme** : $\hat{S}(\omega) = \frac{1}{N} |Y(\omega)|^2$ $\hat{S}(\omega) = \frac{1}{N} Y(\omega) \bar{Y}(\omega)$

Dans le cas $\{y_n\}$ réel : rappel : si $\{y_n\}$ réel, on a $\forall k: Y_{-k} = \bar{Y}_k$ ($\bar{Y}_k \equiv$ conjugué de Y_k et $Y_k = TFD(\{y_n\})$)

$$\hat{S}(\omega) = \frac{1}{N} Y(\omega) \bar{Y}(\omega) \rightarrow \hat{S}(\omega) = \frac{1}{N} \sum_{m=0}^{N-1} y_m e^{-j\omega m} \sum_{n=0}^{N-1} y_n e^{j\omega n} = \frac{1}{N} \sum_{m=-\infty}^{\infty} y_m e^{-j\omega m} \sum_{n=0}^{N-1} y_n e^{j\omega n}$$

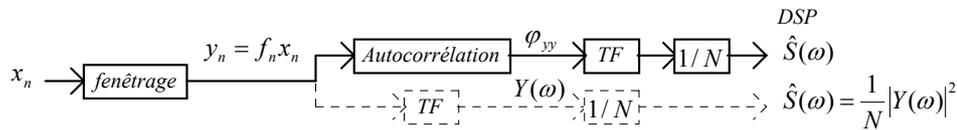
soit, en posant : $k = m - n$: $\hat{S}(\omega) = \frac{1}{N} \sum_{k=-\infty}^{\infty} \sum_{n=0}^{N-1} y_n y_{n+k} e^{-j\omega k} = \frac{1}{N} \sum_{k=-\infty}^{\infty} \varphi_{yy_k} e^{-j\omega k} = \frac{1}{N} TFD[\varphi_{yy_k}]$

où : $\varphi_{yy_k} = \sum_{n=0}^{N-|k|-1} y_n y_{n+k} = \sum_{n=0}^{N-|k|-1} x_n f_n x_{n+k} f_{n+k}$

Le périodogramme $\hat{S}(\omega)$ du signal x_n est la TFD de la séquence d'autocorrélation φ_{yy_k} du signal fenêtré :

$$\hat{S}(\omega) = \frac{1}{N} TFD[\varphi_{yy_k}] \quad \text{avec : } y_n = f_n x_n = \text{signal } x_n \text{ fenêtré}$$

algorithme du calcul de la DSP de x_n par périodogramme



biais de l'estimateur (biais du périodogramme)

On montre que l'estimateur est biaisé, sauf si la fenêtre $\{f_n\}$ est infiniment large ($N = \infty$).

variance de l'estimateur (variance du périodogramme)

On montre par contre que la variance de l'estimateur ne diminue pas quand N augmente.

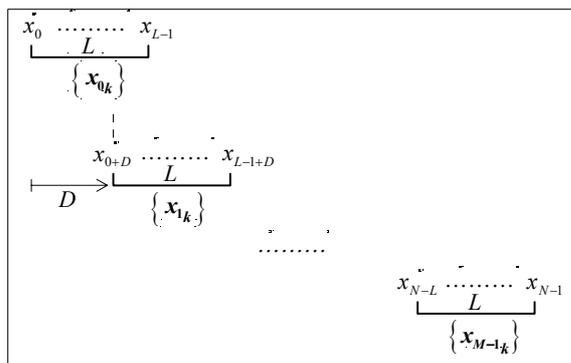
Le périodogramme est donc inconsistant (sauf pour quelques cas particuliers de signaux $\{x_n\}$).

Périodogramme moyenné

Du fait de son inconsistance, en pratique, la méthode du périodogramme simple est peu satisfaisante. On préfère généralement le périodogramme moyenné, qui consiste à calculer M périodogrammes à partir de M sous-suites du signal $\{x_n\}$ et à en effectuer la moyenne, réduisant ainsi, si les M périodogrammes sont indépendants, la variance de l'estimateur (dans le rapport M).

C'est la méthode de **Welch**. On a : $\hat{S}_M(\omega) = \frac{1}{M} \sum_{m=0}^{M-1} \left\{ \frac{1}{L} \left| \sum_{k=0}^{L-1} x_{m_k} f_k e^{-jk\omega} \right|^2 \right\}$

où les $\{x_{m_k}\}$ sont M parties (de durée L) se chevauchant de la séquence $\{x_n\}$ (de durée N) :



avec :

- . x_{m_k} : durée L $0 \leq k \leq M - 1$ M : nombre de sous-suites $\{x_{m_k}\}$ de $\{x_n\}$
- . D : décalage de chaque sous-suite :
 - . si $D = L$: le chevauchement (C) des sous-suites $\{x_{m_k}\}$ est nul : $C = 0$
 - . sinon ($0 < D < L$) : il y a chevauchement (C) des sous-suites $\{x_{m_k}\}$: $C = L - D$
 - (. si $D > L$ on perd des échantillons de $\{x_n\}$, et si $D = 0$ on ne peut balayer $\{x_n\}$).
- . Nombre M de sous-suites :
 - . si $(N - L)$ est divisible par D , on a : $M = \frac{N - L}{D} + 1$
 - . sinon ($(N - L)$ non divisible par D) : $M = E\left(\frac{N - L}{D}\right) + 1$ ou $E(\cdot)$ représente la partie entière.

Si D est grand (proche de L) le chevauchement des séquences $\{x_{m_k}\}$ est faible et celles-ci peuvent donc être considérées comme relativement indépendantes : on a alors M périodogrammes indépendants.

Le compromis entre biais et variance apparaît clairement : pour un N donné, le biais sera d'autant plus faible que L est grand (large fenêtre), et la variance sera d'autant plus faible que M est grand. Il est clair que l'accroissement de M et L rend les sous-suites non indépendantes, et la variance ne sera pas réduite. (Welch suggère le compromis: $M \approx \frac{2N}{L}$).

2.2.2. Estimateur de DSP : méthode du corrélogramme (ou encore méthode de Blackman-Tukey)

On a vu d'une part qu'il existe un estimateur consistant des séquences d'autocorrélation et d'autre part, que le périodogramme est la Transformée de Fourier de cette séquence d'autocorrélation.

On peut donc d'abord estimer l'autocorrélation puis en prendre la Transformée de Fourier. C'est le principe du corrélogramme :

$$\hat{S}(\omega) = \frac{1}{N} \sum_{k=-N}^N \hat{\phi}_{xx_k} f_k e^{-j\omega k} = \frac{1}{N} TF[\hat{\phi}_{xx_n} f_n]$$

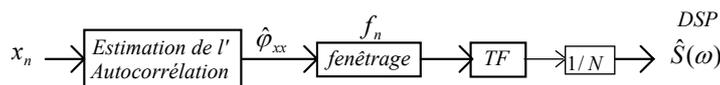
On a placé une fenêtre $\{f_k\}$ de largeur $2M + 1$ pour éviter de fausser l'estimation de $\hat{S}(\omega)$.

En effet pour $|k|$ élevé, $\hat{\phi}_{xx_k}$ peut avoir des valeurs dispersées.

D'autre part, $\{\hat{\phi}_{xx_k}\}$ étant paire, $\{f_k\}$ doit l'être également, sinon $\hat{S}(\omega)$ ne le sera pas.

On montre que l'estimateur corrélogramme est consistant.

algorithme du calcul de la DSP de x_n par corrélogramme



2.2.3. Estimation de la DSP à partir du modèle AR du signal ≡ analyse LPC

(THEORIE MODERNE DE L'ESTIMATION SPECTRALE basée sur les processus générateurs de signal)

On a vu qu'un signal peut être représenté par un modèle générateur (filtre formeur) excité par un bruit blanc. Le modèle pouvant être du type AR, MA, ARMA ou d'Etat (Gauss-Markov).

A partir de la connaissance des paramètres du modèle, on pourra alors calculer la DSP du signal.

L'avantage par rapport aux techniques classiques précédentes est une meilleure résolution spectrale et la possibilité d'identification en temps réel des paramètres du signal par filtrage adaptatif.

On sait que la Fonction de Transfert d'un filtre AR est :

$$H(z) = \frac{Y(z)}{X(z)} = \frac{1}{1 + \sum_{n=1}^N a_n z^{-n}}$$

correspondant à l'équation récurrente :

$$Y_n = X_n - \sum_{k=1}^N a_k y_{n-k}$$

Prenons $T = 1$ pour période d'échantillonnage pour simplifier les écritures.

Les DSPs d'entrée et de sortie sont reliées par la relation :

$$\Phi_{YY}(\omega) = |H(\omega)|^2 \Phi_{XX}(\omega)$$

où : $\Phi_{XX}(\omega) = TF[\varphi_{xx_n}]$ et $\Phi_{YY}(\omega) = TF[\varphi_{yy_n}]$ avec : φ_{xx_n} et φ_{yy_n} : autocorrélations.

soit :

$$\Phi_{YY}(\omega) = \frac{\Phi_{XX}(\omega)}{\left| 1 + \sum_{n=1}^N a_n e^{-jn\omega} \right|^2}$$

Si $\{X_n\}$ est une séquence blanche de variance V ($\Phi_{XX}(\omega) = V$) :

$$\Phi_{YY}(\omega) = \frac{V}{\left| 1 + \sum_{n=1}^N a_n e^{-jn\omega} \right|^2}$$

$\Phi_{YY}(\omega)$ est obtenu à partir de V et des N paramètres a_n du modèle AR (N n'est évidemment pas connu a priori).

Il faut donc préalablement « identifier » les N paramètres, qui sont avec V , les solutions des $N + 1$ équations linéaires de **Yule-Walker** :

$$\begin{cases} \varphi_{yy_0} + \sum_{i=1}^N a_i \varphi_{yy_i} = V \\ \varphi_{yy_1} + \sum_{i=1}^N a_i \varphi_{yy_{i-1}} = 0 \\ \dots \\ \varphi_{yy_N} + \sum_{i=1}^N a_i \varphi_{yy_{N-i}} = 0 \end{cases} \Leftrightarrow \text{pour } 0 \leq k \leq N : \sum_{n=0}^N \varphi_{yy_{k-n}} a_n = \begin{cases} V & \text{si } k = 0 \\ 0 & \text{si } 1 \leq k \leq N \end{cases} \quad (a_0 = 1) \quad (\varphi_{yy_k} \text{ paire})$$

soit, sous forme matricielle : ce sont les équations de **Yule-Walker** :

$$\begin{bmatrix} \varphi_{yy_0} & \varphi_{yy_1} & \dots & \varphi_{yy_N} \\ \varphi_{yy_1} & \varphi_{yy_0} & \dots & \varphi_{yy_{N-1}} \\ \vdots & \vdots & \ddots & \vdots \\ \varphi_{yy_N} & \varphi_{yy_{N-1}} & \dots & \varphi_{yy_0} \end{bmatrix} \begin{bmatrix} 1 \\ a_1 \\ \vdots \\ a_N \end{bmatrix} = \begin{bmatrix} V \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad \text{ou encore : } \boxed{\mathbf{R} \underline{a} = \underline{v}}$$

Pour réaliser ce calcul, une inversion matricielle est possible mais on utilise généralement l'algorithme efficace itératif de **Levinson**, ou encore la méthode de **Burg**, qui évitent une inversion matricielle.

En résumé, le calcul de la DSP de la séquence $\{Y_n\}$ requiert :

. l'estimation de séquence d'autocorrélation $\{\varphi_{yy_k}\}$, à insérer dans le système :

$$\begin{bmatrix} \varphi_{yy_0} & \varphi_{yy_1} & \cdots & \varphi_{yy_N} \\ \varphi_{yy_1} & \varphi_{yy_0} & \cdots & \varphi_{yy_{N-1}} \\ \vdots & \vdots & \vdots & \vdots \\ \varphi_{yy_N} & \varphi_{yy_{N-1}} & \cdots & \varphi_{yy_0} \end{bmatrix} \begin{bmatrix} 1 \\ a_1 \\ \vdots \\ a_N \end{bmatrix} = \begin{bmatrix} V \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

. le calcul des paramètres a_n ($1 \leq n \leq N$) et de V à l'aide de l'algorithme de Levinson (ou par inversion matricielle).

. l'utilisation de la relation : $\Phi_{YY}(\omega) = \frac{V}{\left| 1 + \sum_{n=1}^N a_n e^{-jn\omega} \right|^2}$ pour obtenir la DSP cherchée.

3. CONCLUSION

Comme toujours en Traitement du Signal, plus on a d'informations sur le signal à traiter, meilleur est le traitement réalisé. Ainsi, des méthodes plus puissantes d'analyse spectrale (Pisarenko, Prony, ...) mettent à profit la connaissance de la structure du signal à caractériser, ou plus généralement, à traiter.

TP 3. Caractérisation. Estimation

CARACTERISATION

I. TRANSFORMATIONS FREQUENTIELLES

TFD (signaux stationnaires)

$x_n = x(nT)$: Signal à TD, échantillonné à la cadence T, et d'une certaine durée $T_0 := M \cdot T$

Il s'agit d'un signal de parole (enregistrement de la voyelle "a").

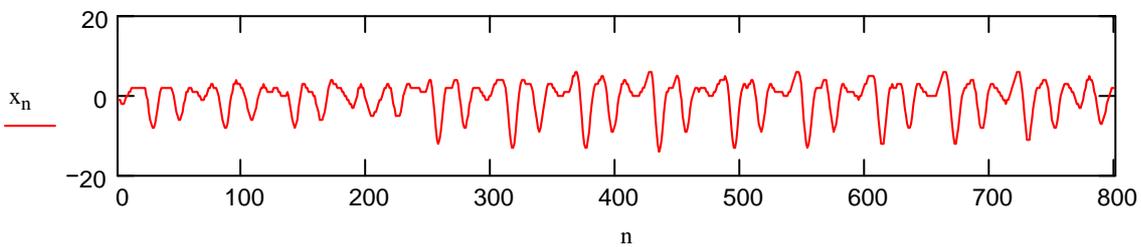
La fréquence d'échantillonnage est : $F = 1/T = 8$ kHz.

Le nombre d'échantillons est : $M = 800$.

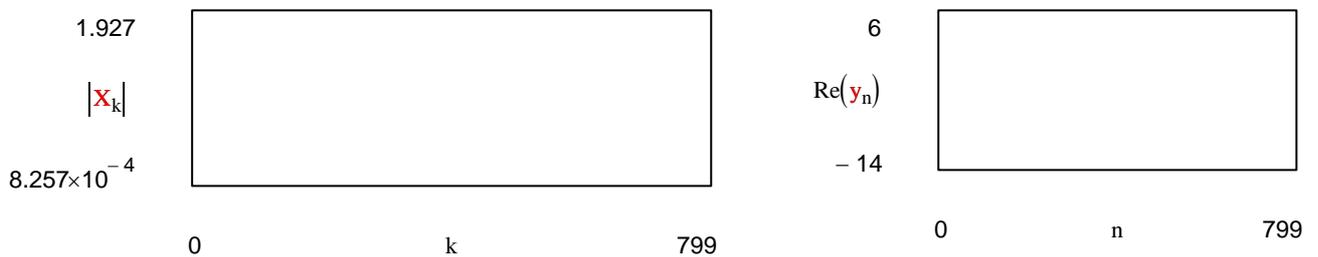
La durée de l'enregistrement est : $T_0 = 0.1$ s.

Chaque échantillon est codé en mono sur 1 octet.

```
. Fichier : M := 800 n := 0..M - 1 x := READWAV("o1.wav") x := x - 128
```



```
X :=  k := 0..M - 1 <- Fonctions utilisateur -> y := TFDI(X)
```



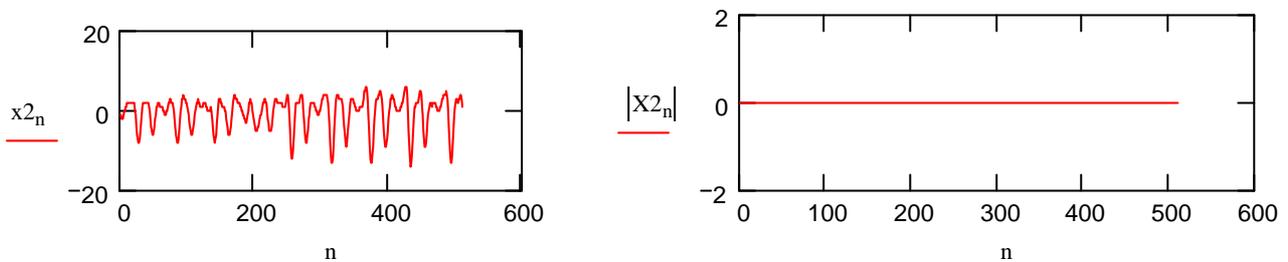
Conclusion :

Transformée de Fourier rapide : FFT

Le nombre de points M doit être une puissance de 2, soit ici 512 points.

Du fait de la périodicité de M/2 de la Transformée, le calcul de la FFT est limité à M/2.

```
M := 512 n := 0..M - 1 x2 := submatrix(x, 0, M - 1, 0, 0) X2 :=  X2_{M-1} := 0
```



ONDELETTES (caractérisation de signaux non stationnaires)

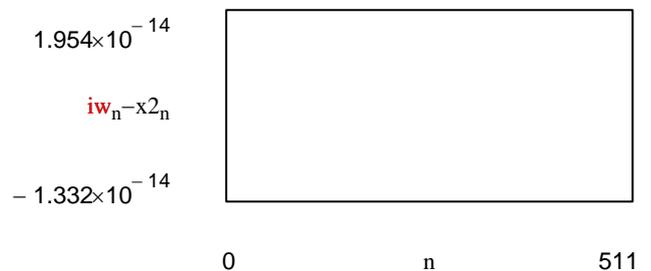
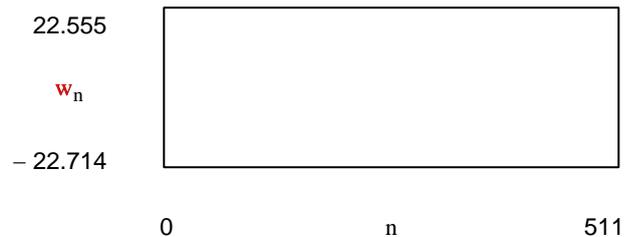
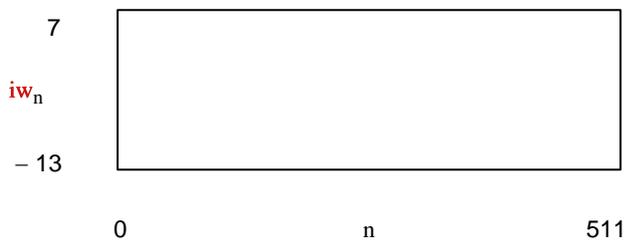
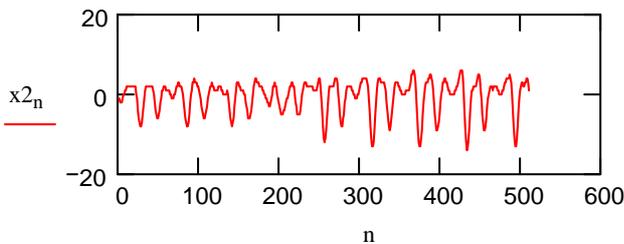
La fonction de Transformée rapide en ondelettes requiert un nombre d'échantillons égal à une puissance de 2 : on prend donc les 512 1ers points du signal original comptant 800 points.

```
M := 512  n := 0..M - 1  x2 :=
  for n ∈ 0..M - 1
    x2_n ← x_n
  x2
```

Transformée en ondelettes : `w :=`

Transformée en ondelettes inverse : `iw :=`

^ Fonctions intégrées MathCad (1er niveau de décomposition)

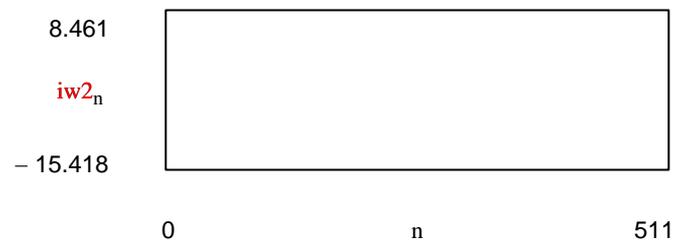
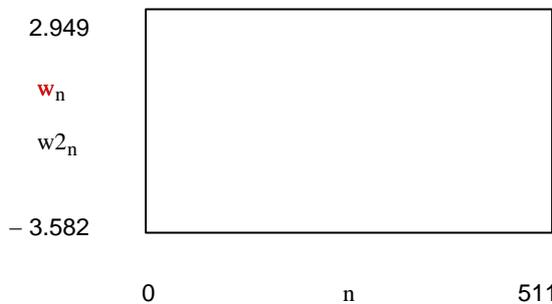
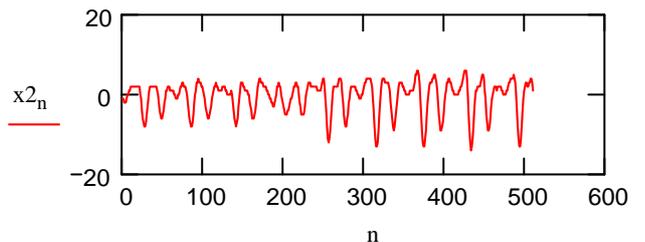


Conclusion :

Compression par ondelettes *Exemple : compression d'un facteur 3*

On peut utiliser seulement les M/3 premiers points de w = Transfo en ondelettes de x2 pour caractériser x2. Pour le voir, annuler la transformée en ondelettes w_n sur M/3 < n < M pour fabriquer le signal w2 et prendre ensuite la transformée en ondelettes inverse iw2 et comparer iw2 à x2 en traçant la différence iw2-x2. Conclusion.

```
w2 :=
  iw2 := iwave(w2)
```



Visualisation de la compression : compression d'images :

```
I := READ_IMAGE("bato2.bmp")
```

Nombre de colonnes : $N_x := \text{cols}(I)$ Nombre de lignes : $N_y := \text{rows}(I)$

$N_x = 128$

$N_y = 128$

Transformée en ondelettes :

Traitement 1D (colonne)

$$\text{Wave1D}(f) := \begin{cases} g \leftarrow \text{wave}(f) \\ g \end{cases}$$

Transformée en ondelettes de chaque colonne de l'image :

```
Wave2D(I) :=
```

```
K := Wave2D(I)
```

Transformée en ondelettes inverse :

Traitement 1D (colonne)

$$\text{IWave1D}(f) := \begin{cases} g \leftarrow \text{iwave}(f) \\ g \end{cases}$$

Transformée en ondelettes inverse de chaque colonne de l'image :

```
IWave2D(I) :=
```

```
L := IWave2D(K)
```

Image originale



Image ondelettes

Image ondelettes inverse

I

K

L

Compression d'un facteur N :

```
N := 2
```

```
Compress1D(u) :=
```

Transformée en ondelettes :

Compression 1D (colonne) $Wave1Dcomp(f) := \begin{cases} g \leftarrow wave(f) \\ g \leftarrow \mathbf{Compress1D}(g) \\ g \end{cases}$

Transformée en ondelettes de chaque colonne de l'image

$$Wave2Dcomp(I) := \begin{cases} \text{for } x \in 0..cols(I) - 1 \\ J^{(x)} \leftarrow \mathbf{Wave1Dcomp}(I^{(x)}) \\ \text{return } J \end{cases}$$

$$K := \mathbf{Wave2Dcomp}(I) \quad L := \mathbf{IWave2D}(K)$$

Image originale

Image ondelettes

Image ondelettes inverse



I

K

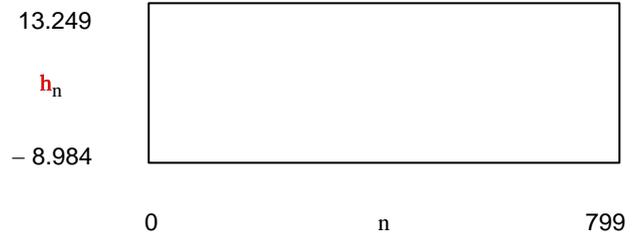
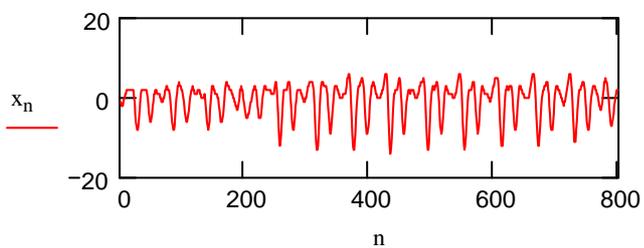
L

TRANSFORMEE DE HILBERT (Transformation temporelle)

$$M := 800 \quad n := 0..M - 1$$

$$h := \blacksquare$$

<- Fonction intégrée à MathCad car la fonction utilisateur Hilbert() est trop lente.



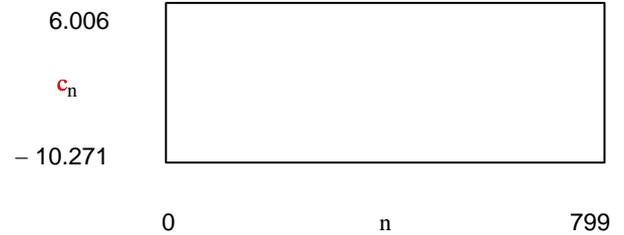
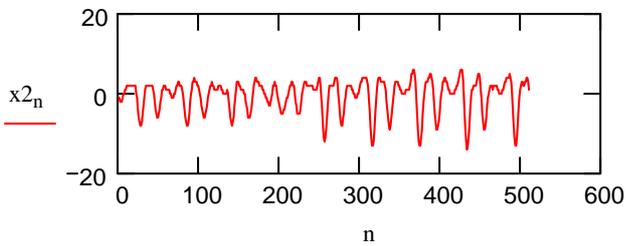
II. ANALYSE CEPSTRALE (Transformation temporelle)

CEPSTRE

$M := 512$ $n := 0..M - 1$

Cepstre : $c := \blacksquare$

<- Fonction intégrée MathCad



Conclusion :

III. ANALYSE LPC (Transformation temporelle)

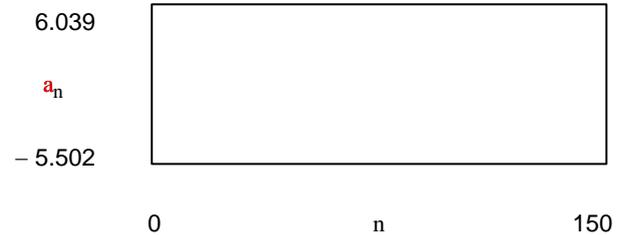
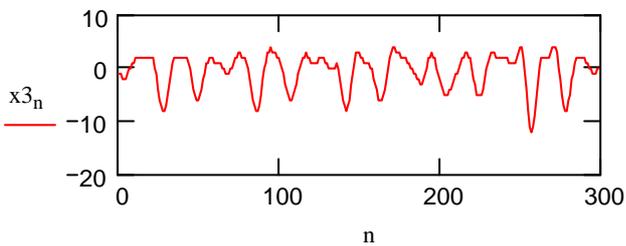
$M := 300$ $n := 0..M - 1$ $x3 := \begin{cases} \text{for } n \in 0..M - 1 \\ x3_n \leftarrow x_n \\ x3 \end{cases}$

$\phi_{xx} := \blacksquare$
(vecteur autocovariance)

$N := 80$ $a := \blacksquare$
(coeffs du filtre formeur)

$V := a_0$ $V = \blacksquare$

^~ Nombre de points limité car sinon les fonctions utilisateur r() et YuleWalker() sont trop lentes.



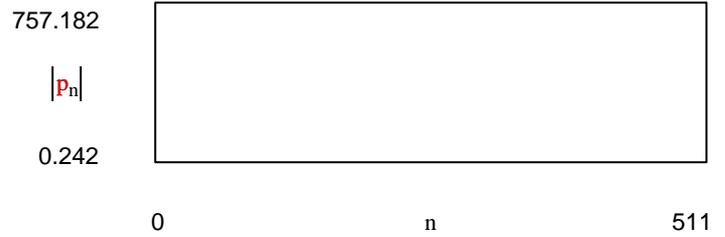
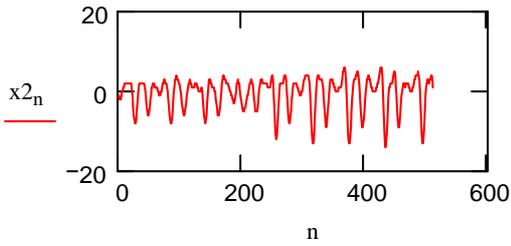
ESTIMATION

IV. ANALYSE SPECTRALE (Transformation fréquentielle)

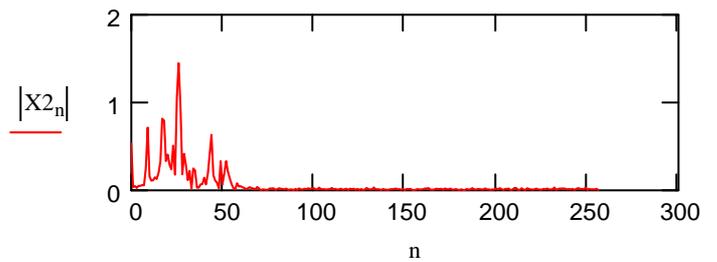
Périodogramme

M := 512 n := 0..M - 1 cor := ■

Périodogramme : p := ■



Spectre du signal aléatoire dans sa globalité, à comparer au spectre direct de x2 sur 1 seule réalisation : X2 := FFT(x2)



Fonctions Bibliothèque :

Moyenne m, Fonction d'Autocovariance r et Variance V du signal x de dim K par la méthode d'ergodicité :

. Moyenne :

$$\text{moy}(x) \equiv \begin{cases} K \leftarrow \text{length}(x) \\ m \leftarrow \frac{1}{K} \cdot \sum_{n=0}^{K-1} x_n \\ m \end{cases}$$

. Vecteur d'AutoCovariance :

$$\text{vautocov}(x) \equiv \begin{cases} K \leftarrow \text{length}(x) \\ m \leftarrow \text{moy}(x) \\ \text{for } k \in 0..K-1 \\ r_k \leftarrow \frac{1}{K} \cdot \sum_{n=0}^{K-k-1} (x_{n+k} - m) \cdot (x_n - m) \\ r \end{cases}$$

. Variance :

$$\text{Var}(x) \equiv \text{vautocov}(x)_0$$

Fonction YuleWalker(ϕ_{xx}, N) : Résolution du système de Yule-Walker $Ra = v$ par inversion matricielle

Paramètres d'entrée : . ϕ_{xx} : vecteur fonction d'autocovariance (dim K) du signal x (dim K) : $\phi_{xx0}, \phi_{xx1} \dots \phi_{xxK-1}$
 . N : dim de l'ordre du filtre (formeur/LPC) AR

Paramètre de sortie : . le vecteur a (dim N+1) :
 $a_0 = V$: variance (du bruit blanc si filtre formeur); a_1, a_2, \dots, a_N : paramètres du filtre (formeur/LPC) AR d'ordre N

```

YuleWalker( $\phi_{xx}, N$ ) ≡
    for i ∈ 0..N
        for j ∈ 0..N
             $R_{i,j} \leftarrow \phi_{xx} |_{i-j}$ 
         $b_0 \leftarrow 0$ 
        for i ∈ 1..N
             $b_i \leftarrow \phi_{xx_i}$ 
             $\alpha \leftarrow \left( \frac{-1}{\phi_{xx0}} \right) \cdot R^{-1} \cdot b$ 
             $a_0 \leftarrow \frac{\phi_{xx0}}{\phi_{xx0} \cdot \alpha_0 + 1}$ 
        for i ∈ 1..N
             $a_i \leftarrow a_0 \cdot \alpha_i$ 
    a
    
```

Exemple : $K := 3$ $r_0 := 6$
 $N := 2$ $r_1 := 3$
 $r_2 := 0$

$a := \text{YuleWalker}(r, N)$

$$V := a_0$$

$$a = \begin{pmatrix} 4 \\ -0.667 \\ 0.333 \end{pmatrix}$$

$$V = 4$$

$$i := 1..N$$

$$a_i =$$

-0.667
0.333

Fonction YuleW(x,N) : Résolution du système de Yule-Walker $Ra = v$ par inversion matricielle

Paramètres d'entrée : . x : signal x (dim K) : $x_0, x_1 \dots x_{K-1}$
 . N : dim de l'ordre du filtre (formeur/LPC) AR

Paramètre de sortie : . le vecteur a (dim N+1) :
 $a_0 = V$: variance (du bruit blanc si filtre formeur); $a_1, a_2, \dots a_N$: paramètres du filtre (formeur/LPC) AR d'ordre N

$$\text{YuleW}(x,N) \equiv \begin{cases} r \leftarrow \text{vautocov}(x) \\ a \leftarrow \text{YuleWalker}(r,N) \\ a \end{cases}$$

Bruit blanc de K points de moyenne m et de variance V :

Bruit blanc de K points centré de variance 0.083 :

$$\text{WhiteNoise}(K) \equiv \begin{cases} A \leftarrow 281 \\ P \leftarrow 31357 \\ v_0 \leftarrow 100 \\ \text{for } n \in 1..N-1 \\ \quad v_n \leftarrow \text{mod}(A \cdot v_{n-1}, P) \\ \text{for } n \in 0..N-1 \\ \quad w_n \leftarrow \frac{1}{P} \cdot v_n - 0.5 \\ w \end{cases}$$

Bruit blanc de K points de moyenne m et de variance V :

$$\text{BruitBlanc}(K, m, V) \equiv \sqrt{12 \cdot V} \cdot \text{whiten}(K) + m$$

$$\text{BruitBlanc2}(K, m, V) \equiv \sqrt{12 \cdot V} \cdot \text{WhiteNoise}(K) + m$$

Réponse reponse (vecteur dim. K) du filtre AR d'ordre N (coeffs a_i $0 \leq i \leq N$) au signal x (vecteur dim. K)

$$\text{reponse}(x, a, N) \equiv \begin{cases} K \leftarrow \text{length}(x) \\ \text{for } k \in 0..K-1 \\ \quad \begin{cases} y_k \leftarrow x_k & \text{if } k = 0 \\ y_k \leftarrow x_k - \sum_{i=0}^k a_i \cdot y_{k-i} & \text{if } 1 \leq k < N \\ y_k \leftarrow x_k - \sum_{i=0}^N a_i \cdot y_{k-i} & \text{if } k \geq N \end{cases} \\ y \end{cases}$$

TFD

$$\text{TFD}(x) \equiv \left| \begin{array}{l} M \leftarrow \text{length}(x) \\ \text{for } k \in 0..M-1 \\ X_k \leftarrow \frac{1}{M} \cdot \sum_{n=0}^{M-1} x_n \cdot e^{-i \cdot 2 \cdot \pi \cdot k \cdot \frac{n}{M}} \\ X \end{array} \right.$$

ou encore :

$$\text{TFD2}(x) \equiv \left| \begin{array}{l} M \leftarrow \text{length}(x) \\ \text{for } k \in 0..M-1 \\ R \leftarrow 0 \\ I \leftarrow 0 \\ \text{for } n \in 0..M-1 \\ R \leftarrow R + x_n \cdot \cos\left(2 \cdot \pi \cdot k \cdot \frac{n}{M}\right) \\ I \leftarrow I - x_n \cdot \sin\left(2 \cdot \pi \cdot k \cdot \frac{n}{M}\right) \\ X_k \leftarrow \frac{1}{M} \cdot (R + i \cdot I) \\ \text{return } X \end{array} \right.$$

TFDI

$$\text{TFDI}(X) \equiv \left| \begin{array}{l} M \leftarrow \text{length}(X) \\ \text{for } n \in 0..M-1 \\ x_n \leftarrow \sum_{k=0}^{M-1} X_k \cdot e^{i \cdot 2 \cdot \pi \cdot k \cdot \frac{n}{M}} \\ x \end{array} \right.$$

Convolution causale (signaux de même longueur)

$$\text{convcausal2}(x, h) \equiv \left| \begin{array}{l} M \leftarrow \text{length}(x) \\ \text{for } n \in 0..M-1 \\ y_n \leftarrow \sum_{k=0}^n x_k \cdot h_{n-k} \\ y \end{array} \right.$$

Convolution causale (signaux de longueur différente)

$$\text{convcausal}(x, h) \equiv \left| \begin{array}{l} M \leftarrow \text{length}(x) \\ N \leftarrow \text{length}(h) \\ \text{for } n \in 0..M-1 \\ y_n \leftarrow \sum_{k=0}^n x_k \cdot h_{n-k} \text{ if } n < N \\ y_n \leftarrow 0 \text{ otherwise} \\ y \end{array} \right.$$

Transformée de Hilbert

$$\text{Hilbert}(x) \equiv \left| \begin{array}{l} M \leftarrow \text{length}(x) \\ f_0 \leftarrow 0 \\ \text{for } n \in 1..M-1 \\ f_n \leftarrow \frac{1}{\pi \cdot n} \\ \text{TH} \leftarrow \text{convcausal}(x, f) \\ \text{TH} \end{array} \right.$$

4. Conditionnement - Filtrage. Détection

I. CONDITIONNEMENT - FILTRAGE

1. Prétraitements (Mise en forme, amplification, fenêtrage, filtrage ...)

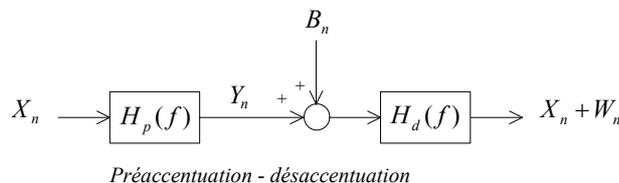
Préaccentuation et désaccentuation (Débruitage - Télécommunications - Traitement de la parole)

Le principe du débruitage par préaccentuation-désaccentuation consiste à améliorer le Rapport Signal sur Bruit.

Lors de la transmission d'un signal à travers un canal soumis à un bruit, on améliore le Rapport Signal sur Bruit (SNR) en utilisant à l'émission un filtre dit de *préaccentuation* de FT $H_p(f)$ et à la réception le filtre inverse de *désaccentuation* de FT $H_d(f) = H_p^{-1}(f)$.

Soit un signal aléatoire à TD X_n , supposé stationnaire au 2nd ordre, centré, de DSP connue $S_X(f)$. Ce signal est transmis à travers un canal soumis à un bruit B_n stationnaire au 2nd ordre, centré, de DSP $S_B(f)$ connue.

On effectue à l'émission le filtrage $H_p(f)$ et, afin que l'opération laisse le signal utile X_n inchangé, on effectue à la réception le filtrage inverse $H_d(f) = H_p^{-1}(f)$:



Par conséquent, le signal reçu a pour expression : $X_n + W_n$ où W_n représente le bruit obtenu par filtrage du bruit B_n . Le but est de déterminer le couple $\{H_p(f), H_d(f)\}$ qui minimise la puissance de W_n . Mais il faut s'imposer une contrainte. En effet, on pourrait rendre aussi petite que l'on veut la puissance de W_n en multipliant $H_p(f)$ par un facteur A élevé et en divisant $H_d(f)$ par le même facteur. Cela marcherait mais conduirait tôt ou tard à une saturation, une distorsion. La contrainte choisie consiste à fixer la puissance P_Y en sortie du filtre d'émission $H_p(f)$, ce qui est raisonnable en pratique.

Remarque : Une simple soustraction du bruit, à supposer que celui-ci soit connu, ne marche évidemment pas, du fait de son caractère aléatoire.

Détermination des filtres $H_p(f)$ et $H_d(f)$

La DSP du signal Y_n en sortie du filtre $H_p(f)$ donne la puissance P_Y de Y_n : (cf. chapitre 1)

$$P_Y = E[Y_n^2] = R_{Y Y_0} = TF^{-1}[S_Y(f)]_{\text{en } \tau=0} = \int_{-\infty}^{\infty} S_Y(f) df \quad \rightarrow \quad P_Y = \int_{-\infty}^{\infty} |H_p(f)|^2 S_X(f) df$$

Un même calcul appliqué au filtre $H_d(f)$ excité par le bruit B_n seul donne pour la puissance P_W de W_n :

$$P_W = E[W_n^2] = \int_{-\infty}^{\infty} |H_d(f)|^2 S_B(f) df$$

Il faut minimiser P_W avec la contrainte de fixer la puissance P_Y à une valeur P_0 .

Une méthode classique de minimisation sous contrainte est la technique des *multiplicateurs de Lagrange*. Dans cette méthode, on additionne à l'expression à minimiser, λ fois la contrainte, ce qui conduit à minimiser $P_W + \lambda P_Y$ ou, ce qui revient au même, à minimiser $P_W + \lambda(P_Y - P_0)$: (en fixant P_Y à la valeur P_0 , λ est un multiplicateur de Lagrange)

Il faut donc minimiser :
$$\int_{-\infty}^{\infty} |H_d(f)|^2 S_B(f) df + \lambda \left(\int_{-\infty}^{\infty} |H_p(f)|^2 S_X(f) df - P_0 \right)$$

En remplaçant $H_p(f)$ par $H_d^{-1}(f)$, cette expression s'écrit :
$$\int_{-\infty}^{\infty} \left(|H_d(f)|^2 S_B(f) + \lambda \frac{S_X(f)}{|H_d(f)|^2} \right) df - \lambda P_0$$

Pour en obtenir le minimum, il faut choisir $|H_d(f)|^2$ qui rend minimum l'expression sous le signe de l'intégrale :

$|H_d(f)|^2 S_B(f) + \lambda \frac{S_X(f)}{|H_d(f)|^2}$ et donc qui annule sa dérivée par rapport à $|H_d(f)|^2$.

Il vient :
$$|H_d(f)|^2 = \sqrt{\lambda \frac{S_X(f)}{S_B(f)}}$$

Le calcul de λ se fait en exprimant la contrainte :
$$P_0 = \int_{-\infty}^{\infty} |H_p(f)|^2 S_X(f) df = \int_{-\infty}^{\infty} \frac{S_X(f)}{|H_d(f)|^2} df = \frac{1}{\sqrt{\lambda}} \int_{-\infty}^{\infty} \sqrt{S_B(f) S_X(f)} df$$

Finalement, il vient pour le filtre de désaccentuation :

$$|H_d(f)|^2 = \sqrt{\lambda} \cdot \frac{\sqrt{S_X(f)}}{\sqrt{S_B(f)}} = \frac{\int_{-\infty}^{\infty} \sqrt{S_B(f) S_X(f)} df}{P_0} \cdot \frac{\sqrt{S_X(f)}}{\sqrt{S_B(f)}}$$

$$\rightarrow \boxed{|H_d(f)|^2 = \frac{1}{P_0} \int_{-\infty}^{\infty} \sqrt{S_B(f) S_X(f)} df \sqrt{\frac{S_X(f)}{S_B(f)}}$$

et pour le filtre de préaccentuation :

$$\boxed{|H_p(f)|^2 = |H_d^{-1}(f)|^2 = P_0 \frac{1}{\int_{-\infty}^{\infty} \sqrt{S_B(f) S_X(f)} df} \sqrt{\frac{S_B(f)}{S_X(f)}}$$

II. DETECTION

1. Filtrage adapté (matched filtering)

Dans le cadre du filtrage adapté, le problème est de détecter la présence d'un signal, connu, dans du bruit (et non le débruitage de ce signal).

Nous allons traiter ce problème dans le cas discret, le plus important pour les applications.

Position du problème

On dispose de la mesure bruitée $\{x_n\}$ qui, selon la présence ou non du signal $\{s_n\}$ à détecter, mêlé au bruit $\{b_n\}$, peut prendre 2 formes :

- . $x_n = s_n + b_n$ (présence)
- . $x_n = b_n$ (absence)

où : $\{s_n\}$ est un signal déterministe de durée $N + 1$

$\{B_n\}$ est un signal (VA) aléatoire centré, stationnaire au 2nd ordre (moments d'ordre 1 (moyenne) et 2 (variance) stationnaires), dont $\{b_n\}$ est une observation, une épreuve, une réalisation.

Il s'agit de déterminer un filtre linéaire (causal), de RI $\{h_n\}$, d'entrée/sortie respective $\{x_n\}$ et $\{y_n\}$, qui rend maximal le Rapport Signal sur Bruit (SNR : Signal to Noise Ratio), noté ρ

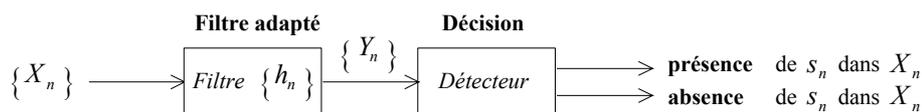
(le but n'est pas d'éliminer le bruit mais de détecter la présence du signal utile dans le signal bruité) .

→ Le filtre adapté (h) constitue donc une aide à la décision. Il consiste en accroissement du Rapport Signal/Bruit.

En notant : $y_N(s) = s_n * h_n |_{n=N} = \sum_{n=0}^N s_{N-n} h_n$: Réponse du filtre à $\{s_n\}$ à l'instant $n = N$

et : $y_N(b) = b_n * h_n |_{n=N} = \sum_{n=0}^N b_{N-n} h_n$: Réponse du filtre à $\{b_n\}$ à l'instant $n = N$

on a pour le SNR : $\rho = \frac{y_N^2(s)}{E[Y_N^2(B)]} = \rho(N, \{h_n\})$: Rapport des puissances instantanées à l'instant $n = N$



On note ρ^* ($= \rho^*(N)$) la valeur maximale de ρ ($= \rho(N, \{h_n\})$) et $\{h_n^*\}$ la RI optimale correspondante ($=$ fournissant ρ^*) du filtre cherché : $\rho^* = \rho(N, \{h_n^*\})$.

Remarques

. Afin de faciliter la tâche du détecteur, on a choisi un critère de coût (ρ) qui correspond à l'apparition d'un pic sur y_n pour $n = N$.

. On a supposé implicitement que le signal $\{s_n\}$ apparaît à l'instant $n = 0$. Cette hypothèse n'est nullement restrictive puisque le bruit est supposé stationnaire.

. La valeur du maximum ρ^* de ρ est libre de choix : on peut laisser sa valeur indéfinie, auquel cas la RI $\{h_n\}$ est alors définie à une Constante près, ou bien fixer sa valeur par normalisation :

Normalisation

$$y_N^*(s) = 1 \quad \text{soit :} \quad y_N^*(s) = \sum_{n=0}^N s_{N-n} h_n^* = 1$$

Solution

Afin de simplifier les notations et les calculs, on pose (où l'indice R signifie *Reverse*):

$$\underline{\mathbf{s}}_R = \begin{bmatrix} s_N \\ s_{N-1} \\ \dots \\ s_0 \end{bmatrix} \quad \underline{\mathbf{b}}_R = \begin{bmatrix} b_N \\ b_{N-1} \\ \dots \\ b_0 \end{bmatrix} \quad \underline{\mathbf{h}} = \begin{bmatrix} h_0 \\ h_1 \\ \dots \\ h_N \end{bmatrix}$$

Les équations:
$$\begin{cases} y_N(s) = \sum_{n=0}^N s_{N-n} h_n \\ y_N(b) = \sum_{n=0}^N b_{N-n} h_n \\ \sum_{n=0}^N s_{N-n} h_n^* = 1 \end{cases}$$
 peuvent alors être réécrites (la convolution est commutative):
$$\begin{cases} y_N(s) = \underline{\mathbf{s}}_R^T \underline{\mathbf{h}} = \underline{\mathbf{h}}^T \underline{\mathbf{s}}_R \\ y_N(b) = \underline{\mathbf{b}}_R^T \underline{\mathbf{h}} = \underline{\mathbf{h}}^T \underline{\mathbf{b}}_R \\ \underline{\mathbf{s}}_R^T \underline{\mathbf{h}}^* = 1 \end{cases}$$

Dans ces conditions (en tenant compte du fait que $\underline{\mathbf{h}}$ est déterministe) :

$$\begin{cases} y_N^2(s) = (\underline{\mathbf{s}}_R^T \underline{\mathbf{h}})^2 = \underline{\mathbf{h}}^T \underline{\mathbf{s}}_R \underline{\mathbf{s}}_R^T \underline{\mathbf{h}} \\ y_N^2(b) = (\underline{\mathbf{b}}_R^T \underline{\mathbf{h}})^2 = \underline{\mathbf{h}}^T \underline{\mathbf{b}}_R \underline{\mathbf{b}}_R^T \underline{\mathbf{h}} \end{cases} \rightarrow E[Y_N^2(B)] = E[\underline{\mathbf{h}}^T \underline{\mathbf{B}}_R \underline{\mathbf{B}}_R^T \underline{\mathbf{h}}] = \underline{\mathbf{h}}^T E[\underline{\mathbf{B}}_R \underline{\mathbf{B}}_R^T] \underline{\mathbf{h}} = \underline{\mathbf{h}}^T \phi_{BB} \underline{\mathbf{h}}$$

où : $\phi_{BB} = E[\underline{\mathbf{B}}_R \underline{\mathbf{B}}_R^T]$ est la matrice $(N+1) \times (N+1)$ d'autocorrélation (\equiv d'autocovariance dans le cas stationnaire) du bruit $\{B_n\}$ (le terme général de ϕ_{BB} est $\phi_{BB_{i-j}}$ (matrice de Toeplitz))

Ainsi :
$$\rho = \frac{y_N^2(s)}{E[Y_N^2(B)]} = \frac{\underline{\mathbf{h}}^T \underline{\mathbf{s}}_R \underline{\mathbf{s}}_R^T \underline{\mathbf{h}}}{\underline{\mathbf{h}}^T \phi_{BB} \underline{\mathbf{h}}} \quad \rho^* = \frac{\underline{\mathbf{h}}^{*T} \underline{\mathbf{s}}_R \underline{\mathbf{s}}_R^T \underline{\mathbf{h}}^*}{\underline{\mathbf{h}}^{*T} \phi_{BB} \underline{\mathbf{h}}^*}$$

Comme ρ^* est maximal par rapport à $\underline{\mathbf{h}}$, on a nécessairement : $\rho \leq \rho^*$ donc :

$$a(\underline{\mathbf{h}}) \stackrel{\Delta}{=} \underline{\mathbf{h}}^T \underline{\mathbf{s}}_R \underline{\mathbf{s}}_R^T \underline{\mathbf{h}} - \rho^* \underline{\mathbf{h}}^T \phi_{BB} \underline{\mathbf{h}} \leq 0 \quad \forall \underline{\mathbf{h}} \quad \text{car : } \underline{\mathbf{h}}^T \underline{\mathbf{s}}_R \underline{\mathbf{s}}_R^T \underline{\mathbf{h}} = (\underline{\mathbf{s}}_R^T \underline{\mathbf{h}})^2 \geq 0 \quad \text{et} \quad \underline{\mathbf{h}}^T \phi_{BB} \underline{\mathbf{h}} = E[Y_N^2(B)] \geq 0$$

et : $a(\underline{\mathbf{h}}) = 0$ pour $\underline{\mathbf{h}} = \underline{\mathbf{h}}^*$

$a(\underline{\mathbf{h}})$ étant une fonction (de plusieurs variables) qui passe par un maximum pour $\underline{\mathbf{h}} = \underline{\mathbf{h}}^*$, on a :

$$\underline{\mathbf{a}}'(\underline{\mathbf{h}}^*) = \underline{\mathbf{0}} \quad \text{où :} \quad \underline{\mathbf{a}}' = \begin{bmatrix} \frac{\partial a}{\partial h_0} \\ \frac{\partial a}{\partial h_1} \\ \dots \\ \frac{\partial a}{\partial h_N} \end{bmatrix}$$

On rappelle les règles de dérivation matricielle: $\frac{d(\underline{\mathbf{h}}^2)}{d\underline{\mathbf{h}}} = 2\underline{\mathbf{h}}$, $\frac{d(\underline{\mathbf{s}}_R^T \underline{\mathbf{h}})}{d\underline{\mathbf{h}}} = \underline{\mathbf{s}}_R$, $\frac{d[(\underline{\mathbf{s}}_R^T \underline{\mathbf{h}})^2]}{d\underline{\mathbf{h}}} = 2\underline{\mathbf{s}}_R \underline{\mathbf{s}}_R^T \underline{\mathbf{h}}$ et $\frac{d(\underline{\mathbf{h}}^T \phi_{BB} \underline{\mathbf{h}})}{d\underline{\mathbf{h}}} = 2\phi_{BB} \underline{\mathbf{h}}$

d'où : $\underline{\mathbf{a}}'(\underline{\mathbf{h}}) = 2\underline{\mathbf{s}}_R \underline{\mathbf{s}}_R^T \underline{\mathbf{h}} - 2\rho^* \phi_{BB} \underline{\mathbf{h}}$ et comme : $\underline{\mathbf{a}}'(\underline{\mathbf{h}}^*) = \underline{\mathbf{0}}$ et : $\underline{\mathbf{s}}_R^T \underline{\mathbf{h}}^* = 1$

on a : $\underline{\mathbf{a}}'(\underline{\mathbf{h}}^*) = 2\underline{\mathbf{s}}_R \underline{\mathbf{s}}_R^T \underline{\mathbf{h}}^* - 2\rho^* \phi_{BB} \underline{\mathbf{h}}^* = \underline{\mathbf{0}} \rightarrow \underline{\mathbf{s}}_R = \rho^* \phi_{BB} \underline{\mathbf{h}}^*$

Si ϕ_{BB}^{-1} existe (ϕ_{BB} régulière \equiv de déterminant non nul), on a donc :

$$\underline{\mathbf{h}}^* = \frac{1}{\rho^*} \phi_{BB}^{-1} \underline{\mathbf{s}}_R$$

On peut calculer ρ^* d'après la relation : $\underline{\mathbf{s}}_R^T \underline{\mathbf{h}}^* = 1$ et en utilisant la relation précédente : $\underline{\mathbf{s}}_R^T \underline{\mathbf{h}}^* = \frac{1}{\rho^*} \underline{\mathbf{s}}_R^T \phi_{BB}^{-1} \underline{\mathbf{s}}_R = 1$

soit :

$$\rho^* = \underline{\mathbf{s}}_R^T \phi_{BB}^{-1} \underline{\mathbf{s}}_R$$

Cas particulier important : $\{B_n\}$ est un bruit blanc de variance σ^2 :

Pour un bruit blanc de variance σ^2 , on a : $\phi_{BB} = \sigma^2 \mathbf{I}$

Et les formules établies : $\rho^* = \underline{\mathbf{s}}_R^T \phi_{BB}^{-1} \underline{\mathbf{s}}_R$ et : $\underline{\mathbf{h}}^* = \frac{1}{\rho^*} \phi_{BB}^{-1} \underline{\mathbf{s}}_R$ deviennent :

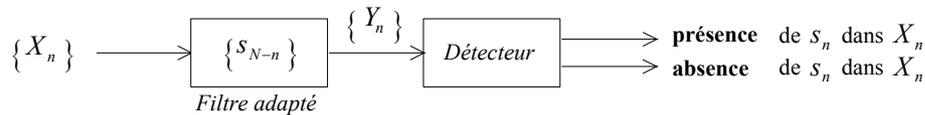
$$\rho^* = \frac{\underline{\mathbf{s}}_R^T \underline{\mathbf{s}}_R}{\sigma^2} = \frac{\sum_{n=0}^N s_n^2}{\sigma^2}$$

$$\underline{\mathbf{h}}^* = \frac{1}{\sum_{n=0}^N s_n^2} \underline{\mathbf{s}}_R$$

Comme $\underline{\mathbf{h}}^*$ est défini à une Constante près, on peut prendre : $\underline{\mathbf{h}}^* = \underline{\mathbf{s}}_R$ soit : $h_n^* = s_{N-n} \quad n = 0, \dots, N$

ce qui revient à prendre : $\sum_{n=0}^N s_n^2 = 1$.

Le filtre adapté au signal $\{s_n\}$ (au sens de la maximisation du Rapport Signal sur Bruit à l'instant N) a donc pour RI l'image (dans un miroir) du signal $\{s_n\}$ (image obtenue en effectuant un renversement du temps).



Remarques

▪ La connaissance du signal $\{s_n\}$ à détecter est nécessaire, ainsi que l'estimation de l'autocovariance du bruit.

▪ Le filtre adapté effectue la convolution : $Y_n = \sum_{i=0}^N X_{n-i} s_{N-i}$ soit en posant : $m = N - i$:

$$Y_n = \sum_{m=0}^N X_{m+n-N} s_m$$

A un coefficient près multiplicateur $\frac{1}{N}$, cette relation a la structure d'un corrélateur, qui effectuerait l'intercorrélacion entre $\{X_n\}$ et $\{s_n\}$. C'est pourquoi le filtre adapté est parfois assimilé à un intercorrélacion.

Il faut cependant bien noter les différences de fonctionnement :

- dans un intercorrélacion, N est grand et $n \ll N$ alors que :
- pour un filtre adapté, N peut être faible et $0 \leq n \leq N$.

TP 4. Conditionnement - Filtrage. Détection

CONDITIONNEMENT - FILTRAGE

$N := 255$ $n := 0..N$

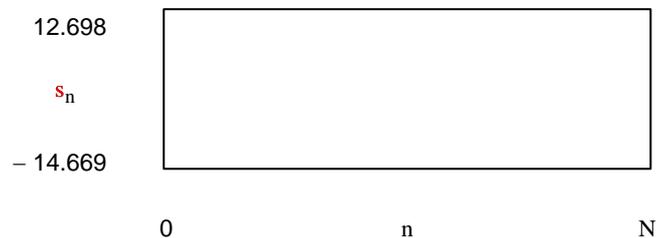
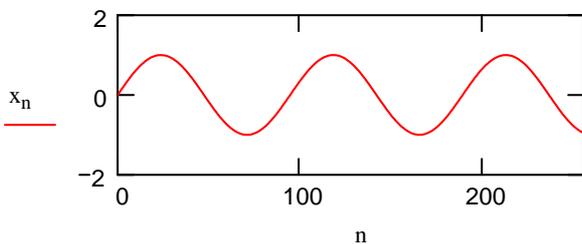
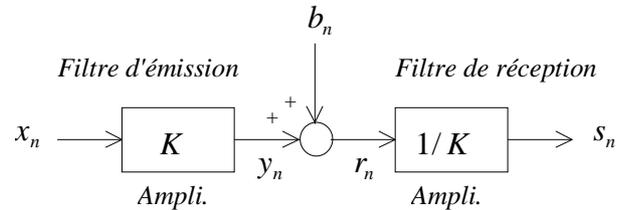
Types de bruit : $b1 := 5 \cdot \text{gaussn}(N + 1)$ $b2 := 5 \cdot \text{whiten}(N + 1)$

Signal émis (non bruité) : $x_n := \sin\left(\frac{n}{15}\right)$

Bruitage (canal de transmission) : $b := b1$

1. Amplification (amont) / atténuation (aval) :

$K := 1$ $y_n := \blacksquare$ $r_n := \blacksquare$ $s_n := \blacksquare$



Conclusion :

2. Préaccentuation / désaccentuation :

$P0 := 1$ $X := \text{fft}(\text{vautocov}(x))$ $B := \text{fft}(\text{vautocov}(b))$

$L := \text{length}(X)$

$m := 0..L - 1$ FT filtre désaccentuation: $Hd_m := \blacksquare$

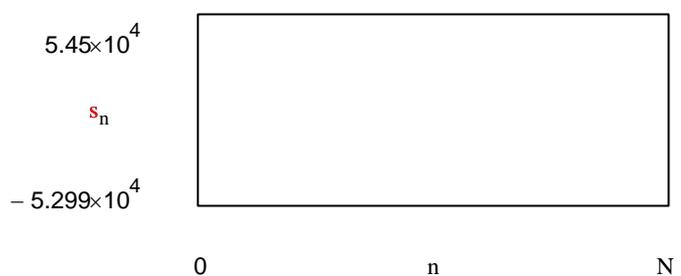
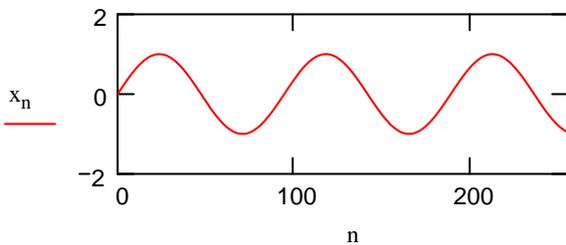
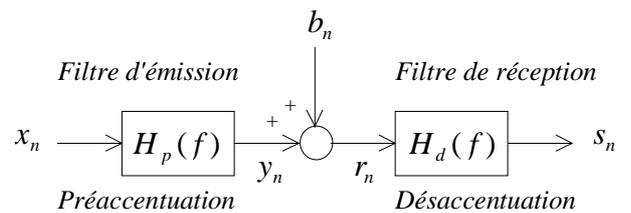
FT filtre préaccentuation: $Hp_m := \blacksquare$

RI filtre désaccentuation: $hd := \blacksquare$

RI filtre préaccentuation: $hp := \blacksquare$

$y := \blacksquare$ $r_n := \blacksquare$

$s := \blacksquare$



Conclusion :

DETECTION

RAZ : $s := \text{submatrix}(s, 0, 0, 0, 0)$

1. Filtrage adapté :

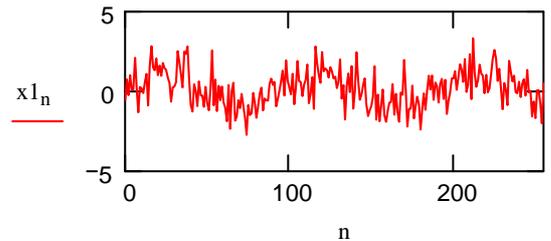
$N := 255$ $n := 0..N$

Signal non bruité : $s_n := \sin\left(\frac{n}{15}\right)$

Types de bruit : $b1 := \text{gaussn}(N + 1)$ $V := 1$ $b2 := \text{BruitBlanc}(N + 1, 0, V)$

Méthode générale :

Signal reçu : $b := b1$ $x1_n := s_n + b_n$ $x2_n := b_n$



Matrice d'autocorrélation du bruit :

$br_n := b_{N-n}$ $\Phi := \blacksquare$

SNR :

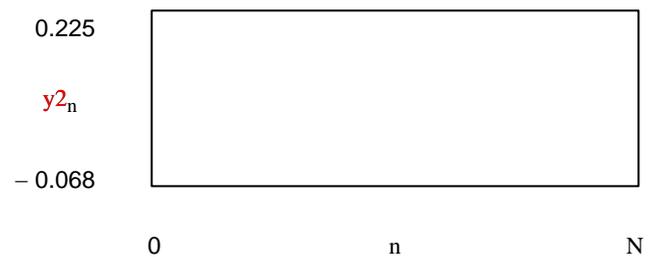
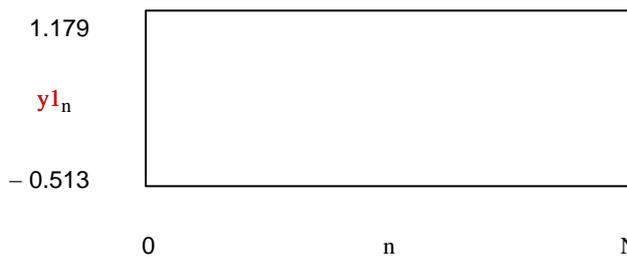
$sr_n := s_{N-n}$ $\rho := \blacksquare$

RI du filtre adapté :

$h := \blacksquare$

Sortie du filtre adapté :

$y1 := \text{convol}(x1, h)$ $y2 := \text{convol}(x2, h)$

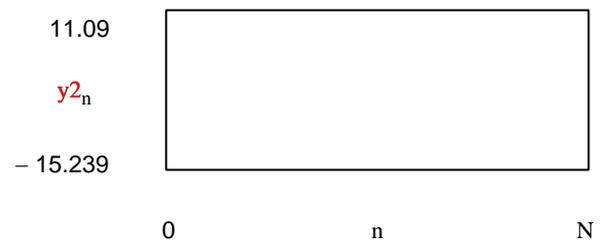
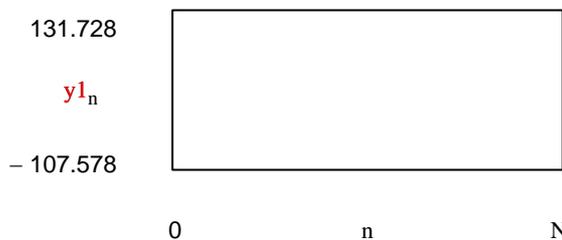


Conclusion:

Cas particulier : bruit blanc de variance V :

Signal reçu : $b := b2$ $x1_n := s_n + b_n$ $x2_n := b_n$

$sr_n := s_{N-n}$ $\rho := \blacksquare$ $h := \blacksquare$ $y1 := \text{convol}(x1, h)$ $y2 := \text{convol}(x2, h)$



Fonctions Bibliothèque :

Moyenne m, Vecteur r et Matrice d'Autocovariance r et Variance V du signal x de dim K (par ergodicité) :

. Moyenne :

$$\text{moy}(x) \equiv \begin{cases} K \leftarrow \text{length}(x) \\ m \leftarrow \frac{1}{K} \cdot \sum_{n=0}^{K-1} x_n \\ m \end{cases}$$

. Vecteur d'AutoCovariance :

$$\text{vautocov}(x) \equiv \begin{cases} K \leftarrow \text{length}(x) \\ m \leftarrow \text{moy}(x) \\ \text{for } k \in 0..K-1 \\ r_k \leftarrow \frac{1}{K} \cdot \sum_{n=0}^{K-k-1} (x_{n+k} - m) \cdot (x_n - m) \\ r \end{cases}$$

. Matrice d'AutoCovariance :

$$\text{mautocov}(x) \equiv \begin{cases} K \leftarrow \text{length}(x) \\ r \leftarrow \text{vautocov}(x) \\ \text{for } i \in 0..K- \\ \text{for } j \in 0..K- \\ R_{i,j} \leftarrow r_{|i-j|} \\ R \end{cases}$$

. Variance :

$$\text{Var}(x) \equiv \text{vautocov}(x)_0$$

Fonction YuleWalker(φ_{xx}, N) : Résolution du système de Yule-Walker Ra = v par inversion matricielle

Paramètres d'entrée : . φ_{xx} : vecteur fonction d'autocovariance (dim K) du signal x (dim K) : φ_{xx0}, φ_{xx1} ... φ_{xxK-1}
 . N : dim de l'ordre du filtre (formeur/LPC) AR

Paramètre de sortie : . le vecteur a (dim N+1) :
 a₀ = V : variance (du bruit blanc si filtre formeur); a₁, a₂, ... a_N : paramètres du filtre (formeur/LPC) AR d'ordre N

$$\text{YuleWalker}(\phi_{xx}, N) \equiv \begin{cases} \text{for } i \in 0..N \\ \text{for } j \in 0..N \\ R_{i,j} \leftarrow \phi_{xx} |_{i-j} \\ b_0 \leftarrow 0 \\ \text{for } i \in 1..N \\ b_i \leftarrow \phi_{xx_i} \\ \alpha \leftarrow \left(\begin{matrix} -1 \\ \phi_{xx_0} \end{matrix} \right) \cdot R^{-1} \cdot b \\ a_0 \leftarrow \frac{\phi_{xx_0}}{\phi_{xx_0} \cdot \alpha_0 + 1} \\ \text{for } i \in 1..N \\ a_i \leftarrow a_0 \cdot \alpha_i \\ a \end{cases}$$

Exemple : K := 3 r₀ := 6
 N := 2 r₁ := 3
 r₂ := 0

a := YuleWalker(r, N)

V := a₀

$$a = \begin{pmatrix} 4 \\ -0.667 \\ 0.333 \end{pmatrix}$$

V = 4
 i := 1..N
 a_i =

-0.667
0.333

Fonction YuleW(x,N) : Résolution du système de Yule-Walker $Ra = v$ par inversion matricielle

Paramètres d'entrée : . x : signal x (dim K) : $x_0, x_1 \dots x_{K-1}$
 . N : dim de l'ordre du filtre (formeur/LPC) AR

Paramètre de sortie : . le vecteur a (dim N+1) :
 $a_0 = V$: variance (du bruit blanc si filtre formeur); $a_1, a_2, \dots a_N$: paramètres du filtre (formeur/LPC) AR d'ordre N

$$\text{YuleW}(x,N) \equiv \begin{cases} r \leftarrow \text{vautocov}(x) \\ a \leftarrow \text{YuleWalker}(r,N) \\ a \end{cases}$$

Bruit blanc de K points de moyenne m et de variance V :

Bruit blanc de K points centré de variance 0.083 :

$$\text{WhiteNoise}(K) \equiv \begin{cases} A \leftarrow 281 \\ P \leftarrow 31357 \\ v_0 \leftarrow 100 \\ \text{for } n \in 1..N-1 \\ \quad v_n \leftarrow \text{mod}(A \cdot v_{n-1}, P) \\ \text{for } n \in 0..N-1 \\ \quad w_n \leftarrow \frac{1}{P} \cdot v_n - 0.5 \\ w \end{cases}$$

Bruit blanc de K points de moyenne m et de variance V :

$$\text{BruitBlanc}(K, m, V) \equiv \sqrt{12 \cdot V} \cdot \text{whiten}(K) + m$$

$$\text{BruitBlanc2}(K, m, V) \equiv \sqrt{12 \cdot V} \cdot \text{WhiteNoise}(K) + m$$

Réponse reponse (vecteur dim. K) du filtre AR d'ordre N (coeffs a_i $0 \leq i \leq N$) au signal x (vecteur dim. K)

$$\text{reponse}(x, a, N) \equiv \begin{cases} K \leftarrow \text{length}(x) \\ \text{for } k \in 0..K-1 \\ \quad \begin{cases} y_k \leftarrow x_k & \text{if } k = 0 \\ y_k \leftarrow x_k - \sum_{i=0}^k a_i \cdot y_{k-i} & \text{if } 1 \leq k < N \\ y_k \leftarrow x_k - \sum_{i=0}^N a_i \cdot y_{k-i} & \text{if } k \geq N \end{cases} \\ y \end{cases}$$

TFD

$$\text{TFD}(x) \equiv \left| \begin{array}{l} M \leftarrow \text{length}(x) \\ \text{for } k \in 0..M-1 \\ X_k \leftarrow \frac{1}{M} \cdot \sum_{n=0}^{M-1} x_n \cdot e^{-i \cdot 2 \cdot \pi \cdot k \cdot \frac{n}{M}} \\ X \end{array} \right.$$

ou encore :

$$\text{TFD2}(x) \equiv \left| \begin{array}{l} M \leftarrow \text{length}(x) \\ \text{for } k \in 0..M-1 \\ R \leftarrow 0 \\ I \leftarrow 0 \\ \text{for } n \in 0..M-1 \\ R \leftarrow R + x_n \cdot \cos\left(2 \cdot \pi \cdot k \cdot \frac{n}{M}\right) \\ I \leftarrow I - x_n \cdot \sin\left(2 \cdot \pi \cdot k \cdot \frac{n}{M}\right) \\ X_k \leftarrow \frac{1}{M} \cdot (R + i \cdot I) \\ \text{return } X \end{array} \right.$$

TFDI

$$\text{TFDI}(X) \equiv \left| \begin{array}{l} M \leftarrow \text{length}(X) \\ \text{for } n \in 0..M-1 \\ x_n \leftarrow \sum_{k=0}^{M-1} X_k \cdot e^{i \cdot 2 \cdot \pi \cdot k \cdot \frac{n}{M}} \\ x \end{array} \right.$$

Convolution causale (signaux de même longueur)

$$\text{convcausal2}(x, h) \equiv \left| \begin{array}{l} M \leftarrow \text{length}(x) \\ \text{for } n \in 0..M-1 \\ y_n \leftarrow \sum_{k=0}^n x_k \cdot h_{n-k} \\ y \end{array} \right.$$

Convolution causale (signaux de longueur différente)

$$\text{convcausal}(x, h) \equiv \left| \begin{array}{l} M \leftarrow \text{length}(x) \\ N \leftarrow \text{length}(h) \\ \text{for } n \in 0..M-1 \\ y_n \leftarrow \sum_{k=0}^n x_k \cdot h_{n-k} \text{ if } n < N \\ y_n \leftarrow 0 \text{ otherwise} \\ y \end{array} \right.$$

Transformée de Hilbert

$$\text{Hilbert}(x) \equiv \left| \begin{array}{l} M \leftarrow \text{length}(x) \\ f_0 \leftarrow 0 \\ \text{for } n \in 1..M-1 \\ f_n \leftarrow \frac{1}{\pi \cdot n} \\ \text{TH} \leftarrow \text{convcausal}(x, f) \\ \text{TH} \end{array} \right.$$

5. Transmission

Codage. Egalisation. Extraction de signal immergé dans du bruit. Filtrage adaptatif

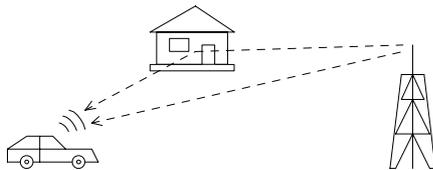
INTRODUCTION

Objectif

Le but est de transmettre un message porteur d'information d'un dispositif de départ appelé *émetteur* vers un dispositif d'arrivée appelé *récepteur* par l'intermédiaire d'un *canal de transmission*.

Les dégradations que peut subir le signal dans le canal de transmission sont multiples : distorsions, déformations spectrales, bruit, écho, ...

La figure suivante illustre un cas concret de canal introduisant un écho : la transmission d'un message d'une tour émettrice vers un véhicule :



Dans cette situation, l'onde émise peut emprunter 2 chemins pour atteindre le récepteur : un trajet direct et un trajet réfléchi sur un bâtiment. Les temps de propagation étant différents sur ces 2 trajets, le récepteur observe la superposition de 2 versions décalées du signal émis.

Par exemple, si l'on émet une suite de symboles a_k (k est entier et désigne le temps), on reçoit : $\alpha a_k + \beta a_{k-m}$ (α, β sont réels et m désigne un retard en nombre d'échantillons) : il y a interférence entre les symboles a_k et a_{k-m} .

L'objectif de l'égalisation est de restituer le message émis à partir de cette observation (qui est un version filtrée de la séquence transmise) : c'est l'annulation d'écho.

Moyens

Dans le vide, les ondes électromagnétiques se propagent très bien et les sons très mal. Dans l'eau, c'est l'inverse.

Il est ainsi évident que la propagation dépend essentiellement du milieu : il est généralement plus facile de transmettre un message par un câble coaxial de 1 m de long qu'entre la terre et la lune !

Dans tous les cas, **il est impératif d'adapter le message au canal de communication et de traiter le signal brut reçu pour en extraire le message originel.**

Pour ce faire, de nombreux dispositifs intermédiaires sont nécessaires à un système de télécommunications :

Emetteur

- . Un *dispositif de numérisation* des données.
- . Un *dispositif de compression et de codage*: la redondance du message émis autorise la compression (avec ou sans perte) :
 - . codage de source : suppression de la redondance du message à transmettre
 - . codage canal : introduction d'une redondance du message à transmettre (si le canal perturbe la transmission) en vue de la détection et correction des erreurs de transmission au niveau du récepteur.
- . Un *modulateur* : son rôle est d'adapter le signal aux propriétés spectrales du canal de transmission. Par exemple, les fréquences audio (de 20 Hz à 20 kHz) se propagent très mal sous la forme d'ondes électromagnétiques et il est nécessaire de translater le spectre d'un message audio autour d'une fréquence plus élevée pour en assurer la transmission. Cette translation doit naturellement s'effectuer sans altérer l'information portée par le signal. La modulation est également justifiée par le fait qu'un même canal de transmission peut être utilisé par plusieurs systèmes de communication (multiplexage).
- . Une *antenne* : elle rayonne et diffuse le signal issu du modulateur dans le milieu pour lui permettre de se propager (antennes paraboliques pour les ondes Hertiennes, hauts-parleurs ou hydrophones pour les ondes acoustiques, ...).

Récepteur

- . Une *antenne* dont le rôle est de capter le message.
- . Un *démodulateur* destiné à ramener le signal reçu vers la bande de base (bande spectrale du message avant modulation).
- . Un *dispositif de récupération de porteuse* permettant de compenser un éventuel décalage entre la fréquence porteuse utilisée par le modulateur et celle utilisée par le démodulateur.
- . Un *dispositif d'échantillonnage*, préliminaire nécessaire à tout traitement numérique. Cet échantillonnage doit en général être commandé par un *dispositif de synchronisation symbole* permettant de synchroniser échantillonnage en réception avec le rythme d'émission des symboles.
- . Un *égaliseur* dont le rôle est de compenser les distorsions introduites par le canal et les différents dispositifs électroniques de transmission. Sa structure exacte dépend de la complexité de l'application considérée.
- . Des *systèmes de décodage et de décompression*.

Il est clair que la description précédente ne donne qu'une image grossière de ce que peut être un système concret. En particulier, la décomposition du traitement global en modules indépendants est souvent artificielle. Nombre de dispositifs efficaces estiment conjointement plusieurs des paramètres nécessaires au bon fonctionnement du récepteur.

Modélisation mathématique

Du point de vue mathématique, les opérations décrites précédemment s'écrivent grossièrement de la manière suivante :

Les symboles porteurs d'information

Le signal à transmettre est une suite $\{a_k\}$ de valeurs complexes appartenant à un ensemble fini.

Exemple : Modulation de phase (PSK Phase Shift Keying)

$a_k \in \{-1, +1\}$ pour une modulation MDP2, Modulation par Déplacement de Phase à 2 états (*BPSK Binary Phase Shift Keying*). $a_k \in \{-1, +1, -i, +i\}$ pour une modulation MDP4 à 4 états.

Exemple : Modulation de 2 porteuses en quadrature (QAM Quadrature Amplitude Modulation)

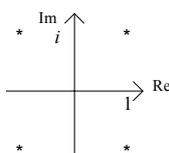
$a_k \in \{\pm 1 \pm i, \pm 2 \pm i, \pm 2 \pm 2i, \pm 1 \pm 2i\}$ pour une modulation QAM à 16 états. Lorsqu'une seule des 2 porteuses est utilisée, la modulation QAM se réduit à une simple modulation d'amplitude. Par exemple, $a_k \in \{-2, -1, +1, +2\}$ correspond à une modulation d'amplitude (*AM Amplitude Modulation, ASK Amplitude Shift Keying*) à 4 états.

Données M – aires . Constellation

En général, les a_k peuvent prendre M valeurs différentes. On parle de données M – aires . Les différents états possibles peuvent être représentés dans le plan complexe. La représentation dans le plan complexe des différentes valeurs prises par le signal au cours du temps s'appelle une constellation.

Constellation d'un signal : représentation dans le plan complexe des différentes valeurs prises (alphabet) par ce signal.

Exemple : Constellation d'une modulation à 4 états de phase $a_k \in \{1 + i, -1 + i, -1 - i, 1 - i\}$



La mise en forme

La suite $\{a_k\}$ des symboles à transmettre doit être séquencée dans le temps. On émet les symboles a_k à un rythme régulier. La durée séparant l'émission de 2 symboles successifs est notée T_0 . Elle s'appelle la durée symbole. A partir de la suite des valeurs complexes $\{a_k\}$, on construit le signal à TC $a(t)$:

$$a(t) = \sum_k a_k \delta(t - kT_0)$$

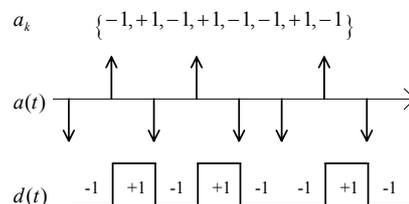
La convolution de $a(t)$ par une impulsion de mise en forme $\Pi(t)$ fournit le signal à TC :

$$d(t) = \sum_k a_k \Pi(t - kT_0)$$

La fonction de mise en forme Π la plus simple est la porte :

$$\Pi_{T_0}(x) = \begin{cases} 1 & \text{pour } 0 \leq x < T_0 \\ 0 & \text{sinon} \end{cases}$$

Exemple : Mise en forme, par une fonction porte, de 8 symboles consécutifs d'une modulation à 2 états (± 1) :



Mise en forme d'un message numérique

Remarque :

Il est en général préférable d'utiliser une impulsion de mise en forme $\Pi(t)$ plus douce que la porte, afin de limiter l'encombrement spectral. En effet, plus une fonction est lisse (absence de transitions rapides, de fronts raides, de marches d'escalier), plus sa Transformée de Fourier (donc son spectre) décroît rapidement, limitant ainsi l'encombrement spectral.

La modulation

La modulation vise à translater le spectre du signal Basse-Fréquence (BF) à émettre $d(t)$ (signal modulant) autour d'une fréquence plus élevée ν_0 (porteuse) tout en préservant l'information supportée par ce signal.

La translation du spectre de $d(t)$ autour d'une fréquence porteuse ν_0 conduit au signal Haute-Fréquence (HF) $e(t)$ effectivement transmis dans le milieu physique.

Partant d'une fonction sinusoidale porteuse $A \cos(2\pi\nu_0 t + \varphi)$, l'idée de base d'une modulation consiste à faire varier lentement par rapport à ν_0 (variation comme le signal à transmettre), l'un des paramètres A , ν_0 ou φ . On parle respectivement de modulation d'amplitude, de fréquence ou de phase.

Lorsque les variations du paramètre choisi sont effectivement lentes par rapport à la fréquence de la porteuse ν_0 , le signal obtenu (signal modulé) est à bande étroite autour de ν_0 .

Les modulations sont généralement linéaires car alors l'égalisation est facilitée. Dans ce cas, la translation en fréquence (convolution par un Dirac) s'effectue en multipliant $d(t)$ par une exponentielle complexe en temps. Le signal réellement émis s'obtient en prenant la partie réelle (ce qui a pour effet de symétriser le spectre du signal réellement émis).

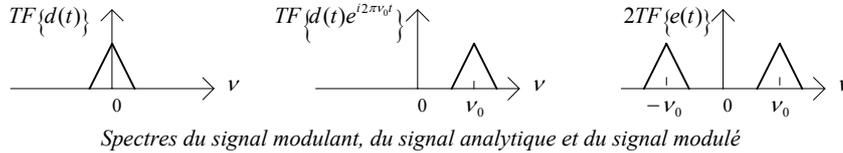
Ainsi le signal HF émis s'écrit : $e(t) = \text{Re}\{d(t)e^{i2\pi\nu_0 t}\}$

En résumé :

$d(t)$ est un signal BF à transmettre

$d(t)e^{i2\pi\nu_0 t}$ est à bande étroite autour de ν_0

$e(t)$ comporte 2 bandes de fréquences : l'une autour de ν_0 et l'autre autour de $-\nu_0$.



On note P et Q les parties réelle et imaginaire du signal transmis :

$$a_k = P_k + iQ_k \quad k \text{ entier}$$

$$d(t) = P_t + iQ_t \quad t \text{ réel}$$

Le signal modulé $e(t)$ s'écrit :

$$e(t) = \text{Re}\{d(t)e^{i2\pi\nu_0 t}\} = P_t \cos(2\pi\nu_0 t) - Q_t \sin(2\pi\nu_0 t) = \sqrt{P_t^2 + Q_t^2} \cos\left\{2\pi\nu_0 t + \arctan\left(\frac{Q_t}{P_t}\right)\right\}$$

Cette relation montre que $e(t)$ peut être vu :

- . comme la modulation de 2 porteuses en quadrature
- . comme une modulation de phase et d'amplitude.

Evidemment, il est possible de n'exploiter que l'une de ces 2 possibilités pour réaliser une modulation d'amplitude ou une modulation de phase. (Par exemple, si le signal $d(t)$ est réel, seule l'amplitude est modulée).

Pour une modulation de fréquence, $\arctan\left(\frac{Q_t}{P_t}\right)$ peut être pris comme fonction linéaire non stationnaire du temps :

$$\arctan\left(\frac{Q_t}{P_t}\right) = R_t t$$

Le canal de transmission

Le signal $e(t)$ se propage dans le canal de transmission. Ce canal est presque toujours supposé linéaire. Ainsi, sa sortie s'exprime comme convolution de son entrée par sa RI. Un canal linéaire peut être caractérisé de manière équivalente par sa Réponse en Fréquence (TF de sa RI). Le signal $e(t)$ injecté dans le canal étant à bande étroite autour de ν_0 , le signal $r(t)$ arrivant au niveau du récepteur l'est aussi lorsque l'hypothèse de linéarité est vérifiée. Il s'écrit :

$$r(t) = \text{Re}\{\tilde{r}(t)e^{i2\pi\nu_0 t}\} \quad \text{où } \tilde{r}(t) \text{ est un signal BF complexe.}$$

La démodulation

La démodulation est l'opération inverse de la modulation. Le but est de restituer au mieux le signal modulant, BF, à partir du signal modulé. Pour cela, il est nécessaire de translater $r(t)$ vers la bande de base (bande de fréquence initiale avant modulation). En notant $r_a(t) = \tilde{r}(t)e^{i2\pi\nu_0 t}$ le signal analytique associé à $r(t)$, la démodulation produit le signal :

$$x(t) = r_a(t)e^{-i(2\pi\nu_0 t + \phi_t)} \quad \text{où } \phi_t \text{ la phase } \phi_t \text{ traduit le déphasage entre les porteuses de l'émetteur et du récepteur (} \phi_t \text{ varie linéairement au cours du temps).}$$

Les traitements en bande de base

L'échantillonnage du signal démodulé $x(t)$ conduit à une suite de valeurs $x(\tau_k + kT_0)$.

Lorsque le canal et tous les systèmes de traitement sont quasi parfaits, le signal n'a subi que peu de distorsions. De ce fait, l'estimation du message émis découle immédiatement de :

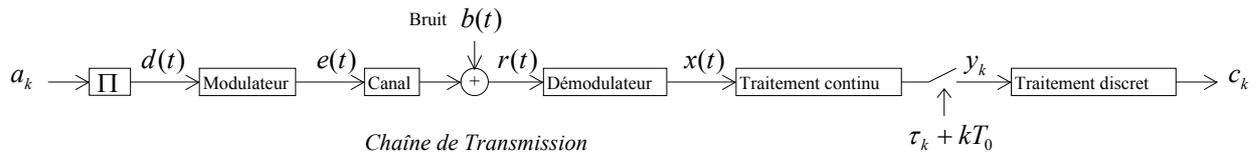
$$x(t) \approx d(t) \quad \text{d'où :} \quad x(\tau_k + kT_0) \approx a_k$$

En pratique, ce n'est jamais le cas. Les principales sources d'erreur sont les suivantes :

1. La RI du canal n'est pas idéale (RI idéale \equiv égale à une impulsion de Dirac)
2. L'échantillonneur à la réception n'est pas synchrone avec celui utilisé lors de l'émission ou à une période d'échantillonnage différente de celle (T_0) de l'échantillonneur d'émission (τ_k non constant).
3. La fréquence porteuse ν utilisée par le démodulateur diffère de la fréquence ν_0 utilisée par le modulateur (fading).

Il est donc impératif de traiter le signal de manière à estimer au mieux les symboles émis. Les principaux systèmes permettant de compenser les distorsions sont les suivants, dont le numéro indique le défaut précédent correspondant :

1. L'égalisation réduit les distorsions liées essentiellement aux imperfections du canal de transmission.
2. La synchronisation ou récupération de rythme compense la désynchronisation entre les échantillonneurs émetteur et récepteur.
3. La récupération de porteuse lutte contre le fait que la fréquence porteuse utilisée par le modulateur diffère de celle qu'utilise le démodulateur. Le système de récupération de porteuse permet également de compenser les déphasages liés à l'effet Doppler (l'effet Doppler apparaît quand l'émetteur ou le récepteur ou les deux sont en mouvement : exemple : son (de fréquence variable pour un piéton) émis par une voiture dépassant le piéton).



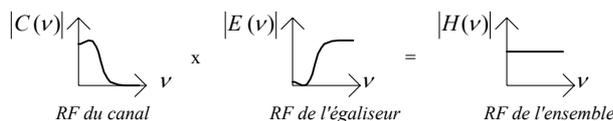
Egalisation

Dans le domaine des télécommunications, les données émises traversent un canal de transmission et les différents dispositifs électroniques associés à l'émetteur et au récepteur. Le rôle de l'égaliseur est alors de réduire au mieux les distorsions apportées par ces éléments.

Aspect fréquentiel

En l'absence de bruit, l'égaliseur est simplement un filtre inverse dont le rôle est de rendre plate la Réponse Fréquentielle (RF) de l'ensemble canal-égaliseur.

Lorsque le canal de transmission est bruité, un simple filtre inverse amplifie considérablement le bruit (le bruit occupe les Hautes Fréquences et comme le canal a une RF du type passe-bas, son inverse est passe-haut amplifiant ainsi le bruit) et un autre critère est utilisé dans le même but (un critère quadratique (moindres carrés) minimisant l'énergie de l'erreur entre la réponse du canal-égaliseur réel et celle du canal-égaliseur idéal).



Aspect temporel

Lors d'une transmission, des symboles élémentaires sont émis régulièrement (toutes les T_0 secondes). Si par exemple, la durée de la RI du canal est inférieure à l'écart de temps entre 2 symboles consécutifs, ceux-ci restent temporellement disjoints en sortie du canal de transmission. Dans le cas contraire, on ne peut plus les distinguer simplement les uns des autres : un traitement du signal (égalisation) s'impose.

Diagramme en oeil

Considérons un exemple cas d'école : le signal émis se compose d'une suite aléatoire blanche de 1 et de -1. La représentation adoptée ici utilise 16 points par durée symbole. La FT du canal comporte 1 pôle α et s'écrit :

$$H(z) = \frac{1 - \alpha}{1 - \alpha z^{-1}} \quad \text{où } z^{-1} \text{ représente ici un retard de } 1/16 \text{ de symbole.}$$

La valeur de α utilisée est complexe : $\alpha = 0.95 e^{j}$, induisant un comportement oscillant de la réponse.

La RI du canal est très oscillante, traduisant une résonance marquée de la RF.

La représentation temporelle des signaux d'entrée et de sortie du canal et celle des constellations (constellation d'un signal \equiv représentation dans le plan complexe des différentes valeurs prises par ce signal) permettent de se faire une première idée de la nature du problème.

Par exemple, si la constellation en sortie se comporte de 2 tâches distinctes, il est clair qu'il sera facile de restituer le signal émis. Il suffit pour cela d'utiliser une fonction non-linéaire instantanée de type seuillage. Cette situation se traduit également par le fait que la représentation temporelle du signal de sortie est très proche de celle du signal d'entrée.

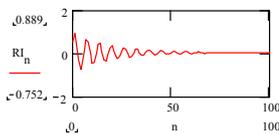
Il s'avère souvent utile de disposer d'autre modes de représentation des signaux. En particulier, il faut noter que l'un des défauts majeurs de la représentation sous la forme de constellation réside dans la totale disparition de l'aspect temporel.

Une manière commode de visualiser le taux d'interférences entre symboles consiste à représenter sur la durée de 1 ou 2 symboles, toutes les configurations possibles de la sortie, associées aux différentes suites de symboles possibles en entrée. Une telle représentation porte le nom de diagramme en oeil.

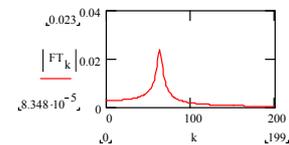
Diagramme en oeil : représentation sur la durée de 1 ou 2 symboles de toutes les configurations possibles de la sortie, associées aux différentes suites de symboles possibles en entrée.

En pratique, l'ouverture de l'oeil constitue un bon indicateur du degré d'interférences entre symboles : plus l'oeil est fermé, plus le taux d'interférences entre symboles est important.

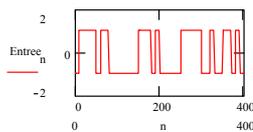
RI du canal



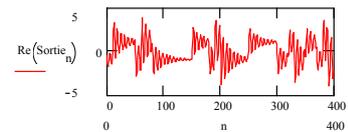
Spectre d'amplitude du canal



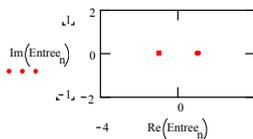
Entrée du canal (16 points/symbole)



Sortie du canal (partie réelle)



Constellation du signal d'entrée du canal



Constellation du signal de sortie du canal

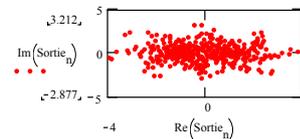
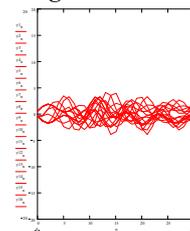


Diagramme en oeil



I. CODAGE - DECODAGE (MODULATION - DEMODULATION)

Modulations analogiques

Pour que l'occupation en fréquence soit minimale, on utilise généralement une porteuse sinusoïdale (la porteuse n'occupe alors qu'une seule fréquence : la fréquence de la sinusoïde).

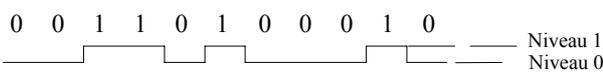
3 façons d'insérer le signal à émettre BF (Basse Fréquence) dans la porteuse HF (Haute Fréquence) :

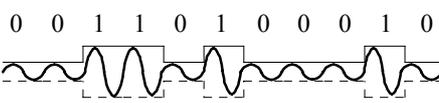
On insère :

- la BF dans l'amplitude de la porteuse : *Modulation d'amplitude AM (Amplitude Modulation)* ou encore *ASK (Amplitude Shift Keying)*
- la BF dans la fréquence de la porteuse : *Modulation de fréquence FM (Frequency Modulation)* ou encore *FSK (Frequency Shift Keying)*
- la BF dans la phase de la porteuse : *Modulation de phase PM (Phase Modulation)* ou encore *PSK (Phase Shift Keying)*

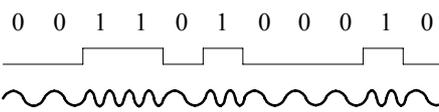
Ex. Transmission d'un signal numérique :

(porteuse sinusoïdale : )

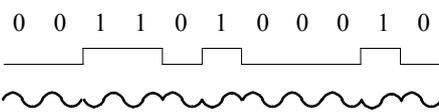
Signal à transmettre :  Niveau 1
Niveau 0

Modulation d'amplitude AM : 

Remarque : Portée d'émission importante (réflexion sur les couches hautes de l'atmosphère) mais au prix d'une moindre qualité (le bruit se greffe directement sur le signal utile, enveloppe du signal modulé)

Modulation de fréquence FM : 

Avantage : Excellente immunité au bruit (le bruit ne se greffe pas directement sur le signal utile mais sur l'enveloppe du signal modulé)

Modulation de phase PM : 

Avantages : Meilleur débit de transmission en plus d'une excellente immunité au bruit (le bruit ne se greffe pas directement sur le signal utile mais sur l'enveloppe du signal modulé)

• porteuse : $s_p(t) = A \cos(2\pi f_p t + \varphi)$ la porteuse est quasi-toujours sinusoïdale

$A \rightarrow A(t) = A + s_m(t)$: c'est la modulation AM ($s_m(t)$: signal à transmettre)

$f_p \rightarrow f_p(t) = f_p + s_m(t)$: c'est la modulation FM

$\varphi \rightarrow \varphi(t) = \varphi_0(t) + s_m(t)$: c'est la modulation PM ($\varphi_0(t) = 2\pi f_p t + \varphi$)

3 types de modulation :

- Modulation analogique continue
- Modulation analogique par impulsions
- Modulation par impulsions codées (Modulation Numérique)

HF : signal porteur, noté $s_p(t)$

BF : signal utile, signal modulant, signal à transmettre noté $s_m(t)$

Pour la modulation d'amplitude : $s_p(t)$ sinusoïdale de fréquence allant de qq. 100 kHz à qq. MHz.

Pour la modulation de fréquence : $s_p(t)$ sinusoïdale de fréquence de l'ordre de 100 MHz.

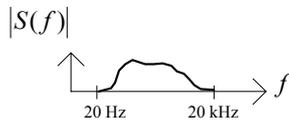
→ Avec une porteuse de fréquence 1 MHz, l'encombrement spectral du signal audio de l'exemple précédent occupant de 20 Hz à 20 kHz, est translaté de 1 MHz et devient maintenant :
de 1.00005 MHz à 1.02 MHz → l'antenne est alors adaptée à toute la plage (et une dimension raisonnable).

Traitement du Signal 5. Transmission (Codage. Egalisation. Extraction de signal immergé dans du bruit. Filtrage adaptatif)

Il suffit à la réception de s'accorder sur la fréquence porteuse, autour de laquelle *la variation de fréquence relative est faible* → différenciation aisée, ≠ à sans modulation : la détection, ou démodulation, consiste alors en un simple filtrage fréquentiel (passe-bas, pour éliminer la porteuse HF et ne conserver que le signal utile BF).

Exemple : Emission sans modulation de 2 signaux de musique (leur spectre va de 20 Hz à 20 kHz) :

Spectre d'amplitude



Spectre occupé (occupation en fréquence)



Les 2 signaux se mélangent si on les transmet tels quels car, à la réception, il est impossible de les séparer.

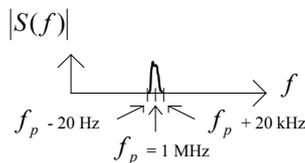
Le principe de la séparation consiste en un filtrage fréquentiel : le récepteur se cale sur (sélectionne) la fréquence (ou la gamme de fréquences) du signal à détecter et élimine le reste des fréquences. Si des signaux occupent la même plage de fréquences, il est donc impossible de les séparer à la réception.

La modulation consiste à effectuer une transposition de fréquences du signal à émettre autour de la fréquence f_p de la porteuse (porteuse sinusoïdale de fréquence $f_p \sim 1$ MHz).

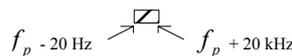
Ainsi, la variation relative de fréquence du signal modulé est très faible et autorise une détection (démodulation, filtrage fréquentiel) permettant de récupérer le signal original en se calant sur la fréquence de la porteuse. Pour transmettre plusieurs signaux BF, il suffit alors d'utiliser pour chacun une fréquence de porteuse différente autour de laquelle le récepteur se calera pour la détection.

Exemple : Emission avec modulation d'un signal de musique (spectre allant de 20 Hz à 20 kHz) ($f_p \sim 1$ MHz) :

Spectre d'amplitude



Spectre occupé (occupation en fréquence)



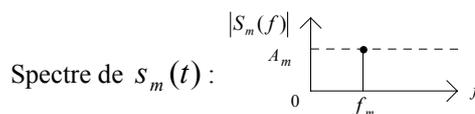
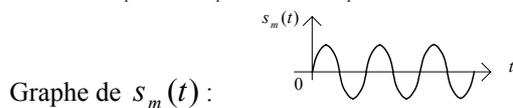
2 porteuses ne peuvent donc avoir la même fréquence sinon on ne peut démoduler (→ l'espace des fréquences est règlementé).

La porteuse constitue donc une enveloppe (au sens du courrier « postal ») dans laquelle on insère le signal à transmettre (c'est la modulation) pour ne pas que les signaux à émettre se mélangent durant le transport. (L'enveloppe est elle-même un signal, le signal porteur, de fréquence unique et de HF). La démodulation doit se charger d'éliminer cette « enveloppe » par filtrage fréquentiel passe-bas (les HF sont éliminées).

Si le signal à transmettre est caractérisé par un spectre très étroit (même à BF), il est clair qu'il n'a pas besoin de porteuse. Mais la plupart des informations ont pour support un signal BF de spectre de fréquences large en valeur relative et doivent donc être modulés.

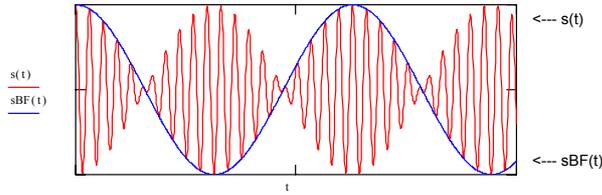
Modulation par multiplication (encore appelée modulation « sans porteuse » car $|S(f)|=0$ à $f = f_p$)

Porteuse: $s_p(t) = A_p \cdot \cos(2\pi f_p t)$ Signal BF (supposé sinusoïdal dans un 1^{er} temps): $s_m(t) = A_m \cdot \cos(2\pi f_m t)$



Représentation temporelle du signal modulé $s(t)$: (signal émis)

$$s(t) = k \cdot s_p(t) \cdot s_m(t) = k A_p A_m \cdot \cos(2\pi f_p t) \cdot \cos(2\pi f_m t)$$

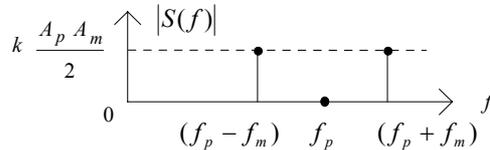


Le signal $s_m(t)$ est, au facteur $k A_p$ près, le signal $s_{BF}(t)$, qui est l'enveloppe, au sens graphique, de $s(t)$.

Spectre $S(f)$ de $s(t)$

$$s(t) = k \frac{A_p A_m}{2} [\cos\{2\pi(f_p + f_m)t\} + \cos\{2\pi(f_p - f_m)t\}]$$

d'où le spectre d'amplitude $|S(f)|$ de $s(t)$ (décomposition en série de Fourier réduite ici à une simple linéarisation



du produit de cosinus) :

Démodulation du signal : $s(t) \rightarrow s_m(t)$

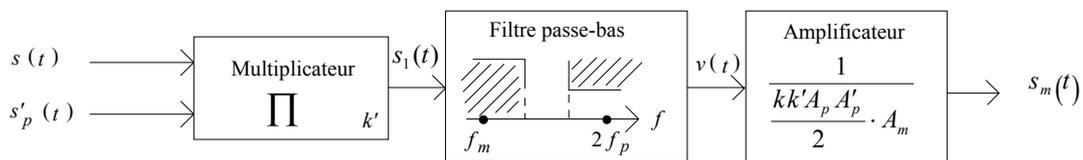
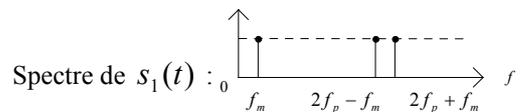
Elle peut être obtenue en multipliant $s(t)$ par une porteuse locale (à la réception) : $s'_p(t)$ avec :

$$s'_p(t) = A'_p \cdot \cos(2\pi f_p t) \text{ puis en filtrant}$$

$$s_1(t) = k' \cdot s'_p(t) \cdot s(t) = k k' A'_p A_p A_m \cos^2(2\pi f_p t) \cdot \cos(2\pi f_m t) = \frac{k k' A'_p A'_p}{2} \cdot A_m [1 + \cos(4\pi f_p t)] \cos(2\pi f_m t)$$

$$s_1(t) = k' \cdot s'_p(t) \cdot s(t) = \underbrace{\frac{k k' A'_p A'_p}{2} \cdot A_m \cdot \cos(2\pi f_m t)}_{v(t)} + \underbrace{A_m \cdot \frac{k k' A'_p A'_p}{4} [\cos\{2\pi(2f_p - f_m)t\} + \cos\{2\pi(2f_p + f_m)t\}]}_{\text{éliminé par un filtre passe-bas}}$$

$$v(t) = s_m(t) \text{ au facteur } \frac{k k' A'_p A'_p}{2} \cdot A_m \text{ près.}$$



Modulation d'Amplitude (AM)

Représentation temporelle du signal modulé : $s(t)$

Ex. : $s_m(t)$ sinusoïdal : $s_m(t) = A_m \cdot \cos(2\pi f_m t)$

$$s(t) = [A_p + s_m(t)] \cdot \cos(2\pi f_p t) = [A_p + A_m \cdot \cos(2\pi f_m t)] \cdot \cos(2\pi f_p t)$$

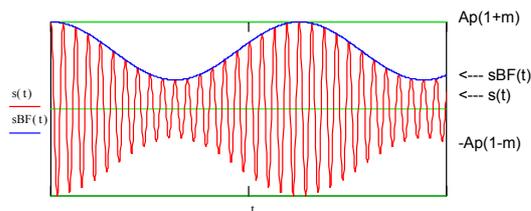
En posant : $m = \frac{A_m}{A_p}$: indice, ou taux de modulation (en %) $m = k A_m$ avec : $k = \frac{1}{A_p}$

on a : $s(t) = A_p [1 + m \cdot \cos(2\pi f_m t)] \cdot \cos(2\pi f_p t)$

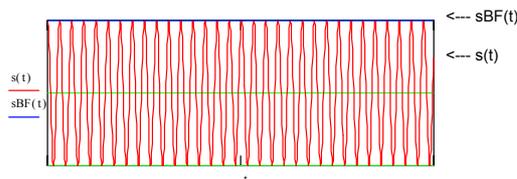
$$s_p(t) = s_p(A_p, \phi, t) = A_p \cos(2\pi f_p t + \phi) \rightarrow s(t) = s_p(A(t), 0, t) \text{ avec } A(t) = A_p + s_m(t)$$

• Réglage du taux de modulation m :

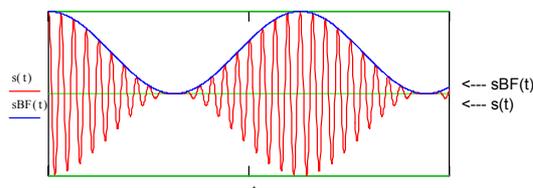
$0 < m < 1$: L'enveloppe > 0 du signal $s(t)$ est le signal à transmettre $s_m(t)$ (à un facteur près et un offset près) : $A_p(1-m) \leq$ enveloppe positive de $s(t) \leq A_p(1+m)$. La modulation est correctement réalisée.



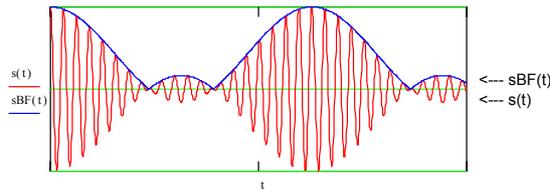
$m = 0$: L'enveloppe > 0 du signal $s(t)$ ne témoigne pas suffisamment du signal à transmettre $s_m(t)$: il y a sous-modulation (l'information utile est indétectable)



$m = 1$: L'enveloppe > 0 du signal $s(t)$ est le signal à transmettre $s_m(t)$ (à un facteur près et un offset près) : on est à la limite de la surmodulation : $0 \leq$ enveloppe positive de $s(t) \leq 2A_p$



$m > 1$: L'enveloppe > 0 du signal $s(t)$ ne témoigne plus fidèlement du signal à transmettre $s_m(t)$ (perte d'informations à la transmission, donc à la réception) : il y a *surmodulation*



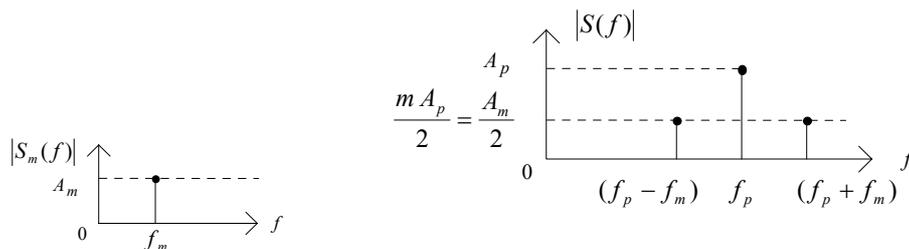
Spectre du signal modulé AM

Ex. : $s_m(t)$ sinusoïdal : $s_m(t) = A_m \cdot \cos(2\pi f_m t)$

$$s(t) = [A_p + s_m(t)] \cdot \cos(2\pi f_p t) = [A_p + A_m \cdot \cos(2\pi f_m t)] \cdot \cos(2\pi f_p t)$$

En linéarisant le produit de cosinus, on a : (avec : $m = \frac{A_m}{A_p}$)

$$s(t) = A_p \cos(2\pi f_p t) + \frac{mA_p}{2} \{ \cos[2\pi(f_p + f_m)t] + \cos[2\pi(f_p - f_m)t] \} \quad \text{d'où les spectres :}$$

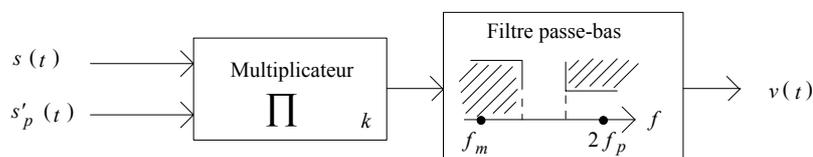


Démodulation du signal AM

Détection synchrone

On utilise un oscillateur local à la réception, oscillant exactement à la fréquence f_p et délivrant le signal :

$$s'_p(t) = A'_p \cdot \cos(2\pi f_p t)$$



En effet : $k s(t) \cdot s'_p(t) = k A_p A'_p [1 + m \cos(2\pi f_m t)] \cos^2(2\pi f_p t)$ d'où :

$$k s(t) \cdot s'_p(t) = \underbrace{\frac{k A_p A'_p}{2} \left\{ 1 + m \cos(2\pi f_m t) \right\}}_{v(t)} + \underbrace{\frac{k A_p A'_p}{2} \left\{ \cos(4\pi f_p t) + \frac{m}{2} \cos[2\pi(2f_p - f_m)t] + \frac{m}{2} \cos[2\pi(2f_p + f_m)t] \right\}}_{\text{éliminé par le filtre passe-bas}}$$

$v(t)$ est bien proportionnel à $s_m(t)$ après élimination de la composante continue.

Boucle à Verrouillage de Phase (Phase Locked Loop : PLL)

Pour éviter le problème de fading (décalage entre les oscillateurs à l'émission et à la réception), on utilise de préférence un asservissement PLL pour reconstituer $s'_p(t)$ à partir de $s(t)$ (pour la détection synchrone) :

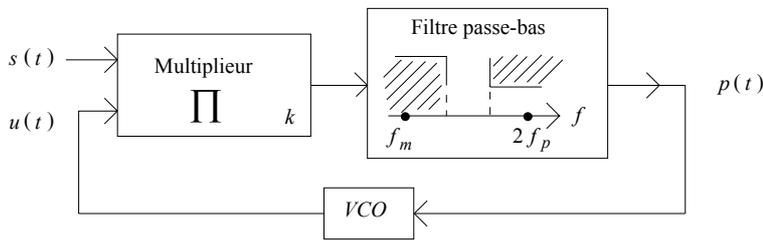
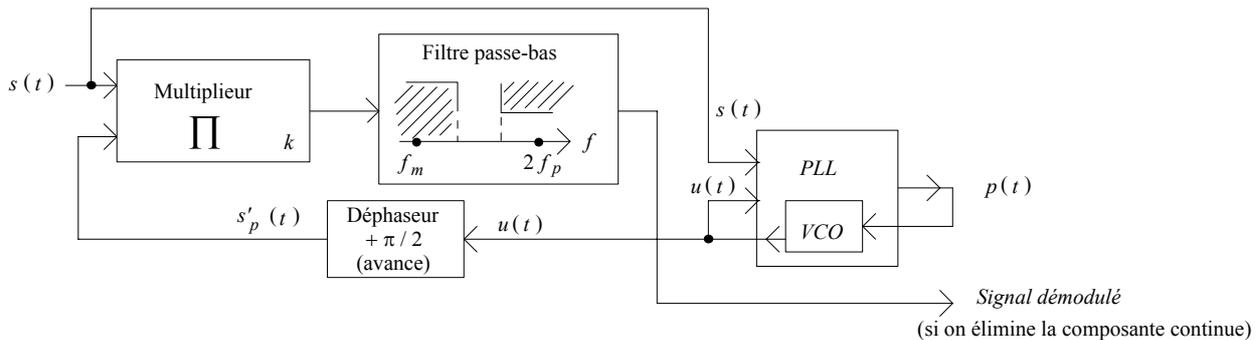


Schéma synoptique d'une PLL

L'oscillateur commandé en tension (VCO) (Voltage Controlled Oscillator) délivre le signal: $u(t) = A'_p \cdot \sin[2\pi f_p t - \varphi(t)]$ avec $\varphi(t)$ dépendant de la tension de commande $p(t)$ par la relation :

$$\frac{d\varphi(t)}{dt} = a p(t) \quad \text{avec : } a : \text{C}^{\text{te}} \quad \text{et} \quad \left| \frac{d\varphi(t)}{dt} \right| \ll 2\pi f_p$$



Détection synchrone à PLL

Modulation Bande Latérale Unique par multiplication (BLU)

On a vu que la multiplication du signal utile $s_m(t) = A_m \cos(2\pi f_m t)$ par la porteuse $s_p = A_p \cos(2\pi f_p t)$ provoque 2 composantes fréquentielles sinusoïdales d'amplitudes $k \frac{A_p A_m}{2}$ et de fréquences $(f_p + f_m)$ et $(f_p - f_m)$.

La modulation BLU ne conserve qu'une seule de ces 2 composantes :

Ex. : la Bande Latérale Supérieure (BLS) → signal émis : $s(t) = \frac{k A_p A_m}{2} \cdot \cos[2\pi(f_p + f_m)t]$

Avantage : La BLU occupe ainsi une bande de fréquences limitée par rapport à la modulation par multiplication (occupation fréquentielle réduite).

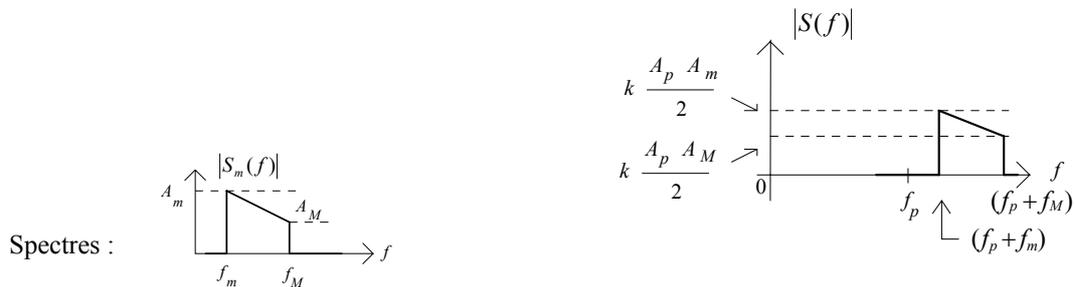
Représentation Temporelle de la BLU

Si $s_m(t)$ est sinusoïdal de fréquence f_m → $s(t)$: sinusoïde d'amplitude C^{te} de fréquence $f_p + f_m$

→ Modulation BLU ≡ Translation du signal dans le domaine fréquentiel (translation du spectre, décalage, transposition de fréquences).

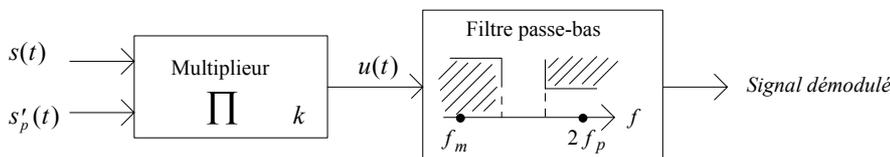
Spectre du signal BLU

Si $s_m(t)$ occupe la bande de fréquences $[f_m, f_M]$ → le signal modulé BLU occupe la bande $[f_p + f_m, f_p + f_M]$:



→ En ne conservant qu'une seule Bande Latérale, on limite la bande de fréquences nécessaire à la transmission.

Démodulation de la BLU



$s'_p(t)$: oscillateur local : $s'_p(t) = A'_p \cdot \cos(2\pi f_p t)$

$s(t) = \text{BLU} = k_0 A_p A_m \cos[2\pi(f_p + f_m)t]$

→ $u(t) = k_0 k A_p A_m A'_p \cos[2\pi(f_p + f_m)t] \cos(2\pi f_p t) = \frac{k_0 k A_p A_m A'_p}{2} \left\{ \underbrace{\cos[2\pi(2f_p + f_m)t]}_{\text{terme à éliminer (filtrage)}} + \cos(2\pi f_m t) \right\}$

Modulation de Fréquence (FM) et Modulation de Phase (PM)

On insère le signal BF dans la fréquence de la porteuse.

Avantage : La FM est moins sensible que la modulation d'amplitude aux bruits parasites car ceux-ci agissent directement sur l'amplitude du signal modulé (bruit additif) donc du signal utile à la réception, et quasiment pas sur sa fréquence.

Modulation de Phase et Modulation de Fréquence (PM et FM)

Le signal modulé (≡ signal émis) peut s'écrire : $s(t) = A(t) \cdot \cos \theta(t)$ avec : $\theta(t)$: phase de $s(t)$.

En modulation AM : $s_m(t)$ intervient dans $A(t)$ Ex. : $A(t) = k A_p A_m \cos(2\pi f_m t) = k A_p s_m(t)$
 $\theta(t) = 2\pi f_p t$

En modulation PM et FM : $s_m(t)$ intervient dans $\theta(t)$: $\theta(t) = 2\pi f_p t + \varphi(t)$ et : $A(t) = A_p$

- en PM: $\varphi(t) = k s_m(t)$ → $s(t) = A_p \cos(2\pi f_p t + k s_m(t))$ $s(t) = s_p(A_p, \varphi(t), t)$ avec $\varphi(t) = k s_m(t)$

- en FM: $\frac{d\varphi(t)}{dt} = k s_m(t)$ → $s(t) = A_p \cos\left(2\pi f_p t + k \int_0^t s_m(\tau) d\tau\right)$ $s(t) = s_p(A_p, \varphi(t), t)$ avec $\varphi(t) = k \int_0^t s_m(\tau) d\tau$

Rappel : $s_p(t) = s_p(A_p, \varphi, t) = A_p \cos(2\pi f_p t + \varphi)$ et $s_m(t) = A_m \cos(2\pi f_m t)$ si le signal modulant est sinusoïdal.

Dans le cas où le signal utile (\equiv signal à transmettre, encore appelé signal modulant) $s_m(t)$ est sinusoïdal (cas intéressant pour une étude élémentaire) : $s_m(t) = A_m \cos(2\pi f_m t)$, on a, pour expression du signal modulé $s(t)$:

- en FM :
$$s(t) = A_p \cos\left[2\pi f_p t + \frac{kA_m}{2\pi f_m} \sin(2\pi f_m t)\right] = A_p \cos\left[2\pi f_p t + m \cdot \sin(2\pi f_m t)\right]$$

avec : $m = \frac{kA_m}{2\pi f_m}$: indice de modulation.

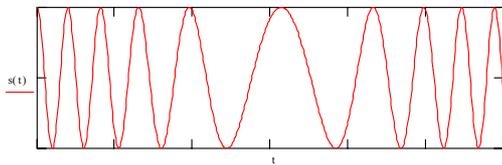
- en PM :
$$s(t) = A_p \cos\left(2\pi f_p t + kA_m \cos(2\pi f_m t)\right) = A_p \cos\left(2\pi f_p t + m \cdot \cos(2\pi f_m t)\right)$$

avec : $m = kA_m$: indice de modulation.

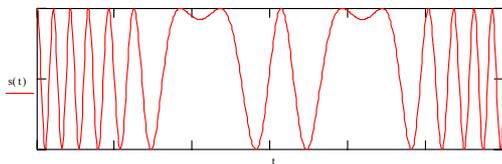
Forme du signal FM

Pour $s_m(t)$ sinusoïdal :

- modulation correctement réalisée : (m faible et $\neq 0$)



- surmodulation (perte d'informations) : (m élevé)



Spectre du signal FM

Cas simple où $m \ll 1$

$$s(t) = A_p \cos\{2\pi f_p t + m \sin(2\pi f_m t)\} = A_p \underbrace{\{\cos(2\pi f_p t) \cdot \cos(m \cdot \sin(2\pi f_m t))\}}_{\neq 0} - \underbrace{\sin(2\pi f_p t) \cdot \sin(m \cdot \sin(2\pi f_m t))}_{\neq 0}$$

$$s(t) \approx A_p \cos(2\pi f_p t) - A_p \sin(2\pi f_p t) \cdot m \cdot \sin(2\pi f_m t)$$

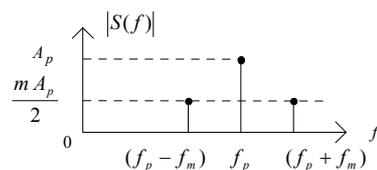
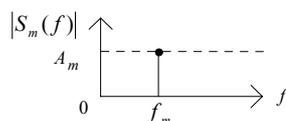
$$\rightarrow s(t) \approx A_p \cos(2\pi f_p t) + \frac{m A_p}{2} \cdot \cos\{2\pi(f_p + f_m)t\} - \frac{m A_p}{2} \cdot \cos\{2\pi(f_p - f_m)t\}$$

↑
porteuse

↑
BLS

↑
BLI

Spectres :



m quelconque

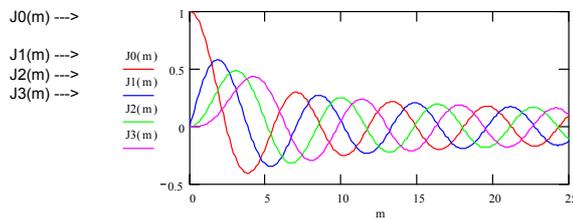
$$s(t) = A_p \cos\{2\pi f_p t + m \sin(2\pi f_m t)\} = A_p \cdot \text{Re}[e^{i\{2\pi f_p t + m \sin(2\pi f_m t)\}}] \quad (\text{formule d'Euler})$$

Décomposition de Bessel : $e^{i m \sin(2\pi f_m t)} = \sum_{n=-\infty}^{\infty} J_n(m) \cdot e^{i n 2\pi f_m t}$ où $J_n(m)$ sont les fonctions de Bessel (réelles)

$$s(t) = A_p \cdot \text{Re}\left[\sum_{n=-\infty}^{\infty} J_n(m) \cdot e^{i 2\pi (f_p + n f_m) t}\right] \rightarrow s(t) = A_p \sum_{n=-\infty}^{\infty} [J_n(m) \cdot \cos(2\pi (f_p + n f_m) t)]$$

Le spectre de $s(t)$ est infini ($n : -\infty \rightarrow \infty$) : spectre doublement infini si $s_m(t)$ n'est pas sinusoïdal.

Fonctions de Bessel :



Démodulation de Fréquence

On transforme la variation de fréquence $\Delta\omega$ en variation d'amplitude $\Delta A \rightarrow$ On peut ensuite effectuer une démodulation d'amplitude (à PLL par exemple).

Modulations numériques (Transmissions Numériques)

Les modulations d'impulsions

En télécommunications, le signal représentant l'information à transmettre subit une transposition de fréquence de façon à occuper une bande de fréquence de largeur relativement faible autour d'une porteuse HF.

Cette transposition est effectuée par modulation d'un des paramètres caractérisant la porteuse : amplitude, fréquence ou phase.

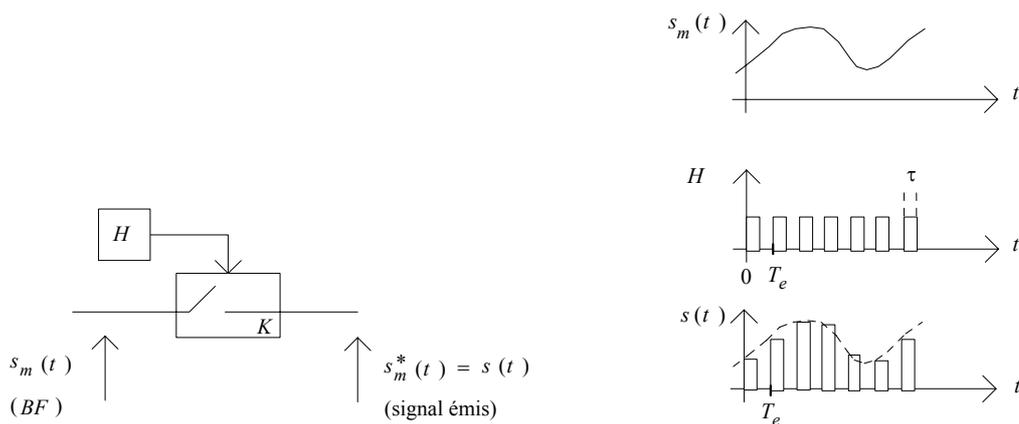
Si la porteuse est constituée d'impulsions, la modulation présente l'avantage de pouvoir transporter plusieurs signaux à la fois : c'est le multiplexage temporel (multiplexage \equiv sélection, commutation de 1 signal parmi N).

Les impulsions de la porteuse ont pour fréquence F_e : fréquence des échantillons du signal numérique. A la réception, il suffit de se caler sur la fréquence d'échantillonnage F_e pour démoduler.

La modulation d'impulsions en amplitude

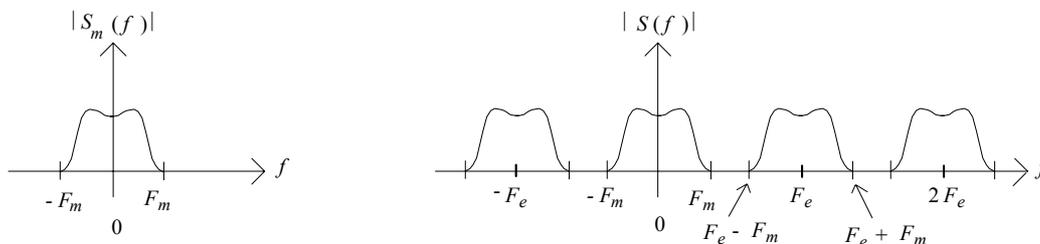
On utilise un échantillonneur rythmé par une horloge H de période T_e (période d'échantillonnage $T_e = \frac{1}{F_e}$) et

fournissant des impulsions de largeur τ . Un échantillonneur \equiv interrupteur, commutateur analogique K commandé par l'horloge H



Spectre de $s(t)$:

On montre à l'aide de la Transformée de Fourier que l'échantillonnage temporel a pour effet de périodiser le spectre de $s_m(t)$: (à la période (fréquentielle) F_e) :



Traitement du Signal 5. Transmission (Codage. Egalisation. Extraction de signal immergé dans du bruit. Filtrage adaptatif)

La porteuse (≡ l'horloge) doit respecter la condition de Shannon : (théorème de Shannon) :

$$F_e \geq 2 F_m \quad \text{où } F_m \text{ est la fréquence maximale contenue dans le spectre de } s_m(t),$$

sans quoi il y a recouvrement des lobes du spectre de $s(t)$ et impossibilité, à la réception, de reconstituer $s_m(t)$ à partir des échantillons $s_m^*(t)$.

Ex. : Signal de musique : $20 \text{ Hz} < f < 20 \text{ kHz} \rightarrow F_m = 20 \text{ kHz} \rightarrow F_e \geq 40 \text{ kHz}$

Signal de parole : $300 \text{ Hz} < f < 4 \text{ kHz} \rightarrow F_m = 4 \text{ kHz} \rightarrow F_e \geq 8 \text{ kHz}$.

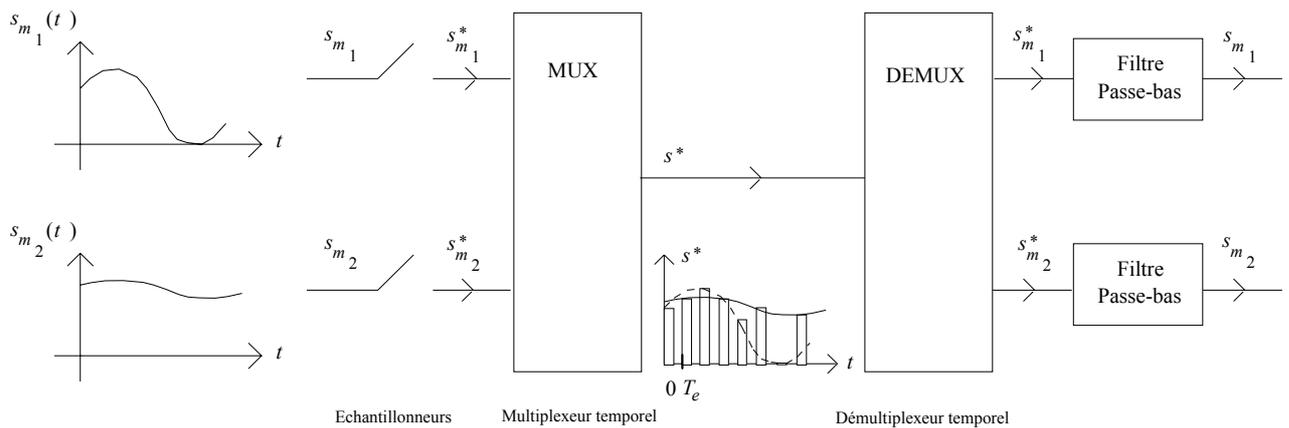
Démodulation

La démodulation se fait simplement par filtrage passe-bas de $s(t)$ avec F_m pour fréquence de coupure.

Multiplexage

On peut, sur un même canal de transmission (≡ une même porteuse (équivalent d'une ligne de transmission pour les transmissions filaires)) transporter plusieurs messages en même temps par multiplexage temporel (≡ sélection d'un signal parmi N).

A la réception, il faut séparer les messages par démultiplexage (≡ récupération des N signaux à partir d'1 signal).

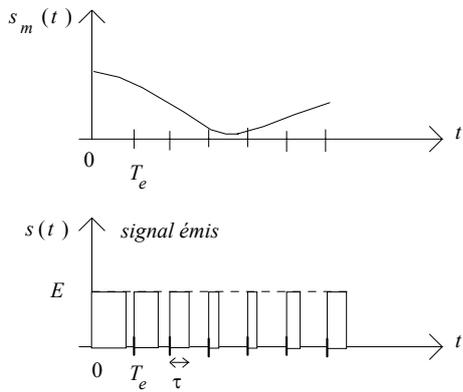


Ce système de multiplexage est utilisé en téléphonie : la fréquence supérieure de la voix est de l'ordre de $F_m = 4 \text{ kHz}$.

L'échantillonnage se fait à $F_e = 8 \text{ kHz}$. On arrive à transporter plusieurs dizaines de voies téléphoniques sur la même ligne de transmission.

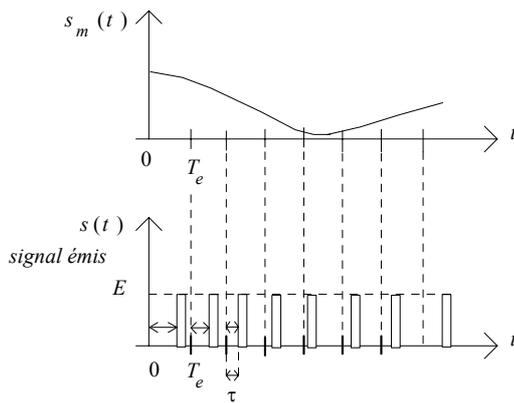
La modulation d'impulsions en durée (PWM : Power Width Modulation)

Les impulsions ont toutes la même amplitude, mais c'est leur durée τ qui est fonction du signal $s_m(t)$ à transporter.



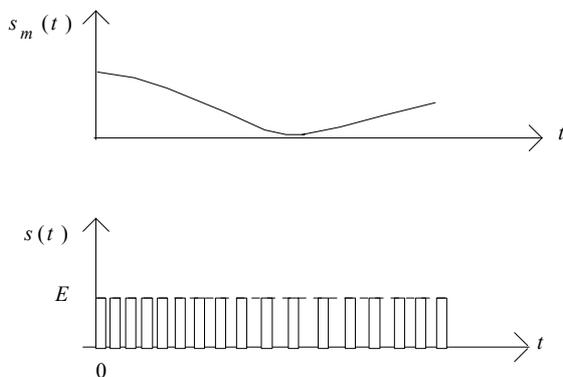
La modulation d'impulsions en position

Les impulsions ont toutes la même amplitude et la même durée, mais leurs fronts montants n'apparaissent tous les T_e secondes, comme dans les cas précédents. Ils sont retardés d'un temps τ qui est fonction du signal $s_m(t)$ à transmettre :



La modulation d'impulsions en densité (moins utilisée car F_e est variable)

Les impulsions ont toutes la même amplitude et la même durée, mais leur densité (\equiv fréquence) est fonction du signal $s_m(t)$ à transmettre :



La modulation d'impulsions codées (MIC)

1 impulsion est codée par 1 mot binaire

Les échantillons successifs du signal à émettre ne sont pas utilisés directement mais sont quantifiés et codés numériquement en un mot binaire à l'aide d'un convertisseur analogique-numérique (CAN).

La quantification va arrondir l'amplitude de l'échantillon (cas de la modulation en amplitude) ou sa durée (cas de la modulation en durée) etc ..., à une valeur discrète parmi $N = 2^m$ valeurs possibles

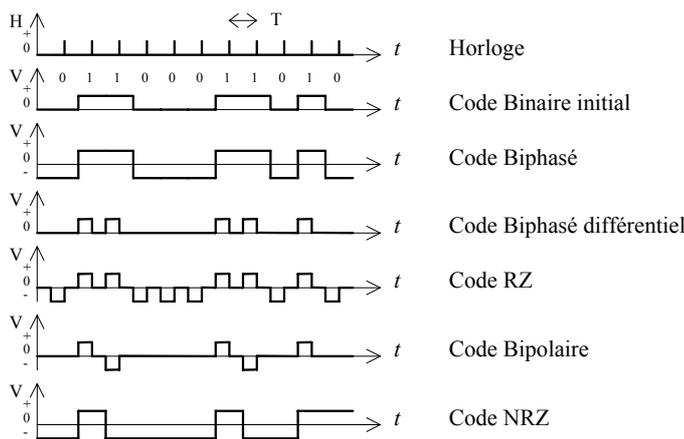
(m : nombre de bits de quantification, N : dynamique de quantification, de codage).

(L'échantillonnage discrétise le temps à la période d'échantillonnage T_e , alors que la quantification discrétise la valeur, l'amplitude de l'échantillon).

Le but du codage peut être de limiter l'occupation spectrale du signal à émettre, ou l'immunisation au bruit, ou encore le compactage, la compression des données ...

Les principaux codages utilisés sont les suivants :

- code binaire biphasé
- code différentiel biphasé
- code RZ (Retour à Zéro)
- code NRZ (Non Retour à Zéro)
- code à modulation de délai (DM ou Miller)
- codes FM
- codes multiniveaux



Le signal numérisé (≡ échantillonné, quantifié et codé) est parfois codé directement suivant le code binaire pur (≡ code binaire initial) ou mieux, suivant le code binaire réfléchi (code de Gray) par exemple, pour autoriser la détection et la correction des erreurs de transmission.

Le code biphasé présente, quant à lui, l'avantage de limiter le spectre de fréquences occupé par le signal à émettre par rapport au code binaire pur.

La Modulation Δ :

Plutôt que de transmettre le mot binaire de chaque échantillon, on ne transmet que la différence d'amplitude entre 2 échantillons consécutifs et pouvant souvent être codée sur un seul bit si la valeur des échantillons ne varie pas trop brutalement.

Avantage : Compression (≡ réduction) de la quantité d'informations à transmettre.

Transmission du signal binaire (≡ numérique)

Transmission en Bande de Base

Les impulsions de courant matérialisant les 0 et les 1 sont appliquées directement pour l'émission, on dit dans ce cas que la transmission se fait en bande de base.

Compte tenu du théorème de Shannon, un signal dont la bande de fréquence utile (≡ occupée) est B nécessite un canal de largeur $2B$.

Transmission par modulation

Le plus souvent, la transmission en bande de base n'est pas possible du fait de la faible Bande Passante du canal de transmission, et il faut adapter le mode de représentation de l'information au canal disponible : c'est la modulation.

- *Modulation ASK (Amplitude Shift Keying) : Modulation d'amplitude*

Un oscillateur de fréquence élevée est modulé en tout ou rien par le signal binaire à transmettre. Le spectre du signal utile est transposé autour de la fréquence porteuse.

- *Modulation FSK (Frequency Shift Keying) : Saut de fréquence*

Aux 2 valeurs 0 et 1 des bits successifs correspondent 2 fréquences différentes de l'oscillateur pilote (≡ porteuse).

- *Modulation PSK (Phase Shift Keying) : Saut de phase*

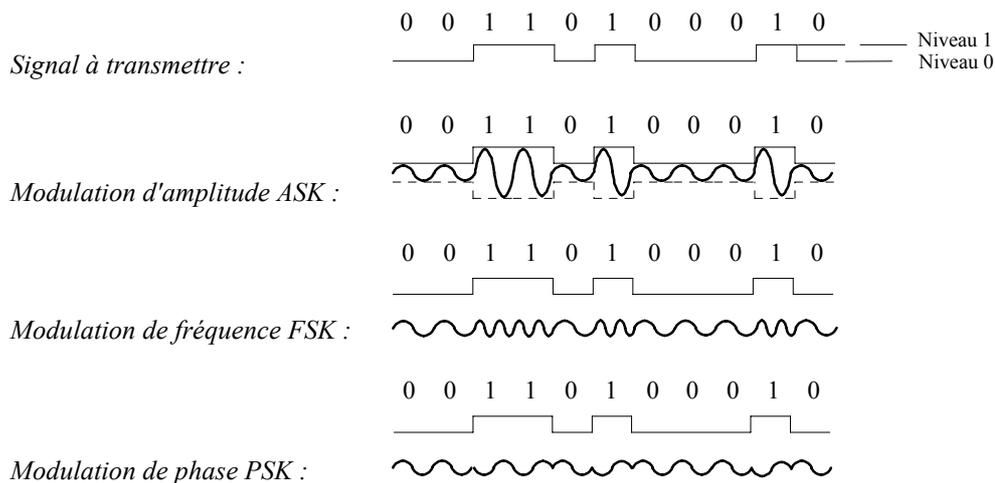
Le signal à transmettre est constitué d'impulsions polaires qui commandent une variation de $\pm \pi / 2$ de la phase de l'oscillateur pilote. Ce procédé nécessite de connaître la phase de référence → transmission nécessairement synchrone.

- *Modulation DPSK (Differential Phase Shift Keying) : Saut de phase différentiel*

Pour éviter d'avoir à transmettre une phase de référence, l'information est transmise sous forme de variations de phase.

Transmission d'un signal numérique :

(porteuse sinusoïdale : )



Codes détecteurs et correcteurs d'erreurs

Le bruit dans les systèmes de transmission binaire peut modifier un bit du mot. Plusieurs codes de détection et de correction existent :

- *Utilisation d'un bit supplémentaire de parité*

(parité d'un mot binaire \equiv parité du nombre de bits à 1 contenus dans le mot)

A chaque mot du message à transmettre, est ajouté un bit choisi de telle sorte que le mot résultant ait une parité paire (ce bit ajouté est appelé bit de parité) :

Code détecteur d'erreurs par test de parité à la réception.

Inconvénient : pas de correction possible car on ne peut pas localiser le bit erroné. De plus, un nombre pair d'erreurs simultanées sur un mot passent inaperçues.

- *Codes à parités entrelacées*

Un tableau 2D (\equiv à 2 Dimensions) est construit avec les mots (tableau de taille $N_x * N_y$). La parité est contrôlée horizontalement et verticalement : code détecteur plus puissant que le précédent car il y a localisation du bit d'erreur et donc correction possible. Comme le précédent, seul un nombre impair d'erreurs simultanées peut être détecté.

- *Codes de Hamming*

Codes détecteurs et correcteurs. Ils ajoutent plusieurs bits à des emplacements bien déterminés dans le mot (chacun de ces bits contrôle un groupe de bits du mot avec recouvrements des groupes) pour détecter la position des erreurs et ainsi les corriger plus efficacement

- *Codes cycliques*

Pour une transmission sûre à 100 % le Codage à Redondance Cyclique (CRC) peut être utilisé, pour lequel les données sont codées sous forme polynomiale.

Ce contrôle consiste à comparer les restes des divisions d'un polynôme par un polynôme fixe pour la donnée initiale et pour la donnée après transmission (les données étant représentées sous forme de polynômes dont la longueur conditionne le choix des polynômes diviseurs).

Plusieurs codes à redondance cyclique sont définis suivant la longueur des mots à transmettre.

- *Codes continus (correction d'erreurs en paquets)*

La plupart des codes détectent des erreurs réparties individuellement. Malheureusement, les erreurs sont parfois (même si c'est rare) groupées en paquets. (*ex.*: poussière sur un Compact Disc \rightarrow plusieurs dizaine de bits disparaissent d'un coup).

Pour combattre ce phénomène, une solution simple mais efficace consiste à effectuer un entrelacement (\equiv écriture des mots dans un ordre différent de celui de la lecture).

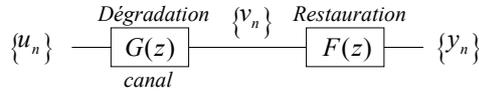
Exemple : entrée horizontale des mots dans une matrice et lecture verticale de ces mots.

Code détecteur et correcteur d'erreur.

Filtrage optimal au sens des moindres carrés

Le filtre égaliseur (*equalizer*) et l'extracteur de signal à partir d'une mesure bruitée sont 2 problèmes différents menant à la même solution formelle. Le premier est posé de manière déterministe, le second de manière stochastique. Les deux se basent sur un critère quadratique (minimisation de l'énergie de l'erreur).

II. EGALISATION



où $G(z)F(z)$ est la FT du canal de transmission dégradé-restauré.

Etant donnée la dégradation $G(z)$, il s'agit de déterminer le filtre de restauration $F(z)$ tel que : $\frac{Y(z)}{U(z)} = D(z)$,

soit : $F(z)G(z) = D(z)$. Le terme « égaliseur » provient du cas particulier $D(z) = 1$. On a alors : $F(z)G(z) = 1$ (La dégradation est connue par modélisation ou identification). (Le rôle de l'égaliseur $F(z)$ est de **compenser** la dégradation).

La solution immédiate : $F(z) = \frac{D(z)}{G(z)}$ est généralement non réalisable.

En effet, si $G(z)$ s'annule pour certaines fréquences, $F(z)$ est infini. Ou bien si $G(z)$ comporte des zéros « instables », $F(z)$ est alors instable. Ou encore, $F(z)$ doit être passe-haut si comme c'est généralement le cas en physique, $G(z)$ est passe-bas, et un filtre physique passe-haut ne peut être passe-haut pour des fréquences infiniment élevées.

Enfin, si $G(z)$ provoque un retard (cas des filtres causaux), $F(z)$ est non causal car il doit provoquer une avance.

On préfère ainsi se limiter à la classe des filtres $F(z)$ du type MA (\equiv RIF : Réponse Impulsionnelle Finie) et minimiser le critère *quadratique* (\equiv *moindres carrés*) pour *signaux déterministes* :

$$V(\{f_n\}) = \sum_{n=-\infty}^{\infty} (d_n - \hat{d}_n)^2$$

où : $\{d_n\}$, $\{f_n\}$ et $\{g_n\}$ sont les RIs de $D(z)$, $F(z)$ et $G(z)$ et $\{\hat{d}_n\}$ la RI du canal restauré.

Comme $\hat{d}_n = g_n * f_n$ (du fait de la relation d'égalisation : $F(z)G(z) = D(z)$), on a :

$$V(\{f_n\}) = \sum_{n=-\infty}^{\infty} (d_n - g_n * f_n)^2 \text{ avec } f_n = 0 \text{ pour } n < 0 \text{ et } n > N \text{ (filtre RIF causal de support de longueur } N + 1)$$

$$\rightarrow V(\{f_n\}) = \sum_{n=-\infty}^{\infty} \left(d_n - \sum_{k=0}^N f_k g_{n-k} \right)^2 = \sum_{n=-\infty}^{\infty} d_n^2 - 2 \sum_{k=0}^N f_k \sum_{n=-\infty}^{\infty} d_n g_{n-k} + \sum_{i=0}^N \sum_{j=0}^N f_i f_j \left[\sum_{n=-\infty}^{\infty} g_{n-i} g_{n-j} \right]$$

$$\text{(on rappelle que : } \left(\sum_{i=0}^N a_i \right)^2 = \sum_{i=0}^N a_i^2 + 2 \sum_{i=0}^{N-1} a_i \sum_{j=i+1}^N a_j = \sum_{i=0}^N a_i^2 + 2 \sum_{i=0}^{N-1} \sum_{j=i+1}^N a_i a_j)$$

$$\text{or : } \sum_{n=-\infty}^{\infty} d_n g_{n-k} = \varphi_{dg_{-k}} \text{ et en posant } n - j = \lambda : \sum_{n=-\infty}^{\infty} g_{n-i} g_{n-j} = \sum_{\lambda=-\infty}^{\infty} g_{\lambda+j-i} g_{\lambda} = \varphi_{gg_{j-i}}$$

φ_{gg} : autocorrélation de g_n φ_{dg} : intercorrélation entre d_n et g_n

$$\text{donc : } V(\{f_n\}) = \sum_{n=-\infty}^{\infty} d_n^2 - 2 \sum_{k=0}^N f_k \varphi_{dg_{-k}} + \sum_{i=0}^N \sum_{j=0}^N f_i f_j \varphi_{gg_{j-i}}$$

$$\text{Posons : } \sum_{n=-\infty}^{\infty} d_n^2 = \alpha^2 \quad \underline{\mathbf{f}} = \begin{bmatrix} f_0 \\ f_1 \\ \dots \\ f_N \end{bmatrix} \quad \underline{\varphi}_{dg} = \begin{bmatrix} \varphi_{dg_0} \\ \varphi_{dg_{-1}} \\ \dots \\ \varphi_{dg_{-N}} \end{bmatrix} \quad \varphi_{gg_{i,j}} = \varphi_{gg_{j-i}}$$

φ_{gg} : matrice d'autocorrélation (Toeplitz)

Alors $V(\underline{\mathbf{f}}) = \alpha^2 - 2 \underline{\mathbf{f}}^T \underline{\varphi}_{dg} + \underline{\mathbf{f}}^T \varphi_{gg} \underline{\mathbf{f}}$ dont le minimum est donné par l'annulation du gradient $\underline{\mathbf{V}}_{\underline{\mathbf{f}}} = -2 \underline{\varphi}_{dg} + 2 \varphi_{gg} \underline{\mathbf{f}} = \underline{\mathbf{0}}$

Donc :

$$\phi_{gs} \mathbf{f} = \phi_{-dg}$$

Egalisation

Dans le cas où $D(z) = 1$: $\{d_n\} = \delta_n$ et $\varphi_{dg_{-k}} = \sum_{n=-\infty}^{\infty} d_n g_{n-k} = g_{-k}$ on a, en posant $\underline{\gamma} = \begin{bmatrix} g_0 \\ g_{-1} \\ \dots \\ g_{-N} \end{bmatrix}$:

(on retrouve une forme proche du système d'équations de **Yule-Walker**. Ce sont les équations de **Wiener-Hopf**)

$$\phi_{gg} \underline{\mathbf{f}} = \underline{\gamma}$$

dont la solution peut être obtenue par inversion matricielle :

$$\underline{\mathbf{f}} = \phi_{gg}^{-1} \underline{\gamma}$$

De plus, dans le cas causal (g_n causale), l'égalisation conduit au système (Yule-Walker) :

$$\phi_{gg} \underline{\mathbf{f}} = \underline{\gamma} \quad \text{avec} \quad \underline{\gamma} = \begin{bmatrix} g_0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Conclusion

Le principe de l'égalisation traitée ici fait partie de l'égalisation supervisée : on la connaissance de la dégradation, directement comme ici, par identification par exemple, ou par apprentissage (réseaux de neurones artificiels (apprentissage numérique), chaînes de Markov (apprentissage numérique), algorithmes génétiques, apprentissage symbolique).

Lorsque la dégradation n'est pas connue a priori, il ne s'agit plus réellement de restauration de signal et on parle alors d'égalisation non supervisée encore appelée égalisation aveugle.

III. EXTRACTION DE SIGNAL IMMERGE DANS DU BRUIT

Le problème de l'extraction de signal immergé dans du bruit est l'analogie en stochastique du problème précédent déterministe de l'égalisation.

Ce type de **débruitage** est basé sur un critère quadratique (minimisation de l'énergie de l'erreur) dans un contexte stochastique.

$$\{u_n\} = \{s_n\} + \{b_n\} \longrightarrow \boxed{F(z)} \longrightarrow \{\hat{s}_n\}$$

A partir de la mesure $u_n = s_n + b_n$ (signal + bruit), il s'agit de déterminer la FT $F(z)$ (ou la séquence RI

$\{f_n\} = TZ^{-1}[F(z)]$) satisfaisant au critère *quadratique* pour *signaux aléatoires* (on passe en *Espérance*) :

$$\boxed{V(\{f_n\}) = E[S_n - \hat{S}_n]^2} \text{ minimum} \quad \text{avec : } \{S_n\} \text{ VA dont une réalisation est } \{s_n\}$$

Comme : $\hat{S}_n = f_n * U_n$ pour qu'il y ait extraction du bruit. On a donc : $V(\{f_n\}) = E[S_n - f_n * U_n]^2$
 ($\{U_n\}$ VA dont une réalisation est $\{u_n\}$)

Le filtre $\{f_n\}$ étant pris comme un filtre RIF (\equiv MA) de support de longueur $N + 1$, on a : $V(\{f_n\}) = E\left[S_n - \sum_{k=0}^N f_k * U_{n-k}\right]^2$

En posant :

$$\underline{\mathbf{f}} = \begin{bmatrix} f_0 \\ f_1 \\ \dots \\ f_N \end{bmatrix} \quad \text{et} \quad \underline{\mathbf{U}} = \begin{bmatrix} U_n \\ U_{n-1} \\ \dots \\ U_{n-N} \end{bmatrix} \quad \text{il vient :} \quad V(\{f_n\}) = V(\underline{\mathbf{f}}) = E[S_n - \underline{\mathbf{f}}^T \underline{\mathbf{U}}]^2$$

Principe d'orthogonalité : l'erreur est orthogonale à l'observation :

$$\text{Le gradient } (\equiv \text{dérivée}) \text{ de } V \text{ donne le minimum de } V(\underline{\mathbf{f}}) \text{ pour :} \quad \underline{\mathbf{V}}_{\underline{\mathbf{f}}} = 2E[(S_n - \underline{\mathbf{f}}^T \underline{\mathbf{U}})\underline{\mathbf{U}}] = \underline{\mathbf{0}}$$

Cette relation exprime le fait que, à l'instant n , l'erreur d'estimation $(S_n - \underline{\mathbf{f}}^T \underline{\mathbf{U}})$ est orthogonale à toutes les données présentes et passées ($\underline{\mathbf{U}}$) (le produit scalaire (\equiv matriciel) est nul). C'est le principe d'**orthogonalité**.

$$\text{On a :} \quad \underline{\mathbf{V}}_{\underline{\mathbf{f}}} = 2E[(S_n - \underline{\mathbf{f}}^T \underline{\mathbf{U}})\underline{\mathbf{U}}] = 2E[\underline{\mathbf{U}}(S_n - \underline{\mathbf{U}}^T \underline{\mathbf{f}})] = 2\{E[S_n \underline{\mathbf{U}}] - E[\underline{\mathbf{U}} \underline{\mathbf{U}}^T] \underline{\mathbf{f}}\} = \underline{\mathbf{0}}$$

Comme le terme général de la matrice $\underline{\mathbf{U}} \underline{\mathbf{U}}^T$ est le terme d'autocorrélation $\varphi_{uu_{i-j}}$, $E[\underline{\mathbf{U}} \underline{\mathbf{U}}^T]$ est la matrice d'autocorrélation φ_{uu} , notée \mathbf{R} : $\mathbf{R} = E[\underline{\mathbf{U}} \underline{\mathbf{U}}^T]$: matrice de Toeplitz.

De même, si on note $\varphi_{su} = E[S_n \underline{\mathbf{U}}]$ le vecteur d'intercorrélacion, on retrouve une forme proche du système d'équations de **Yule-Walker** (sous forme matricielle).

$$\text{Le gradient de } V \text{ s'écrit donc :} \quad \underline{\mathbf{V}}_{\underline{\mathbf{f}}} = 2\{\varphi_{su} - \mathbf{R} \underline{\mathbf{f}}\} = \underline{\mathbf{0}}$$

Ce sont les équations de **Wiener-Hopf** :

$$\boxed{\mathbf{R} \underline{\mathbf{f}} = \varphi_{su}}$$

dont la solution peut être obtenue par inversion matricielle :

$$\boxed{\underline{\mathbf{f}} = \mathbf{R}^{-1} \varphi_{su}}$$

L'intérêt de cette approche est de ne pas nécessiter la connaissance du bruit mais seulement celle du signal à extraire (c'est un minimum !) *Pour filtrer un signal bruité, il faut la connaissance : soit du bruit, soit du signal non bruité !*

Dans un contexte d'*autoadaptation* (*filtrage adaptatif*), la contrainte *Temps Réel* (*calcul en ligne*) impose d'éviter l'inversion matricielle en approchant \mathbf{R} et φ_{su} en utilisant les données passées (cf. filtrage adaptatif (Widrow)).

IV. FILTRAGE ADAPTATIF

Filtre adaptatif de Widrow *version récurrente (temps réel) du filtre extracteur de signal immergé dans du bruit vu avant*

Le filtre adaptatif de **Widrow** représente l'exemple le plus connu de filtre adaptatif : il s'agit d'un filtre MA (\equiv RIF) optimal au sens des moindres carrés. L'ajustement des *paramètres* du filtre (\equiv les coefficients a_i , \equiv les échantillons f_i de RI) se fait en ligne, c'est-à-dire que les paramètres du filtre varient au cours du temps pour s'adapter à la *contrainte (filtrage optimal* : pour le filtre de Widrow, la contrainte est quadratique (moindres carrés)) spécifiée.

Dans l'approche de Widrow, le minimum du critère de performance $V(\underline{\mathbf{f}})$ [$V(\underline{\mathbf{f}}) = \alpha^2 - 2\underline{\mathbf{f}}^T \underline{\varphi}_{dg} + \underline{\mathbf{f}}^T \underline{\phi}_{gg} \underline{\mathbf{f}}$ (pour le problème de l'égalisation) ou $V(\underline{\mathbf{f}}) = E[S_n - \underline{\mathbf{f}}^T \underline{\mathbf{U}}]^2$ (pour le problème de l'extraction de signal immergé dans du bruit)] est obtenu de manière itérative par une méthode de gradient.

Le vecteur gradient est perpendiculaire aux hypersurfaces $V(\underline{\mathbf{f}}) = C^{te}$: en effet, si l'on suppose que $\underline{\mathbf{f}}$ est fonction du temps (filtrage adaptatif), on a, sur une isosurface, $dV = 0 = \underline{\mathbf{V}}_{\underline{\mathbf{f}}}^T \cdot \dot{\underline{\mathbf{f}}} \cdot dt$, donc $V(\underline{\mathbf{f}})$ est perpendiculaire à $\dot{\underline{\mathbf{f}}}$, qui est le vecteur tangent à l'isosurface.

Dans ces conditions, si on itère sur $\underline{\mathbf{f}}$ proportionnellement au gradient de $V(\underline{\mathbf{f}})$ (le gradient de $V(\underline{\mathbf{f}})$ est $\underline{\mathbf{V}}_{\underline{\mathbf{f}}} = 2\{\underline{\varphi}_{su} - \mathbf{R}\underline{\mathbf{f}}\}$) on ira vers le minimum de $V(\underline{\mathbf{f}})$ selon « la plus grande pente ».

Si k est l'indice d'itération, on choisit : (il ne représente absolument pas l'élevation à une puissance)

$$\underline{\mathbf{f}}^{k+1} = \underline{\mathbf{f}}^k - \mu (\mathbf{R}\underline{\mathbf{f}}^k - \underline{\varphi}_{su}) \quad \text{où } \mu \text{ est un paramètre fixant la pas d'itération.}$$

Comme toujours avec la méthode du gradient, si μ est faible, la convergence est lente et si μ est grand, il ya risque de divergence. Pour être plus précis, sur la valeur maximale de μ on peut réécrire la relation précédente :

$$\underline{\mathbf{f}}^{k+1} = [\mathbf{I} - \mu \mathbf{R}] \underline{\mathbf{f}}^k + \mu \underline{\varphi}_{su}$$

C'est une équation vectorielle du 1er ordre, dont la solution converge vers $\mathbf{R}^{-1} \underline{\varphi}_{su}$ si toutes les valeurs propres de $[\mathbf{I} - \mu \mathbf{R}]$ sont de module inférieur à 1, c'est-à-dire si :

$$|1 - \mu \lambda_i| < 1 \quad \text{où : } \lambda_i \text{ sont les valeurs propres de } \underline{\phi}_{uu} = \mathbf{R}$$

d'où une contrainte pour le choix de μ :

$$0 < \mu < \frac{2}{\lambda_{\max}} \quad \text{où : } \lambda_{\max} \text{ est la plus grande valeur propre } \lambda_i \text{ de } \mathbf{R}$$

Il reste à affiner la relation itérative : $\underline{\mathbf{f}}^{k+1} = \underline{\mathbf{f}}^k - \mu (\mathbf{R}\underline{\mathbf{f}}^k - \underline{\varphi}_{su})$ en évitant le calcul, la connaissance de \mathbf{R} et $\underline{\varphi}_{su}$.

Pour cela, au lieu du critère de performance $V(\underline{\mathbf{f}})$ [$V(\underline{\mathbf{f}}) = V(\{f_n\}) = \sum_{n=-\infty}^{\infty} (d_n - \hat{d}_n)^2 = \alpha^2 - 2\underline{\mathbf{f}}^T \underline{\varphi}_{dg} + \underline{\mathbf{f}}^T \underline{\phi}_{gg} \underline{\mathbf{f}}$ (pour le problème de l'égalisation) ou $V(\underline{\mathbf{f}}) = V(\{f_n\}) = E[S_n - \hat{S}_n]^2 = E[S_n - \underline{\mathbf{f}}^T \underline{\mathbf{U}}]^2$ (pour le problème de l'extraction de signal immergé dans du bruit)], on utilise un autre critère de performance, constitué de l'erreur quadratique instantanée :

$$V'(\underline{\mathbf{f}}) = (s_n - \underline{\mathbf{f}}^T \underline{\mathbf{U}})^2 = e_n^2 \quad \text{dont le gradient est : } \underline{\mathbf{V}}'_{\underline{\mathbf{f}}} = -(s_n - \underline{\mathbf{f}}^T \underline{\mathbf{U}}) \underline{\mathbf{U}} = -e_n \underline{\mathbf{U}}$$

L'algorithme d'autoadaptation peut donc s'écrire : (il a prouvé expérimentalement son efficacité)

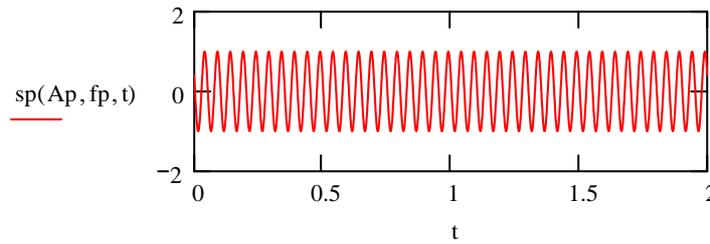
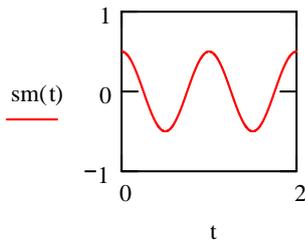
$$\underline{\mathbf{f}}^{k+1} = \underline{\mathbf{f}}^k - \mu e_n \underline{\mathbf{U}} \quad \text{soit, pour la } i \text{-ième composante de } \underline{\mathbf{f}} : \underline{f}_i^{k+1} = \underline{f}_i^k - \mu e_n u_{n-i} \quad \text{avec } 0 \leq i \leq N.$$

TP 5. Transmission

I. CODAGE - DECODAGE

Signal à transmettre (Signal modulant) : $A_m := 0.5$ $f_m := 1$ $sm1(t) := A_m \cdot \cos(2 \cdot \pi \cdot f_m \cdot t)$ $sm2(t) := \begin{cases} A_m & \text{if } 1 < t \leq 1. \\ 0 & \text{otherwise} \end{cases}$
 $t := 0, 10^{-3} \dots 2$ $sm(t) := sm1(t)$

Porteuse : $A_p := 1$ $f_p := 20$ $sp(A_p, \phi, t) := A_p \cdot \cos(2 \cdot \pi \cdot f_p \cdot t + \phi)$

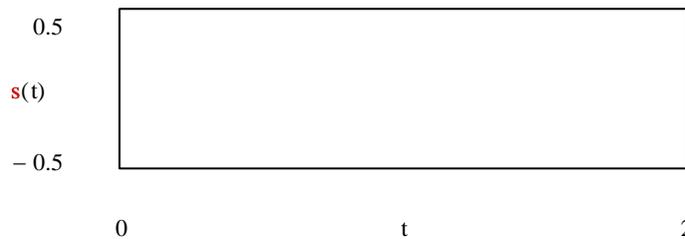


Modulation par multiplication

$k := 1$

$s(t) :=$ ■

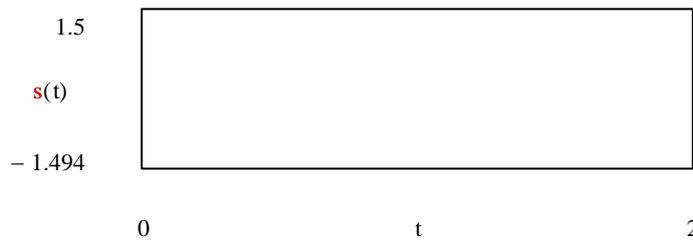
Signal modulé



Modulation d'amplitude (AM)

$A(t) :=$ ■

$s(t) := sp(A(t), 0, t)$ $m := \frac{A_m}{A_p}$ $m = 0.5$

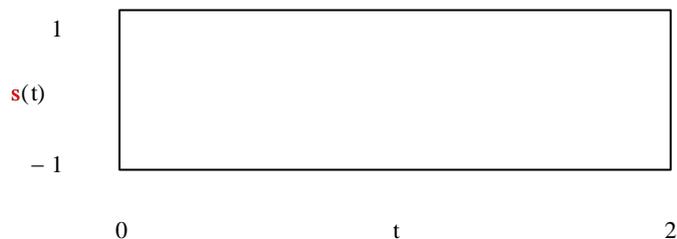


Modulation de phase (PM)

$k := 20$

$\phi(t) :=$ ■

$s(t) := sp(A_p, \phi(t), t)$ $m := k \cdot A_m$ $m = 10$



Modulation de fréquence (FM)

$k := 150$ $\phi(t) := \blacksquare$

$s(t) := \text{sp}(A_p, \phi(t), t)$ $m := \frac{k \cdot A_m}{2 \cdot \pi \cdot f_m}$

$m = 11.937$



II. EGALISATION (1)

$N := 50$ $n := 0..N$

Signal d'entrée : $u_n := \sin\left(\frac{n}{2}\right)$

RI de la dégradation : $g_n := (-1)^n$

Signal dégradé : $v := \blacksquare$

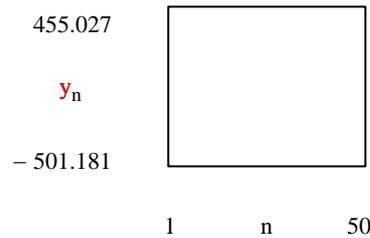
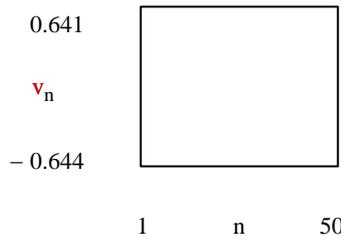
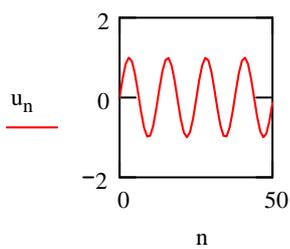
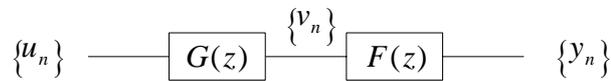
RI de la restauration : $\phi := \blacksquare$

$\gamma_n := 0$

$\gamma_0 := g_0$

$f := \blacksquare$

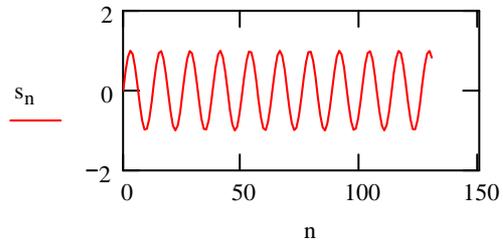
Signal restauré : $y := \blacksquare$



III. EXTRACTION DE SIGNAL IMMERGE DANS DU BRUIT

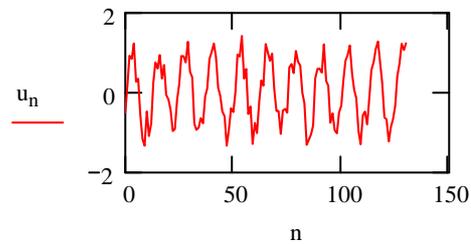
$$N := 130 \quad n := 0..N \quad \{u_n\} = \{s_n\} + \{b_n\} \longrightarrow \boxed{F(z)} \longrightarrow \{y_n\} = \{\hat{s}_n\}$$

Signal d'entrée non bruité : $s_n := \sin\left(\frac{n}{2}\right)$



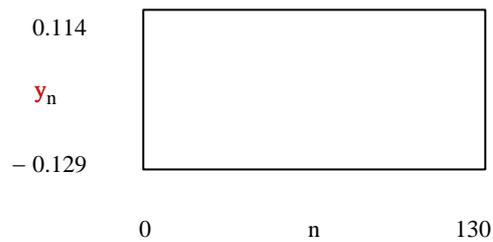
Bruit : $b := \text{whiten}(N + 1)$

Signal bruité : $u := s + b$ *u reverse* : $ub_n := u_{N-n}$



RI du débruitage : $R := \blacksquare$ $\phi su := \blacksquare$ $f := \blacksquare$

Signal débruité : $y := \blacksquare$



Fonctions Bibliothèque :

Moyenne m, Vecteur r et Matrice d'Autocovariance r et Variance V du signal x de dim K (par ergodicité) :

. Moyenne :

$$\text{moy}(x) \equiv \begin{cases} K \leftarrow \text{length}(x) \\ m \leftarrow \frac{1}{K} \cdot \sum_{n=0}^{K-1} x_n \\ m \end{cases}$$

. Vecteur d'AutoCovariance :

$$\text{vautocov}(x) \equiv \begin{cases} K \leftarrow \text{length}(x) \\ m \leftarrow \text{moy}(x) \\ \text{for } k \in 0..K-1 \\ r_k \leftarrow \frac{1}{K} \cdot \sum_{n=0}^{K-k-1} (x_{n+k} - m) \cdot (x_n - m) \\ r \end{cases}$$

. Matrice d'AutoCovariance :

$$\text{mautocov}(x) \equiv \begin{cases} K \leftarrow \text{length}(x) \\ r \leftarrow \text{vautocov}(x) \\ \text{for } i \in 0..K-1 \\ \text{for } j \in 0..K-1 \\ R_{i,j} \leftarrow r_{|i-j|} \\ R \end{cases}$$

. Variance :

$$\text{Var}(x) \equiv \text{vautocov}(x)_0$$

Fonction YuleWalker(φ_{xx}, N) : Résolution du système de Yule-Walker Ra = v par inversion matricielle

Paramètres d'entrée : . φ_{xx} : vecteur fonction d'autocovariance (dim K) du signal x (dim K) : φ_{xx0}, φ_{xx1} ... φ_{xxK-1}
 . N : dim de l'ordre du filtre (formeur/LPC) AR

Paramètre de sortie : . le vecteur a (dim N+1) :
 a₀ = V : variance (du bruit blanc si filtre formeur); a₁, a₂, ... a_N : paramètres du filtre (formeur/LPC) AR d'ordre N

$$\text{YuleWalker}(\phi_{xx}, N) \equiv \begin{cases} \text{for } i \in 0..N \\ \text{for } j \in 0..N \\ R_{i,j} \leftarrow \phi_{xx} |i-j| \\ b_0 \leftarrow 0 \\ \text{for } i \in 1..N \\ b_i \leftarrow \phi_{xx_i} \\ \alpha \leftarrow \left(\frac{-1}{\phi_{xx_0}} \right) \cdot R^{-1} \cdot b \\ a_0 \leftarrow \frac{\phi_{xx_0}}{\phi_{xx_0} \cdot \alpha_0 + 1} \\ \text{for } i \in 1..N \\ a_i \leftarrow a_0 \cdot \alpha_i \\ a \end{cases}$$

Exemple : K := 3 r₀ := 6
 N := 2 r₁ := 3
 r₂ := 0

a := YuleWalker(r, N)

V := a₀

$$a = \begin{pmatrix} 4 \\ -0.667 \\ 0.333 \end{pmatrix}$$

V = 4
 i := 1..N
 a_i =

-0.667
0.333

Fonction YuleW(x,N) : Résolution du système de Yule-Walker $Ra = v$ par inversion matricielle

Paramètres d'entrée : . x : signal x (dim K) : $x_0, x_1 \dots x_{K-1}$
 . N : dim de l'ordre du filtre (formeur/LPC) AR

Paramètre de sortie : . le vecteur a (dim N+1) :
 $a_0 = V$: variance (du bruit blanc si filtre formeur); a_1, a_2, \dots, a_N : paramètres du filtre (formeur/LPC) AR d'ordre N

$$\text{YuleW}(x,N) \equiv \begin{cases} r \leftarrow \text{vautocov}(x) \\ a \leftarrow \text{YuleWalker}(r,N) \\ a \end{cases}$$

Bruit blanc de K points de moyenne m et de variance V :

Bruit blanc de K points centré de variance 0.083 :

$$\text{WhiteNoise}(K) \equiv \begin{cases} A \leftarrow 281 \\ P \leftarrow 31357 \\ v_0 \leftarrow 100 \\ \text{for } n \in 1..N-1 \\ \quad v_n \leftarrow \text{mod}(A \cdot v_{n-1}, P) \\ \text{for } n \in 0..N-1 \\ \quad w_n \leftarrow \frac{1}{P} \cdot v_n - 0.5 \\ w \end{cases}$$

Bruit blanc de K points de moyenne m et de variance V :

$$\text{BruitBlanc}(K, m, V) \equiv \sqrt{12 \cdot V} \cdot \text{whiten}(K) + m$$

$$\text{BruitBlanc2}(K, m, V) \equiv \sqrt{12 \cdot V} \cdot \text{WhiteNoise}(K) + m$$

Réponse reponse (vecteur dim. K) du filtre AR d'ordre N (coeffs $a_i, 0 \leq i \leq N$) au signal x (vecteur dim. K)

$$\text{reponse}(x, a, N) \equiv \begin{cases} K \leftarrow \text{length}(x) \\ \text{for } k \in 0..K-1 \\ \quad \begin{cases} y_k \leftarrow x_k & \text{if } k = 0 \\ y_k \leftarrow x_k - \sum_{i=0}^k a_i \cdot y_{k-i} & \text{if } 1 \leq k < N \\ y_k \leftarrow x_k - \sum_{i=0}^N a_i \cdot y_{k-i} & \text{if } k \geq N \end{cases} \\ y \end{cases}$$

TFD

$$\text{TFD}(x) \equiv \left| \begin{array}{l} M \leftarrow \text{length}(x) \\ \text{for } k \in 0..M-1 \\ \quad X_k \leftarrow \frac{1}{M} \cdot \sum_{n=0}^{M-1} x_n \cdot e^{-i \cdot 2 \cdot \pi \cdot k \cdot \frac{n}{M}} \\ \end{array} \right| X$$

ou encore :

$$\text{TFD2}(x) \equiv \left| \begin{array}{l} M \leftarrow \text{length}(x) \\ \text{for } k \in 0..M-1 \\ \quad R \leftarrow 0 \\ \quad I \leftarrow 0 \\ \quad \text{for } n \in 0..M-1 \\ \quad \quad R \leftarrow R + x_n \cdot \cos\left(2 \cdot \pi \cdot k \cdot \frac{n}{M}\right) \\ \quad \quad I \leftarrow I - x_n \cdot \sin\left(2 \cdot \pi \cdot k \cdot \frac{n}{M}\right) \\ \quad X_k \leftarrow \frac{1}{M} \cdot (R + i \cdot I) \\ \end{array} \right| \text{return } X$$

TFDI

$$\text{TFDI}(X) \equiv \left| \begin{array}{l} M \leftarrow \text{length}(X) \\ \text{for } n \in 0..M-1 \\ \quad x_n \leftarrow \sum_{k=0}^{M-1} X_k \cdot e^{i \cdot 2 \cdot \pi \cdot k \cdot \frac{n}{M}} \\ \end{array} \right| x$$

Convolution causale (signaux de même longueur)

$$\text{convcausal2}(x, h) \equiv \left| \begin{array}{l} M \leftarrow \text{length}(x) \\ \text{for } n \in 0..M-1 \\ \quad y_n \leftarrow \sum_{k=0}^n x_k \cdot h_{n-k} \\ \end{array} \right| y$$

Convolution causale (signaux de longueur différente)

$$\text{convcausal}(x, h) \equiv \left| \begin{array}{l} M \leftarrow \text{length}(x) \\ N \leftarrow \text{length}(h) \\ \text{for } n \in 0..M-1 \\ \quad y_n \leftarrow \sum_{k=0}^n x_k \cdot h_{n-k} \quad \text{if } n < N \\ \quad y_n \leftarrow 0 \quad \text{otherwise} \\ \end{array} \right| y$$

Transformée de Hilbert

```
Hilbert(x) ≡ | M ← length(x)
               | f0 ← 0
               | for n ∈ 1..M - 1
               |   fn ← 1 / (π · n)
               | TH ← convcausal(x, f)
               | TH
```

Vecteur r de covariance des signaux x et y de dim K (par ergodicité) :

. Vecteur de Covariance :

```
vcov(y, x) ≡ | K ← length(x)
               | for k ∈ 0..K - 1
               |   vk ← 1 / K · ∑n=0K-1-k yn+k · xn
               | v
```

```
vcovtermegeneral(x, y) ≡ | K ← length(x)
                           | mx ← moy(x)
                           | my ← moy(y)
                           | for k ∈ 0..K - 1
                           |   rk ← 1 / (K - k) · ∑n=0K-k-1 (xn+k - mx) · (yn - my)
                           | r
```

Vecteur et Matrice nuls :

```
VecteurNul(N) ≡ | for k ∈ 0..N - 1
                  |   Nulk ← 0
                  | Nul
```

```
MatriceNul(Nlignes, Ncols) ≡ | for l ∈ 0..Nlignes - 1
                               |   for c ∈ 0..Ncols - 1
                               |     Null,c ← 0
                               | Nul
```

6. Prédiction

Prédiction

Lissage - Filtrage - Prédiction

Soit n mesures d'un signal. Le traitement de la mesure au temps :

- . $n-1$ ($n-k$ en général, avec $k > 0$) (*passé*) : s'appelle le *lissage*
- . n (*présent*) : s'appelle le *filtrage*
- . $n+1$ ($n+k$ en général, avec $k > 0$) (*futur*) : s'appelle la *prédiction*

PREDICTION LINEAIRE (Codage LPC – Linear Prediction Coding)

Etant donné les N échantillons passés d'un signal comportant M points ($N \leq M$) : $y_{n-1}, y_{n-2}, \dots, y_{n-N}$, il s'agit d'en prédire l'échantillon présent : y_n .

Le plus simple est d'effectuer une prédiction linéaire : $\hat{y}_n = -a_1 y_{n-1} - a_2 y_{n-2} - \dots - a_N y_{n-N}$ $\hat{y}_n = -\sum_{i=1}^N a_i y_{n-i}$

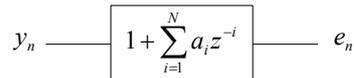
Le filtre de prédiction est donc un filtre AR (AutoRégressif) pur (\equiv sans entrée).

Cela induit que, du fait de la stationnarité du modèle adopté (AR), la prédiction ne sera fiable que si le signal à prédire est relativement stationnaire.

On peut définir l'erreur de prédiction : $e_n \stackrel{\Delta}{=} y_n - \hat{y}_n$ soit $e_n = y_n + a_1 y_{n-1} + a_2 y_{n-2} + \dots + a_N y_{n-N}$

d'où : $e_n = \sum_{i=0}^N a_i y_{n-i}$ (avec $a_0 = 1$) \xrightarrow{TZ} $E(z) = \sum_{i=0}^N a_i z^{-i} Y(z)$

c'est l'équation du filtre « erreur de prédiction » : $\frac{E(z)}{Y(z)} = \sum_{i=0}^N a_i z^{-i} = 1 + \sum_{i=1}^N a_i z^{-i}$:



il s'agit d'un filtre (MA) inverse (au sens de FT inverse) du filtre AR de FT : $\frac{1}{1 + \sum_{i=1}^N a_i z^{-i}}$.

Posant le problème de prédiction en termes stochastiques (vu le nombre important de paramètres en jeu), le filtre « erreur de prédiction » précédent : $e_n = \sum_{i=0}^N a_i y_{n-i}$ s'écrit : $E_n = \sum_{i=0}^N a_i Y_{n-i}$ où les signaux E_n et Y_n sont aléatoires, et on se propose de calculer, pour ce filtre, les paramètres a_i qui minimisent la puissance (\approx la variance) de E_n .

Rappel : . Puissance de X_n : $P_X = E[X_n^2]$ \approx . Variance de X_n : $V_X = E[X_n^2] - E^2[X_n]$

Puissance P_E de E_n : $P_E = E[E_n^2] = ?$

$$E[E_n^2] = E\left[\sum_{i=0}^N a_i Y_{n-i} \cdot \sum_{j=0}^N a_j Y_{n-j}\right] = \sum_{i=0}^N \sum_{j=0}^N a_i a_j E[Y_{n-i} Y_{n-j}] \rightarrow \boxed{E[E_n^2] = \sum_{i=0}^N \sum_{j=0}^N a_i a_j \varphi_{yy_{j-i}}}$$

où φ_{yy_k} est la fonction (vecteur) d'autocovariance (\equiv d'autocorrélation dans le cas stationnaire) du signal y .

Puissance minimale: (on peut en calculant la dérivée 2^{de} de P_E s'assurer que l'annulation de la dérivée 1^{ère} de P_E conduit bien à un minimum de P_E)

$$\frac{\partial E[E_n^2]}{\partial a_n} = \frac{\partial}{\partial a_n} \left[a_n^2 \varphi_{yy_0} + 2a_n \sum_{\substack{m=0 \\ m \neq n}}^N a_m \varphi_{yy_{n-m}} + \text{termes ne contenant pas } a_n \right] = 2a_n \varphi_{yy_0} + 2 \sum_{\substack{m=0 \\ m \neq n}}^N a_m \varphi_{yy_{n-m}} = 0$$

soit (condition d'optimalité) :

$$a_n \varphi_{yy_0} + \sum_{\substack{m=0 \\ m \neq n}}^N a_m \varphi_{yy_{n-m}} = 0 \quad \left(\begin{array}{l} n = 0, \dots, N \\ a_0 = 1 \end{array} \right) \Leftrightarrow a_n \varphi_{yy_0} + \sum_{\substack{m=1 \\ m \neq n}}^N a_m \varphi_{yy_{n-m}} + \left| \begin{array}{l} a_0 \varphi_{yy_n} \\ n \neq 0 \end{array} \right. = 0 \quad \left(\begin{array}{l} n = 0, \dots, N \\ a_0 = 1 \end{array} \right)$$

$$\Leftrightarrow \sum_{m=1}^N a_m \varphi_{yy_{n-m}} + \left| \begin{array}{l} a_0 \varphi_{yy_n} \\ n \neq 0 \end{array} \right. = 0 \quad \left(\begin{array}{l} n = 0, \dots, N \\ a_0 = 1 \end{array} \right) \Leftrightarrow \sum_{m=1}^N a_m \varphi_{yy_{n-m}} = -\varphi_{yy_n} \quad \left(\begin{array}{l} n = 1, \dots, N \\ a_0 = 1 \end{array} \right)$$

ou encore :
$$\boxed{\sum_{m=0}^N a_m \varphi_{yy_{n-m}} = 0} \quad \left(\begin{array}{l} n = 1, \dots, N \\ a_0 = 1 \end{array} \right)$$

Du fait que φ_{yy_k} est paire, on a :
$$E[E_n^2] = \sum_{i=0}^N \sum_{j=0}^N a_i a_j \varphi_{yy_{j-i}} = \sum_{i=0}^N \sum_{j=1}^N a_i a_j \varphi_{yy_{j-i}} + a_0 \sum_{i=0}^N a_i \varphi_{yy_i} \quad (a_0 = 1)$$

$$\rightarrow E[E_n^2] = \sum_{j=1}^N a_j \sum_{i=0}^N a_i \varphi_{yy_{j-i}} + a_0 \sum_{i=0}^N a_i \varphi_{yy_i} \quad (a_0 = 1)$$

En utilisant: $\sum_{m=0}^N a_m \varphi_{yy_{n-m}} = 0 \quad \left(\begin{array}{l} n = 1, \dots, N \\ a_0 = 1 \end{array} \right)$ dans la relation précédente, il vient:
$$\boxed{E[E_n^2]_{\min} = \sum_{i=0}^N a_i \varphi_{yy_i} \stackrel{\Delta}{=} V} \quad (a_0 = 1)$$

On a donc :
$$\boxed{\sum_{m=0}^N a_m \varphi_{yy_{n-m}} = 0} \quad \left(\begin{array}{l} n = 1, \dots, N \\ a_0 = 1 \end{array} \right) \quad \text{et} \quad \boxed{E[E_n^2]_{\min} = \sum_{i=0}^N a_i \varphi_{yy_i} \stackrel{\Delta}{=} V} \quad (a_0 = 1).$$

Ces équations peuvent s'écrire :

$$\left\{ \begin{array}{l} \varphi_{yy_0} + \sum_{i=1}^N a_i \varphi_{yy_i} = V \\ \varphi_{yy_1} + \sum_{i=1}^N a_i \varphi_{yy_{i-1}} = 0 \\ \dots \\ \varphi_{yy_N} + \sum_{i=1}^N a_i \varphi_{yy_{N-i}} = 0 \end{array} \right. \Leftrightarrow \text{pour } 0 \leq k \leq N : \sum_{n=0}^N \varphi_{yy_{k-n}} a_n = \begin{cases} V & \text{si } k = 0 \\ 0 & \text{si } 1 \leq k \leq N \end{cases} \quad (a_0 = 1) (\varphi_{yy_k} \text{ paire})$$

soit, sous forme matricielle : ce sont les équations de **Yule-Walker** :

$$\begin{bmatrix} \varphi_{yy_0} & \varphi_{yy_1} & \dots & \varphi_{yy_N} \\ \varphi_{yy_1} & \varphi_{yy_0} & \dots & \varphi_{yy_{N-1}} \\ \vdots & \vdots & \ddots & \vdots \\ \varphi_{yy_N} & \varphi_{yy_{N-1}} & \dots & \varphi_{yy_0} \end{bmatrix} \begin{bmatrix} 1 \\ a_1 \\ \vdots \\ a_N \end{bmatrix} = \begin{bmatrix} V \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad \text{ou encore : } \boxed{\mathbf{R} \mathbf{a} = \mathbf{v}}$$

La matrice d'autocorrélation \mathbf{R} de terme général $r_{i,j}$ ne dépendant que de $i-j$ (φ_{yy_k} paire) est dite matrice de Toeplitz.

Ainsi, à partir de la connaissance de la séquence d'autocorrélation de $\{Y_n\}$, la relation matricielle permet de déterminer les $N+1$ inconnues : V et les paramètres a_i du filtre prédicteur.

Pour résoudre le système de **Yule-Walker**, une inversion matricielle est possible mais on utilise généralement l'algorithme efficace itératif de **Levinson**, ou la méthode de **Burg**, qui évitent une inversion matricielle.

(La méthode de **Burg**, introduisant la contrainte supplémentaire de stabilité du filtre de prédiction au système de **Yule-Walker**, garantit ainsi un filtre AR de prédiction stable).

Remarque :

D'un point de vue algorithmique, les 2 problèmes de filtre prédicteur et de synthèse AR par filtre formeur sont équivalents. On montre d'ailleurs que, pour N suffisamment grand, $\{E_n\}$ est un bruit blanc.

Application

▪ Compression de signal par codage LPC

On considère un signal que l'on souhaite compresser par codage LPC.

Le principe du codage LPC consiste à modéliser un signal x comme sortie d'un filtre AR d'ordre N (modèle tout pôle, sans entrée, tournant sur ses seules conditions initiales) :

$$x_n = \sum_{k=1}^N -a_k x_{n-k} \quad a_k : N \text{ coefficients du filtre AR}$$

La minimisation de l'erreur (dite erreur de prédiction) entre le signal original et le signal issu du filtre AR conduit à la détermination des coefficients $a_1, a_2 \dots a_N$ du filtre AR.

Cette détermination met en oeuvre un système d'équations linéaires (système de Yule-Walker), dont la résolution peut se faire de plusieurs façons :

- directement (méthode itérative de Yule-Walker)
- par la méthode itérative de Burg

où s est une fenêtre glissante (sans recouvrement) sur le signal original x à compresser, et N l'ordre désiré pour le filtre AR ($1 < N < M$, avec $M = \text{length}(s)$)

(l'avantage de la méthode de Burg est de garantir un filtre AR stable).

Ainsi, pour une fenêtre s donnée (de M points), après avoir déterminé les N coefficients a_k du filtre AR prédicteur, on peut reconstruire une approximation \hat{s} de s par l'équation :

$$n = N \dots M - 1 : \hat{s}_n = \sum_{k=1}^N -a_k \hat{s}_{n-k} \text{ avec l'initialisation : } \hat{s}_n = s_n \text{ pour } n = 0 \dots N - 1$$

Taux de compression

Pour compresser une fenêtre de M points, on a besoin de N coefficients a_k et de N points d'initialisation.

Le taux de compression peut donc être décrit par l'évaluation du rapport : $2N / M$.

Taux de distorsion

L'évaluation du taux de distorsion entre le signal décompressé (ou encore prédit) \hat{x} et le signal original x peut se décrire par l'évaluation du rapport Signal sur Bruit global sur tout le signal x (SNROV) (Signal to Noise Ratio Overall)

$$\text{SNROV} = 10 \log \left(\frac{\sum_{n=0}^{\text{length}(x)-1} x_n^2}{\sum_{n=0}^{\text{length}(x)-1} b_n^2} \right) = 10 \log \left(\frac{\sum_{n=0}^{\text{length}(x)-1} x_n^2}{\sum_{n=0}^{\text{length}(x)-1} (x_n - \hat{x}_n)^2} \right) \text{ avec } b_n = x_n - \hat{x}_n \quad (\text{SNROV en dB})$$

▪ Caractérisation : Analyse spectrale par codage LPC [Voir Chapitre Caractérisation]

TP 6. Prédiction

1. Prédiction (sur signal très stationnaire)

Signal à prédire : $K := 30$ $n := 0, 1.. K - 1$ $y_n := \sin\left(\frac{n}{5}\right)$

Ordre du filtre AR (taille du passé pour la prédiction) : $N :=$

Centrer le signal avant prédiction et le décentrer après

$m := \text{moy}(y)$ $y := y - m$

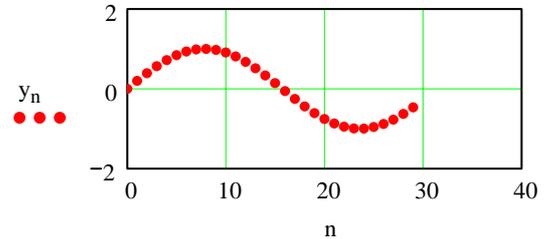
Prédiction à 1 pas :

Coefficients a du filtre AR de LPC :

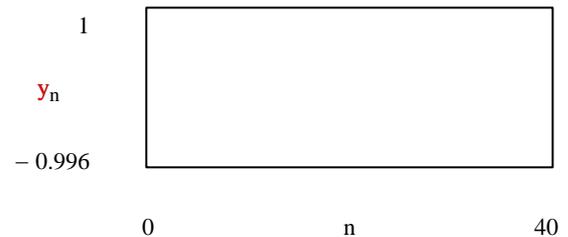
$\phi :=$ $a :=$ $a =$

ou : $a := \text{YuleW}(y, N)$ $a = \begin{pmatrix} 0.01 \\ -1.722 \\ 0.766 \end{pmatrix}$

ou ENCORE (fonction intégrée Mathcad) : $a := \text{yulew}(y, N)$ $a = \begin{pmatrix} 1 \\ -1.722 \\ 0.766 \end{pmatrix}$

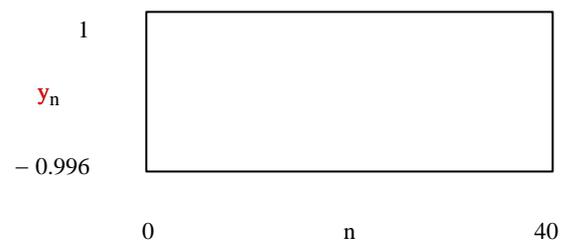


Filtre AR de LPC : $y_K :=$ $n := 0, 1.. K$ $y := y + m$



Fonction Prédiction à 1 pas : $\text{Predict1pas}(v, N)$

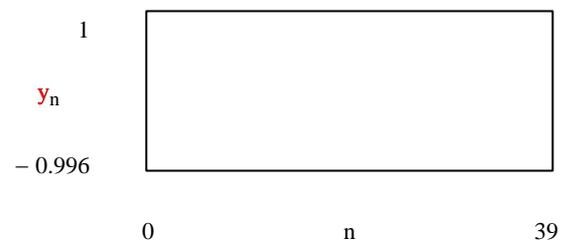
$y := \text{predict1pas}(y, N)$ $n := 0, 1.. K$



Fonction Prédiction à Npas pas : $\text{PredictNpas}(v, N, Npas)$ $Npas := 10$

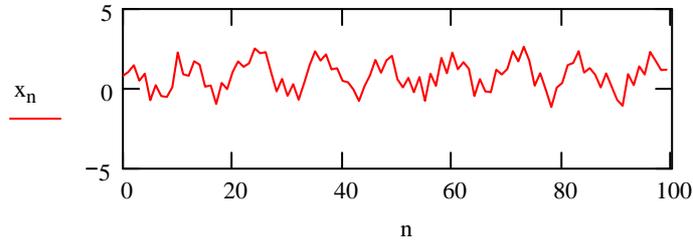
$y := \text{predictNpas}(y, N, Npas)$ $n := 0, 1.. K - 1 + Npas$

Conclusion :

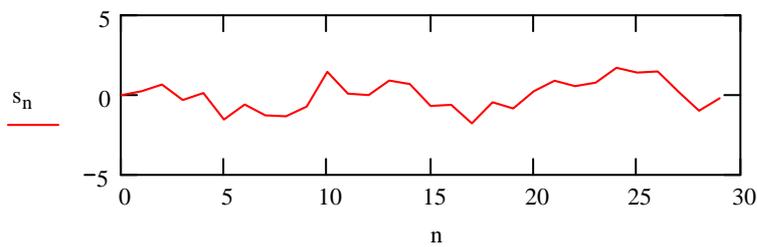


2. Application (sur signal peu stationnaire) : Compression de signal par LPC : 2 méthodes : YuleWalker & Burg

Signal x original (L points) $L := 100$ $n := 0..L - 1$ $x_n := \cos\left(\frac{2 \cdot \pi \cdot n}{12}\right) + \text{rnd}(2) - .2$



Fenêtrage s du signal original (M points) + Centrage $M := 30$ $n := 0..M - 1$ $s_n := x_n$ $m := \text{moy}(s)$ $s := s -$



Ordre N du filtre AR de prediction $N := \text{trunc}\left(\frac{M}{2} - 5\right)$ $N = 10$

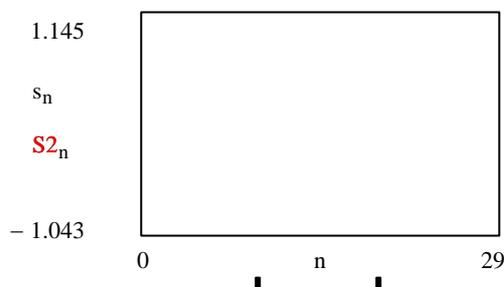
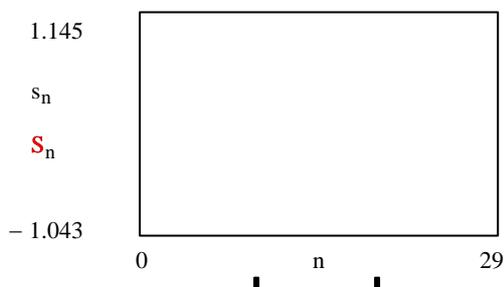
Coefficients a du filtre AR de prediction (2 méthodes Yule-Walker et Burg)
 Yule-Walker $a := \text{yulew}(s, N)$ Burg $a2 := \text{burg}(s, N)$

Signal S issu du filtre AR de prediction

Initialisation $n := 0..N - 1$ $S_n := \blacksquare$ $S2_n := \blacksquare$

Calcul $n := N..M - 1$ $S_n := \blacksquare$ $S2_n := \blacksquare$

Comparaison entre le signal original s et le signal prédit S $n := 0..M - 1$



Rapport Signal/Bruit sur une fenêtre

SNROVdB := ■

SNROV2dB := ■

SNROVdB = ■

SNROV2dB = ■

Taux de compression

Nombre d'échantillons de la fenêtre du signal original :

M = 30

Nombre de points du signal compressé, prédit, pour une fenêtre :

2·N = 20

Conclusions

. **Methode de Yule-Walker :**

. **Methode de Burg :**

Fonctions Bibliothèque :

Moyenne m, Vecteur r et Matrice d'Autocovariance r et Variance V du signal x de dim K (par ergodicité) :

. Moyenne :

$$\text{moy}(x) \equiv \begin{cases} K \leftarrow \text{length}(x) \\ m \leftarrow \frac{1}{K} \cdot \sum_{n=0}^{K-1} x_n \\ m \end{cases}$$

. Vecteur d'AutoCovariance :

$$\text{vautocov}(x) \equiv \begin{cases} K \leftarrow \text{length}(x) \\ m \leftarrow \text{moy}(x) \\ \text{for } k \in 0..K-1 \\ r_k \leftarrow \frac{1}{K} \cdot \sum_{n=0}^{K-k-1} (x_{n+k} - m) \cdot (x_n - m) \\ r \end{cases}$$

. Matrice d'AutoCovariance :

$$\text{mautocov}(x) \equiv \begin{cases} K \leftarrow \text{length}(x) \\ r \leftarrow \text{vautocov}(x) \\ \text{for } i \in 0..K-1 \\ \text{for } j \in 0..K-1 \\ R_{i,j} \leftarrow r_{|i-j|} \\ R \end{cases}$$

. Variance :

$$\text{Var}(x) \equiv \text{vautocov}(x)_0$$

Fonction YuleWalker(ϕ_{xx}, N) : Résolution du système de Yule-Walker $Ra = v$ par inversion matricielle

Paramètres d'entrée : . ϕ_{xx} : vecteur fonction d'autocovariance (dim K) du signal x (dim K) : $\phi_{xx0}, \phi_{xx1} \dots \phi_{xxK-1}$
 . N : dim de l'ordre du filtre (formeur/LPC) AR

Paramètre de sortie : . le vecteur a (dim N+1) :
 $a_0 = V$: variance (du bruit blanc si filtre formeur); a_1, a_2, \dots, a_N : paramètres du filtre (formeur/LPC) AR d'ordre N

$$\text{YuleWalker}(\phi_{xx}, N) \equiv \begin{cases} \text{for } i \in 0..N \\ \text{for } j \in 0..N \\ R_{i,j} \leftarrow \phi_{xx}|i-j| \\ b_0 \leftarrow 0 \\ \text{for } i \in 1..N \\ b_i \leftarrow \phi_{xx_i} \\ \alpha \leftarrow \left(\frac{-1}{\phi_{xx0}} \right) \cdot R^{-1} \cdot b \\ a_0 \leftarrow \frac{\phi_{xx0}}{\phi_{xx0} \cdot \alpha_0 + 1} \\ \text{for } i \in 1..N \\ a_i \leftarrow a_0 \cdot \alpha_i \\ a \end{cases}$$

Exemple : $K := 3$ $r_0 := 6$
 $N := 2$ $r_1 := 3$
 $r_2 := 0$

$$a := \text{YuleWalker}(r, N)$$

$$V := a_0$$

$$a = \begin{pmatrix} 4 \\ -0.667 \\ 0.333 \end{pmatrix}$$

$$V = 4$$

$$i := 1..N$$

$$a_i =$$

-0.667
0.333

Fonction YuleW(x,N) : Résolution du système de Yule-Walker $Ra = v$ par inversion matricielle

Paramètres d'entrée : . x: signal x (dim K) : $x_0, x_1 \dots x_{K-1}$
 . N : dim de l'ordre du filtre (formeur/LPC) AR

Paramètre de sortie : . le vecteur a (dim N+1) :
 $a_0 = V$: variance (du bruit blanc si filtre formeur); $a_1, a_2, \dots a_N$: paramètres du filtre (formeur/LPC) AR d'ordre N

$$\text{YuleW}(x,N) \equiv \left\{ \begin{array}{l} r \leftarrow \text{vautocov}(x) \\ a \leftarrow \text{YuleWalker}(r,N) \\ a \end{array} \right.$$

Bruit blanc de K points de moyenne m et de variance V :

Bruit blanc de K points centré de variance 0.083 :

$$\text{WhiteNoise}(K) \equiv \left\{ \begin{array}{l} A \leftarrow 281 \\ P \leftarrow 31357 \\ v_0 \leftarrow 100 \\ \text{for } n \in 1..N-1 \\ \quad v_n \leftarrow \text{mod}(A \cdot v_{n-1}, P) \\ \text{for } n \in 0..N-1 \\ \quad w_n \leftarrow \frac{1}{P} \cdot v_n - 0.5 \\ w \end{array} \right.$$

Bruit blanc de K points de moyenne m et de variance V :

$$\text{BruitBlanc}(K, m, V) \equiv \sqrt{12 \cdot V} \cdot \text{whiten}(K) + m$$

$$\text{BruitBlanc2}(K, m, V) \equiv \sqrt{12 \cdot V} \cdot \text{WhiteNoise}(K) + m$$

Réponse reponse (vecteur dim. K) du filtre AR d'ordre N (coeffs $a_i, 0 \leq i \leq N$) au signal x (vecteur dim. K)

$$\text{reponse}(x, a, N) \equiv \left\{ \begin{array}{l} K \leftarrow \text{length}(x) \\ \text{for } k \in 0..K-1 \\ \quad \left\{ \begin{array}{l} y_k \leftarrow x_k \text{ if } k = 0 \\ y_k \leftarrow x_k - \sum_{i=0}^k a_i \cdot y_{k-i} \text{ if } 1 \leq k < N \\ y_k \leftarrow x_k - \sum_{i=0}^N a_i \cdot y_{k-i} \text{ if } k \geq N \end{array} \right. \\ y \end{array} \right.$$

TFD

$$\text{TFD}(x) \equiv \left| \begin{array}{l} M \leftarrow \text{length}(x) \\ \text{for } k \in 0..M-1 \\ \quad X_k \leftarrow \frac{1}{M} \cdot \sum_{n=0}^{M-1} x_n \cdot e^{-i \cdot 2 \cdot \pi \cdot k \cdot \frac{n}{M}} \\ \end{array} \right| X$$

ou encore :

$$\text{TFD2}(x) \equiv \left| \begin{array}{l} M \leftarrow \text{length}(x) \\ \text{for } k \in 0..M-1 \\ \quad R \leftarrow 0 \\ \quad I \leftarrow 0 \\ \quad \text{for } n \in 0..M-1 \\ \quad \quad R \leftarrow R + x_n \cdot \cos\left(2 \cdot \pi \cdot k \cdot \frac{n}{M}\right) \\ \quad \quad I \leftarrow I - x_n \cdot \sin\left(2 \cdot \pi \cdot k \cdot \frac{n}{M}\right) \\ \quad X_k \leftarrow \frac{1}{M} \cdot (R + i \cdot I) \\ \end{array} \right| \text{return } X$$

TFDI

$$\text{TFDI}(X) \equiv \left| \begin{array}{l} M \leftarrow \text{length}(X) \\ \text{for } n \in 0..M-1 \\ \quad x_n \leftarrow \sum_{k=0}^{M-1} X_k \cdot e^{i \cdot 2 \cdot \pi \cdot k \cdot \frac{n}{M}} \\ \end{array} \right| x$$

Convolution causale (signaux de même longueur)

$$\text{convcausal2}(x, h) \equiv \left| \begin{array}{l} M \leftarrow \text{length}(x) \\ \text{for } n \in 0..M-1 \\ \quad y_n \leftarrow \sum_{k=0}^n x_k \cdot h_{n-k} \\ \end{array} \right| y$$

Convolution causale (signaux de longueur différente)

$$\text{convcausal}(x, h) \equiv \left| \begin{array}{l} M \leftarrow \text{length}(x) \\ N \leftarrow \text{length}(h) \\ \text{for } n \in 0..M-1 \\ \quad y_n \leftarrow \sum_{k=0}^n x_k \cdot h_{n-k} \quad \text{if } n < N \\ \quad y_n \leftarrow 0 \quad \text{otherwise} \\ \end{array} \right| y$$

Transformée de Hilbert

```
Hilbert(x) ≡ | M ← length(x)
               | f0 ← 0
               | for n ∈ 1..M - 1
               |   fn ← 1 / (π · n)
               | TH ← convcausal(x, f)
               | TH
```

Vecteur r de covariance des signaux x et y de dim K (par ergodicité) :

. Vecteur de Covariance :

```
vcov(y, x) ≡ | K ← length(x)
               | for k ∈ 0..K - 1
               |   vk ← 1 / K · ∑n=0K-1-k yn+k · xn
               | v
```

```
vcovtermegeneral(x, y) ≡ | K ← length(x)
                           | mx ← moy(x)
                           | my ← moy(y)
                           | for k ∈ 0..K - 1
                           |   rk ← 1 / (K - k) · ∑n=0K-k-1 (xn+k - mx) · (yn - my)
                           | r
```

Vecteur et Matrice nuls :

```
VecteurNul(N) ≡ | for k ∈ 0..N - 1
                  |   Nulk ← 0
                  | Nul
```

```
MatriceNul(Nlignes, Ncols) ≡ | for l ∈ 0..Nlignes - 1
                              | for c ∈ 0..Ncols - 1
                              |   Null,c ← 0
                              | Nul
```

Prédiction à 1 pas

```
predict1pas(v, N) ≡ |
```

Prédiction à Npas pas

```
predictNpas(v, N, Npas) ≡ |
```

7. Filtrage optimal

Filtrage optimal

(Filtrage sous contrainte)

Il existe 3 grandes méthodes de filtrage optimal, c'est-à-dire de filtrage sous contrainte d'un signal altéré par un bruit :

. le filtrage par moindres carrés

Il consiste en une minimisation de l'énergie de l'erreur entre le signal mesuré et l'estimation (par un modèle) du signal non bruité. Le modèle adopté pour le signal non altéré est un modèle linéaire et assez général. Il y a lissage du bruit plus que sa réjection. Le filtrage des moindres carrés se place d'un point de vue déterministe. Il y a contrainte de stationnarité.

. le filtrage de Wiener

Il s'appuie sur la méthode des moindres carrés : le filtrage de Wiener est l'analogie en stochastique du filtrage des moindres carrés qui lui, se place dans un contexte déterministe. Il y a contrainte de stationnarité.

. le filtrage de Kalman

Il consiste en la poursuite d'un modèle d'Etat (modèle de Gauss-Markov) nécessaire et relativement précis du signal non bruité. Il n'y a pas de contrainte de stationnarité. Si le modèle est correct, il y a véritable réjection du bruit et non pas un lissage. Si le modèle n'est pas réaliste, les résultats peuvent être très mauvais.

I. MOINDRES CARRES

Moindres carrés direct

Algorithme LS (*Least Squares*)

Un système linéaire de n équations à n inconnues : $\mathbf{Xh} = \mathbf{y}$

où :

- \mathbf{X} désigne une matrice carrée de dimension $n \times n$
- \mathbf{h} désigne un vecteur de dimension $n \times 1$ inconnu
- \mathbf{y} désigne un vecteur de dimension $n \times 1$

possède une solution unique :

$$\mathbf{h} = \mathbf{X}^{-1} \mathbf{y}$$

si et seulement si \mathbf{X} est inversible (\equiv non singulière) c'est-à-dire $\det(\mathbf{X}) \neq 0$ (\equiv matrice régulière).

Erreur (ou écart) : $\boldsymbol{\varepsilon} = \mathbf{y} - \mathbf{Xh}$ (écart entre la solution vraie et la solution estimée, calculée) : $\boldsymbol{\varepsilon} = 0$

Notons que dans ce cas, l'écart quadratique, défini par : $\boldsymbol{\varepsilon}^2 = (\mathbf{y} - \mathbf{Xh})^T (\mathbf{y} - \mathbf{Xh})$

(plus généralement : $\boldsymbol{\varepsilon}^2 = (\mathbf{y} - \mathbf{Xh})^H (\mathbf{y} - \mathbf{Xh})$ dans le cas complexe, où l'indice H signifie la transposition-conjugaison)

est juste égal à 0 : $\boldsymbol{\varepsilon}^2 = 0$

(Si la matrice \mathbf{X} n'est pas carrée, on définit sa *pseudo-inverse* \mathbf{X}^\dagger en minimisant un critère quadratique (*moindres carrés*)).

L'utilisation d'approximations linéaires fait que, en traitement du signal, de nombreux signaux peuvent se modéliser sous la forme \mathbf{Xh} où :

- \mathbf{X} désigne une matrice connue
- et \mathbf{h} désigne un vecteur inconnu.

Exemple : Filtre RIF de RI \mathbf{h} inconnue.

Dans ce cas : \mathbf{X} est une matrice obtenue à partir de valeurs de l'entrée, et le produit \mathbf{Xh} représente le signal de sortie (le produit scalaire de vecteurs réalise la convolution).

En l'absence de bruit, l'observation de n valeurs du signal de sortie suffit donc à estimer \mathbf{h} :

$$\mathbf{h} = \mathbf{X}^{-1} \mathbf{y} \quad (\text{déconvolution})$$

En pratique, il n'en est jamais ainsi et on se donne plus d'observations (k) que d'inconnues (n) (du fait de la présence d'un bruit de mesure) : il y a alors une « infinité de solutions non exactes » et la méthode des moindres carrés fournit alors le vecteur à estimer (de dimension n) optimal au sens quadratique à partir des k observations.

En notant \mathbf{y} le vecteur d'observation et en supposant le bruit \mathbf{w} additif, on a : $\mathbf{y} = \mathbf{Xh} + \mathbf{w}$

On montre en statistique, que l'estimateur de \mathbf{h} , construit à partir des observations \mathbf{y} et de la matrice \mathbf{X} , est d'autant meilleur que le nombre k de points observations est grand, par conséquent : $k \geq n$

alors la matrice \mathbf{X} est de dimension $k \times n$

et les vecteurs \mathbf{h} de dimension $n \times 1$

\mathbf{y} de dimension $k \times 1$

On a donc plus d'équations k que d'inconnues n : on dit que le système est *surdéterminé*.

La conséquence est qu'il n'existe pas en général de vecteur \mathbf{h} tel que \mathbf{Xh} soit parfaitement égal à l'observation \mathbf{y} .

Une autre façon de le dire est que le vecteur \mathbf{y} , de dimension $k \times 1$, n'appartient pas au sous-espace vectoriel engendré par les n vecteurs-colonnes de \mathbf{X} . Ce sous-espace appelé *image* de \mathbf{X} sera noté $\text{Img}(\mathbf{X})$.

Le problème est de trouver un vecteur \mathbf{h} , de dimension $n \times 1$, tel que l'élément \mathbf{Xh} (qui appartient évidemment à $\text{Img}(\mathbf{X})$) soit le plus proche possible de \mathbf{y} , ce qui se traduit ici par la *minimisation* de l'erreur quadratique (minimisation de l'énergie de l'erreur ε) :

$$\varepsilon^2 = J(\mathbf{h}) = (\mathbf{y} - \mathbf{Xh})^T (\mathbf{y} - \mathbf{Xh}) \quad (\text{plus généralement : } \varepsilon^2 = (\mathbf{y} - \mathbf{Xh})^H (\mathbf{y} - \mathbf{Xh}))$$

La solution est donnée par le *théorème de projection (principe d'orthogonalité)*, qui dit que le vecteur \mathbf{h} est tel que l'écart : $\varepsilon = \mathbf{y} - \mathbf{Xh}$ est orthogonal à $\text{Img}(\mathbf{X})$ et donc en particulier aux n vecteurs-colonnes de la matrice \mathbf{X} :

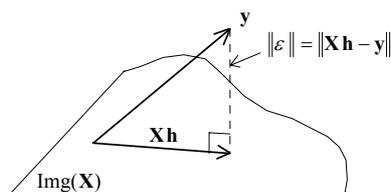
- $\varepsilon^2 = (\mathbf{y} - \mathbf{Xh})^T (\mathbf{y} - \mathbf{Xh})$

- Minimisation de l'énergie de l'erreur $\varepsilon \rightarrow$ Détermination de \mathbf{h} tel que $\frac{\partial \varepsilon^2}{\partial \mathbf{h}} = 0$

(pour effectivement vérifier que l'on a un minimum de ε^2 et non un maximum, il suffit de regarder le signe des dérivées secondes : $\frac{\partial^2 \varepsilon^2}{\partial \mathbf{h}^2} > 0$)

- $\frac{\partial \varepsilon^2}{\partial \mathbf{h}} = -2 \mathbf{X}^T \cdot (\mathbf{y} - \mathbf{Xh}) = 0$ (produit scalaire nul) $\rightarrow \mathbf{X}^T \cdot (\mathbf{y} - \mathbf{Xh}) = 0 \rightarrow \{\varepsilon = (\mathbf{y} - \mathbf{Xh})\} \perp \mathbf{X}^T$

(l'erreur ε est orthogonale à l'observation \mathbf{X}^T (et comme \mathbf{Xh} colinéaire à $\mathbf{X}^T \rightarrow \varepsilon \perp \mathbf{Xh}$) :



Théorème de projection

En exprimant cette condition d'orthogonalité, on obtient :

$$\begin{aligned} \mathbf{X}^T(\mathbf{y} - \mathbf{X}\mathbf{h}) &= \mathbf{0} && \text{(plus généralement : } \mathbf{X}^H(\mathbf{y} - \mathbf{X}\mathbf{h}) = \mathbf{0}\text{)} \\ \text{soit : } \mathbf{X}^T\mathbf{y} &= \mathbf{X}^T\mathbf{X}\mathbf{h} && \text{(plus généralement : } \mathbf{X}^H\mathbf{y} = \mathbf{X}^H\mathbf{X}\mathbf{h}\text{)} \end{aligned}$$

Si la matrice \mathbf{X} est de rang plein (\equiv les n vecteurs-colonnes de \mathbf{X} sont indépendants), la matrice $\mathbf{X}^T\mathbf{X}$ (plus généralement : $\mathbf{X}^H\mathbf{X}$) de dimension $n \times n$, est inversible et la solution cherchée est l'estimateur des moindres carrés (Gauss) :

$$\boxed{\mathbf{h} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}} \quad \text{(plus généralement : } \mathbf{h} = (\mathbf{X}^H\mathbf{X})^{-1}\mathbf{X}^H\mathbf{y}\text{)}$$

A l'origine, l'estimateur des moindres carrés n'est pas à proprement parler de nature probabiliste.

Le filtrage des moindres carrés est l'analogie en déterministe du filtrage de Wiener qui lui, se place dans un contexte stochastique.

De plus, on montre que, sous l'hypothèse que le bruit \mathbf{w} est blanc, l'estimateur des moindres carrés de \mathbf{h} est linéaire, sans biais et de variance minimale (estimateur optimal).

Enfin, la méthode des moindres carrés s'étend sans difficulté à un modèle d'observation non linéaire. Il suffit de remplacer dans le problème précédent $\mathbf{X}\mathbf{h}$ par l'expression supposée de forme connue $\mathbf{X}(\mathbf{h})$ dépendant de façon non linéaire du paramètre \mathbf{h} à estimer. Malheureusement, il n'est alors plus possible de donner une expression analytique de l'argument du minimum de la fonction de coût $J(\mathbf{h}) = [\mathbf{y} - \mathbf{X}(\mathbf{h})]^T[\mathbf{y} - \mathbf{X}(\mathbf{h})]$ (plus généralement $J(\mathbf{h}) = [\mathbf{y} - \mathbf{X}(\mathbf{h})]^H[\mathbf{y} - \mathbf{X}(\mathbf{h})]$) et on se contente d'algorithmes classiques de minimisation numérique. Ces algorithmes souffrent en général de l'existence de minima locaux et posent le problème de la vitesse de convergence. En outre, ces minima ne possèdent pas le caractère optimal précédent.

Moindres carrés récursif Algorithme RLS (Recursive Least Squares)

La relation des moindres carrés direct : $\mathbf{h} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$ fait que le calcul de \mathbf{h} nécessite la manipulation d'une matrice \mathbf{X} de dimension $k \times n$.

Si cette matrice est construite à partir d'un flux continu de données et que l'on veut effectuer le calcul en *temps réel*, le fait que la taille de \mathbf{X} croît de façon illimitée rend impossible un calcul direct à partir de la relation $\mathbf{h} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$.

Heureusement, l'algorithme des moindres carrés récursif fournit une solution pouvant être implémentée en temps réel et qui présente en outre l'avantage de ne pas recourir à une inversion matricielle. Il fonctionne par mise à jour de la valeur \mathbf{h} au fur et à mesure que les données arrivent.

Indexons par i les valeurs obtenues à l'étape i : $\mathbf{h}_i = \mathbf{P}_i \mathbf{X}_i^T \mathbf{y}_i$ avec : $\mathbf{P}_i = (\mathbf{X}_i^T \mathbf{X}_i)^{-1}$

En notant : $\mathbf{X}_{i+1} = \begin{bmatrix} \mathbf{X}_i \\ \mathbf{x}_{i+1}^T \end{bmatrix}$ où : \mathbf{x}_{i+1}^T désigne la $(i+1)^{ième}$ ligne de la matrice \mathbf{X}_{i+1}

on déduit : $\mathbf{X}_{i+1}^T \mathbf{X}_{i+1} = \begin{bmatrix} \mathbf{X}_i^T & \mathbf{x}_{i+1} \end{bmatrix} \begin{bmatrix} \mathbf{X}_i \\ \mathbf{x}_{i+1}^T \end{bmatrix} = \mathbf{X}_i^T \mathbf{X}_i + \mathbf{x}_{i+1} \mathbf{x}_{i+1}^T$

et donc : $\mathbf{P}_{i+1} = (\mathbf{P}_i^{-1} + \mathbf{x}_{i+1} \mathbf{x}_{i+1}^T)^{-1}$

En appliquant à \mathbf{P}_{i+1} l'identité suivante pour une matrice \mathbf{R} et un vecteur \mathbf{u} : $(\mathbf{R} + \mathbf{u}\mathbf{u}^T)^{-1} = \mathbf{R}^{-1} - \frac{\mathbf{R}^{-1} \mathbf{u}\mathbf{u}^T \mathbf{R}^{-1}}{1 + \mathbf{u}\mathbf{R}^{-1} \mathbf{u}^T}$

on a :
$$\mathbf{P}_{i+1} = \mathbf{P}_i - \frac{\mathbf{P}_i \mathbf{x}_{i+1} \mathbf{x}_{i+1}^T \mathbf{P}_i}{1 + \mathbf{x}_{i+1}^T \mathbf{P}_i \mathbf{x}_{i+1}}$$

En reportant dans l'expression de \mathbf{h}_{i+1} donnée par :
$$\mathbf{h}_{i+1} = \mathbf{P}_{i+1} \begin{bmatrix} \mathbf{X}_i^T & \mathbf{x}_{i+1} \end{bmatrix} \begin{bmatrix} y_i \\ y_{i+1} \end{bmatrix} = \mathbf{P}_{i+1} (\mathbf{X}_i^T y_i + \mathbf{x}_{i+1} y_{i+1})$$

il vient finalement :
$$\mathbf{h}_{i+1} = \mathbf{h}_i + \mathbf{K}_i (y_{i+1} - \mathbf{x}_{i+1}^T \mathbf{h}_i) \quad \text{avec :} \quad \mathbf{K}_i = \frac{\mathbf{P}_i \mathbf{x}_{i+1}}{1 + \mathbf{x}_{i+1}^T \mathbf{P}_i \mathbf{x}_{i+1}} \quad (\text{vecteur dim. } nx1)$$

Dans cette expression, le terme $y_{i+1} - \mathbf{x}_{i+1}^T \mathbf{h}_i$ apparaît comme l'erreur entre la valeur observée y_{i+1} et la valeur $\mathbf{x}_{i+1}^T \mathbf{h}_i$ que l'on aurait dû observer si \mathbf{h}_i était exacte.

L'algorithme des moindres carrés récursifs s'écrit donc :

. Valeurs initiales : $i = 1 \quad \mathbf{P}_i = \frac{\mathbf{I}_{(n+1) \times (n+1)}}{\delta} \quad \text{avec } \mathbf{I}_{(n+1) \times (n+1)} : \text{matrice identité dim. } (n+1) \times (n+1)$
 et $\delta \ll 1 \quad (\delta > 0) \quad (\delta \text{ coefficient réel})$

$$\mathbf{h}_i = \mathbf{0}$$

. Pour $i = 1, \dots, k-1$ Répéter :
$$\mathbf{K}_i = \frac{\mathbf{P}_i \mathbf{x}_{i+1}}{1 + \mathbf{x}_{i+1}^T \mathbf{P}_i \mathbf{x}_{i+1}} \quad \text{avec } \mathbf{x}_i^T = [x_{i+n} \quad x_{i+n-1} \quad \dots \quad x_i]$$

$$\mathbf{h}_{i+1} = \mathbf{h}_i + \mathbf{K}_i (y_{i+1} - \mathbf{x}_{i+1}^T \mathbf{h}_i)$$

$$\mathbf{P}_{i+1} = \mathbf{P}_i - \frac{\mathbf{P}_i \mathbf{x}_{i+1} \mathbf{x}_{i+1}^T \mathbf{P}_i}{1 + \mathbf{x}_{i+1}^T \mathbf{P}_i \mathbf{x}_{i+1}}$$

A condition d'avoir k grand, il est inutile de partir des Conditions Initiales exactes faisant intervenir le calcul de \mathbf{P}_1 et \mathbf{h}_1 à partir de la relation : $\mathbf{h}_i = \mathbf{P}_i \mathbf{X}_i^T \mathbf{y}_i$.

Moindres carrés moyens Algorithme LMS (Least Mean Squares)

L'algorithme des Moindres carrés moyens (algorithme LMS), encore appelé gradient stochastique, n'est autre que la version récursive du filtre de Wiener (voir § Filtrage de Wiener).

Identification par la méthode des moindres carrés

▪ Cas monovariante ($n=1$)

Soient k observations $y_i, i \in [1 \dots k]$, d'un signal dépendant linéairement d'un paramètre a à estimer et de variable connue x_i (exemple le temps : x_i = instants de prises de mesures, pour un signal y_i fonction du temps) : $\mathbf{y} = \mathbf{X}a$

L'estimé de a au sens d'un critère d'écart quadratique (\equiv moindres carrés) est donné par \hat{a} .

- . Si nombre d'observations $k <$ nombre n ($n=1$ si cas monovariante) de paramètres à estimer \rightarrow Solution impossible à déterminer
- . Si nombre d'observations $k =$ nombre n ($n=1$ si cas monovariante) de paramètres à estimer \rightarrow Solution exacte : pas de moindres carrés à appliquer
- . Si nombre d'observations $k \geq$ nombre n ($n=1$ si cas monovariante) de paramètres à estimer \rightarrow Solution minimisant l'erreur quadratique = Moindres carrés.

\hat{a} peut être déterminé directement (moindres carrés direct LS) ou récursivement (moindres carrés récursif RLS) :

Calcul direct - Moindres Carrés Direct - Algorithme LS (Least Squares)

$$\hat{a} = \frac{\sum_{i=1}^k x_i y_i}{\sum_{i=1}^k x_i^2} = \frac{\mathbf{X}^T \cdot \mathbf{y}}{\mathbf{X}^T \cdot \mathbf{X}} = [\mathbf{X}^T \cdot \mathbf{X}]^{-1} \cdot \mathbf{X}^T \cdot \mathbf{y}$$

avec :

$$\mathbf{X} = \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_k \end{bmatrix} \quad \text{vecteur: } (k,1) : \text{ variable}$$

$$\hat{a} = \text{scalaire: } (1,1) : \text{ 1 paramètre à estimer}$$

$$\mathbf{X}^T = [x_1 \quad x_2 \quad \dots \quad x_k] \quad \text{vecteur: } (1,k)$$

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \dots \\ y_k \end{bmatrix} \quad \text{vecteur: } (k,1) : k \text{ observations}$$

$$\mathbf{y} = \mathbf{X} a$$

Le calcul direct, dans le cas multivariable, va impliquer une inversion de matrice.

Calcul récursif Temps Réel - Moindres Carrés Récursif - Algorithme RLS (Recursive Least Squares)

On itère sur les k observations $i \in [1 \dots k]$:

Pour $i = 1$ à k :

$$\begin{cases} K_i = \frac{P_{i-1} \cdot x_i}{1 + P_{i-1} \cdot x_i^2} \\ P_i = P_{i-1} - K_i \cdot P_{i-1} \cdot x_i \\ \hat{a}_i = \hat{a}_{i-1} + K_i \cdot [y_i - \hat{a}_{i-1} x_i] \end{cases} \quad \text{avec : } \begin{cases} \hat{a}_i, K_i, P_i \quad \text{"scalaires"} : (1,1) \\ P_k = \frac{1}{\sum_{i=1}^k x_i^2} = \frac{1}{\mathbf{X}^T \cdot \mathbf{X}} \end{cases} \rightarrow \begin{cases} K_k = P_k \cdot x_k \\ \hat{a}_k = \hat{a}_{k-1} + P_k \cdot x_k \cdot [y_k - \hat{a}_{k-1} x_k] \end{cases}$$

On peut implémenter la récurrence sans les indices car il n'est pas nécessaire de mémoriser les résultats intermédiaires :

Pour $i = 1$ à k :

$$\begin{cases} K = \frac{P \cdot x_i}{1 + P \cdot x_i^2} \\ P = P - K \cdot P \cdot x_i \\ \hat{a} = \hat{a} + K \cdot [y_i - \hat{a} x_i] \end{cases} \quad \text{avec : } \hat{a}, K, P \quad \text{scalaires : } (1,1)$$

Un calcul récursif nécessite d'être initialisé. On peut initialiser principalement de 2 façons :

Initialisation

1) On initialise à partir de la mesure d'indice $i = m$:

On effectue un calcul direct à partir des m premières observations, et ensuite on récurse à partir de $m + 1$ jusqu'à k :

$$\text{Calcul direct : } \hat{a} = \hat{a}_m = P_m \cdot Q \quad \text{avec : } \begin{cases} P_m = \frac{1}{\sum_{i=1}^m x_i^2} = \frac{1}{\mathbf{X}^T \cdot \mathbf{X}} \quad \text{avec } \mathbf{X} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix} \quad \text{et } m < k \\ Q = \sum_{i=1}^m x_i y_i = \mathbf{X}^T \cdot \mathbf{y} \quad \text{avec } \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix} \quad \text{et } m < k \end{cases} \quad Q \text{ scalaire : } (1,1)$$

ou :

2) On initialise à partir de la mesure d'indice $i = m = 0$:

On choisit une valeur de P_m suffisamment grande \rightarrow à partir d'un certain nombre d'observations, l'estimation est indépendante de la Condition Initiale \hat{a}_m que l'on peut choisir arbitrairement (en particulier $\hat{a}_m = 0$).

(P_m est une mesure de l'erreur de l'estimation effectuée avec les m premières observations).

(Une valeur faible de P_i indique que \hat{a}_{i-1} converge vers \hat{a}_i).

Avec cette initialisation, il faut suffisamment d'observations pour pouvoir converger avec précision vers la solution exacte \hat{a}_k fournie par le calcul direct.

Le calcul récursif, par nature, implique une propagation d'erreur de troncature.

▪ Cas multivariable

Soient k observations $y_i, i \in [1 \dots k]$, d'un signal dépendant linéairement de n paramètres \mathbf{a} à estimer, avec $k \geq n$ (système surdéterminé) ($\mathbf{a}^T = [a_1 \ a_2 \ \dots \ a_n]$) et de variables connues \mathbf{x}_i :

$$\mathbf{y} = \mathbf{X}\mathbf{a} \quad \text{avec} \quad \mathbf{X} = [\mathbf{x}_i^T] \quad \forall i \in [1 \dots k].$$

L'estimé de \mathbf{a} au sens d'un critère d'écart quadratique (\equiv moindres carrés) est donné par $\hat{\mathbf{a}}$.

- . Si nombre d'observations $k <$ nombre n ($n=1$ si cas monovarié) de paramètres à estimer \rightarrow Solution impossible à déterminer
- . Si nombre d'observations $k =$ nombre n ($n=1$ si cas monovarié) de paramètres à estimer \rightarrow Solution exacte : pas de moindres carrés à appliquer
- . Si nombre d'observations $k \geq$ nombre n ($n=1$ si cas monovarié) de paramètres à estimer \rightarrow Solution minimisant l'erreur quadratique = Moindres carrés.

$\hat{\mathbf{a}}$ peut être déterminé directement (moindres carrés direct) ou récursivement (moindres carrés récursif) :

Calcul direct - Moindres Carrés Direct - Algorithme LS (Least Squares)

$$\hat{\mathbf{a}} = [\mathbf{X}^T \cdot \mathbf{X}]^{-1} \cdot \mathbf{X}^T \cdot \mathbf{y} \quad \text{avec :} \quad \begin{cases} \mathbf{x}_i^T = [x_{i1} \ x_{i2} \ \dots \ x_{in}] \quad \text{vecteur: } (1, n) : \text{ variables } (i = \text{une valeur dans l'intervalle } [1, k]) \\ \mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ \vdots & \vdots & \dots & \vdots \\ x_{i1} & x_{i2} & \dots & x_{in} \\ \vdots & \vdots & \dots & \vdots \\ x_{k1} & x_{k2} & \dots & x_{kn} \end{bmatrix} = [\mathbf{x}_i^T] \quad \forall i \in [1, k] \quad \text{matrice: } (k, n) : \text{ variables} \\ \hat{\mathbf{a}}^T = [\hat{a}_1 \ \hat{a}_2 \ \dots \ \hat{a}_n] \quad \hat{\mathbf{a}} : \text{vecteur: } (n, 1) : n \text{ paramètres à estimer} \\ \mathbf{y}^T = [y_1 \ y_2 \ \dots \ y_k] \quad \mathbf{y} : \text{vecteur: } (k, 1) : k \text{ observations} \\ \mathbf{y} = \mathbf{X}\mathbf{a} \end{cases}$$

Le calcul direct, dans le cas multivariable, implique une inversion de matrice.

Calcul récursif Temps Réel - Moindres Carrés Récursif - Algorithme RLS (Recursive Least Squares)

On itère sur les k observations $i \in [1 \dots k]$:

Pour $i = 1$ à k :

$$\begin{cases} \mathbf{K}_i = \mathbf{P}_{i-1} \cdot \mathbf{x}_i \cdot (1 + \mathbf{x}_i^T \cdot \mathbf{P}_{i-1} \cdot \mathbf{x}_i)^{-1} \\ \mathbf{P}_i = \mathbf{P}_{i-1} - \mathbf{K}_i \cdot \mathbf{x}_i^T \cdot \mathbf{P}_{i-1} \\ \hat{\mathbf{a}}_i = \hat{\mathbf{a}}_{i-1} + \mathbf{K}_i \cdot [y_i - \mathbf{x}_i^T \cdot \hat{\mathbf{a}}_{i-1}] \end{cases} \quad \text{avec :} \quad \begin{cases} (1 + \mathbf{x}_i^T \cdot \mathbf{P}_{i-1} \cdot \mathbf{x}_i) : (1, 1) \quad \text{scalaire} \quad (\text{pas d'inversion de matrice}) \\ \mathbf{P}_i : (n, n) \\ \mathbf{x}_i : (n, 1) \quad \mathbf{x}_i^T = [x_{i1} \ x_{i2} \ \dots \ x_{in}] : n \text{ variables} \\ \mathbf{K}_i : (n, 1) \\ y_i : (1, 1) \quad \text{"scalaire"} : k \text{ observations} \quad 1 \leq i \leq k \quad k : \text{nombre d'observations} \\ \hat{\mathbf{a}}_i : (n, 1) \quad \hat{\mathbf{a}}_i^T = [\hat{a}_1 \ \hat{a}_2 \ \dots \ \hat{a}_n] \quad \text{"vecteur } (n, 1)": n \text{ paramètres à estimer} \end{cases}$$

On peut implémenter la récurrence sans les indices car il n'est pas nécessaire de mémoriser les résultats intermédiaires :

Pour $i = 1$ à k :

$$\begin{aligned} \mathbf{K} &= \mathbf{P} \cdot \mathbf{x}_i \cdot (1 + \mathbf{x}_i^T \cdot \mathbf{P} \cdot \mathbf{x}_i)^{-1} \\ \mathbf{P} &= \mathbf{P} - \mathbf{K} \cdot \mathbf{x}_i^T \cdot \mathbf{P} \\ \hat{\mathbf{a}} &= \hat{\mathbf{a}} + \mathbf{K} \cdot [y_i - \mathbf{x}_i^T \cdot \hat{\mathbf{a}}] \end{aligned}$$

Un calcul récursif nécessite d'être initialisé. On peut initialiser principalement de 2 façons :

Initialisation

1) On initialise à partir de la mesure d'indice $i = m$:

On effectue un calcul direct à partir des m premières observations :

on obtient $\hat{\mathbf{a}}_m$ ($n,1$), \mathbf{P}_m (n,n) et \mathbf{Q} ($n,1$), et ensuite on récurse à partir de $m + 1$ jusqu'à k :

Calcul direct : $\hat{\mathbf{a}} = \hat{\mathbf{a}}_m = \mathbf{P}_m \cdot \mathbf{Q}$ avec :

$$\mathbf{P}_m = [\mathbf{X}^T \cdot \mathbf{X}]^{-1} \quad \text{avec} \quad \mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ \vdots & \vdots & \dots & \vdots \\ x_{i1} & x_{i2} & \dots & x_{in} \\ \vdots & \vdots & \dots & \vdots \\ x_{m1} & x_{m2} & \dots & x_{mn} \end{bmatrix}$$

$$\mathbf{Q} = \mathbf{X}^T \cdot \mathbf{y} \quad \text{avec} \quad \mathbf{y}^T = [y_1 \quad y_2 \quad \dots \quad y_m]$$

ou :

2) On initialise à partir de la mesure d'indice $i = m = 0$:

On choisit la matrice \mathbf{P}_m avec des valeurs grandes sur la diagonale et 0 sinon (\mathbf{P}_m prise diagonale).

A partir d'un certain nombre d'observations, l'estimation est indépendante de la Condition Initiale $\hat{\mathbf{a}}_m$ que l'on peut choisir arbitrairement (en particulier $\hat{\mathbf{a}}_m = \mathbf{0}$).

(\mathbf{P}_m est une mesure de l'erreur de l'estimation effectuée avec les m premières observations).

(Une valeur faible de \mathbf{P}_i indique que $\hat{\mathbf{a}}_{i-1}$ converge vers $\hat{\mathbf{a}}_i$).

Avec cette initialisation, il faut suffisamment d'observations pour pouvoir converger avec précision vers la solution exacte $\hat{\mathbf{a}}_k$ fournie par le calcul direct.

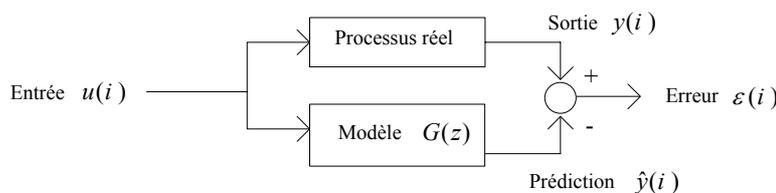
Le calcul récursif, par nature, implique une propagation d'erreur de troncature.

Application : Identification de processus

L'identification par la méthode des moindres carrés permet d'identifier les échantillons de sortie $y(i)$ du système considéré, comme issus d'un processus ARMA (modèle général).

L'intérêt est évidemment de pouvoir ainsi disposer de la FT du processus considéré.

(s'il n'y a pas d'entrée $u(i)$ (ou si elle est non ou mal connue), on se limite à un modèle AR purement récursif) :



Modèle : Processus ARMA :

$$G(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \dots + b_p z^{-p}}{1 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_q z^{-q}}$$

Les paramètres à estimer sont : $a_1, a_2, \dots, a_q, b_0, b_1, \dots, b_p$ ($n = p + q + 1$ paramètres à estimer).

On suppose que la séquence d'entrée $u(i)$ appliquée excite tous les modes du processus.

- séquence d'entrée : $u(0), u(1) \dots u(k-1)$: k observations

- séquence observée : $y(0), y(1) \dots y(k-1)$

Les séquences d'entrée et de sortie sont observées sur $i \in [0 \dots k-1]$. On va appliquer le critère quadratique de moindres carrés pour minimiser l'énergie de l'erreur.

$$\text{On a : } G(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \dots + b_p z^{-p}}{1 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_q z^{-q}} = \frac{\hat{Y}(z)}{U(z)} \quad \text{d'où, par } TZ^{-1} :$$

$$\hat{y}(i) = -a_1 \hat{y}(i-1) - a_2 \hat{y}(i-2) - \dots - a_q \hat{y}(i-q) + b_0 u(i) + b_1 u(i-1) + \dots + b_p u(i-p) \quad (0)$$

$\hat{y}(i)$ est l'estimée de $y(i)$, ou encore sa prédiction, son modèle calculé à partir des observations précédentes.

Pour calculer $\hat{y}(i)$ il faut bien démarrer la récurrence et utiliser les observations. L'équation (0) devient ainsi :

$$\hat{y}(i) = -a_1 y(i-1) - a_2 y(i-2) - \dots - a_q y(i-q) + b_0 u(i) + b_1 u(i-1) + \dots + b_p u(i-p)$$

et l'erreur $\varepsilon(i)$ s'écrit :

$$\varepsilon(i) = y(i) - \hat{y}(i) = y(i) + a_1 y(i-1) + a_2 y(i-2) + \dots + a_q y(i-q) - b_0 u(i) - b_1 u(i-1) - \dots - b_p u(i-p)$$

d'où on tire l'observation :

$$y(i) = -a_1 y(i-1) - a_2 y(i-2) - \dots - a_q y(i-q) + b_0 u(i) + b_1 u(i-1) + \dots + b_p u(i-p) + \varepsilon(i) \quad (1)$$

Les séquences d'entrée et de sortie étant observées sur $i \in [0 \dots k-1]$, on a le système d'équations :

$$\begin{cases} y(0) = b_0 u(0) + \varepsilon(0) \\ y(1) = b_0 u(1) + b_1 u(0) - a_1 y(0) + \varepsilon(1) \\ y(2) = b_0 u(2) + b_1 u(1) + b_2 u(0) - \{a_1 y(1) + a_2 y(0)\} + \varepsilon(2) \\ y(3) = b_0 u(3) + b_1 u(2) + b_2 u(1) + b_3 u(0) - \{a_1 y(2) + a_2 y(1) + a_3 y(0)\} + \varepsilon(3) \\ \vdots \\ y(k-1) = b_0 u(k-1) + b_1 u(k-2) + \dots + b_p u(k-1-p) - \{a_1 y(k-2) + a_2 y(k-3) + \dots + a_q y(k-1-q)\} + \varepsilon(k-1) \end{cases}$$

D'où la relation matricielle :

$$\begin{bmatrix} y(0) \\ y(1) \\ \vdots \\ y(k-1) \end{bmatrix} = \underbrace{\begin{bmatrix} u(0) & 0 & \dots & 0 & \vdots & 0 & \dots & \dots & 0 \\ u(1) & u(0) & 0 & 0 & \vdots & -y(0) & 0 & \dots & 0 \\ u(2) & u(1) & u(0) & 0 & \vdots & -y(1) & -y(0) & 0 & 0 \\ \vdots & \vdots & & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ u(k-1) & u(k-2) & \dots & u(k-1-p) & \vdots & -y(k-2) & -y(k-3) & \dots & -y(k-1-q) \end{bmatrix}}_{(k \times n)} \cdot \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_p \\ \dots \\ a_1 \\ a_2 \\ \vdots \\ a_q \end{bmatrix}_{(n \times 1)} + \begin{bmatrix} \varepsilon(0) \\ \varepsilon(1) \\ \vdots \\ \varepsilon(k-1) \end{bmatrix}_{(k \times 1)} \quad (2)$$

(kx1) (kxn) (nx1) (kx1)

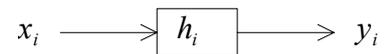
soit :

$$\mathbf{y} = \mathbf{X}\mathbf{a} + \underline{\varepsilon} \quad (3) \quad \text{avec} \quad \mathbf{X} = \begin{bmatrix} u(0) & 0 & \dots & 0 & \vdots & 0 & \dots & \dots & 0 \\ u(1) & u(0) & 0 & 0 & \vdots & -y(0) & 0 & \dots & 0 \\ u(2) & u(1) & u(0) & 0 & \vdots & -y(1) & -y(0) & 0 & 0 \\ \vdots & \vdots \\ u(k-1) & u(k-2) & \dots & u(k-1-p) & \vdots & -y(k-2) & -y(k-3) & \dots & -y(k-1-q) \end{bmatrix}$$

Soit le critère quadratique : $J = \frac{1}{2} \sum_{i=0}^{k-1} \varepsilon^2(i) = \frac{1}{2} \boldsymbol{\varepsilon}^T \cdot \boldsymbol{\varepsilon}$

La solution qui minimise J conduit à la relation des moindres carrés : (calcul direct) : $\hat{\mathbf{a}} = [\mathbf{X}^T \cdot \mathbf{X}]^{-1} \cdot \mathbf{X}^T \cdot \mathbf{y}$

Application : Identification d'un canal de communication



On modélise un canal de communication par un filtre RIF de RI : $\{h_0, h_1, \dots, h_{n-1}\}$

Le problème de l'identification de ce canal est d'estimer la RI $\{h_i\}$ (n inconnues)

La méthode la plus simple consiste à envoyer une séquence connue $\{x_i\}$ de k symboles ($i = 0, \dots, k-1$) ($k \geq n$), dite séquence d'apprentissage, et d'observer la sortie $\{y_i\}$ du canal.

En pratique, comme le canal n'est généralement pas stationnaire (il varie dans le temps), on est conduit à refaire régulièrement cette opération en faisant précéder chaque trame de symboles par une séquence d'apprentissage.

L'observation en sortie du canal s'écrit : (la séquence d'apprentissage et la RI du filtre sont supposées causales)

$$y_i = h_i * x_i + w_i = \sum_{j=0}^{n-1} h_j x_{i-j} + w_i = h_0 x_i + h_1 x_{i-1} + \dots + h_{n-1} x_{i-n+1} + w_i$$

où : $\{w_i\}$ représente un bruit additif qui vient se superposer au signal transmis.

En écrivant la suite des k observations sous forme matricielle, pour i allant de n à $n+k-1$, on a :

$$\begin{bmatrix} y_n \\ \vdots \\ y_{n+k-1} \end{bmatrix} = \begin{bmatrix} x_{n-1} & \dots & x_1 & x_0 \\ x_n & \dots & x_2 & x_1 \\ \vdots & \vdots & \vdots & \vdots \\ x_{n+k-2} & \dots & x_k & x_{k-1} \end{bmatrix} \begin{bmatrix} h_0 \\ \vdots \\ h_{n-1} \end{bmatrix} + \begin{bmatrix} w_n \\ \vdots \\ w_{n+k-1} \end{bmatrix}$$

Le vecteur \mathbf{h} qui minimise l'écart quadratique $\varepsilon^2 = J(\mathbf{h}) = (\mathbf{y} - \mathbf{X}\mathbf{h})^T (\mathbf{y} - \mathbf{X}\mathbf{h})$ est donné par :

$$\mathbf{h} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y} \quad (\text{moindres carrés direct})$$

On peut se demander s'il existe des séquences d'apprentissage meilleures que d'autres.

La réponse est oui.

Ainsi, si on considère que la séquence $\{x_i\}$ est une réalisation d'un processus aléatoire stationnaire ergodique, on peut montrer que les performances sont optimales lorsque ce processus est **blanc** (ce résultat était prévisible, car il y a tout dans un bruit blanc, ce dernier excite donc tous les modes, toutes les fréquences du canal).

On s'efforcera donc à émettre des séquences d'apprentissage qui ressemblent autant que possible à du bruit blanc.

7 ANNEXE. Filtrage optimal

II. FILTRAGE DE WIENER

Considérons un système S dont l'entrée x_n et la sortie y_n sont 2 processus aléatoires, réels, centrés, stationnaires au 2nd ordre.

On suppose connues la fonction d'autocovariance R_{XX_k} de x_n et la fonction de covariance $E[Y_{n+k} X_n]$

(les signaux x_n et y_n sont une réalisation des VA X_n et Y_n).

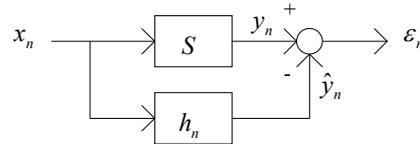
On suppose que y_n et x_n sont de covariance stationnaire, ce qui signifie que $E[Y_{n+k} X_n]$ ne dépend que de k et peut s'écrire R_{YX_k} .

On se propose de déterminer le filtre linéaire de RI Finie (MA) (N points) : $\{h_n\} = \{h_0, h_1, \dots, h_{N-1}\}$ qui soit la meilleure approximation du système S .

On pose :
$$\hat{y}_n = h_n * x_n = \sum_{k=0}^{N-1} h_k x_{n-k} = h_0 x_n + h_1 x_{n-1} + \dots + h_{N-1} x_{n-N+1}$$

et :
$$\varepsilon_n = y_n - \hat{y}_n$$

On cherche la séquence $\{h_n\}$ qui minimise l'énergie (\equiv moyenne statistique de la puissance) de l'erreur ε_n (critère quadratique) : $E[\varepsilon_n^2]$



D'après le *théorème de projection* (vu plus haut, avec les moindres carrés), les coefficients qui minimisent l'écart quadratique entre y_n et \hat{y}_n sont tels que l'erreur ε_n ($= y_n - \hat{y}_n$) est orthogonale à tout élément de l'espace de projection. Or cet espace est engendré par les variables aléatoires x_{n-k} où $k \in \{0, 1, \dots, N-1\}$.

Par conséquent, au sens de l'orthogonalité dans l'espace de Hilbert des VAs de carré sommable, on a, pour tout $k \in \{0, 1, \dots, N-1\}$: $E[(y_n - \hat{y}_n)x_{n-k}] = E[(y_n - h_0 x_n - h_1 x_{n-1} - \dots - h_{N-1} x_{n-N+1})x_{n-k}] = 0$

soit : $\forall k \in \{0, 1, \dots, N-1\}$: $E[(h_0 x_n + h_1 x_{n-1} + \dots + h_{N-1} x_{n-N+1})x_{n-k}] = E[y_n x_{n-k}]$

Cette relation peut s'écrire sous forme matricielle (équation de **Wiener-Hopf**) : $\mathbf{R} \mathbf{h} = \mathbf{r}$

en posant :

$$\mathbf{x}_n = [x_n \quad \dots \quad x_{n-N+1}]^T$$

$$\mathbf{h} = [h_0 \quad \dots \quad h_{N-1}]^T$$

et où par définition :

$$\mathbf{R} = E[\mathbf{x}_n \mathbf{x}_n^T]$$

$$\mathbf{r} = E[y_n \mathbf{x}_n]$$

\mathbf{R} et \mathbf{r} s'écrivent à partir des fonctions de covariance R_{XX_k} et R_{YX_k} :

$$\mathbf{R} = \begin{bmatrix} R_{XX_0} & \dots & R_{XX_{N-1}} \\ \vdots & \dots & \vdots \\ R_{XX_{N-1}} & \dots & R_{XX_0} \end{bmatrix} \quad \mathbf{r} = \begin{bmatrix} R_{YX_0} \\ \vdots \\ R_{YX_{N-1}} \end{bmatrix}$$

Le filtre de **Wiener**, solution de l'équation de **Wiener-Hopf**, s'obtient aisément :

$$\mathbf{h} = \mathbf{R}^{-1} \mathbf{r}$$

Evidemment, dans le cas où le système S est un filtre linéaire RIF de RI g_n , de durée finie N , h_n est exactement égale à g_n et le minimum de $E[\varepsilon_n^2]$ est égal à 0.

(On peut le montrer en rappelant que, pour un filtre linéaire, la fonction de covariance R_{yX_k} entre l'entrée et la sortie est donnée par

$$R_{yX_k} = h_k * R_{XX_k}$$

La solution trouvée : $\mathbf{h} = \mathbf{R}^{-1} \mathbf{r}$ nécessite une inversion matricielle. On peut toutefois l'éviter par mise en oeuvre de l'*algorithme du gradient*.

Algorithme du gradient (version récurrente du filtre de Wiener)

Considérons l'équation récurrente : $\mathbf{h}(n) = \mathbf{h}(n-1) + \mu[\mathbf{r} - \mathbf{R}\mathbf{h}(n-1)]$

où : $\mathbf{h}(n)$ désigne le vecteur obtenu au pas d'itération n . Le coefficient réel μ est appelé le *pas du gradient*.

A condition de choisir : $0 < \mu < \frac{2}{\max_i(\lambda_i)}$ (λ_i désignent les valeurs propres de \mathbf{R})

on montre que $\mathbf{h}(n)$ converge vers le filtre de Wiener.

(En effet la solution de l'équation récurrente : $\mathbf{h}(n) = \mathbf{h}(n-1) + \mu[\mathbf{r} - \mathbf{R}\mathbf{h}(n-1)]$ est donnée par : $\mathbf{h}(n) = (\mathbf{I} - \mathbf{A}^n)\mathbf{R}^{-1}\mathbf{r} + \mathbf{A}^n\mathbf{h}(0)$ où on a posé : $\mathbf{A} = (\mathbf{I} - \mu\mathbf{R})$ et utilisé l'identité : $(\mathbf{I} + \mathbf{A} + \dots + \mathbf{A}^{n-1})(\mathbf{I} - \mathbf{A}) = \mathbf{I} - \mathbf{A}^n$)

Moindres carrés moyens **Algorithme LMS** (Least Mean Squares)

Si on connaît \mathbf{R} et \mathbf{r} , l'algorithme du gradient fournit une solution au problème posé.

Malheureusement, \mathbf{R} et \mathbf{r} sont généralement inconnus et doivent être estimés à partir des données observées.

L'algorithme des *moindres carrés moyens* (algorithme gradient stochastique ou encore LMS - Least Mean Squares), algorithme adaptatif imaginé par **Widrow** (1985), évite le calcul de ces estimations.

Posons : $\mathbf{x}_n = [x_n \ \dots \ x_{n-N+1}]^T$ et : $e(n) = y_n - \mathbf{x}_n^T \mathbf{h}(n-1)$

On peut écrire : $\mathbf{r} - \mathbf{R}\mathbf{h}(n-1) = E[y_n \mathbf{x}_n] - E[\mathbf{x}_n \mathbf{x}_n^T] \mathbf{h}(n-1) = E[\mathbf{x}_n e(n)]$

Notons que $e(n)$ représente la différence entre la valeur y_n à l'instant n en sortie du système S et la valeur calculée avec les coefficients obtenus à l'instant $n-1$.

Si $e(n)$ est nulle, on peut considérer que les coefficients ont la bonne valeur, sinon il faut les corriger. L'erreur $e(n)$ est différente de l'erreur $\varepsilon_n = y_n - \hat{y}_n$ définie plus haut et qui fait intervenir la limite cherchée.

En conséquence, l'équation récurrente a pour expression : $\mathbf{h}(n) = \mathbf{h}(n-1) + \mu E[\mathbf{x}_n e(n)]$

Une idée, dont le mérite revient à *Widrow*, est de remplacer dans l'équation récurrente, l'estimée de $E[\mathbf{x}_n e(n)]$ par sa valeur « instantanée » $\mathbf{x}_n e(n)$. Et ça marche ! A condition d'accepter une oscillation résiduelle autour de la solution.

On obtient 2 équations qui constituent l'*algorithme du gradient stochastique* ou *algorithme LMS* :

Algorithme LMS

. Valeur initiale : $\mathbf{h}(0) = \mathbf{0}$

. Répéter :

$$e(n) = y_n - \mathbf{x}_n^T \mathbf{h}(n-1)$$

$$\mathbf{h}(n) = \mathbf{h}(n-1) + \mu \mathbf{x}_n e(n)$$

III. FILTRAGE DE KALMAN

L'approche de Kalman permet de s'affranchir de l'hypothèse de stationnarité des signaux. Elle met à profit la **Représentation d'Etat**.

Exemple :

Soit un véhicule se déplaçant sur une droite avec une vitesse v constante.

Partant de la position initiale d_0 , sa position à l'instant t est donnée par : $d(t) = d_0 + vt$

Cette équation de mouvement peut encore s'écrire (t est **discret**) :

$$\begin{cases} d(t+1) = d(t) + v(t) \\ v(t+1) = v(t) \quad (= v) \end{cases}$$

avec comme CIs (Conditions Initiales) :

$$\begin{cases} d(0) = d_0 \\ v(0) = v \end{cases}$$

En fait, on n'a pas vraiment confiance dans l'hypothèse selon laquelle la vitesse est constante (le modèle d'évolution de la vitesse dont on dispose est imprécis). Pour prendre en compte d'éventuelles variations, on suppose que :

$$\begin{cases} d(t+1) = d(t) + v(t) + b_1(t) \\ v(t+1) = v(t) + b_2(t) \end{cases}$$

où : $b_1(t)$ et $b_2(t)$ sont 2 processus aléatoires figurant un *bruit de modèle*.

L'équation d'évolution a donc pour expression matricielle :

$$\begin{bmatrix} d(t+1) \\ v(t+1) \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} d(t) \\ v(t) \end{bmatrix} + \begin{bmatrix} b_1(t) \\ b_2(t) \end{bmatrix}$$

Supposons à présent que la position $d(t)$ soit observée à la sortie d'un dispositif bruité qui délivre la valeur :

$$y(t) = d(t) + u(t)$$

où : $u(t)$ est un processus aléatoire figurant un *bruit d'observation*, un *bruit de mesure*.

Le problème se ramène donc alors au système d'équations :

$$\begin{cases} \begin{bmatrix} d(t+1) \\ v(t+1) \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} d(t) \\ v(t) \end{bmatrix} + \begin{bmatrix} b_1(t) \\ b_2(t) \end{bmatrix} \\ y(t) = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} d(t) \\ v(t) \end{bmatrix} + u(t) \end{cases}$$

En passant à une écriture matricielle, on a :

$$\begin{cases} \mathbf{x}(t+1) = \mathbf{A} \mathbf{x}(t) + \mathbf{b}(t) & \text{équation d'Etat} \\ \mathbf{y}(t) = \mathbf{C} \mathbf{x}(t) + \mathbf{u}(t) & \text{équation d'observation} \end{cases}$$

en posant :

- . Etat ($n \times 1$): $\mathbf{x}(t) = \begin{bmatrix} d(t) \\ v(t) \end{bmatrix}$
- . Observation vectorielle ($p \times 1$) : $\mathbf{y}(t)$
- . $\mathbf{A} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$ ($n \times n$)
- . $\mathbf{b}(t) = \begin{bmatrix} b_1(t) \\ b_2(t) \end{bmatrix}$ ($n \times 1$)
- . $\mathbf{C} = \begin{bmatrix} 1 & 0 \end{bmatrix}$ ($p \times n$)
- . $\mathbf{u}(t) = u(t)$ ($p \times 1$)

Le filtre de Kalman fournit la meilleure estimation en *moyenne quadratique* de $\mathbf{x}(t)$ à partir de l'observation $\mathbf{y}(t)$.

Dans sa forme la plus simple, on fait les hypothèses suivantes (bruits de modèle et d'observation les pires possibles : bruits blancs) :

. les composantes de $\mathbf{b}(t)$ et $\mathbf{u}(t)$ sont des processus aléatoires *centrés, blancs et non corrélés entre eux*.

Leurs matrices de covariance respectives ont pour expression : (t et τ sont *discrets*)

$$\mathbf{R}_b(\tau) = E[\mathbf{b}(t+\tau)\mathbf{b}^T(t)] = \begin{bmatrix} \sigma_{b_1}^2 & 0 & \dots & 0 \\ 0 & \sigma_{b_2}^2 & 0 & \vdots \\ \vdots & \vdots & \vdots & 0 \\ 0 & \dots & 0 & \sigma_{b_n}^2 \end{bmatrix} \delta(\tau)$$

$$\mathbf{R}_u(\tau) = E[\mathbf{u}(t+\tau)\mathbf{u}^T(t)] = \begin{bmatrix} \sigma_{u_1}^2 & 0 & \dots & 0 \\ 0 & \sigma_{u_2}^2 & 0 & \vdots \\ \vdots & \vdots & \vdots & 0 \\ 0 & \dots & 0 & \sigma_{u_p}^2 \end{bmatrix} \delta(\tau)$$

et : $E[\mathbf{u}(t+\tau)\mathbf{b}^T(t)] = 0$

Le filtre de Kalman fournit à chaque instant t l'estimation $\hat{\mathbf{x}}(t)$, fonction *linéaire* de $\mathbf{y}(1), \mathbf{y}(2), \dots, \mathbf{y}(t)$ qui minimise l'erreur quadratique $E[\mathbf{x}(t) - \hat{\mathbf{x}}(t)]^2$.

Dans le cas gaussien, cette estimation linéaire est la meilleure parmi toutes les fonctions de $\mathbf{y}(1), \mathbf{y}(2), \dots, \mathbf{y}(t)$.

Filtre de Kalman récursif

Le calcul direct fait intervenir l'inverse d'une matrice de covariance de dimension t . Par conséquent, quand t croît, la dimension de cette matrice croît de façon « explosive ».

L'avantage du filtre de Kalman est d'être un algorithme récursif.

Il effectue le calcul de $\hat{\mathbf{x}}(t)$ à l'instant t par une mise à jour de la valeur $\hat{\mathbf{x}}(t-1)$ obtenue à l'instant $t-1$ en tenant compte de la dernière valeur observée $\mathbf{y}(t)$.

Algorithme du filtre récursif de Kalman

. Valeurs initiales : $\hat{\mathbf{x}}(0) = E[\mathbf{x}(0)] \qquad \mathbf{K}(0) = E[\mathbf{x}(0)\mathbf{x}^T(0)]$

. Répéter pour $t \geq 1$:

$$\mathbf{G}(t) = \mathbf{K}(t-1)\mathbf{C}^T [\mathbf{C}\mathbf{K}(t-1)\mathbf{C}^T + \mathbf{R}_u(0)]^{-1} \tag{1}$$

$$\hat{\mathbf{x}}(t) = \mathbf{A}\hat{\mathbf{x}}(t-1) + \mathbf{G}(t)[\mathbf{y}(t) - \mathbf{C}\mathbf{A}\hat{\mathbf{x}}(t-1)] \tag{2}$$

$$\mathbf{K}(t) = \mathbf{A}[\mathbf{K}(t-1) - \mathbf{G}(t)\mathbf{C}\mathbf{K}(t-1)\mathbf{A}^T + \mathbf{R}_b(0)] \tag{3}$$

L'expression de cet algorithme appelle quelques remarques :

. $\mathbf{G}(t)$ s'appelle le *gain de Kalman*. Il peut être calculé à l'avance. En effet les équations qui le déterminent ((1) et (3)) ne dépendent pas des données observées. Toutefois, à part dans le cas scalaire (état de dimension 1), la détermination de $\mathbf{G}(t)$ requiert la résolution d'une équation récurrente complexe.

. La signification de l'équation (2) est claire. D'après l'équation d'Etat, s'il n'y avait pas de bruit, la valeur de $\mathbf{x}(t)$ à l'instant t serait $\mathbf{A}\mathbf{x}(t-1)$. C'est le 1er terme de l'équation (2). Mais à cause du bruit, il faut corriger cette valeur par une quantité proportionnelle à l'écart entre la valeur observée $\mathbf{y}(t)$ et la valeur que l'on aurait dû avoir si l'équation d'observation n'était pas bruitée, à savoir $\mathbf{C}\mathbf{x}(t) = \mathbf{C}\mathbf{A}\mathbf{x}(t-1)$.

. Le filtre de Kalman nécessite la connaissance de $\mathbf{R}_b(0)$ et $\mathbf{R}_u(0)$. Cependant la théorie peut être généralisée et inclure la situation où il faut estimer ces quantités à partir des données observées. Dans ce cas la suite des gains n'est plus pré-calculable.

. En résumé, le filtre de Kalman conduit à l'erreur quadratique minimale. Il nécessite une description détaillée du modèle de signal sans toutefois imposer de contraintes de stationnarité. Lorsque l'on s'écarte de ce modèle, les résultats obtenus peuvent être très mauvais. Le filtre de Kalman est linéaire et récursif. Le nombre d'opérations par point (≡ échantillon) est important, conduisant à un temps de calcul coûteux qui a parfois limité sa mise en oeuvre.

TD 7. Filtrage optimal

I. MOINDRES CARRES

Identification par la méthode des moindres carrés

1. Résolution d'un système d'équations surdéterminé au sens d'un critère d'écart quadratique [Voir TP]

Considérons le système d'équations :

$$\begin{cases} 2\alpha + 3\beta = 6 \\ 3\alpha + 4\beta = 7 \\ \alpha + \beta = 2 \end{cases}$$

Déterminer les paramètres $\hat{\mathbf{a}} = \begin{bmatrix} \hat{\alpha} \\ \hat{\beta} \end{bmatrix}$ par la méthode des moindres carrés direct (LS).

2. Approximation polynomiale [Voir TP]

On cherche à approcher une série de k mesures (température de 10 jours du mois dernier) par un polynôme d'ordre $n-1$.



z_i (abscisse)	1	2	4	7	9	12	13	16	17	19
y_i (ordonnée)	3	5	10	6	9	11	15	18	21	19

0. On choisit pour commencer un polynôme d'ordre 0 (droite moyenne, $n=1$) pour approcher la série de mesures.

Rechercher le paramètre \hat{a} (ordonnée à l'origine de la droite moyenne) :
 - par LS
 - par RLS

- A l'aide de ce modèle, prédire la température du 20^{ème} jour, du 30^{ème} jour.

1. On choisit maintenant pour modèle une droite passant par l'origine ($n=1$) pour identifier la série de mesures.

Rechercher le paramètre \hat{a} (pente de la droite passant par l'origine) :
 - par LS
 - par RLS

- A l'aide de ce modèle, prédire la température du 20^{ème} jour, du 30^{ème} jour.

Ibis. On choisit ici pour modèle un polynôme du 1^{er} ordre (droite de régression linéaire, $n=2$) pour l'identification.

Rechercher le paramètre $\hat{\mathbf{a}} = \begin{bmatrix} \hat{a}_0 \\ \hat{a}_1 \end{bmatrix}$ (\hat{a}_0 ordonnée à l'origine et \hat{a}_1 pente de la droite) :
 - par LS

- par RLS

- A l'aide de ce modèle, prédire la température du 20^{ème} jour, du 30^{ème} jour.

2. On choisit ici pour modèle un polynôme du 2nd degré ($n=3$) pour l'identification.

Rechercher les paramètres $\hat{\mathbf{a}} = \begin{bmatrix} \hat{a}_0 \\ \hat{a}_1 \\ \hat{a}_2 \end{bmatrix}$ de la courbe qui « passe » par l'ensemble des points (z_i, y_i) :
 - par LS

- par RLS

- A l'aide de ce modèle, prédire la température du 20^{ème} jour, du 30^{ème} jour.

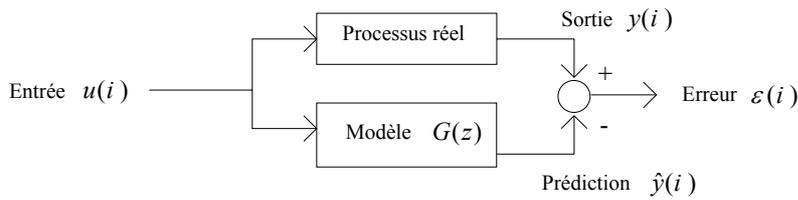
3. On choisit ici pour modèle un polynôme d'ordre n pour l'identification.

Rechercher les paramètres $\hat{\mathbf{a}}^T = [\hat{a}_0 \quad \hat{a}_1 \quad \dots \quad \hat{a}_{n-1}]$ du polynôme :
 - par LS
 - par RLS

- A l'aide de ce modèle, prédire la température du 20^{ème} jour, du 30^{ème} jour.

3. Identification de processus [Voir TP]

Considérons un processus échantillonné et son modèle :



1. On prend comme modèle : $G(z) = b_0 + b_1z^{-1}$ Filtre MA d'ordre $n = 2$

On a donc : $\hat{y}(i) = b_0u(i) + b_1u(i-1)$ et $\varepsilon(i) = y(i) - \hat{y}(i) = y(i) - b_0u(i) - b_1u(i-1)$

On enregistre les k données suivantes :

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
$u(i)$	1.0	0.8	0.6	0.4	0.2	0.0	0.2	0.4	0.6	0.8	1.0	0.8	0.6	0.4	0.2
$y(i)$	0.9	2.5	2.4	1.3	1.2	0.8	0.0	0.9	1.4	1.9	2.3	2.4	2.3	1.3	1.2

- En utilisant la méthode des moindres carrés, déterminer les paramètres optimaux du modèle \hat{b}_0 et \hat{b}_1 .
- Tracer les réponses $y(i)$ et $\hat{y}(i)$ ainsi que la commande $u(i)$ sur un même graphe.

2. Mêmes questions en prenant un modèle plus général : $G(z) = \frac{b_0 + b_1z^{-1} + \dots + b_pz^{-p}}{1 + a_1z^{-1} + \dots + a_qz^{-q}}$ Filtre ARMA d'ordre $n = p + q + 1$

TD 7 Annexe. Filtrage optimal

I. MOINDRES CARRÉS - II. FILTRAGE DE WIENER - III. FILTRAGE DE KALMAN

Détermination de l'inverse stable d'un filtre RIF - Egalisation - Moindres carrés - Débruitage d'un signal par filtrage de Kalman

On considère un filtre linéaire causal RIF de RI h_n à $N + 1$ coefficients et de FT $H(z) = \sum_{i=0}^N h_i z^{-i}$.

Son inverse de FT : $G(z) = H^{-1}(z)$ est un filtre RII (de RI g_n) pouvant être instable ($G(z)$ a des pôles) s'il est implémenté sous forme causale.

On peut toutefois assurer sa stabilité quitte à en prendre une forme non causale. On perd alors l'implémentation « temps réel » (calcul en ligne) du filtre (le filtrage se fait alors en temps différé).

Une autre solution consiste à approcher g_n par un filtre RIF de support de longueur $L + 1$ pris causal généralement.

Application

On considère le filtre RIF de RI h_n à $N + 1$ coefficients : ($N = 2$) :
$$\begin{cases} h_0 = 1 \\ h_1 = -\frac{5}{2} \\ h_2 = 1 \end{cases}$$
 Sa FT est $H(z)$.

Détermination de l'inverse stable d'un filtre RIF

1. Mettre en évidence le caractère instable de la RI $g_n = TZ^{-1}[G(z) = H^{-1}(z)]$ prise causale.
2. Déterminer la RI $g_n = TZ^{-1}[G(z) = H^{-1}(z)]$ stable mais non causale.

Egalisation

3. Dans le cas où on limite le support de g_n à une longueur $L + 1$ (filtre RIF) : g_0, g_1, \dots, g_L (filtre causal) montrer que le problème peut se mettre sous la forme d'un système linéaire, en écrivant la relation d'égalisation :

$$g_n * h_n = \sum_{k=0}^N h_k g_{n-k} = h_0 g_n + h_1 g_{n-1} + h_2 g_{n-2} = \delta_n \quad \text{pour } n \text{ allant de } n = 0 \text{ jusqu'à } n = L.$$

On implémentera la méthode avec $L = 3$.

Moindres carrés

4. Implémenter la méthode *des moindres carrés* pour déterminer le filtre restaurateur $\{g_n\} = \{g_0, g_1, \dots, g_L\}$:
 - . moindres carrés direct (*LS*)
 - . moindres carrés récursifs (*RLS*)

Débruitage d'un signal par filtrage de Kalman

On observe le signal : $y(t) = x(t) + u(t)$ où $t \in \mathbf{Z}$.

On dispose du modèle suivant d'évolution du signal (modèle AR du 1er ordre) :

. équation d'Etat : $x(t) = a x(t-1) + b(t)$ avec : $a = 0.9$

. équation d'observation : $y(t) = x(t) + u(t)$

Les bruits de modèle $b(t)$ et d'observation $u(t)$ sont centrés, blancs et non corrélés entre eux.

Leur variance respective est : $\sigma_b^2 = 1$ $\sigma_u^2 = 9$

On pose : $\rho = \frac{\sigma_b^2}{\sigma_u^2}$

5.1. Déterminer l'expression de la puissance P_0 correspondant à la solution stationnaire de l'équation d'Etat. On prendra cette valeur comme expression initiale de $E[x^2(0)]$.

5.2. Ecrire un programme qui met en oeuvre le filtre de Kalman sur la trajectoire proposée.

TP 7. Filtrage optimal

I. MOINDRES CARRÉS

1. Résolution d'un système de k équations à n inconnues [Voir TD]

(La méthode des Moindres Carrés requiert toujours une surdétermination : $k \geq n$.)

Si $k=n$, ce n'est plus un problème d'optimisation et la solution est exacte, obtenue par inversion matricielle (alors qu'on a une pseudo-inverse en ce qui concerne le système surdéterminé)).

(cas multivariable)

LS $X := \begin{bmatrix} \vdots \\ \vdots \\ \vdots \end{bmatrix}$ $y := \begin{bmatrix} \vdots \\ \vdots \\ \vdots \end{bmatrix}$ $a := \begin{bmatrix} \vdots \\ \vdots \\ \vdots \end{bmatrix}$ $a = \begin{bmatrix} \vdots \\ \vdots \\ \vdots \end{bmatrix}$

Vérification : prédiction Y : $Y := X \cdot a$ $Y = \begin{bmatrix} \vdots \\ \vdots \\ \vdots \end{bmatrix}$

2. Modélisation d'une série de k données par un polynôme d'ordre n [Voir TD]

(La méthode des Moindres Carrés requiert toujours une surdétermination : $k \geq n$.)

Si $k=n$, ce n'est plus un problème d'optimisation et la solution est exacte : il s'agit du polynôme de Lagrange d'ordre $n=k$, ou encore une courbe spline, Bézier ...).

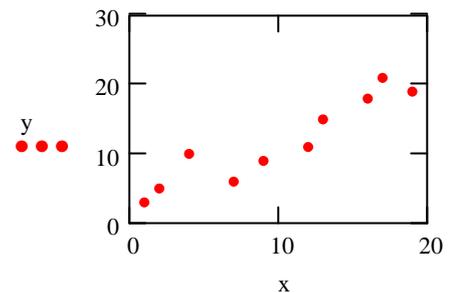
Données x y(x)

data :=

1	3
2	5
4	10
7	6
9	9
12	11
13	15
16	18
17	21
19	19

$x := \text{data}^{\langle 0 \rangle}$ $y := \text{data}^{\langle 1 \rangle}$ $k := \text{length}(x)$ $k = 10$

$$x^T = \begin{bmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 0 & 1 & 2 & 4 & 7 & 9 & 12 & 13 & 16 & 17 & 19 \end{bmatrix}$$

$$y^T = \begin{bmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 0 & 3 & 5 & 10 & 6 & 9 & 11 & 15 & 18 & 21 & 19 \end{bmatrix}$$


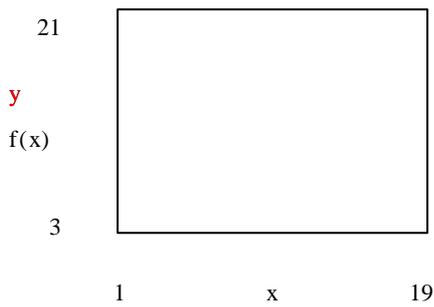
0. Polynôme d'ordre 0 (droite moyenne) (cas monovariante)

LS $X := \mathbf{1}$ $y := \mathbf{y}$ $a := \mathbf{a}$ $\mathbf{a} = \mathbf{a}$ $a := a_0$

$\mathbf{a} = \mathbf{a}$ $f(x) := 0 \cdot x + \mathbf{a}$

$$x^T = \begin{array}{|c|c|c|c|c|c|c|c|c|c|c|} \hline & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ \hline 0 & 1 & 2 & 4 & 7 & 9 & 12 & 13 & 16 & 17 & 19 \\ \hline \end{array}$$

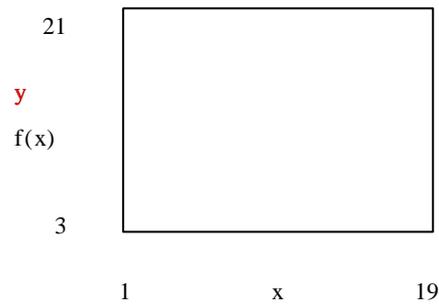
$y^T = \mathbf{y}$



Vérification : Fonction intégrée Mathcad

$a := \text{mean}(y)$ $\mathbf{a} = \mathbf{a}$

RLS $a := \mathbf{a}$ $\mathbf{a} = \mathbf{a}$ $f(x) := 0 \cdot x + \mathbf{a}$



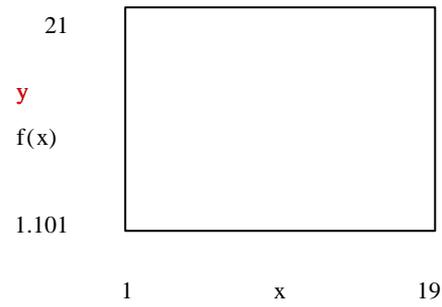
Prédiction $x_k := 20$ $y_k := f(x_k)$ $y_k = \mathbf{a}$

1. Droite passant par l'origine (cas monovariante)

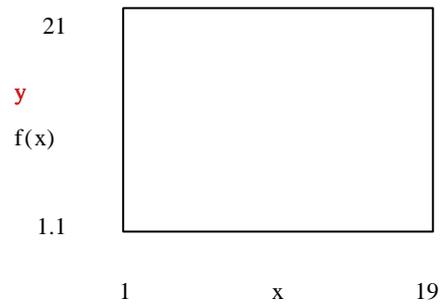
LS $X := \mathbf{■}$ $y := \mathbf{■}$ $a := \mathbf{■}$ $a := a_0$ $a = \mathbf{■}$ $f(x) := a \cdot x$

$$X^T = \mathbf{■}$$

$$y^T = \mathbf{■}$$



RLS $a := \mathbf{■}$ $a = \mathbf{■}$ $f(x) := a \cdot x$



Prédiction $x_k := 20$ $y_k := f(x_k)$ $y_k = \mathbf{■}$

1bis. Polynôme du 1er ordre (droite de régression linéaire) (cas multivariable)

LS $X^{(0)} := \mathbf{1}$ $X^{(1)} := \mathbf{x}$ $y := \mathbf{y}$ $\mathbf{a} := \mathbf{a}$

$\mathbf{a} = \mathbf{a}$ $f(x) := \mathbf{a}_1 \cdot x + \mathbf{a}_0$

$\mathbf{X}^T = \mathbf{X}$

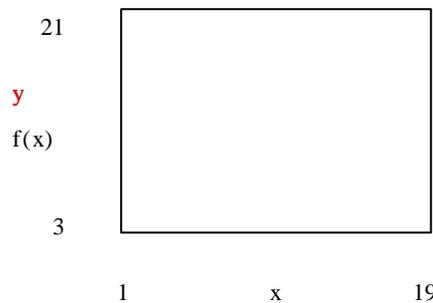
$\mathbf{y}^T = \mathbf{y}$

Vérification : Fonctions intégrées Mathcad

$\mathbf{a}_0 := \text{intercept}(X^{(1)}, y)$

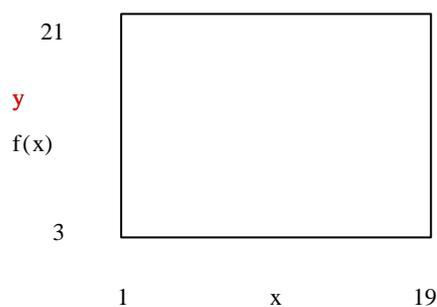
$\mathbf{a}_1 := \text{slope}(X^{(1)}, y)$

$\mathbf{a} = \mathbf{a}$



RLS $\mathbf{a} := \mathbf{a}$

$\mathbf{a} = \mathbf{a}$ $f(x) := \mathbf{a}_1 \cdot x + \mathbf{a}_0$



Prédiction $x_k := 20$

$y_k := f(x_k)$

$y_k = \mathbf{a}$

2. Polynôme du 2nd ordre (cas multivariable)

LS $X := \mathbf{\cdot}$

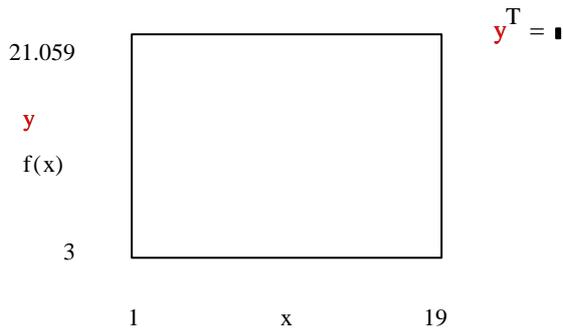
$y := \mathbf{\cdot}$

$a := \mathbf{\cdot}$

$a = \mathbf{\cdot}$

$$f(x) := a_2 \cdot x^2 + a_1 \cdot x + a_0$$

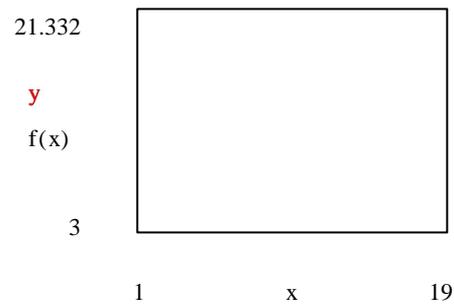
$X^T = \mathbf{\cdot}$



RLS $a := \mathbf{\cdot}$

$a = \mathbf{\cdot}$

$$f(x) := a_2 \cdot x^2 + a_1 \cdot x + a_0$$



Prédiction $x_k := 20$

$y_k := f(x_k)$

$y_k = \mathbf{\cdot}$

3. Polynôme d'ordre n (cas multivariable)

n := 5

k = 10

LS X := ■

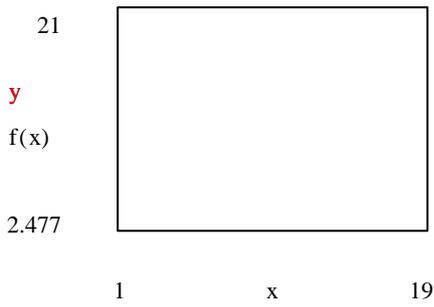
y := ■

a := ■

$$f(x) := \sum_{i=0}^n a_i \cdot x^i$$

$$a^T = \blacksquare$$

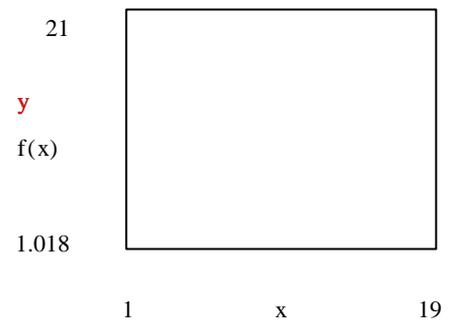
$$y^T = \blacksquare$$



RLS a := ■

$$a^T = \blacksquare$$

$$f(x) := \sum_{i=0}^n a_i \cdot x^i$$



Prédiction xk := 20

yk := f(xk)

yk = ■

3. Identification de la Fonction de Transfert à n coeffs d'un processus [Voir TD]

TD]

à partir de k mesures de sa sortie (et de k valeurs de l'entrée correspondante)

(La méthode des Moindres Carrés requiert toujours une surdétermination : $k \geq n$.

Si $k=n$, ce n'est plus un problème d'optimisation, la solution est exacte, obtenue par déconvolution).

Données k u(k) y(k)

data :=

0	1	0.9
1	0.8	2.5
2	0.6	2.4
3	0.4	1.3
4	0.2	1.2
5	0	0.8
6	0.2	0
7	0.4	0.9
8	0.6	1.4
9	0.8	1.9
10	1	2.3
11	0.8	2.4
12	0.6	2.3
13	0.4	1.3
14	0.2	1.2

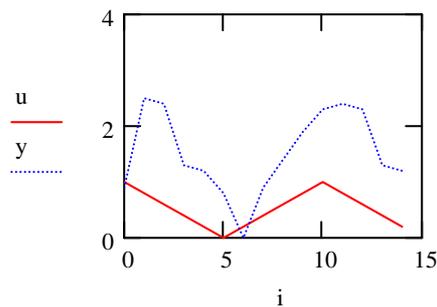
i := data^{<0>}

u := data^{<1>}

y := data^{<2>}

k := length(u)

k = 15



1. FT à n=2 coeffs (cas multivariable)

LS X := **a**

y := **b**

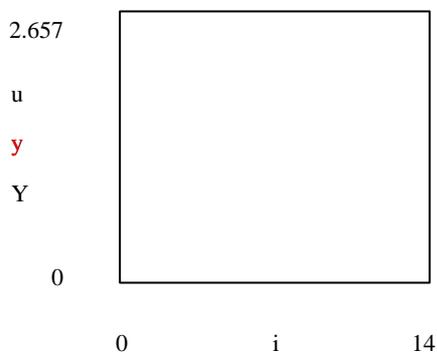
a := **a**

a = **a**

Coeffs du filtre : b := **a**

b = **a**

Vérification : Sortie prédite Y :



$$Y := \begin{cases} \text{for } i \in 0..k-1 \\ \left| \begin{array}{l} y_i \leftarrow b_0 \cdot u_i \text{ if } i = 0 \\ y_i \leftarrow b_0 \cdot u_i + b_1 \cdot u_{i-1} \text{ otherwise} \end{array} \right. \\ y \end{cases}$$

2. FT à $n=p+q+1$ coeffs (cas multivariable)

LS $p := 1$ $q := 0$ $X := \mathbf{1}$

$n := p + q + 1$

$n = 2$

$y := \mathbf{1}$

$a := \mathbf{1}$

$\mathbf{a}^T = \mathbf{1}$

Coeffs du filtre :

$$b := \begin{cases} \text{for } j \in 0..p \\ b_j \leftarrow a_j \\ b \end{cases}$$

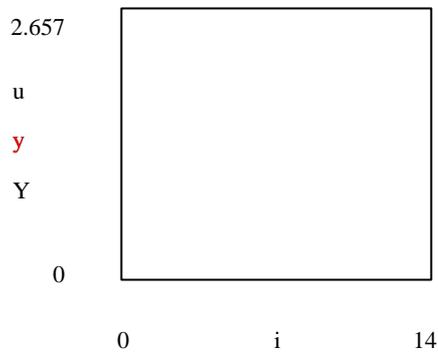
$\mathbf{b}^T = \mathbf{1}$

$$a := \begin{cases} \text{for } j \in p+1..p+q \text{ if } p+q \geq p+1 \\ c_{j-p} \leftarrow a_j \\ c \end{cases}$$

$\mathbf{a}^T = \mathbf{1}$

Vérification : Sortie prédite Y :

$$Y := \begin{cases} \text{for } i \in 0..k-1 \\ \left| \begin{array}{l} y_i \leftarrow \sum_{j=0}^{\min(p,i)} b_j \cdot u_{i-j} - \sum_{j=1}^{\min(q,i)} a_j \cdot y_{i-j} \text{ if } \min(q,i) \geq 1 \\ y_i \leftarrow \sum_{j=0}^{\min(p,i)} b_j \cdot u_{i-j} \text{ otherwise} \end{array} \right. \\ y \end{cases}$$



TP 7 ANNEXE. Filtrage optimal

I. MOINDRES CARRÉS

1. Filtrage LS (Least Squares - moindres carrés direct)

Identification de la RI h (dim. n) d'un canal à partir de la mesure bruitée de sa réponse y (dim. k) à un signal x (dim. k)

$$\begin{pmatrix} y_n \\ \vdots \\ y_{n+k-1} \end{pmatrix} = \begin{pmatrix} x_{n-1} & \dots & x_1 & x_0 \\ x_n & \dots & x_2 & x_1 \\ \vdots & \vdots & \vdots & \vdots \\ x_{n+k-2} & \dots & x_k & x_{k-1} \end{pmatrix} \cdot \begin{pmatrix} h_0 \\ \vdots \\ h_{n-1} \end{pmatrix} + \begin{pmatrix} w_n \\ \vdots \\ w_{n+k-1} \end{pmatrix}$$

Y Sortie vecteur $k \times 1$ X matrice $k \times n$ h RI vecteur $n \times 1$ W Bruit vecteur $k \times 1$

Solution : (Moindres carrés direct) $h := (X^T \cdot X)^{-1} \cdot X^T \cdot Y$ Il faut le système surdéterminé: $k \geq n$

Programmation : k : nombre de points d'observation de y RAZ : $X := \text{submatrix}(X, 0, 0, 0, 0)$
 n : nombre de points inconnus de h $Y := \text{submatrix}(Y, 0, 0, 0, 0)$

Faire croître k à partir de la valeur $k = n$ et comparer la RI vraie du canal avec celle obtenue par identification

$k := 3$ $n := 3$ $i := 0..k-1$ $j := 0..n-1$ $p := 0..n+k-2$ Réserve mémoire : $y_p := 0$ $Y_i := 0$

Exemple de signal d'entrée : $x := \text{whiten}(n+k-1)$ Bruit blanc d'entrée = meilleur apprentissage possible

Exemple de RI (vraie) du canal : $j := 0..n-1$ $h_j := j$ $h^T = (0 \ 1 \ 2)$

Sortie : $p := 0..n+k-2$ $y_p := \text{for } m \in 0..n-1$
 $y_p \leftarrow y_p + h_m \cdot x_{p-m}$ if $m \leq p$

Bruitage de la sortie (canal) : $w_p := \text{rnd}(0.2)$ $y_p := y_p + w_p$

Constitution de la matrice X : Réserve mémoire : $X_{i,j} := 0$ $X_{i,j} := \text{for } i \in 0..k-1$
for $j \in 0..n-1$

Constitution du vecteur $Y[0 \rightarrow k-1] = y[n \rightarrow n+k-1]$:
 $Y_i := \text{for } i \in 0..k-1$ ou encore :
 $Y_i \leftarrow y_{i+n-1}$ $Y := \text{submatrix}(y, n-1, n+k-2, 0, 0)$ RAZ :

Moindres carrés (direct) : $h := \blacksquare$ $h^T = \blacksquare$

Méthode directe (déconvolution) (seulement si $k = n$) : $Y := \text{submatrix}(Y, 0, n-1, 0, 0)$ $X := \text{submatrix}(X, 0, n-1, 0, n-1)$

$h := X^{-1} \cdot Y$ $h^T = (0.137 \ 1.031 \ 1.882)$

Conclusion: . pour $k = n$:
. pour $k > n$:

II. FILTRAGE DE WIENER

Identification de la RI h (dim. n) d'un canal à partir de la mesure bruitée de sa réponse y (dim. k) à un signal x (dim. k)

1. Filtrage de Wiener

$P := n - 1$ $N := P + 1$ Nombre de points de la RI inconnue
 $n := N - 1$ entre 0 et $N-1$
 $x := \text{whiten}(N)$ $k := 0..N - 1$ $xr_k := x_{N-1-k}$ $x = \begin{pmatrix} 0.489 \\ -0.381 \\ -0.491 \end{pmatrix}$ $xr = \begin{pmatrix} -0.491 \\ -0.381 \\ 0.489 \end{pmatrix}$

$r := \text{vcov}(y, xr)$ $R := \text{mautocov}(x)$ $R = \begin{pmatrix} 0.192 & -0.021 & -0.075 \\ -0.021 & 0.192 & -0.021 \\ -0.075 & -0.021 & 0.192 \end{pmatrix}$

$r = \begin{pmatrix} -4.452 \times 10^{-3} \\ 0.046 \\ 0.159 \end{pmatrix}$ $h := \blacksquare$ $h^T = \blacksquare$

III. FILTRAGE DE KALMAN

1. Filtrage de Kalman

$N := 100$ $t := 0..N$
 $mb := 0$ $Vb := 1$ $b := \text{BruitBlanc}(N + 1, mb, Vb)$ Bruit de modèle
 $mu := 0$ $Vu := 9$ $u := \text{BruitBlanc}(N + 1, mu, Vu)$ Bruit d'observation
 $a := 0.9$ $\rho := \frac{Vb}{Vu}$ Etat initial : $x0 := 20$

Signal bruité :

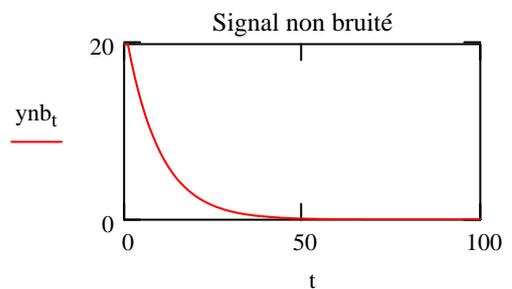
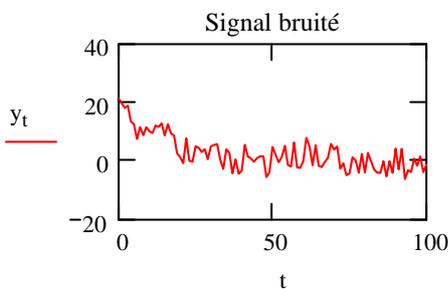
Signal non bruité :

$$x := \begin{cases} \text{for } t \in 0..N \\ \quad x_t \leftarrow x_0 \text{ if } t = 0 \\ \quad x_t \leftarrow x_0 + b_t \text{ if } t = 1 \\ \quad x_t \leftarrow a \cdot x_{t-1} + b_t \text{ if } t > 1 \end{cases}$$

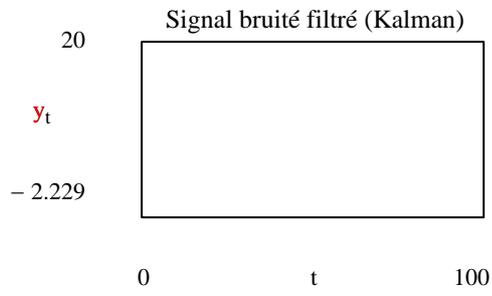
$$xnb := \begin{cases} \text{for } t \in 0..N \\ \quad xnb_t \leftarrow x_0 \text{ if } t = 0 \\ \quad xnb_t \leftarrow xnb_0 \text{ if } t = 1 \\ \quad xnb_t \leftarrow a \cdot xnb_{t-1} \text{ if } t > 1 \end{cases}$$

$$y_t := x_t + u_t$$

$$y_{nb}_t := xnb_t$$



Filtre de Kalman : Matrices : $A := a$ $C := 1$



```

y :=
| K_0 ← Vb / (1 - a^2)
| for t ∈ 0..N
| x
    
```

Fonctions Bibliothèque :

Moyenne m, Vecteur r et Matrice d'Autocovariance r et Variance V du signal x de dim K (par ergodicité) :

. Moyenne :

$$\text{moy}(x) \equiv \begin{cases} K \leftarrow \text{length}(x) \\ m \leftarrow \frac{1}{K} \cdot \sum_{n=0}^{K-1} x_n \\ m \end{cases}$$

. Vecteur d'AutoCovariance :

$$\text{vautocov}(x) \equiv \begin{cases} K \leftarrow \text{length}(x) \\ m \leftarrow \text{moy}(x) \\ \text{for } k \in 0..K-1 \\ r_k \leftarrow \frac{1}{K} \cdot \sum_{n=0}^{K-k-1} (x_{n+k} - m) \cdot (x_n - m) \\ r \end{cases}$$

. Matrice d'AutoCovariance :

$$\text{mautocov}(x) \equiv \begin{cases} K \leftarrow \text{length}(x) \\ r \leftarrow \text{vautocov}(x) \\ \text{for } i \in 0..K-1 \\ \text{for } j \in 0..K-1 \\ R_{i,j} \leftarrow r_{|i-j|} \\ R \end{cases}$$

. Variance :

$$\text{Var}(x) \equiv \text{vautocov}(x)_0$$

Fonction YuleWalker(φ_{xx}, N) : Résolution du système de Yule-Walker Ra = v par inversion matricielle

Paramètres d'entrée : . φ_{xx} : vecteur fonction d'autocovariance (dim K) du signal x (dim K) : φ_{xx0}, φ_{xx1} ... φ_{xxK-1}
 . N : dim de l'ordre du filtre (formeur/LPC) AR

Paramètre de sortie : . le vecteur a (dim N+1) :
 a₀ = V : variance (du bruit blanc si filtre formeur); a₁, a₂, ... a_N : paramètres du filtre (formeur/LPC) AR d'ordre N

$$\text{YuleWalker}(\phi_{xx}, N) \equiv \begin{cases} \text{for } i \in 0..N \\ \text{for } j \in 0..N \\ R_{i,j} \leftarrow \phi_{xx} |i-j| \\ b_0 \leftarrow 0 \\ \text{for } i \in 1..N \\ b_i \leftarrow \phi_{xx_i} \\ \alpha \leftarrow \begin{pmatrix} -1 \\ \phi_{xx_0} \end{pmatrix} \cdot R^{-1} \cdot b \\ a_0 \leftarrow \frac{\phi_{xx_0}}{\phi_{xx_0} \cdot \alpha_0 + 1} \\ \text{for } i \in 1..N \\ a_i \leftarrow a_0 \cdot \alpha_i \\ a \end{cases}$$

Exemple : K := 3 r₀ := 6
 N := 2 r₁ := 3
 r₂ := 0

a := YuleWalker(r, N)

V := a₀

$$a = \begin{pmatrix} 4 \\ -0.667 \\ 0.333 \end{pmatrix}$$

V = 4
 i := 1..N
 a_i =

-0.667
0.333

Fonction YuleW(x,N) : Résolution du système de Yule-Walker $Ra = v$ par inversion matricielle

Paramètres d'entrée : . x: signal x (dim K) : $x_0, x_1 \dots x_{K-1}$
 . N : dim de l'ordre du filtre (formeur/LPC) AR

Paramètre de sortie : . le vecteur a (dim N+1) :
 $a_0 = V$: variance (du bruit blanc si filtre formeur); a_1, a_2, \dots, a_N : paramètres du filtre (formeur/LPC) AR d'ordre N

$$\text{YuleW}(x,N) \equiv \left\{ \begin{array}{l} r \leftarrow \text{vautocov}(x) \\ a \leftarrow \text{YuleWalker}(r,N) \\ a \end{array} \right.$$

Bruit blanc de K points de moyenne m et de variance V :

Bruit blanc de K points centré de variance 0.083 :

$$\text{WhiteNoise}(K) \equiv \left\{ \begin{array}{l} A \leftarrow 281 \\ P \leftarrow 31357 \\ v_0 \leftarrow 100 \\ \text{for } n \in 1..N-1 \\ \quad v_n \leftarrow \text{mod}(A \cdot v_{n-1}, P) \\ \text{for } n \in 0..N-1 \\ \quad w_n \leftarrow \frac{1}{P} \cdot v_n - 0.5 \\ w \end{array} \right.$$

Bruit blanc de K points de moyenne m et de variance V :

$$\text{BruitBlanc}(K, m, V) \equiv \sqrt{12 \cdot V} \cdot \text{whiten}(K) + m$$

$$\text{BruitBlanc2}(K, m, V) \equiv \sqrt{12 \cdot V} \cdot \text{WhiteNoise}(K) + m$$

Réponse reponse (vecteur dim. K) du filtre AR d'ordre N (coeffs $a_i, 0 \leq i \leq N$) au signal x (vecteur dim. K)

$$\text{reponse}(x, a, N) \equiv \left\{ \begin{array}{l} K \leftarrow \text{length}(x) \\ \text{for } k \in 0..K-1 \\ \quad \left\{ \begin{array}{l} y_k \leftarrow x_k \text{ if } k = 0 \\ y_k \leftarrow x_k - \sum_{i=0}^k a_i \cdot y_{k-i} \text{ if } 1 \leq k < N \\ y_k \leftarrow x_k - \sum_{i=0}^N a_i \cdot y_{k-i} \text{ if } k \geq N \end{array} \right. \\ y \end{array} \right.$$

TFD

$$\text{TFD}(x) \equiv \left| \begin{array}{l} M \leftarrow \text{length}(x) \\ \text{for } k \in 0..M-1 \\ \quad X_k \leftarrow \frac{1}{M} \cdot \sum_{n=0}^{M-1} x_n \cdot e^{-i \cdot 2 \cdot \pi \cdot k \cdot \frac{n}{M}} \\ \end{array} \right| X$$

ou encore :

$$\text{TFD2}(x) \equiv \left| \begin{array}{l} M \leftarrow \text{length}(x) \\ \text{for } k \in 0..M-1 \\ \quad R \leftarrow 0 \\ \quad I \leftarrow 0 \\ \quad \text{for } n \in 0..M-1 \\ \quad \quad R \leftarrow R + x_n \cdot \cos\left(2 \cdot \pi \cdot k \cdot \frac{n}{M}\right) \\ \quad \quad I \leftarrow I - x_n \cdot \sin\left(2 \cdot \pi \cdot k \cdot \frac{n}{M}\right) \\ \quad X_k \leftarrow \frac{1}{M} \cdot (R + i \cdot I) \\ \end{array} \right| \text{return } X$$

TFDI

$$\text{TFDI}(X) \equiv \left| \begin{array}{l} M \leftarrow \text{length}(X) \\ \text{for } n \in 0..M-1 \\ \quad x_n \leftarrow \sum_{k=0}^{M-1} X_k \cdot e^{i \cdot 2 \cdot \pi \cdot k \cdot \frac{n}{M}} \\ \end{array} \right| x$$

Convolution causale (signaux de même longueur)

$$\text{convcausal2}(x, h) \equiv \left| \begin{array}{l} M \leftarrow \text{length}(x) \\ \text{for } n \in 0..M-1 \\ \quad y_n \leftarrow \sum_{k=0}^n x_k \cdot h_{n-k} \\ \end{array} \right| y$$

Convolution causale (signaux de longueur différente)

$$\text{convcausal}(x, h) \equiv \left| \begin{array}{l} M \leftarrow \text{length}(x) \\ N \leftarrow \text{length}(h) \\ \text{for } n \in 0..M-1 \\ \quad y_n \leftarrow \sum_{k=0}^n x_k \cdot h_{n-k} \quad \text{if } n < N \\ \quad y_n \leftarrow 0 \quad \text{otherwise} \\ \end{array} \right| y$$

Transformée de Hilbert

```
Hilbert(x) ≡ | M ← length(x)
               | f0 ← 0
               | for n ∈ 1..M - 1
               |   fn ← 1 / (π · n)
               | TH ← convcausal(x, f)
               | TH
```

Vecteur r de covariance des signaux x et y de dim K (par ergodicité) :

. Vecteur de Covariance :

```
vcov(y, x) ≡ | K ← length(x)
               | for k ∈ 0..K - 1
               |   vk ← 1 / K · ∑n=0K-1-k yn+k · xn
               | v
```

```
vcovtermegeneral(x, y) ≡ | K ← length(x)
                           | mx ← moy(x)
                           | my ← moy(y)
                           | for k ∈ 0..K - 1
                           |   rk ← 1 / (K - k) · ∑n=0K-k-1 (xn+k - mx) · (yn - my)
                           | r
```

Vecteur et Matrice nuls :

```
VecteurNul(N) ≡ | for k ∈ 0..N - 1
                  |   Nulk ← 0
                  | Nul
```

```
MatriceNul(Nlignes, Ncols) ≡ | for l ∈ 0..Nlignes - 1
                               |   for c ∈ 0..Ncols - 1
                               |     Null,c ← 0
                               | Nul
```

Prédiction à 1 pas

```
predict1pas(v, N) ≡ | L ← length(v)
                    | m ← moy(v)
                    | v ← v - m
                    | φ ← vautocov(v)
                    | a ← YuleWalker(φ, N)
                    |   vL ← - ∑i=1N ai · vL-i
                    | v ← v + m
                    | v
```

Prédiction à Npas pas

```
predictNpas(v, N, Npas) ≡ | for n ∈ 1..Npas
                           |   v ← predict1pas(v, N)
                           | v
```

TRAITEMENT DU SIGNAL

ANNEXE

8 ANNEXE. Décision (Classification, Reconnaissance des Formes)

I. CAS BI-CLASSE

Dans certains cas, la finalité de la mesure d'un signal est la reconnaissance du phénomène physique qui en est la source.

Lorsque celui-ci fait partie d'un ensemble de situations déjà connues *à priori*, le problème se ramène à la Reconnaissance Des Formes (RDF) et relève de la théorie de la décision.

Un cas simple, fréquemment rencontré et particulièrement étudié correspond au cas de la décision entre 2 classes.

L'observateur est alors en présence d'une source qui génère une sortie caractéristique d'un phénomène parmi 2 possibles, que l'on appelle des *classes* (ou *hypothèses*) H_1 et H_2 .

Cette source est alors perturbée par un canal de transmission qui rend impossible une détermination sûre des hypothèses.

Le signal mesuré est alors utilisé pour construire un élément X d'un *espace de représentation* (ou d'*observation*) \mathcal{X} du phénomène observé. Cet espace de représentation doit correspondre à une statistique suffisante du phénomène observé, c'est-à-dire que la connaissance du vecteur d'observation X doit permettre d'avoir tous les éléments pour prendre une décision correcte. Dans la mesure du possible, aucune variable de décision ne doit donc être volontairement occultée.

La constitution de cet espace permet ensuite de concevoir et de qualifier une *règle de décision*, dont la fonction est de déterminer au mieux à quelle hypothèse correspond l'élément X construit à partir du signal observé.

Cette règle de décision réalise donc une partition de l'espace de représentation en 2 sous-espaces disjoints et complémentaires \mathcal{X}_1 et \mathcal{X}_2 , c'est-à-dire un découpage de l'espace de représentation en 2 sous-espaces tels que :

$$\mathcal{X}_1 \cup \mathcal{X}_2 = \mathcal{X} \quad \text{et} \quad \mathcal{X}_1 \cap \mathcal{X}_2 = \emptyset$$

Une fois cette règle de décision constituée, il ne reste alors plus qu'à l'appliquer aux individus, pour déterminer la classe à laquelle ils appartiennent.

Cas où les densités de probabilité du vecteur d'observation conditionnellement à chaque classe sont connues

Lorsque les densités de probabilité $\rho(X|H_1)$ et $\rho(X|H_2)$ du vecteur d'observation conditionnellement à chaque classe sont connues, il est possible de les utiliser pour déterminer la règle de décision.

Il faut pour cela considérer les 4 cas possibles de décisions et d'hypothèses, et leurs probabilités respectives :

	Décider 1	Décider 2
dans la classe H_1	$P(D_1 H_1) = \int_{\mathcal{X}_1} \rho(X H_1)dX$	$P(D_2 H_1) = \int_{\mathcal{X}_2} \rho(X H_1)dX$
dans la classe H_2	$P(D_1 H_2) = \int_{\mathcal{X}_1} \rho(X H_2)dX$	$P(D_2 H_2) = \int_{\mathcal{X}_2} \rho(X H_2)dX$

Par exemple, si l'hypothèse H_1 correspond à un système en bon état, et l'hypothèse H_2 à un dysfonctionnement de ce système, alors $P(D_2|H_2)$ correspond à la probabilité de détecter le défaut, $P(D_2|H_1)$ correspond à la probabilité de fausse alarme, et $P(D_1|H_2)$ à la probabilité de ne pas détecter un défaut.

Fonction de coût R

Ces 4 probabilités sont ensuite utilisées pour construire une *fonction de risque* (ou de coût) R égale à une somme pondérée de chacune des hypothèses, et que l'on souhaite rendre la plus petite possible :

$$R = C_{11}P(D_1|H_1)P(H_1) + C_{12}P(D_1|H_2)P(H_2) + C_{21}P(D_2|H_1)P(H_1) + C_{22}P(D_2|H_2)P(H_2)$$

Comme les décisions erronées sont plus pénalisées que les décisions correctes, les coefficients de pondération doivent être choisis de façon à vérifier les 2 relations d'ordre suivantes :

$$C_{21} > C_{11} > 0 \quad \text{et} \quad C_{12} > C_{22} > 0$$

Cette fonction de risque admet un minimum absolu, car elle peut être minorée par la valeur :

$$R_{\min} = C_{11}P(H_1) + C_{22}P(H_2) \quad \text{car, pour } R_{\min}, \text{ on a le cas idéal : } \begin{cases} P(D_1|H_1) = P(D_2|H_2) = 1 \\ P(D_1|H_2) = P(D_2|H_1) = 0 \end{cases}$$

De plus, elle peut également s'écrire sous une nouvelle forme, obtenue à partir des densités de probabilités conditionnelles, décrites plus haut :

$$P(D_2|H_1) = 1 - P(D_1|H_1) \quad \text{et} \quad P(D_2|H_2) = 1 - P(D_1|H_2)$$

$$\rightarrow R = C_{11}P(D_1|H_1)P(H_1) + C_{12}P(D_1|H_2)P(H_2) + C_{21}[1 - P(D_1|H_1)]P(H_1) + C_{22}[1 - P(D_1|H_2)]P(H_2)$$

$$\rightarrow R = C_{21}P(H_1) + C_{22}P(H_2) + (C_{12} - C_{22})P(H_2)P(D_1|H_2) + (C_{11} - C_{21})P(H_1)P(D_1|H_1)$$

$$\rightarrow R = C_{21}P(H_1) + C_{22}P(H_2) + \gamma \quad \text{en posant : } \gamma = (C_{12} - C_{22})P(H_2)P(D_1|H_2) + (C_{11} - C_{21})P(H_1)P(D_1|H_1)$$

$$\text{avec : } \gamma = \int_{\chi_1} \{(C_{12} - C_{22})P(H_2)\rho(X|H_2) - (C_{21} - C_{11})P(H_1)\rho(X|H_1)\} dX$$

En comparant les expressions de R et de R_{\min} , on a :

$$R = C_{21}P(H_1) + C_{22}P(H_2) + \gamma \quad \text{avec } C_{21} > C_{11} > 0$$

$$\text{et } R_{\min} = C_{11}P(H_1) + C_{22}P(H_2)$$

soit, pour que $R = R_{\min}$, il faut : $\gamma < 0$ (une probabilité \in à l'intervalle $[0, 1]$)

R est donc minimale lorsque la quantité intégrée : $\{(C_{12} - C_{22})P(H_2)\rho(X|H_2) - (C_{21} - C_{11})P(H_1)\rho(X|H_1)\}$ est < 0 dans la totalité du domaine χ_1 .

Ceci amène donc à choisir l'ensemble χ_1 des valeurs de X pour lesquelles on décide de choisir le cas 1, comme l'ensemble de toutes les observations pour lesquelles :

$$\frac{\rho(X|H_2)}{\rho(X|H_1)} < \frac{(C_{21} - C_{11})P(H_1)}{(C_{12} - C_{22})P(H_2)} \quad (\text{car } C_{12} > C_{22} \quad \text{et} \quad \rho \geq 0 \text{ (probabilité } = \int_{\chi_1} \rho dX \geq 0))$$

Dans le cas contraire, le vecteur X correspond à un élément de χ_2 , et donc à la décision du cas 2.

Le critère bayésien conduit donc à une *règle de décision* appelé le *test du rapport de vraisemblance* :

Test du rapport de vraisemblance

Décider 1 si $L(X) < \mu$

Décider 2 si $L(X) > \mu$

$$\text{avec : } L(X) = \frac{\rho(X|H_2)}{\rho(X|H_1)} \quad \text{et} \quad \mu = \frac{(C_{21} - C_{11})P(H_1)}{(C_{12} - C_{22})P(H_2)}$$

Cette règle de décision consiste donc à comparer une fonction du vecteur d'observation $L(X)$, appelée *récepteur*, à un *seuil de décision* μ qui dépend des probabilités *a priori* des hypothèses et des coûts fixés par l'utilisateur.

Dans certains cas, notamment lorsque les densités de probabilité correspondant aux 2 hypothèses sont gaussiennes, on préférera utiliser le logarithme du rapport de vraisemblance, qui devra être comparé au logarithme de μ .

Si les 2 hypothèses sont considérées comme symétriques et si le risque correspond à la probabilité d'erreur totale, alors :

$$C_{11} = C_{22} = 0 \quad \text{et} \quad C_{12} = C_{21} = 1$$

soit $R = P(D_1|H_2)P(H_2) + P(D_2|H_1)P(H_1)$

Le critère de vraisemblance correspond alors à choisir l'hypothèse dont la probabilité *a posteriori* est la plus élevée :

$$\begin{array}{ll} \text{Décider 1} & \text{si} \quad \frac{\rho(X|H_2)}{\rho(X|H_1)} < \frac{P(H_1)}{P(H_2)} \\ \text{Décider 2} & \text{si} \quad \frac{\rho(X|H_2)}{\rho(X|H_1)} > \frac{P(H_1)}{P(H_2)} \end{array}$$

Cette règle de décision correspond donc à choisir l'hypothèse qui, compte tenu du vecteur d'observation X obtenu, se révèle être la plus probable.

Cas où les densités de probabilité du vecteur d'observation conditionnellement à chaque classe ne sont pas connues

Enfin, si les probabilités *a priori* des 2 hypothèses ne sont pas connues, on peut cependant fixer expérimentalement le seuil d'une règle de décision construite empiriquement en suivant l'approche bayésienne.

En exprimant le critère de risque en fonction de la probabilité de l'hypothèse 2, $P(H_2)$, de la probabilité de « fausse alarme », $P(D_2|H_1)$, et de la probabilité de détection, $P(D_2|H_2)$, on montre que le risque dépend linéairement de la probabilité $P(H_2)$:

$$R = C_{11} + (C_{21} - C_{11})P(D_2|H_1) + P(H_2) \{ (C_{12} - C_{11}) - (C_{12} - C_{22})P(D_2|H_2) - (C_{21} - C_{11})P(D_2|H_1) \}$$

Règle du minimax

La règle de décision qui minimise le risque maximum est donc celle pour laquelle le coefficient de $P(H_2)$ est nul :

$$\begin{aligned} & (C_{12} - C_{11}) - (C_{12} - C_{22})P(D_2|H_2) - (C_{21} - C_{11})P(D_2|H_1) = 0 \\ \rightarrow & P(D_2|H_2) = \frac{C_{12} - C_{11}}{C_{12} - C_{22}} - \frac{C_{21} - C_{11}}{C_{12} - C_{22}} P(D_2|H_1) \end{aligned}$$

Ce choix est appelé *règle du minimax*. Elle exprime le compromis admissible entre la probabilité de détection et la probabilité de fausse alarme.

Courbe COR

Dans la pratique, on estime les fréquences relatives des cas de détection et de fausse alarme en fonction du seuil de décision μ . On obtient alors une courbe, appelée *courbe COR* (caractéristique de l'opérateur de réception $L(X)$) :

$$P(D_2|H_2) = f [P(D_2|H_1)]$$

Cette courbe passe nécessairement par les points (0,0) et (1,1) puisque, lorsque $\mu \rightarrow 0$, le cas 2 est toujours retenu, tandis que, lorsque $\mu \rightarrow \infty$, le cas 2 n'est jamais retenu :

$$\begin{aligned} \lim_{\mu \rightarrow 0} P(D_2|H_1) &= 1 & \lim_{\mu \rightarrow \infty} P(D_2|H_1) &= 0 \\ \lim_{\mu \rightarrow 0} P(D_2|H_2) &= 1 & \lim_{\mu \rightarrow \infty} P(D_2|H_2) &= 0 \end{aligned}$$

De plus, l'expression du détecteur permet de déduire 2 inégalités :

$$\begin{cases} P(D_1|H_2) < \mu P(D_1|H_1) \\ P(D_1|H_2) > \mu P(D_2|H_1) \end{cases} \leftrightarrow \begin{cases} P(D_2|H_2) > 1 - \mu - \mu P(D_2|H_1) \\ P(D_2|H_2) > \mu P(D_2|H_1) \end{cases}$$

II. CAS BI-CLASSE AVEC NON-DECISION

Pour certains problèmes de Reconnaissance Des Formes, il est possible (ou préférable) de ne pas attribuer une prise de décision à chaque valeur du vecteur d'observation, afin d'obtenir une détection plus sûre.

Par rapport au cas bi-classe avec décision précédent, cela revient à ajouter une possibilité supplémentaire de non décision parmi les cas possibles de décisions et d'hypothèses :

	<i>Ne rien décider</i>	<i>Décider 1</i>	<i>Décider 2</i>
<i>dans la classe H_1</i>	$P(D_0 H_1)$	$P(D_1 H_1)$	$P(D_2 H_1)$
<i>dans la classe H_2</i>	$P(D_0 H_2)$	$P(D_1 H_2)$	$P(D_2 H_2)$

Cette nouvelle situation entraîne donc une modification de l'expression de la *fonction de risque* R :

$$R = C_{01}P(D_0|H_1)P(H_1) + C_{11}P(D_1|H_1)P(H_1) + C_{21}P(D_2|H_1)P(H_1) + C_{02}P(D_0|H_2)P(H_2) + C_{12}P(D_1|H_2)P(H_2) + C_{22}P(D_2|H_2)P(H_2)$$

Puisqu'une décision erronée présente un coût plus élevé qu'une non-décision, les différents coefficients de pénalité sont choisis de façon à vérifier les relations d'ordre suivantes :

$$C_{21} > C_{01} > C_{11} > 0 \quad \text{et} \quad C_{12} > C_{02} > C_{22} > 0$$

Cas où les densités de probabilité du vecteur d'observation conditionnellement à chaque classe sont connues

L'expression des probabilités de non-décision en fonction des probabilités de décision :

$$P(D_0|H_1) = 1 - P(D_1|H_1) - P(D_2|H_1) \quad \text{et} \quad P(D_0|H_2) = 1 - P(D_1|H_2) - P(D_2|H_2)$$

conduit alors à une nouvelle expression de la fonction de coût R :

$$R = C_{01}P(H_1) + (C_{11} - C_{01})P(D_1|H_1)P(H_1) + (C_{21} - C_{01})P(D_2|H_1)P(H_1) + C_{02}P(H_2) + (C_{12} - C_{02})P(D_1|H_2)P(H_2) + (C_{22} - C_{02})P(D_2|H_2)P(H_2)$$

$$\rightarrow R = C_{01}P(H_1) + C_{02}P(H_2) + \gamma_1 + \gamma_2 \quad \text{en posant :}$$

$$\begin{aligned} \gamma_1 &= \int_{\mathcal{X}_1} \{(C_{12} - C_{02})P(H_2)\rho(X|H_2) - (C_{01} - C_{11})P(H_1)\rho(X|H_1)\}dX \\ \text{et} \quad \gamma_2 &= \int_{\mathcal{X}_2} \{(C_{21} - C_{01})P(H_1)\rho(X|H_1) - (C_{02} - C_{22})P(H_2)\rho(X|H_2)\}dX \end{aligned}$$

La minimisation de R conduit à la définition de 2 sous-espaces de décision \mathcal{X}_1 et \mathcal{X}_2 , qui ne sont plus complémentaires:

$$\begin{aligned} \mathcal{X}_1 &= \{X \in \mathcal{X}, \quad (C_{12} - C_{02})P(H_2)\rho(X|H_2) - (C_{01} - C_{11})P(H_1)\rho(X|H_1) < 0\} \\ \mathcal{X}_2 &= \{X \in \mathcal{X}, \quad (C_{21} - C_{01})P(H_1)\rho(X|H_1) - (C_{02} - C_{22})P(H_2)\rho(X|H_2) < 0\} \end{aligned}$$

Règle de décision

On en déduit la règle de décision :

Test du rapport de vraisemblance **Décider 1** **si** $\Lambda(X) < \mu_1$
Décider 2 **si** $\Lambda(X) > \mu_2$
Décider 0 **sinon**

$$\text{avec : } \Lambda(X) = \frac{\rho(X|H_2)}{\rho(X|H_1)}, \quad \mu_1 = \frac{(C_{01} - C_{11})P(H_1)}{(C_{12} - C_{02})P(H_2)} \quad \text{et}$$

$$\mu_2 = \frac{(C_{01} - C_{21})P(H_1)}{(C_{22} - C_{02})P(H_2)}$$

Cette règle de décision consiste donc à comparer une fonction du vecteur d'observation à 2 seuils, aucune décision n'étant prise lorsqu'aucune de ces 2 inégalités n'est vérifiée.

Cas où les densités de probabilité du vecteur d'observation conditionnellement à chaque classe ne sont pas connues

Enfin, lorsque l'on ne dispose pas des expressions analytiques des densités de probabilité conditionnelles, on peut chercher à minimiser la fonction de risque en fonction des probabilités *a priori* des hypothèses H_1 et H_2 :

$$R = (C_{11} - C_{01})P(D_1|H_1) + (C_{21} - C_{01}) + P(D_2|H_1) + C_{01}P(H_1) + P(H_2) [(C_{02} - C_{01}) + (C_{12} - C_{02})P(D_1|H_2) + (C_{01} - C_{11})P(D_1|H_1)] - P(H_2) [(C_{21} - C_{01})P(D_2|H_1) + (C_{02} - C_{22})P(D_2|H_2)]$$

Ceci conduit alors à prendre le terme en facteur de $P(H_2)$ égal à 0. Les 2 seuils de décision peuvent alors être déterminés indépendamment en prenant les intersections des 2 courbes COR : $P(D_1|H_1) = f [P(D_1|H_2)]$ et $P(D_2|H_2) = f [P(D_2|H_1)]$ des 2 droites déduites du *minimax* :

$$P(D_1|H_1) = \frac{\Delta_1}{C_{01} - C_{11}} - \frac{C_{12} - C_{02}}{C_{01} - C_{21}} P(D_1|H_2) \quad \text{et} \quad P(D_2|H_2) = \frac{\Delta_2}{C_{02} - C_{12}} - \frac{C_{21} - C_{01}}{C_{02} - C_{22}} P(D_2|H_1)$$

$$\text{avec : } \Delta_1 = (C_{12} - C_{02}) + \Gamma(C_{21} - C_{12}) \quad \text{et} \quad \Delta_2 = (C_{21} - C_{01}) + \Gamma(C_{12} - C_{21})$$

où : $\Gamma(x)$ représente l'échelon de Heaviside.

TD 8 ANNEXE. Décision (Classification, Reconnaissance des Formes)**I. CAS BI-CLASSE****1. Canal de transmission**

Une source qui émet soit une tension nulle (hypothèse 1 H_1), soit une tension constante $V \neq 0$ (hypothèse 2 H_2), est observée à la sortie $y(t)$ d'un canal de transmission, qui introduit un bruit $n(t)$ additif gaussien, blanc et centré :

$$\begin{array}{ll} H_1 : y(t) = n(t) & H_2 : y(t) = V + n(t) \\ \text{avec : } E[n(t)] = 0 & \text{et : } \text{Var}[n(t)] = \sigma^2 \end{array}$$

Construire le récepteur basé sur l'observation de N valeurs de $y(t)$.

TRAITEMENT DU SIGNAL

PROJETS

- Les énoncés de projet détaillés se trouvent dans le répertoire : CORRIGES\GA\TS\TSPROJET
- Le travail demandé est à programmer sous Mathcad, par groupe de 3 élèves maximum.
- Le travail à rendre (sous la forme d'un fichier zip dénommé avec l'intitulé du sujet et tous les noms des membres du groupe de projet, fichier à déposer sur Arelv2) doit contenir le programme Mathcad avec toutes les données nécessaires, les commentaires et rapport intégrés au document Mathcad.
- Des questions de cours individuelles seront posées lors de la soutenance du projet.

Traitement du Signal - Projets

- *Enoncés de projet détaillés se trouvant dans le répertoire : CORRIGES/GA/TS/TSPROJET*

01. Implementation of discrete-time systems - Parole & image
02. Speech signal compression with FFT
03. Image signal compression with FFT
04. Compression parole-image par ondelettes Daubechies
05. Compression parole-image par filtres formeurs
06. Compression parole-image par prediction lineaire (LPC)
07. Transformee en ondelettes Haar - Compression parole
08. Transformee en ondelettes Haar - Compression image
09. Estimation des formants d'un signal parole par LPC
10. Fractal image compression
11. Segmentation parole en mots (detection activite vocale)
12. Repliement de spectre
13. Transformations 1D et 2D - Separabilite
14. Detection de fin de mots par mesure d'energie
15. Suppression d'echo par transformee cepstrale complexe
16. Estimation des formants d'un signal parole par cepstre
17. Methodes d'inversion de transformee de cepstre reel
18. Projet finance - Estimation par les moindres carres

Bioinformatique

- 1 Analyse de signaux ECG (Electro-CardioGramme) et EEG (Electro-EncéphaloGramme) (analyse FFT et Ondelettes).
-