ESIEE

Cité Descartes
2 boulevard Blaise Pascal
BP 99

93162 NOISY LE GRAND Cedex

Introduction au filtrage adaptatif

I4-TTS 2003

J.-F. Bercher & P. Jardin

1 INTRODUCTION

1.1 Objectifs et déroulement du cours

De façon générale les filtres adaptatifs sont des systèmes appliqués sur des données bruitées pour obtenir une information utile à un certain instant t ,ces systèmes étant mis en œuvre dans trois configurations :

- Le filtrage c'est à dire l'extraction de cette information au temps t à partir des données bruitées mesurées jusqu'au temps t inclus.
- Le lissage qui utilisera aussi les données postérieures au temps t.
- La prédiction qui ne se sert que des données jusqu'au temps t-τ pour déduire l'information qui nous intéresse au temps t.

Dans la première partie de ce cours nous exposerons l'approche statistique du problème (**filtrage de Wiener**) qui suppose la disponibilité de certaines grandeurs statistiques (moyenne et autocorrélation) du signal utile et du bruit. L'approche consiste alors à minimiser la moyenne statistique du carré de l'erreur (EQM ou MSE en anglais) entre l'information désirée et la sortie du filtre.
Ce filtrage de Wiener est inadéquat pour les situations dans lesquelles le signal ou le bruit sont non stationnaires . Dans de telles situations le filtre optimal doit être variable dans le temps. La solution à ce problème est fournie par le filtrage de Kalman.
Le filtrage adaptatif pourra aussi être utilisé dans ce cas. Par rapport au filtrage classique le filtrage adaptatif comporte une mise à jour récursive des paramètres (coefficients) du filtre. L'algorithme part de conditions initiales prédéterminées et modifie de façon récursive les coefficients du filtre pour s'adapter au processus. Si celui-ci est stationnaire l'algorithme doit converger vers la solution optimale de Wiener, sinon il présentera une capacité à suivre des variations des grandeurs statistiques du processus si celles-ci sont suffisamment lentes.
Pour présenter ces techniques adaptatives nous présenterons tout d'abord **l'algorithme du gradient** qui fournit un algorithme récursif de calcul des coefficients du filtre . Nous donnerons ensuite une version dans laquelle les grandeurs statistiques impliquées sont remplacées par des valeurs instantanées,on obtient alors l'algorithme très fréquemment utilisé du **gradient stochastique (LMS)** .Nous re-formulerons le problème en termes **de moindres carrés** et présenterons un des algorithmes récursifs basé sur cette approche :l'algorithme des **moindres carrés récursifs (RLS) (**dont il existe de nombreuses variantes non présentées dans le cadre de ce cours).

1.2 Choix de l'algorithme

Le choix de l'algorithme se fera en fonction des critères suivants :

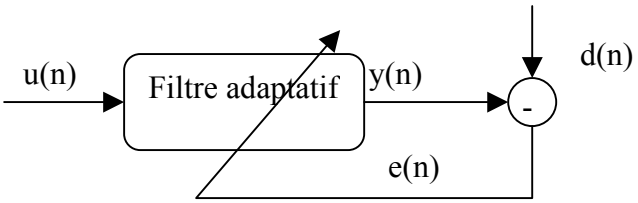
- La rapidité de convergence qui sera le nombre d'itérations nécessaires pour converger « assez près » de la solution optimale de Wiener dans le cas stationnaire.
- La mesure de cette « proximité » entre cette solution optimale et la solution obtenue.
- La capacité de poursuite (tracking) des variations (non stationnarités) du processus. On examinera quels sont les algorithmes vraiment adaptatifs.
- La robustesse au bruit
- La complexité (en nombre de MIPS)
- La structure (se prêtant plus ou moins à l'implémentation en VLSI)
- Les propriétés numériques (stabilité –précision) dans le cas d'une précision limitée sur les données et les coefficients (problèmes d'implémentation en virgule fixe).

Nous ne nous intéresserons dans le cadre de ce cours qu'aux trois premiers critères de choix .
Remarques :

1. La théorie des filtres incluant celle des filtres de Wiener et Kalman a été développée pour des signaux à temps continu et à temps discret mais nous considérerons par la suite uniquement le cas discret où les signaux sont échantillonnés et les filtres sont numériques. Nous nous limiterons de plus aux filtres à réponse impulsionnelle finie (FIR)
2. Le début du cours est écrit pour des signaux complexes. Les formules obtenues étant un peu lourdes, la suite du cours est écrite pour des signaux réels. Vous pouvez donc, lors de la lecture, ignorer les conjugaisons ()* et translater les conjuguées hermitiennes $()^+$ ou $()^H$ en transposées simples $()^T$.

2 FILTRAGE LINEAIRE OPTIMAL – FILTRAGE DE WIENER

Considérons la figure suivante :



Le problème du filtrage optimal de trouver le « meilleur » filtre c'est à dire celui permettant d'obtenir en sortie une réponse *y(n)* la plus « proche » possible d'une réponse désirée *d(n)* lorsque l'entrée est une certaine séquence *u(n)*.

On note $e(n) = y(n) - d(n)$ l'erreur entre la réponse désirée *d(n)* et la sortie *y(n)*. On note également *w(n)* la réponse impulsionnelle du filtre .

Il pourra aussi nous arriver de noter que $y(n) = \hat{d}(n / \underline{X}_n)$ où $\underline{X}_n$ est un vecteur d'état contenant toute l'information utilisée pour prédire d(n) :
$\underline{X}_n$=[u(0),…,u(n),e(0),…,e(n-1),w(0),…,w(M-1),y(0),…y(n-1)].

Le problème consiste donc à rechercher le filtre assurant l'erreur la plus faible *e(n)*, au sens d'une certaine fonction de coût :

$$\underline{\overset{\Delta}{w}} = \arg\min_{\underline{w}} J(e(n))$$

De nombreux choix sont possibles en ce qui concerne la fonction de coût ; par exemple

- erreur quadratique moyenne,
- erreur L_1,
- erreur L_K,
- erreur L_∞,
- $E[f(e_n)]$, où $f(\,)$ est une fonction non linéaire.

Parmi celles-ci, l'erreur quadratique moyenne est la plus utilisée, car elle conduit à des développements mathématiques complets et simples, fournit la solution en fonction des caractéristiques au second ordre des variables aléatoires, caractéristiques qui sont les plus

simples à estimer, et enfin fournit une solution unique. C'est sur l'estimation linéaire en moyenne quadratique que repose le filtrage de Wiener.

2.1 Relations d'orthogonalité - Equation de Wiener-Hopf

La sortie du filtre s'écrit

$$y(n) = \sum_{k=0}^{M-1} w_k^* u(n-k) \tag{0.1}$$

et l'erreur est quant à elle

$$e(n) = d(n) - y(n)$$

Le **filtre de Wiener** est celui qui minimise l'erreur quadratique moyenne (EQM ou MSE en anglais)

$$J = E(|e(n)|^2) \tag{0.2}$$

En introduisant les vecteurs

$\underline{w}^T = [w_0 ... w_{M-1}]$ et $\underline{u}(n)^T = [u(n) ... u(n-M+1)]$

on a alors

$$e(n) = d(n) - \underline{w}^H \underline{u}(n) = d(n) - \underline{u}^T(n)\underline{w}^* \tag{0.3}$$

d'où

$$\begin{aligned} J &= E((d(n) - \underline{w}^H \underline{u}(n))(d^*(n) - \underline{w}^T \underline{u}^*(n))) \\ &= E(|d(n)|^2) - \underline{w}^H E(\underline{u}(n) d^*(n)) - \underline{w}^T E(\underline{u}^*(n) d(n)) + \underline{w}^H E(\underline{u}(n)\underline{u}^H(n))\underline{w} \end{aligned}$$

soit

$$J = \sigma_d^2 - \underline{w}^H \underline{R}_{ud} - \underline{w}^T \underline{R}_{ud}^* + \underline{w}^H \underline{\underline{R}}_{uu} \underline{w} \tag{0.4}$$

avec $\underline{\underline{R}}_{uu} = E(\underline{u}(n)\underline{u}^H(n))$ qui est la matrice d'autocorrélation de l'entrée u(n) . Cette matrice est définie positive , de Toeplitz et à symétrie hermitienne : $\underline{\underline{R}}_{uu} = \underline{\underline{R}}_{uu}^H$

et $\underline{R}_{ud} = E(\underline{u}(n) d^*(n))$ le vecteur d'intercorrélation entre la sortie désirée d(n) et l'entrée u(n).

Le vecteur optimum $\underline{w}^{\Delta}$ est celui qui annule le gradient du critère :

$$\partial J / \partial \underline{w} = \underline{0}$$

En écrivant *J* sous la forme $J = E(e_n e_n^*)$ on a $\frac{\partial J}{\partial \underline{w}} = E(e_n \frac{\partial e_n^*}{\partial \underline{w}}) + E(e_n^* \frac{\partial e_n}{\partial \underline{w}})$

Or (voir note n°2 sur le gradient vectoriel complexe), $\frac{\partial e_n^*}{\partial \underline{w}} = \underline{0}$ et $\frac{\partial e_n}{\partial \underline{w}} = 2\underline{u}(n)$

par conséquent, à l'optimum, on a :

$$E(e^*(n)\underline{u}(n)) = \underline{0} \tag{0.5}$$

C'est le **principe d'orthogonalité** signifiant que toutes les entrées u(n) sont décorrélées de $e^*(n)$.
En développant cette dernière équation on obtient $E(\underline{u}(n)(d^*(n) - \underline{u}^H(n)\underline{w})) = 0$ soit:

$$\underline{\underline{R}}_{uu}\underline{w} = \underline{R}_{ud} \tag{0.6}$$

Cette relation est appelée Formule de Wiener ou **équation de Wiener Hopf**
La solution est le filtre optimal $\overset{\Delta}{\underline{w}}$:

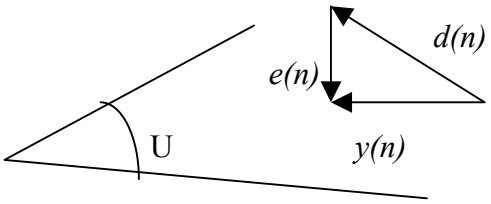
$$\boxed{\overset{\Delta}{\underline{w}} = (\underline{\underline{R}}_{uu})^{-1}\underline{R}_{ud}} \tag{0.7}$$

Pour ce vecteur $\overset{\Delta}{\underline{w}}$ optimal on obtient à partir de (0.4)et (0.6) l'Erreur Quadratique Moyenne minimale :

$$J_{\min} = \sigma_d^2 - \overset{\Delta}{\underline{w}}^H \underline{\underline{R}}_{uu}\overset{\Delta}{\underline{w}} = \sigma_d^2 - \sigma_{\hat{d}}^2 \tag{0.8}$$

si on note : $\hat{d} = \overset{\Delta}{\underline{w}}^H \underline{u}(n)$ le signal filtré optimal et $\sigma_{\hat{d}}^2$ la variance de ce signal
Cette relation (1.9) montre que pour le filtre optimal , l'EQM est la différence entre la variance du signal désiré et celle de l'estimée de ce signal produite par le filtre.
Notes :
1. Interprétation géométrique pour la relation (0.8) : Théorème de projection

Schéma



Comme e(n) est orthogonale à l'hyperplan généré par les vecteurs {u(n) ,…,u(n-M+1)},on a le théorème de Pythagore

$$E(|d(n)|^2) = E(|y(n)|^2) + E(|e(n)|^2) \text{ (équivalent de (0.8))}$$

y(n) est la projection de *d(n)* sur cet hyperplan .
2. Gradient vectoriel complexe
On utilise habituellement le gradient dans le cas réel. Dans le cas complexe, on définit la dérivation par : $\frac{\partial}{\partial z} = (\frac{\partial}{\partial z_R} + j\frac{\partial}{\partial z_I})$

Cette définition conduit à la propriété peu naturelle : $\frac{\partial z}{\partial z} = 0$ et $\frac{\partial z^*}{\partial z} = 2$

La différentiation par rapport à un vecteur étend les définitions précédentes :

$$\frac{\partial}{\partial \underline{w}} = \begin{vmatrix} \frac{\partial}{\partial w_{0R}} + j\frac{\partial}{\partial w_{0I}} \\ \frac{\partial}{\partial w_{1R}} + j\frac{\partial}{\partial w_{1I}} \\ . \\ . \\ \frac{\partial}{\partial w_{M-1R}} + j\frac{\partial}{\partial w_{M-1I}} \end{vmatrix}$$

Avec ces définitions et en dérivant l'EQM donnée par l'équation (0.4) on obtient bien l'équation de Wiener Hopf

2.2 Applications

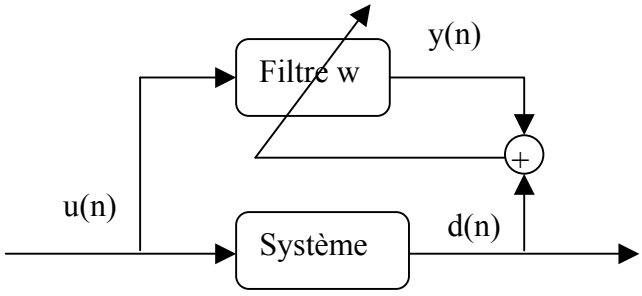
Le filtrage de Wiener adaptatif est un outil puissant en traitement du signal , communications numériques et contrôle automatique . Les applications sont diverses mais présentent toujours les caractéristiques exposées plus haut : on dispose d'une entrée u ainsi que de la réponse désirée d et l'erreur e entre la sortie y et d sert à contrôler (adapter) les valeurs des coefficients du filtre w. Ce qui différencie essentiellement les applications provient de la façon de définir la réponse désirée d. On peut distinguer quatre grandes classes d'applications :

- L'identification de systèmes
- La prédiction
- La modélisation inverse
- L'annulation d'interférences

Nous donnons ci après les schémas correspondant à ces quatre classes .

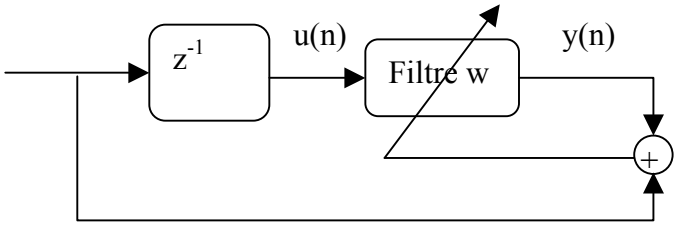
2.2.1 Identification de Systèmes :

d(n) est la sortie du système que l'on souhaite identifier



2.2.2 Prédiction :

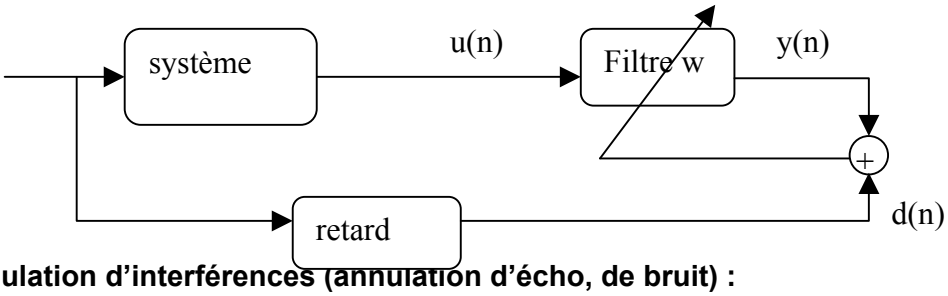
d(n) est le signal à l'instant n et y(n) le signal prédit à partir du signal aux instants précédents.



d(n)

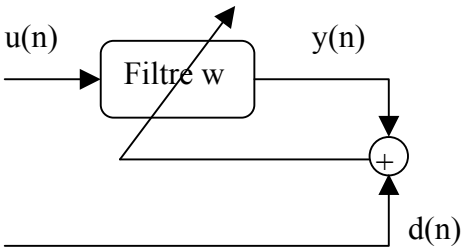
2.2.3 Modélisation inverse (égalisation, déconvolution) :

d(n) est l'entrée (retardée) du système que l'on cherche à « inverser ».



2.2.4 Annulation d'interférences (annulation d'écho, de bruit) :

d(n) est un signal primaire qui comporte les interférences à annuler. u(n) est le signal de référence dénué (ou presque) d'information et obtenu par un capteur proche de celui qui fournit d(n).



3 ALGORITHME DU GRADIENT

L'équation de Wiener Hopf (0.6) qui permet de calculer le filtre de Wiener optimal conduit à résoudre un système de M équations à M inconnues :

$\underline{\underline{R}}_{uu}\underline{w} = \underline{R}_{ud}$

Une méthode d'inversion directe est en $O(M^3)$, ce qui est très coûteux. Il peut être préférable de résoudre ce système par une méthode itérative, notamment en se souvenant que la fonction de coût est quadratique, ce qui entraîne que le minimum est unique.

La méthode du gradient consiste à :

- choisir un vecteur initial de coefficients $\underline{w}(0)$
- ayant un vecteur candidat $\underline{w}(n)$ à l'étape *n*, obtenir un meilleur candidat en incrémentant $\underline{w}(n)$ dans la direction opposée au gradient du coût J .

$$\underline{w}(n+1) = \underline{w}(n) - \frac{1}{2}\mu_n \underline{\nabla} J\Big|_{\underline{w}=\underline{w}(n)} \qquad (0.9)$$

La séquence $\{\mu_n\}$ est une série de coefficients appelés pas d'adaptation.
On distingue l'algorithme à pas d'adaptation constant $\mu_n=\mu$ et les algorithmes à pas décroissant.
Le gradient est égal à : $\underline{\nabla} J\Big|_{\underline{w}=\underline{w}(n)} = 2\underline{\underline{R}}_{uu}\underline{w}(n) - 2\underline{R}_{du}$

La forme explicite de l'algorithme est donc

$$\boxed{\underline{w}(n+1) = \underline{w}(n) + \mu_n(\underline{R}_{du} - \underline{\underline{R}}_{uu}\underline{w}(n))} \qquad (0.10)$$

Notons que cette forme est déterministe et nécessite la connaissance des grandeurs statistiques $\underline{\underline{R}}_{uu}$ et $\underline{R}_{du}$

Le paramètre μ_n contrôle l'importance de la correction apportée au vecteur de coefficients $\underline{w}$ lors de la $n^{ième}$ itération. Le paragraphe suivant porte sur l'étude de la convergence de l'algorithme c'est à dire sur son aptitude à être stable et à tendre vers la solution optimale $\underline{\overset{\Delta}{w}}$.
Les deux facteurs influençant à priori cette convergence sont le pas d'adaptation μ_n et la matrice d'autocorrélation $\underline{\underline{R}}_{uu}$.

3.1 Convergence de l'algorithme du gradient

Nous commençons en étudiant le comportement du vecteur d'erreur entre le vecteur optimal $\underline{\overset{\Delta}{w}}$ et le vecteur déterminé à la $n^{ième}$ itération de l'algorithme $\underline{w}(n)$.

On note $\underline{v}(n) = \underline{w}(n) - \underline{\overset{\Delta}{w}}$
A partir des équations (0.8) et (0.10), on obtient

$$\underline{w}(n+1) = \underline{w}(n) + \mu_n(\underline{\underline{R}}_{uu}(\underline{\overset{\Delta}{w}} - \underline{w}(n)))$$

D'où :

$$\underline{v}(n+1) = (\underline{\underline{I}} - \mu_n\underline{\underline{R}}_{uu})\underline{v}(n) \qquad (0.11)$$

Introduisons maintenant la décomposition propre de la matrice d'autocorrélation :

$\underline{\underline{R}}_{uu} = \underline{\underline{Q}}\,\underline{\underline{\Lambda}}\,\underline{\underline{Q}}^H$, où Q est unitaire.

Soit en multipliant les membres de l'équation (0.11) par $\underline{\underline{Q}}^H$ et en notant $\underline{r}(n) = \underline{\underline{Q}}^H \underline{v}(n)$:

$$\underline{r}(n+1) = (\underline{\underline{I}} - \mu_n \underline{\underline{\Lambda}})\ \underline{r}(n) \qquad (0.12)$$

De cette façon, on fait apparaître les modes propres de l'algorithme.

L'algorithme converge si $\underline{r}(n) \to 0$, ce qui entraîne bien que $\underline{v}(n) \to 0$, soit $\underline{w}(n) \to \overset{\Delta}{\underline{w}}$

Le problème est donc de donner les conditions qui assurent cette convergence, de quantifier la rapidité de convergence, et de choisir le pas d'adaptation.

3.1.1 Conditions de convergence pour un pas μ constant

L'équation (0.12) nous donne par récurrence : $\underline{r}(n) = (\underline{\underline{I}} - \mu \underline{\underline{\Lambda}})^n \underline{r}(0)$

Chacune des composantes s'exprime alors sous la forme $r_k(n) = (1 - \mu\lambda_k)^n r_k(0)$
où λ_k est la $k^{\text{ième}}$ valeur propre.

Les différentes composantes convergent à 0 pourvu que $\boxed{|1 - \mu\lambda_k| < 1 \quad \forall k \in \{0,1,...,M-1\}}$

Nous en déduisons alors **la condition de convergence de l'algorithme du gradient :**
(en notant λ_{max} la valeur propre maximale)

$$\boxed{0 < \mu < 2/\lambda_{\max}} \qquad (0.13)$$

3.1.2 Rapidité de convergence :

Nous nous intéressons maintenant à la vitesse de convergence et choisissons de suivre l'évolution du critère J à minimiser (nous aurions pu continuer de nous intéresser à la convergence des coefficients vers l'optimum):

D'après l'équation (0.4) et avec $\underline{\underline{R}}_{uu}\,\overset{\Delta}{\underline{w}} = \underline{R}_{ud}$, on peut écrire

$$J = \sigma_d^2 - \underline{w}^H \underline{\underline{R}}_{uu}\,\overset{\Delta}{\underline{w}} - \overset{\Delta}{\underline{w}}^H \underline{\underline{R}}_{uu}\underline{w} + \underline{w}^H \underline{\underline{R}}_{uu}\underline{w} = \sigma_d^2 - \overset{\Delta}{\underline{w}}^H \underline{\underline{R}}_{uu}\,\overset{\Delta}{\underline{w}} + (\underline{w} - \overset{\Delta}{\underline{w}})^H \underline{\underline{R}}_{uu}(\underline{w} - \overset{\Delta}{\underline{w}})$$

D'après l'équation (0.8), on a donc

$$J = J_{\min} + (\underline{w} - \overset{\Delta}{\underline{w}})^H \underline{\underline{R}}_{uu}(\underline{w} - \overset{\Delta}{\underline{w}})$$

En utilisant la décomposition propre de la matrice d'autocorrélation et en utilisant de nouveau $\underline{r}(n) = \underline{\underline{Q}}^H(\underline{w}(n) - \overset{\Delta}{\underline{w}}(n))$ on obtient $J = J_{\min} + \underline{r}^H \underline{\underline{\Lambda}}\,\underline{r}$

En utilisant maintenant $\underline{r}(n) = (\underline{\underline{I}} - \mu \underline{\underline{\Lambda}})^n \underline{r}(0)$ nous pouvons
écrire $J = J_{\min} + \underline{r}(0)^H (\underline{\underline{I}} - \mu \underline{\underline{\Lambda}})^n \underline{\underline{\Lambda}} (\underline{\underline{I}} - \mu \underline{\underline{\Lambda}})^n \underline{r}(0)$

Toutes les matrices étant diagonales,

$$J(n) = J_{\min} + \sum_{k=0}^{M-1} \lambda_k (1-\mu\lambda_k)^{2n} \mid r_k(0) \mid^2 \qquad (0.14)$$

Lorsque l'algorithme est convergent c'est à dire lorsque : $0 < \mu < 2/\lambda_{\max}$,

$\lim_{n\to\infty} J(n) = J_{\min}$ quelque soient les conditions initiales $\underline{w}(0)$

La courbe obtenue en traçant J(n) en fonction du nombre d'itérations n est appelée courbe d'apprentissage. D'après l'équation (0.14) cette courbe consiste en une somme d'exponentielles décroissantes, chacune d'elle correspondant à un mode propre de l'algorithme. La vitesse de convergence du mode k est liée à $(1-\mu\lambda_k)^{2n}$.

Le mode le plus lent est lié à la valeur propre la plus petite et le mode le plus rapide est lié à la valeur propre la plus grande.

A une valeur propre, on associe une constante de temps, telle que
$|1-\mu\lambda_k|^2 = exp(-1/\tau_k)$

La constante de temps de l'erreur quadratique est alors

$$\frac{-1}{2\log(|1-\mu\lambda_{\max}|)} \le \tau \le \frac{-1}{2\log(|1-\mu\lambda_{\min}|)} \qquad (0.15)$$

Ceci montre que la convergence est d'autant plus lente que le pas est faible.

remarque : pour $\mu << 1$ on peut approcher la constante de temps par $\tau_k = 1/2\mu\lambda_k$.

3.1.3 Pas optimal

La rapidité de convergence est gouvernée, pour chacun des modes, par $\delta_k = |1-\mu\lambda_k|$

La solution optimale consiste à minimiser le plus grand des δ_k : il s'agit donc d'un problème de type minimax : $\mu_{opt} = \arg\min_{\mu} \max_{k} (|1-\mu\lambda_k|)$

Pour $\mu > \mu_{opt}$ $\quad 1-\mu\lambda_k < 1-\mu_{opt}\lambda_k \quad \Rightarrow \quad \max(|1-\mu\lambda_k|) = \mu\lambda_{\max} - 1$

Pour $\mu < \mu_{opt}$ $\quad 1-\mu\lambda_k > 1-\mu_{opt}\lambda_k \quad \Rightarrow \quad \max(|1-\mu\lambda_k|) = 1-\mu\lambda_{\min}$

Donc pour $\mu = \mu_{opt}$ $\quad 1-\mu\lambda_{\min} = \mu\lambda_{\max} - 1$

on a par conséquent :

$$\mu_{opt} = \frac{2}{\lambda_{\min} + \lambda_{\max}} \qquad (0.16)$$

On peut noter que ce pas appartient bien au domaine des pas qui assurent la convergence, c'est à dire $\mu \in [0, 2/\lambda_{\max}]$

Cependant, calculer les valeurs propres de la matrice de corrélation est aussi compliqué que d'inverser la matrice. Le choix du pas fait donc perdre l'avantage de prendre un algorithme « économique ».
Une mesure de compromis consiste à prendre :

$$\tilde{\mu} = \frac{2}{\sum_{k=1}^{M} \lambda_k} = \frac{2}{Trace(\underline{\underline{R}}_{uu})}$$

La trace de $\underline{\underline{R}}_{uu}$ est simple à calculer (somme des éléments sur la diagonale principale). La matrice étant de Toeplitz, ou approximativement, on peut aussi voir que
$Trace\ (\underline{\underline{R}}_{uu}) = M\ r_{uu}(0)$
et $r_{uu}(0)$ est la puissance de u(n) : $r_{uu}(0) = E(|u_n|^2)$

On constate facilement que $\tilde{\mu} < \mu_{\max}$ et, avec un peu de chance, $\tilde{\mu}$ n'est pas très loin de μ_{opt}.

3.1.4 Pas optimal et conditionnement de la matrice d'autocorrélation

D'après $\underline{r}(n) = (\underline{\underline{I}} - \mu\underline{\underline{\Lambda}})^n \underline{r}(0)$,la convergence de $\underline{r}(n)$ est gouvernée par la valeur propre la plus grande de $\underline{\underline{A}} = \underline{\underline{I}} - \mu\underline{\underline{\Lambda}}$

On peut remarquer que pour *n* assez grand on a : $\underline{r}(n) \simeq \rho(\underline{\underline{A}})^n \underline{r}(0)$, où $\rho(\underline{\underline{A}})$ est le rayon spectral (la plus grande valeur propre) de $\underline{\underline{A}}$.

Pour $\mu = \mu_{opt}$, $\rho(\underline{\underline{A}}) = 1 - \mu\lambda_{\min} = \frac{\lambda_{\max} - \lambda_{\min}}{\lambda_{\max} + \lambda_{\min}} = \frac{C(\underline{\underline{R}}) - 1}{C(\underline{\underline{R}}) + 1}$

où $C(\underline{\underline{R}}) = \frac{\lambda_{\max}}{\lambda_{\min}}$ est le conditionnement de $\underline{\underline{R}}$.

Ce résultat montre que la vitesse de convergence est directement liée au conditionnement de $\underline{\underline{R}}$:

Si $C(\underline{\underline{R}})$ est proche de 1, $\rho(\underline{\underline{A}})$ est proche de 0, et la vitesse de convergence est très élevée. Au contraire, si $C(\underline{\underline{R}}) \to \infty$, $\rho(\underline{\underline{A}}) \to 1$, et l'algorithme est très lent à converger.

En d'autres termes, l'algorithme converge d'autant plus lentement que la matrice de covariance est mal conditionnée.

3.2 Autre présentation de la méthode du gradient

Cette autre présentation s'appuie sur le résultat suivant :

$$\forall \mu < \mu_{\max} \quad , \underline{\underline{R}}^{-1} = \mu \sum_{k=0}^{\infty} (\underline{\underline{I}} - \mu\underline{\underline{R}})^k$$

(on peut montrer cette égalité de la même façon que pour établir la somme d'une suite géométrique convergente).

Dans ce cas,

$$\overset{\Delta}{\underline{w}} = \underline{\underline{R}}_{uu}^{-1}\underline{R}_{du} = \mu\sum_{k=0}^{\infty}(\underline{\underline{I}} - \mu\underline{\underline{R}})^k\,\underline{R}_{du}$$

En posant alors

$$\underline{w}(n) = \mu\sum_{k=0}^{n}(\underline{\underline{I}} - \mu\underline{\underline{R}})^k\,\underline{R}_{du}$$

on a :

$$\underline{w}(n+1) = \mu\sum_{k=0}^{n+1}(\underline{\underline{I}} - \mu\underline{\underline{R}})^k\,\underline{R}_{du} = (\underline{\underline{I}} - \mu\underline{\underline{R}})[\mu\sum_{k=0}^{n}(\underline{\underline{I}} - \mu\underline{\underline{R}})^k\,\underline{R}_{du}] + \mu(\underline{\underline{I}} - \mu\underline{\underline{R}})^0\,\underline{R}_{du}$$

$$= (\underline{\underline{I}} - \mu\underline{\underline{R}})\underline{w}(n) + \mu\underline{R}_{du}$$

soit $\boxed{\underline{w}(n+1) = \underline{w}(n) + \mu_n(\underline{R}_{du} - \underline{\underline{R}}_{uu}\underline{w}(n))}$

On retrouve donc l'algorithme du gradient, qui, vu comme ceci, consiste à inverser la matrice $\underline{\underline{R}}_{uu}$ à l'aide du développement en série de $\underline{\underline{R}}_{uu}^{-1}$.

4 ALGORITHME DU GRADIENT STOCHASTIQUE

L'algorithme du gradient stochastique, ou plus exactement la famille des algorithmes de gradient stochastique, consiste à remplacer le gradient $\underline{\nabla} J\big|_{\underline{w}=\underline{w}(n)} = 2\underline{\underline{R}}_{uu}\underline{w}(n) - 2\underline{R}_{du}$, quantité déterministe, exacte, par une approximation du gradient calculée à partir des données disponibles. Les données étant considérées comme aléatoires, le gradient estimé devient lui aussi une quantité aléatoire et l'algorithme devient stochastique. Parmi cette famille d'algorithmes, le plus connu est l'algorithme LMS (Least Mean Square).

Le problème majeur dans le gradient déterministe est que $\underline{\underline{R}}_{uu}$ et $\underline{R}_{du}$ sont évidemment inconnus. On approchera donc ces grandeurs déterministes par des estimées $\hat{\underline{\underline{R}}}_{uu}(n)$ et $\hat{\underline{R}}_{du}(n)$ à l'instant *n*.
Dans le cas du LMS, on choisit les estimées les plus simples possibles, à savoir :
$\hat{\underline{\underline{R}}}_{uu}(n) = \underline{u}(n)\underline{u}^H(n)$ et $\hat{\underline{R}}_{du}(n) = d(n)\underline{u}(n)$
Ce sont simplement les estimées instantanées des corrélations.
(Exercice : montrer que ces estimées sont non biaisées.)

La relation permettant la mise à jour du filtre $\underline{w}$ est ainsi maintenant

$$\boxed{\underline{w}(n+1) = \underline{w}(n) + \mu\,\underline{u}(n)(d^*(n) - \underline{u}^H(n)\underline{w}(n))} \qquad (0.17)$$

Remarque : il est immédiat de montrer que cette approche est celle qui consiste à prendre l'algorithme du gradient sur le critère instantané :minimiser $|e(n)|^2$ (au lieu du critère statistique : minimiser $E(|e(n)|^2)$).

L'algorithme peut en fait se décomposer en
- calcul de la sortie du filtre : $y(n) = \underline{w}^H(n)\underline{u}(n)$
- calcul de l'erreur d'estimation : $e(n)=d(n)-y(n)$
- mise à jour : $\underline{w}(n+1) = \underline{w}(n) + \mu e^*(n)\underline{u}(n)$

Au temps *n*, on calcule donc la sortie du filtre et l'erreur d'estimation, en utilisant le filtre courant $\underline{w}(n)$.
Le filtre est ensuite mis à jour en ajoutant au filtre courant, le terme correctif $\mu e^*(n)\underline{u}(n)$

On remarquera que $\underline{w}(n)$ est maintenant un processus aléatoire, ceci signifie que pour des jeux de données différents, les trajectoires obtenues seront différentes.

On peut s'intéresser à la convergence de la trajectoire moyenne $E(\underline{w}(n))$ et étudier:

$$E(\underline{w}(n+1)) = E(\underline{w}(n)) + \mu E(\underline{u}(n)(d^*(n) - \underline{u}^H(n)\underline{w}(n)))$$

Si on suppose que $\underline{w}(n)$ est indépendant de $\underline{u}(n)$ (ce qui est très faux) on a :
$E(\underline{u}(n)\underline{u}^H(n)\underline{w}(n)) = \underline{\underline{R}}_{uu}E(\underline{w}(n))$ et la trajectoire moyenne suit l'algorithme du gradient déterministe .

Dans les autres cas (c'est-à-dire de façon réaliste), on n'a pas
$E(\underline{u}(n)\underline{u}^H(n)\underline{w}(n)) = \underline{\underline{R}}_{uu}(n)\underline{w}(n)$ car $\underline{w}(n)$ a toutes les chances d'être corrélé à $\underline{u}(n)$.

La trajectoire moyenne n'est donc pas nécessairement celle du gradient déterministe. On peut cependant espérer qu'elle en soit proche, et que le caractère aléatoire se traduise par des fluctuations autour de cette valeur moyenne.

Comme les estimées instantanées sont très variables, on pourrait penser que l'algorithme LMS a des performances très pauvres. Cependant, comme l'algorithme est par nature récursif, il moyenne ces estimées au cours des itérations, ce qui permet d'obtenir des performances acceptables.

4.1 Convergence et stabilité

On étudie depuis des dizaines d'années la convergence en s'appuyant sur l'hypothèse d'indépendance entre $\underline{u}(n)$ et $\underline{w}(n)$. Rappelons que cette hypothèse est atrocement fausse. Les résultats corrects (très techniques) sont établis dans le livre d'O. Macchi (*).
Dans ce cas (indépendance entre $\underline{u}(n)$ et $\underline{w}(n)$) si on s'intéresse à la convergence de la trajectoire moyenne $E(\underline{w}(n))$ on a simplement : $E(\underline{w}(n+1)) = E(\underline{w}(n)) + \mu(\underline{R}_{du} - \underline{\underline{R}}_{uu}E(\underline{w}(n))$
Ceci étant simplement l'algorithme du gradient déterministe, on aura convergence si $\mu<2/\lambda_{max}$ (pour un pas fixe).

(*) O. Macchi, « Adaptive Processing : the LMS approach with applications in transmission », Wiley, New York, 1995.

Une étude plus précise (en moyenne quadratique), mais qui repose également sur des hypothèses contestables conduit à la condition plus contraignante : $\mu < \dfrac{2}{\alpha\, Trace(\underline{\underline{R}}_{uu})}$
où l'on pourra estimer la trace comme $Mr_{uu}(0)$ si M est la dimension de la matrice. En pratique, on prendra α de l'ordre de 2 ou 3.

4.1.1 Comportement de l'erreur

L'erreur définie par $e(n) = d(n) - \underline{u}^T(n)\underline{w}$ peut être exprimée en fonction de l'erreur optimale obtenue à partir du filtre optimal $\overset{\Delta}{e}(n) = d(n) - \underline{u}^T(n)\overset{\Delta}{\underline{w}}$:

$$e(n) = d(n) - \underline{u}^T(n)\overset{\Delta}{\underline{w}} - \underline{u}^T(n)(\underline{w}(n) - \overset{\Delta}{\underline{w}}) = \overset{\Delta}{e}(n) - \underline{u}^T(n)\underline{v}(n)$$

où $\underline{v}(n)$ est l'écart ou filtre optimal.
Cet écart vérifie l'équation récursive

$$\underline{v}(n+1) = (\underline{\underline{I}} - \mu\underline{u}(n)\underline{u}^T(n))\underline{v}(n) + \mu\overset{\Delta}{e}(n)\underline{u}(n)$$

On peut montrer que la solution s'écrit sous la forme :

$\underline{v}(n+1) = \underline{\underline{Q}}_n \underline{v}(0) + \underline{\Lambda}_{n+1}$

avec $\underline{\underline{Q}}_n = \prod_{i=0}^{n} (\underline{\underline{I}} - \mu \underline{u}(j)\underline{u}^T(j))$

et $\underline{\Lambda}$ qui suit: $\underline{\Lambda}_{n+1} = (\underline{\underline{I}} - \mu \underline{u}(n)\underline{u}^T(n))\,\underline{\Lambda}_n + \mu \overset{\Delta}{e}(n)\underline{u}(n)$

Cette relation fait apparaître deux termes :

- $\underline{\underline{Q}}_n \underline{v}(0)$ représente la contribution des conditions initiales et correspond à une réponse transitoire
- $\underline{\Lambda}_{n+1}$ résulte de l'excitation permanente et représente une fluctuation autour de la solution optimale.

On note alors : $\underline{v}(n+1) = \underline{v}(n+1)^{tr} + \underline{v}(n+1)^{fl}$

L'erreur d'estimation devient alors :

$$e(n) = \overset{\Delta}{e}(n) - \underline{u}^T(n)\underline{v}(n)^{tr} - \underline{u}^T(n)\underline{v}(n)^{fl}$$

On montre que le transitoire tend vers 0 lorsque n tend vers l'infini presque sûrement.
L'erreur moyenne quadratique, lorsque n tend vers l'infini est alors :

$$J = E(\overset{\Delta}{e}(n)^2) + E(|\,\underline{u}^T(n)\underline{v}(n)^{fl}\,|^2) = J_{\min} + E(e^{fl}(n)^2) \quad (0.18)$$

L'erreur quadratique liée à la fluctuation est bornée :

$$E(e^{fl}(n)^2) \le \mu J_{\min} \frac{Tr(\underline{\underline{R}}_{uu})}{2 - \mu\alpha Tr(\underline{\underline{R}}_{uu})} \text{ avec } \alpha \ge 1$$

Notons bien que cette fluctuation provient du fait que la trajectoire est aléatoire, et que c'est la moyenne de $\underline{w}(n)$ qui converge vers $\overset{\Delta}{\underline{w}}$.

En effet, si à l'étape n, n étant aussi grand que l'on veut, $\underline{w}(n) = \overset{\Delta}{\underline{w}}$, alors, à l'étape $(n+1)$:

$\underline{w}(n+1) = \overset{\Delta}{w} + \mu \underline{u}(n)(d(n) - \underline{u}^T(n)\overset{\Delta}{w})$

qui est différent de $\overset{\Delta}{\underline{w}}$, sauf cas très particulier du fait que l'excitation est aléatoire.

De ces résultats, on déduit que :

- plus μ est grand, mais vérifiant la condition de stabilité, plus la convergence est rapide, mais plus la variance résiduelle est importante, ce qui se traduit par des fluctuations importantes autour de la trajectoire moyenne.
- Pour de faibles valeurs de μ, la convergence est lente, mais l'erreur résiduelle a une faible variance, c'est-à-dire que l'on a une trajectoire « presque » déterministe.
- La convergence de l'algorithme requiert donc un compromis entre vitesse et fidélité.

4.1.2 Convergence de l'algorithme à pas variable

Afin d'accélérer la convergence, il est intéressant de faire varier le pas d'adaptation μ :
- μ doit être grand au départ, lorsque l'on est loin de l'optimum
- μ doit être faible lorsque l'on se retrouve au voisinage de l'optimum.

On peut alors utiliser une séquence $\{\mu_n\}$de pas variable.

Dans ce cas, l'algorithme converge en moyenne quadratique , c'est à dire que
$E(|\underline{w}(n)-\overset{\Delta}{\underline{w}}|^2)\xrightarrow[n\to\infty]{}0$ si :(conditions suffisantes de Robbin et Monroe)
- $\mu_n > 0$ quelque soit n
- $\Sigma\mu_n$ diverge
- $\Sigma\mu^2_n$ converge

4.2 Poursuite – Adaptativité

L'algorithme présenté jusqu'ici a été présenté sous l'angle d'une implantation récursive d'un filtre de Wiener, pour lequel on a implicitement supposé les signaux stationnaires. On s'est alors intéressé à la convergence vers la solution $\overset{\Delta}{\underline{w}}$.

A partir de l'implantation récursive, il est également possible de voir l'algorithme comme un algorithme adaptatif, c'est-à-dire capable de suivre des modifications lentes, en permanence, du filtre optimal.
Celui-ci est alors dépendant du temps : $\overset{\Delta}{\underline{w}}\to\overset{\Delta}{\underline{w}}(n)$

On peut tout de suite noter que si l'on désire que le filtre soit adaptatif, on ne peut pas le laisser converger en prenant une séquence $\{\mu_n\}$ avec $\mu_n\to 0$ lorsque $n\to\infty$: l'algorithme doit conserver une « capacité de réaction » permettant la poursuite des non-stationnarités.

On considère donc que $\overset{\Delta}{\underline{w}}(n)$ évolue maintenant suivant une marche aléatoire :
$\overset{\Delta}{\underline{w}}(n)=\overset{\Delta}{\underline{w}}(n-1)+\underline{\varepsilon}(n)$ où $\underline{\varepsilon}(n)$ est un bruit vectoriel, de matrice de covariance $\underline{\underline{\Gamma}}$.

On définit alors un nouvel écart au filtre optimal : $\underline{v}(n)=(\underline{w}(n)-\overset{\Delta}{\underline{w}}(n))$
on a toujours $e(n)=d(n)-\underline{u}^T(n)\overset{\Delta}{\underline{w}}(n)-\underline{u}^T(n)(\underline{w}(n)-\overset{\Delta}{\underline{w}}(n))=\overset{\Delta}{e}(n)-\underline{u}^T(n)\underline{v}(n)$

En résolvant récursivement, on a : $\underline{v}(n+1)=\underline{\underline{Q}}_n\underline{v}(0)+\underline{\Lambda}_{n+1}+\underline{E}_{n+1}$

où $\underline{E}_n$ est la suite aléatoire « supplémentaire », correspondant à l'ajout de $\underline{\varepsilon}(n)$ dans le modèle et qui obéit à l'équation récursive : $\underline{E}_{n+1}=(\underline{\underline{I}}-\mu\underline{u}(n)\underline{u}^T(n))\underline{E}_n-\underline{\varepsilon}(n+1)$

Il s'en suit que $e(n)=\overset{\Delta}{e}(n)+e^{tr}(n)+e^{fl}(n)+e^r(n)$, et une fois disparu le transitoire ,

$$J=J_{\min}+E(e^{fl}(n)^2)+E(e^r(n)^2) \qquad (0.19)$$

avec $E(e^r(n)^2)=\dfrac{Tr(\underline{\underline{\Gamma}})}{\mu(2-\mu\alpha Tr(\underline{\underline{R}}_{uu}))}$ avec $\alpha\geq 1$

on rappelle que $E(e^{fl}(n)^2) \le \mu J_{\min} \frac{Tr(\underline{\underline{R}}_{uu})}{2-\mu\alpha Tr(\underline{\underline{R}}_{uu})}$ avec $\alpha \ge 1$

La minimisation de l'erreur quadratique conduit alors à choisir un pas optimal qui effectue un compromis entre ces deux erreurs.

4.3 Algorithme du signe.

Parmi les variantes simplifiées de l'algorithme LMS, l'algorithme du signe est intéressant : On remplace simplement l'erreur d'estimation *e(n)* par son signe, ce qui conduit à la récurrence :

$$\underline{w}(n+1) = \underline{w}(n) + \mu \underline{u}(n) signe(e(n)) \qquad (0.20)$$

On évite ainsi d'effectuer la multiplication dans la relation d'actualisation des coefficients, ce qui conduit ainsi à une complexité de M multiplications par itération.

Il est à noter qu'en dépit de cette simplification extraordinaire, la convergence reste convenable dans beaucoup d'applications, et en particulier en égalisation adaptative.

4.4 Algorithme LMS normalisé

Dans la récurrence de l'algorithme LMS standard (avec des coefficients de filtre réels)
$\underline{w}(n+1) = \underline{w}(n) + \mu \underline{u}(n)(d(n) - \underline{u}^T(n)\underline{w}(n))$
la dynamique des fluctuations dépend de la dynamique du signal d'entrée $\underline{u}(n)$. On dit qu'on a une amplification du bruit par le gradient.

Reprenons l'équation du gradient déterministe : $\underline{w}(n+1) = \underline{w}(n) - \frac{1}{2}\mu_n \underline{\nabla} J\big|_{\underline{w}=\underline{w}(n)}$

avec $\underline{\nabla} J\big|_{\underline{w}=\underline{w}(n)} = 2\underline{\underline{R}}_{uu}\underline{w}(n) - 2\underline{R}_{du} = \underline{\nabla}_n$.

A l'instant n+1 , on a

$$J(n+1) = E((e(n+1))^2)$$

avec $e(n+1) = d(n+1) - \underline{w}^T(n+1)\underline{u}(n+1) = d(n+1) - \underline{w}^T(n)\underline{u}(n+1) + \frac{1}{2}\mu_n \underline{\nabla}^T(n)\underline{u}(n+1)$.

Si on cherche le pas optimal μ_n qui minimise J(n+1), on doit avoir :

$$\frac{\partial J(n+1)}{\partial \mu_n} = 2E(e(n+1)\frac{\partial e(n+1)}{\partial \mu_n}) = 0$$

soit :

$$E(\underline{\nabla}_n^T d(n+1)\underline{u}_{n+1} - \underline{w}_n^T \underline{u}_{n+1}\underline{u}_{n+1}^T \underline{\nabla}_n + \frac{1}{2}\mu_n \nabla_n^T \underline{u}_{n+1}\underline{u}_{n+1}^T \underline{\nabla}_n) = 0$$

Dans l'algorithme du gradient déterministe $\underline{\nabla}_n$ est certain , la relation précédente s'écrit donc :

$$\underline{\nabla}_n^T \underline{R}_{du} - \underline{w}_n^T \underline{\underline{R}}_{uu}\underline{\nabla}_n + \frac{1}{2}\mu_n \nabla_n^T \underline{\underline{R}}_{uu}\underline{\nabla}_n = 0$$

Soit finalement : $\overset{\Delta}{\mu}_n = \frac{2\left(\underline{w}_n^T \underline{\underline{R}}_{uu}\underline{\nabla}_n - \underline{\nabla}_n^T \underline{R}_{du}\right)}{\nabla_n^T \underline{\underline{R}}_{uu}\underline{\nabla}_n}$

L'estimée instantanée de $\underline{\underline{R}}_{uu}$ est $\hat{\underline{\underline{R}}}_{uu}(n) = \underline{u}(n)\underline{u}^T(n)$,celle de $\underline{R}_{du}$ est $d(n)\underline{u}_n$ et celle de $\underline{\nabla}_n$ est $\hat{\underline{\nabla}}_n = 2(d(n)\underline{u}_n - \underline{u}_n\underline{u}_n^T\underline{w}_n)$.

En remplaçant $\underline{\underline{R}}_{uu}$, $\underline{R}_{du}$ et $\underline{\nabla}_n$ par leurs estimées respectives il vient (à faire en exercice) :

$$\boxed{\overset{\Delta}{\mu}_n = \frac{1}{\underline{u}_n^T \underline{u}_n}} \qquad (0.21)$$

Ce qui revient à prendre un pas inversement proportionnel à la puissance « instantanée » du signal d'entrée.

5 FORMULATION DU FILTRAGE DE WIENER DANS LE CADRE DES MOINDRES CARRES.

Nous avons résolu précédemment le problème du filtrage de Wiener en nous appuyant sur la fonction de coût déterministe

$J = E(|e(n)|^2)$

L'inconvénient de cette approche est que les grandeurs statistiques sont difficiles à obtenir. D'autre part comme nous l'avons déjà signalé , la solution optimale obtenue n'est valable que pour un signal stationnaire. L'approche du gradient stochastique nous a permis, à partir de ce critère statistique de nous affranchir de ces inconvénients mais nous pouvons choisir de reformuler le problème dans le cadre des moindres carrés, c'est-à-dire rechercher le filtre qui minimise la somme des carrés des erreurs, jusqu'à l'instant considéré *n* .
Nous donnerons dans le paragraphe « Algorithme des moindres carrés récursifs » une approche récursive de ce problème (autorisant une évolution pour un fonctionnement adaptatif) .
Dans un premier temps nous allons considérer le cas dans lequel on connaît le signal d'entrée u_n sur N points et trouver le filtre optimal pour ce bloc de données.
Remarque : pour simplifier les équations nous considérerons maintenant le cas d'un filtre à coefficients réels.

5.1 Critère des moindres carrés pour un bloc de données de N points ($u_0,u_1,\ldots,u_{N-1}$)

$$J_{MC} = \sum_i (e(i))^2 = \sum_i (y(i) - d(i))^2$$

$$\text{avec } y(i) = \sum_{l=0}^{M-1} w_l u(i-l)$$

Pour préciser les bornes de la sommation sur *i* , il faut prendre en compte le fait que pour *i<M-1* et *i>N-1*, *y(i)* dépend de valeurs d'entrée inconnues.

Ainsi :

$$\begin{vmatrix} y(0) \\ \vdots \\ y(M-1) \\ \vdots \\ y(N-1) \\ \vdots \\ y(N+M-2) \end{vmatrix} = \begin{bmatrix} u(0) & ? & ? \\ & \ddots & ? \\ \vdots & & \\ u(M-1) & \cdots & u(0) \\ \vdots & & \\ u(N-1) & \cdots & u(N-M) \\ ? & \ddots & \vdots \\ ? & ? & u(N-1) \end{bmatrix} * \begin{vmatrix} w_0 \\ w_1 \\ \vdots \\ w_{M-1} \end{vmatrix}$$

Différentes hypothèses sont possibles concernant les valeurs inconnues de l'entrée :
- soit considérer *u(i) = 0* pour *i < 0*
- soit considérer *u(i) = 0* pour *i > N-1*
- soit ignorer ces valeurs de l'entrée

On note la relation précédente : $\underline{y} = \underline{\underline{A}}^T \underline{w}$

La matrice $\underline{\underline{A}}$ est appelée matrice de données.

$$\underline{\underline{A}} = \begin{bmatrix} u(0) & \cdots & u(M-1) & \cdots & u(N-1) & & (0) \\ & \ddots & \vdots & & \vdots & \ddots & \\ (0) & & u(0) & \cdots & u(N-M) & \cdots & u(N-1) \end{bmatrix}$$

Suivant les hypothèses, on pourra extraire plusieurs matrices :

- si on refuse d'affecter les valeurs inconnues, on extrait une matrice de Toeplitz de dimension *(M) x (N-M+1)*, qui ne contient que les valeurs connues de l'entrée. En d'autres termes, on ne pourra calculer l'erreur que pour *i = M-1* à *i = N-1*. Cette forme est appelée « forme covariance ».
- si on affecte la valeur 0 aux échantillons de l'entrée pour *i < 0*, on parle de préfenêtrage, et on extrait la forme préfenêtrée, de dimension *M x N*. On peut calculer *e(i)* pour *i = 0* à *N-1*
- si on affecte 0 aux valeurs inconnues pour *i > N-1*, on a une forme post-fenêtrée, et on peut calculer *e(i)* pour *i = M-1* à *N+M-2*.
- Enfin, en utilisant à la fois le pré et post fenêtrage, on obtient la forme « autocorrélation » de dimension *M x (N+M-2)*, et *e(i)* est calculable pour *i = 0* à *N+M-2*.

Les termes de covariance et autocorrélation sont consacrés, en particulier en traitement de la parole. Cependant, il n'y a là aucun (ou peu) de rapport avec les définitions des fonctions et matrice de covariance et d'autocorrélation.

Notons de plus que le fait de compléter la séquence par des zéros n'est pas la seule solution. En particulier, il est possible de périodiser la séquence, ce qui conduit à des matrices circulantes.

Suivant les hypothèses faites sur le fenêtrage, on ne minimisera donc pas nécessairement l'erreur sur tout l'horizon :

$$J = \sum_{i=M-1}^{N-1} (e(i))^2$$ pour la forme covariance, par exemple.

On note $\underline{d}$ le vecteur de la sortie désirée, $\underline{y}$ le vecteur de sortie, et $\underline{\underline{A}}$ la matrice de donnée.

Le vecteur d'erreur $\underline{e}$ vaut : $\underline{e} = \underline{y} - \underline{d} = \underline{\underline{A}}^T \underline{w} - \underline{d}$

Le critère J s'écrit : $J = e^T e = (\underline{\underline{A}}^T \underline{w} - d)^T (\underline{\underline{A}}^T \underline{w} - d)$

Il ne reste plus qu'à exprimer le gradient de J par rapport à $\underline{w}$:

$$\underline{\nabla} J\big|_{\underline{w}} = 2\underline{\underline{A}}(\underline{\underline{A}}^T \underline{w} - \underline{d})$$

Et on en déduit que

$$\underline{\underline{A}}\underline{\underline{A}}^T \underline{w} = \underline{\underline{A}}\underline{d}$$

En développant $\underline{\underline{A}}\underline{\underline{A}}^T$ et $\underline{\underline{A}}\underline{d}$, il est facile de voir que les termes génériques valent :

$$(\underline{\underline{A}}\underline{\underline{A}}^T)_{ij} = \sum_l u(l-i)u(l-j) = \hat{r}_{uu}(i-j)$$

$$(\underline{\underline{A}}\underline{d})_i = \sum_l u(l-i)d(l) = \hat{r}_{du}(i)$$

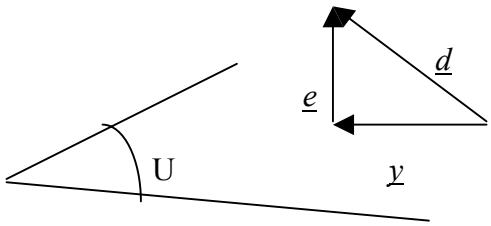
On retrouve ainsi les équations normales (équation de Wiener Hopf) en utilisant des corrélations empiriques (ou temporelles).

Notons que suivant les hypothèses effectuées sur le fenêtrage, ces estimées ne seront pas identiques.
En particulier, la matrice de corrélation empirique est de Toeplitz avec la méthode d'autocorrélation, alors qu'elle ne l'est pas forcément dans les autres hypothèses. On peut encore noter que $\hat{\underline{\underline{R}}}_{uu} = \underline{\underline{A}}\underline{\underline{A}}^T$ est par construction définie non négative (c'est un « carré » matriciel) et symétrique (à symétrie hermitienne).

5.1.1 Théorème de projection

A l'optimum, le gradient s'annule : $\underline{\underline{A}}(\underline{\underline{A}}^T \underline{w} - \underline{d}) = \underline{0} \Leftrightarrow \underline{\underline{A}}\underline{e} = \underline{0}$

Par conséquent, en multipliant par $\overset{\Delta}{\underline{w}}^T$ on obtient : $\underline{y}^T \underline{e} = 0$



on retrouve ainsi le théorème de Pythagore pour les moindres carrés.

5.2 Solution de l'équation normale :

A partir de $\underline{\underline{A}}\underline{\underline{A}}^T \underline{w} = \underline{\underline{A}}\underline{d}$

on obtient $\overset{\Delta}{\underline{w}}_{MC} = (\underline{\underline{A}}\underline{\underline{A}}^T)^{-1} \underline{\underline{A}}\underline{d}$ si $\underline{\underline{A}}\underline{\underline{A}}^T$ est inversible.

$\overset{\Delta}{\underline{w}}_{MC}$ est donc le filtre optimal pour le critère des moindres carrés J_{MC} pour le bloc de données de N points $(u_0, u_1, \ldots, u_{N-1})$.
Les inconvénients de cette approche sont les suivants :

- La solution optimale s'obtient en inversant la matrice $\underline{\underline{A}}\underline{\underline{A}}^T$ ce qui représente une grosse charge de calcul
- Le fonctionnement en bloc ne permet pas à priori de prendre en compte une nouvelle donnée u_N pour ajuster le filtre sans tout recalculer .
- L'algorithme peut être vu comme adaptatif si on considère des blocs successifs de N points mais ne peut pas suivre des non stationnarités à évolution rapide.

Ces différents inconvénients justifient la recherche d'une approche récursive pour ce critère des moindres carrés.

6 ALGORITHME DES MOINDRES CARRES RECURSIFS

L'approche des moindres carrés consiste à minimiser la somme des carrés des erreurs $J_{MC} = \sum_{i=i_1}^{i_2} e(i)^2$. On obtient alors la solution des moindres carrés (les bornes de la somme dépendent des hypothèses de fenêtrage).

$\underline{\overset{\Delta}{w}}_{MC} = \arg\min_{\underline{w}} (\underline{\underline{A}}^T \underline{w} - \underline{d})^T (\underline{\underline{A}}^T \underline{w} - \underline{d})$ où $\underline{\underline{A}}$ est la matrice de données.

La solution exacte s'écrit $\underline{\overset{\Delta}{w}}_{MC} = (\underline{\underline{A}}\,\underline{\underline{A}}^T)^{-1} \underline{\underline{A}}\,\underline{d}$

Nous avons vu que la matrice $(\underline{\underline{A}}\,\underline{\underline{A}}^T)$ correspond à une estimation de la matrice de corrélation $\underline{\underline{R}}_{uu}$, et le produit $\underline{\underline{A}}\,\underline{d}$ est l'estimée de l'intercorrélation $\underline{R}_{ud}$.
L'approche des moindres carrés est par essence une approche bloc, c'est-à-dire qu'on traite un bloc de données pour construire $\underline{\underline{A}}$, puis en déduire $\underline{\overset{\Delta}{w}}_{MC}$.
L'inversion de $\underline{\underline{A}}\,\underline{\underline{A}}^T$ est coûteuse (de l'ordre de M^3).
Par ailleurs, la nature « bloc » de l'algorithme empêche à priori de prendre en compte simplement une nouvelle donnée, en profitant du résultat déjà obtenu : si on a résolu

$\underline{\overset{\Delta}{w}}_{MC}^{(n)} = \arg\min_{\underline{w}} \sum_{i=i_1}^{n} e(i)^2$

peut-on relier $\underline{\overset{\Delta}{w}}_{MC}^{(n+1)} = \arg\min_{\underline{w}} \sum_{i=i_1}^{n+1} e(i)^2$ à $\underline{\overset{\Delta}{w}}_{MC}^{(n)}$?

La réponse est bien entendu affirmative. L'algorithme correspondant est l'algorithme des moindres carrés récursifs, qui permet de mettre à jour la solution au cours du temps. Notons dès à présent qu'il ne s'agit pas, tel quel, d'un algorithme adaptatif : dans la solution $\underline{\overset{\Delta}{w}}_{MC}^{(n)}$ est contenue toute l'histoire du signal jusqu'à l'instant *n*, avec une hypothèse implicite de stationnarité. Pour obtenir un algorithme adaptatif on pourra introduire un « facteur d'oubli » sur les données les plus anciennes (voir paragraphe correspondant).

6.1 Construction de l'algorithme des MCR.

On pose $\underline{\overset{\Delta}{w}}^{(n)} = \underline{\underline{R}}_{uu}^{(n)^{-1}} \underline{R}_{ud}^{(n)}$ où $\underline{\underline{R}}_{uu}^{(n)}$ et $\underline{R}_{ud}^{(n)}$ sont la matrice $\underline{\underline{A}}\,\underline{\underline{A}}^T$ et le vecteur $\underline{\underline{A}}\,\underline{d}$ construits à partir des données jusqu'à l'instant n.
Dorénavant nous considérerons de plus que $i_1=0$ (hypothèse préfenêtrée)

La matrice $\underline{\underline{R}}_{uu}^{(n)}$ possède alors la structure suivante : $\underline{\underline{R}}_{uu}^{(n)} = \sum_{i=0}^{n} \underline{u}(i)\underline{u}(i)^T$

Dans ces conditions, à l'instant *(n+1)*, $\underline{\underline{R}}_{uu}^{(n+1)} = \underline{\underline{R}}_{uu}^{(n)} + \underline{u}(n+1)\underline{u}(n+1)^T$ et on dispose d'une équation récursive pour estimer la matrice de corrélation.

Le problème est maintenant que pour obtenir une formulation récursive pour le vecteur optimal, il faut avoir une formulation récursive sur l'inverse de la matrice de corrélation.

Pour cela, on utilise le **lemme d'inversion matriciel**
Si $\underline{\underline{A}}\ \underline{\underline{B}}\ \underline{\underline{C}}\ \underline{\underline{D}}$ sont des matrices de dimensions convenables,

$$\boxed{(\underline{\underline{A}}+\underline{\underline{B}}\underline{\underline{C}}\underline{\underline{D}})^{-1}=\underline{\underline{A}}^{-1}-\underline{\underline{A}}^{-1}\underline{\underline{B}}(\underline{\underline{C}}^{-1}+\underline{\underline{D}}\underline{\underline{A}}^{-1}\underline{\underline{B}})^{-1}\underline{\underline{D}}\underline{\underline{A}}^{-1}}$$

Ce lemme se démontre par vérification directe.

Dans notre cas, il s'agit d'inverser $\underline{\underline{R}}_{uu}^{(n+1)}=\underline{\underline{R}}_{uu}^{(n)}+\underline{u}(n+1)\underline{u}(n+1)^T$

on prendra donc $\underline{\underline{A}}=\underline{\underline{R}}_{uu}^{(n)}$ $\underline{\underline{B}}=\underline{u}(n+1)$ $\underline{\underline{C}}=1$ $\underline{\underline{D}}=\underline{u}(n+1)^T$

En appliquant le lemme d'inversion matricielle, il vient alors :

$$(\underline{\underline{R}}_{uu}^{(n+1)})^{-1}=(\underline{\underline{R}}_{uu}^{(n)})^{-1}-\frac{(\underline{\underline{R}}_{uu}^{(n)})^{-1}\underline{u}(n+1)\underline{u}(n+1)^T(\underline{\underline{R}}_{uu}^{(n)})^{-1}}{1+\underline{u}(n+1)^T(\underline{\underline{R}}_{uu}^{(n)})^{-1}\underline{u}(n+1)}$$

Pour alléger l'écriture, on pose : $\underline{\underline{K}}_n=(\underline{\underline{R}}_{uu}^{(n)})^{-1}$
on a alors

$$\boxed{\underline{\underline{K}}_{n+1}=\underline{\underline{K}}_n-\frac{\underline{\underline{K}}_n\underline{u}(n+1)\underline{u}(n+1)^T\underline{\underline{K}}_n}{1+\underline{u}(n+1)^T\underline{\underline{K}}_n\underline{u}(n+1)}}\quad(0.22)$$

On notera que cette formule permet de calculer l'inverse de la matrice de corrélation, sans calcul explicite d'inverse.

Le vecteur d'intercorrélation $\underline{\underline{R}}_{ud}^{(n)}$ s'écrit quant à lui, sous forme récursive, selon
$\underline{\underline{R}}_{ud}^{(n+1)}=\underline{\underline{R}}_{ud}^{(n)}+d(n+1)\underline{u}(n+1)$

Le filtre optimal à l'itération n+1 s'écrit donc :
$\underline{w}^{\Delta(n+1)}=\underline{\underline{R}}_{uu}^{(n+1)^{-1}}\underline{\underline{R}}_{ud}^{(n+1)}=\underline{\underline{K}}_{n+1}(\underline{\underline{R}}_{ud}^{(n)}+d(n+1)\underline{u}(n+1))$

Pour simplifier plus loin nous explicitons le dernier terme de cette expression : $\underline{\underline{K}}_{n+1}\underline{u}(n+1)$

$$\underline{\underline{K}}_{n+1}\underline{u}(n+1)=(\underline{\underline{K}}_n-\frac{\underline{\underline{K}}_n\underline{u}(n+1)\underline{u}(n+1)^T\underline{\underline{K}}_n}{1+\underline{u}(n+1)^T\underline{\underline{K}}_n\underline{u}(n+1)})\underline{u}(n+1)$$

$$=\underline{\underline{K}}_n\underline{u}(n+1)(1-\frac{\underline{u}(n+1)^T\underline{\underline{K}}_n\underline{u}(n+1)}{1+\underline{u}(n+1)^T\underline{\underline{K}}_n\underline{u}(n+1)})$$

$$=\frac{\underline{\underline{K}}_n\underline{u}(n+1)}{1+\underline{u}(n+1)^T\underline{\underline{K}}_n\underline{u}(n+1)}$$

Nous avons donc :

$$\underline{w}^{\Delta(n+1)} = (\underline{\underline{K}}_n - \frac{\underline{\underline{K}}_n \underline{u}(n+1)\underline{u}(n+1)^T \underline{\underline{K}}_n}{1+\underline{u}(n+1)^T \underline{\underline{K}}_n \underline{u}(n+1)})(\underline{R}_{ud}^{(n)} + d(n+1)\underline{u}(n+1))$$

$$= \underline{w}^{\Delta(n)} - \frac{\underline{\underline{K}}_n \underline{u}(n+1)\underline{u}(n+1)^T \underline{\underline{K}}_n}{1+\underline{u}(n+1)^T \underline{\underline{K}}_n \underline{u}(n+1)} \underline{R}_{ud}^{(n)} + d(n+1)\frac{\underline{\underline{K}}_n \underline{u}(n+1)}{1+\underline{u}(n+1)^T \underline{\underline{K}}_n \underline{u}(n+1)}$$

$$= \underline{w}^{\Delta(n)} - \frac{\underline{\underline{K}}_n \underline{u}(n+1)}{1+\underline{u}(n+1)^T \underline{\underline{K}}_n \underline{u}(n+1)}(\underline{u}(n+1)^T \underline{\underline{K}}_n \underline{R}_{ud}^{(n)} - d(n+1))$$

Soit finalement :

$$\boxed{\underline{w}^{\Delta(n+1)} = \underline{w}^{\Delta(n)} + \underline{\underline{K}}_{n+1}\underline{u}(n+1)(d(n+1) - \underline{u}(n+1)^T \underline{w}^{\Delta(n)})} \qquad (0.23)$$

Cette formule, définissant le nouveau filtre optimal à partir du filtre est aussi appelée, par analogie, formule de Kalman.
Les équations (0.22) et (0.23) constituent **l'algorithme des moindres carrés récursifs** :

$$\boxed{\underline{\underline{K}}_{n+1} = \underline{\underline{K}}_n - \frac{\underline{\underline{K}}_n \underline{u}(n+1)\underline{u}(n+1)^T \underline{\underline{K}}_n}{1+\underline{u}(n+1)^T \underline{\underline{K}}_n \underline{u}(n+1)}}$$

$$\boxed{\underline{w}^{\Delta(n+1)} = \underline{w}^{\Delta(n)} + \underline{\underline{K}}_{n+1}\underline{u}(n+1)(d(n+1) - \underline{u}(n+1)^T \underline{w}^{\Delta(n)})}$$

Dans l'algorithme développé ci-dessus, on n'a pas utilisé toutes les caractéristiques de la matrice de corrélation, et en particulier le fait que la matrice de corrélation (dans le cas de l'hypothèse autocorrélation) est de Toeplitz.

On peut alors obtenir une formule de renouvellement plus simple.
Par ailleurs, on peut effectuer à la fois une récurrence sur le temps, mais aussi sur l'ordre (Order Recursive Adaptive Least Squares).

On peut également trouver une récurrence sur le gain $\underline{\underline{K}}_{n+1}\underline{u}(n+1)$ plutôt que sur l'inverse de la matrice de corrélation $\underline{\underline{K}}_{n+1}$. Les algorithmes obtenus sont les algorithmes des moindres carrés récursifs rapides, qui peuvent poser des problèmes d'instabilité numérique.

On notera que le terme d'erreur s'écrit

$$\xi(n+1) = d(n+1) - \underline{u}(n+1)^T \underline{w}^{\Delta(n)}$$

alors qu'on avait $e(n+1) = d(n+1) - \underline{u}(n+1)^T \underline{w}(n+1)$ dans le cas des algorithmes du gradient.

La première erreur est l'erreur à priori, c'est-à-dire l'erreur effectuée avant d'avoir mis à jour le filtre avec la nouvelle donnée *u(n+1)*.
Cette erreur est différente de l'erreur à posteriori, dans laquelle le filtre a été mis à jour.

6.2 Problèmes d'initialisation :

La mise en œuvre pratique des algorithmes des MCR requiert le choix des valeurs initiales $\underline{\underline{K}}_0$ et $\underline{\overset{\Delta}{w}}^{(0)}$

La matrice de corrélation, à l'étape *n* s'exprimant comme une somme de dyades ,

$$\underline{\underline{R}}_{uu}^{(n)} = \sum_{i=0}^{n} \underline{u}(i)\underline{u}(i)^T$$

Cette matrice est singulière tant que $n < M$, ce qui empêche de démarrer la récurrence.

Deux solutions sont alors possibles :
- évaluer la matrice de corrélation et son inverse pour un $n > M$
- modifier légèrement la formule d'estimation de $\underline{\underline{R}}_{uu}^{(n)}$ en introduisant un « talon » :

$$\underline{\underline{R}}_{uu}^{(n)} = \sum_{i=0}^{n} \underline{u}(i)\underline{u}(i)^T + \delta \underline{\underline{I}}$$

où δ est un scalaire positif très faible.

Il s'en suit que $\underline{\underline{R}}_{uu}^{(0)} = \delta \underline{\underline{I}}$ et $\underline{\underline{K}}_n = \frac{1}{\delta}\underline{\underline{I}}$

Ce choix consiste en fait à modifier le critère initial en

$$J' = \delta \left\| \underline{w}^{(n)} \right\|^2 + \sum_{i=0}^{n} e(i)^2$$

l'intervention de la constante δ introduit une régularisation de l'algorithme, en pénalisant la norme du vecteur optimal (ce qui est équivalent à effectuer la minimisation de J_{MC} (n), sous contrainte $\left\| \underline{w}^{(n)} \right\|^2 < \eta$).

Il reste à choisir $\underline{\overset{\Delta}{w}}^{(0)}$, que l'on choisit usuellement comme le vecteur nul.

6.3 Version adaptative – pondération exponentielle

L'algorithme précédent n'est pas adaptatif. Pour le rendre adaptatif, il faut « oublier » le passé du signal lors de l'évolution de l'algorithme.

Pour cela, une approche classique consiste à donner un poids plus important aux valeurs les plus récentes : on modifie le critère en introduisant une pondération exponentielle :

$$J' = \sum_{i=0}^{n} \lambda^{n-i} e(i)^2$$

On montre alors aisément que seule l'équation de renouvellement de l'inverse de la corrélation est modifiée :

$$\boxed{\underline{\underline{K}}_{n+1} = \frac{1}{\lambda}\underline{\underline{K}}_n - \frac{1}{\lambda} \frac{\underline{\underline{K}}_n \underline{u}(n+1)\underline{u}(n+1)^T \underline{\underline{K}}_n}{1 + \frac{1}{\lambda}\underline{u}(n+1)^T \underline{\underline{K}}_n \underline{u}(n+1)}} \quad (0.24)$$

Cet algorithme ne converge plus (la matrice de corrélation reste aléatoire lorsque $n \to \infty$), mais ceci est normal puisque l'on se place dans une perspective adaptative, en permettant à l'algorithme de réagir à des non stationnarités.

6.4 Equation récursive sur le critère somme quadratique

A l'instant *n*, pour le vecteur optimal $\underline{\overset{\Delta}{w}}^{(n)}$, on a

$$J(n) = \sum_{i=0}^{n} \lambda^{n-i} e(i)^2$$

$$= \sum_{i=0}^{n} \lambda^{n-i} (d(i) - \underline{\overset{\Delta}{w}}^{T(n)} \underline{u}(i))^2$$

$$= \sum_{i=0}^{n} \lambda^{n-i} (d(i))^2 - \underline{\hat{R}}_{ud}^{T(n)} \underline{\overset{\Delta}{w}}^{(n)}$$

où $\underline{\hat{R}}_{ud}^{T(n)}$ est l'intercorrélation « exponentielle » $\underline{\hat{R}}_{ud}^{(n)} = \sum_{i=0}^{n} \lambda^{n-i} \underline{u}(i) d(i)$.

En posant $\hat{\sigma}_d^2(n) = \sum_{i=0}^{n} \lambda^{n-i} (d(i))^2$

On a $\hat{\sigma}_d^2(n) = \lambda \hat{\sigma}_d^2(n-1) + (d(n))^2$

et $\underline{\hat{R}}_{ud}(n) = \lambda \underline{\hat{R}}_{ud}(n-1) + d(n)\underline{u}(n)$

En utilisant ces deux relations, il vient (après quelques lignes de calcul ...)

$$J(n) = \lambda J(n-1) + \xi_n e_n \qquad (0.25)$$

Notons que $e(n) = d(n) - \underline{w}^{T(n)} \underline{u}(n) = d(n) - (\underline{w}^{(n-1)} + \underline{\underline{K}}_n \underline{u}(n)\xi(n))\underline{u}(n)$

soit $e(n) = (1 - \underline{u}^T(n)\underline{\underline{K}}_n \underline{u}(n))\xi(n)$

Le rapport entre les erreurs à posteriori et à priori s'appelle facteur de conversion :

$$\gamma(n) = \frac{e(n)}{\xi(n)} = 1 - \underline{u}^T(n)\underline{\underline{K}}_n \underline{u}(n)$$

6.5 Convergence de l'algorithme, en moyenne

On supposera ici que

$d(n) = \underline{\overset{\Delta}{w}}^T \underline{u}(n) + \overset{\Delta}{e}(n)$

où $\underline{\overset{\Delta}{w}}$ est le vecteur filtre exact et $\overset{\Delta}{e}(n)$ un bruit d'observation blanc centré, décorrélé de $\underline{u(n)}$ (notez que ce modèle est restrictif : rien ne dit, dans le filtrage de Wiener que *d(n)* est issu d'une filtrée linéaire de *u(n)*).

En écrivant l'algorithme, on a

$$\boxed{\underline{w}^{(n)} = \underline{w}^{(n-1)} + \underline{\underline{K}}_n \underline{u}(n)(d(n) - \underline{u}(n)^T \underline{w}^{(n-1)})}$$

Exprimons $\underline{R}_{ud}^{(n)}$ en remplaçant *d(i)* par son expression,

$$\underline{R}_{ud}{}^{(n)} = \sum_{i=0}^{n} \overset{\Delta}{\underline{w}}{}^{T}\, \underline{u}(i)\underline{u}(i) + \overset{\Delta}{e}(i)\underline{u}(i) = \underline{\underline{R}}_{uu}{}^{(n)}\, \overset{\Delta}{\underline{w}} + \sum_{i=0}^{n} \overset{\Delta}{e}(i)\underline{u}(i)$$

En multipliant par $\underline{\underline{K}}_n$ il vient

$$\underline{w}^{(n)} = \overset{\Delta}{\underline{w}} + \underline{\underline{K}}_n \sum_{i=0}^{n} \overset{\Delta}{e}(i)\underline{u}(i)$$

En utilisant maintenant la composition des espérances : $E_{X,Y}(X(Y)) = E_Y(E_{X/Y}(X/Y))$

et grâce à la décorrélation (l'indépendance) entre $\overset{\Delta}{e}(n)$ et *u(n)*, et au fait que $\overset{\Delta}{e}(n)$ soit centré, on obtient :

$$E(\underline{w}(n)) = \overset{\Delta}{\underline{w}} \tag{0.26}$$

L'algorithme converge donc en moyenne, ce qui est rassurant, et il converge à nombre d'itérations fini : il n'est pas nécessaire d'avoir $n \to \infty$ pour obtenir la convergence (cas du LMS).

6.6 Erreur quadratique moyenne

Avec des hypothèses irréalistes, mais qui permettent d'aboutir à un résultat bien vérifié en pratique, on montre que

$$E\left(\left|\underline{w}(n) - \overset{\Delta}{\underline{w}}\right|^2\right) \propto \frac{1}{n} \sum_{i=0}^{M-1} \frac{1}{\lambda_i}$$

Ce résultat montre que l'algorithme converge en *1/n*, et que l'erreur quadratique est inversement proportionnelle aux valeurs propres. Une fois de plus, le mauvais conditionnement limite la vitesse de convergence.

6.7 Algorithmes sous optimaux

L'idée des algorithmes sous optimaux est de simplifier l'équation de renouvellement de $\underline{\underline{K}}_n$. En particulier, on obtient des algorithmes du type LMS en effectuant la simplification ultime :

$$\underline{\underline{K}}_{n+1} = \mu_{n+1}\underline{\underline{I}}$$

Bibliographie sommaire :

- Méthodes adaptatives pour le signal, F. Michaut, Hermès
- Adaptive filter theory, S. Haykin, Prentice Hall.