

### 3 Filtrage Adaptatif

#### 3.1 Introduction et principaux résultats

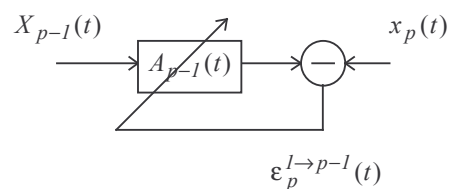
L'attrait que suscite chez les électroniciens l'utilisation du filtrage adaptatif ne tient pas qu'aux intérêts applicatifs de cette technique, pourtant considérables; il s'explique aussi, pour une part non négligeable, par la séduction intellectuelle qu'exerce sur les esprits la mise au point de systèmes artificiels capables de s'adapter de manière indépendante aux diverses situations qu'ils sont appelés à rencontrer.

Selon les disciplines où elle se rencontre, cette capacité d'adaptation est définie de manière différente. En informatique, par exemple, elle se résume souvent à une aptitude à mettre en œuvre des raisonnements logiques selon un certain nombre de règles prédéfinies ou, éventuellement évolutives; c'est là le domaine de l'intelligence artificielle. En électronique, elle se réduit plutôt à l'adaptation, selon un critère mathématique, des paramètres d'un filtre, le plus souvent numérique; il s'agit alors du domaine du filtrage adaptatif.

Les solutions techniques basées sur l'optimisation de filtres adaptatifs se rencontrent dans une variété de problèmes de détection, d'estimation et d'identification. C'est dans les années 60 [1] et dans le domaine de l'égalisation des voies de transmission de données que des solutions adaptatives ont, pour la première fois, été utilisées à grande échelle. Depuis lors, de telles approches ont gagné bien d'autres domaines d'applications [2].

Il existe plusieurs structures possibles, plusieurs types de paramètres et plusieurs critères d'adaptation pour réaliser un filtre adaptatif. Ce cours ne considère que le cas des filtres transverses adaptés selon le critère des moindres carrés.

Un filtre adaptatif transverse peut être représenté par la structure suivante :



- Schéma général d'un filtre adaptatif -

A partir d'un vecteur de  $p-1$  termes :

$$X_{p-1}(t) = \begin{pmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_{p-1}(t) \end{pmatrix} \quad (45.)$$

Le filtre de  $p-1$  coefficients :

$$A_{p-l}(t) = \begin{pmatrix} a_1(t) \\ a_2(t) \\ \vdots \\ a_{p-l}(t) \end{pmatrix} \quad (46.)$$

va calculer l'estimée  $\hat{x}_p(t)$  du signal  $x_p(t)$ .

Ce terme  $\hat{x}_p(t)$  représente donc la prédiction de  $x_p(t)$  à partir de  $\{x_1(t) \ x_2(t) \ \dots \ x_{p-l}(t)\}$ .

La différence entre  $x_p(t)$  et  $\hat{x}_p(t)$  est égale à l'erreur de prédiction de  $x_p(t)$  à partir de  $\{x_1(t) \ x_2(t) \ \dots \ x_{p-l}(t)\}$ . Elle est ici notée  $\varepsilon_p^{l \rightarrow p-l}(t)$ .

L'indice  $p$  indique qu'il s'agit de l'erreur de prédiction de  $x_p(t)$ , l'exposant  $l \rightarrow p-l$  indique que la base de prédiction est composée des valeurs  $x_l(t)$  jusqu'à  $x_{p-l}(t)$ . Cette notation peut paraître un peu surchargée mais elle va permettre de développer de manière unifiée des algorithmes issus d'approches très différentes. L'erreur de prédiction s'écrit finalement :

$$\varepsilon_p^{l \rightarrow p-l}(t) = x_p(t) - A_{p-l}^T(t) X_{p-l}(t) \quad (47.)$$

Dans cette expression l'exposant  $T$  indique que le vecteur  $A_{p-l}(t)$  est transposé et conjugué.

Le critère des moindres carrés consiste à minimiser la fonction de coût  $J_p(t)$  suivante :

$$J_p(t) = \sum_{i=0}^t \lambda^{t-i} \varepsilon_p^{l \rightarrow p-l}(i) \varepsilon_p^{l \rightarrow p-l T}(i) \quad (48.)$$

Expression dans laquelle le scalaire  $\lambda$  représente un facteur d'oubli inférieur à un.

La solution optimale, obtenue au sens de ce critère, est alors donnée par l'expression suivante :

$$A_{p-l}(t) = R_{p-l}^{-1}(t) r_p^{l \rightarrow p-l}(t) \quad (49.)$$

expression dans laquelle  $R_{p-l}(t)$  représente la matrice d'autocorrélation estimée de  $X_{p-l}(t)$  et  $r_p^{l \rightarrow p-l}(t)$  le vecteur d'intercorrélations estimé de  $X_{p-l}(t)$  avec  $x_p(t)$  :

$$R_{p-l}(t) = \sum_{i=0}^t \lambda^{t-i} X_{p-l}(i) X_{p-l}^T(i) \quad (50.)$$

$$r_p^{l \rightarrow p-l}(t) = \sum_{i=0}^t \lambda^{t-i} X_{p-l}(i) x_p^T(i) \quad (51.)$$

la prédiction du signal  $x_p(t)$  s'écrit alors :

$$\hat{x}(t) = r_p^{l \rightarrow p-l T}(t) R_{p-l}^{-1}(t) X_{p-l}(t) \quad (52.)$$

**Remarque :**

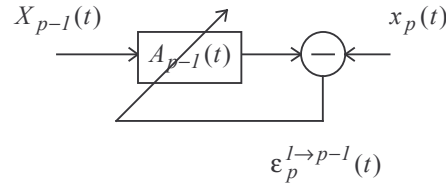
On peut choisir de définir l'erreur de prédiction de manière inversée, c'est à dire par :

$$\varepsilon_p^{l \rightarrow p-1}(t) = A_{p-1}^T(t)X_{p-1}(t) - x_p(t) \quad (53.)$$

La solution optimale ne change pas, seule l'équation de l'erreur de prédiction est inversée. Les équations d'algorithmes d'adaptation des coefficients pourront aussi être inversées.

### 3.2 Démonstration de la solution optimale

On rappelle ci-dessous le schéma général d'un filtre adaptatif :



L'erreur de filtrage  $\varepsilon_p^{l \rightarrow p-1}(t)$  s'écrit de manière vectorielle :

$$\varepsilon_p^{l \rightarrow p-1}(t) = x_p(t) - A_{p-1}^T(t)X_{p-1}(t)$$

avec :

$$X_{p-1}(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_{p-1}(t) \end{bmatrix} \text{ et } A_{p-1}(t) = \begin{bmatrix} a_1(t) \\ a_2(t) \\ \vdots \\ a_{p-1}(t) \end{bmatrix}$$

En développant l'espérance carrée de cette erreur, il vient :

$$E \left[ \varepsilon_p^{l \rightarrow p-1}(t) \varepsilon_p^{l \rightarrow p-1 T}(t) \right] = E \left[ \left( x_p(t) - A_{p-1}^T(t)X_{p-1}(t) \right) \left( x_p(t) - A_{p-1}^T(t)X_{p-1}(t) \right)^T \right]$$

(l'exposant <sup>T</sup> signifiant transconjugué)

$$E \left[ \varepsilon_p^{l \rightarrow p-1}(t) \varepsilon_p^{l \rightarrow p-1 T}(t) \right] = E \left[ \left( x_p(t)x_p^T(t) - A_{p-1}^T(t)X_{p-1}(t)x_p^T(t) - x_p(t)X_{p-1}^T(t)A_{p-1}(t) + A_{p-1}^T(t)X_{p-1}(t)X_{p-1}^T(t)A_{p-1}(t) \right) \right]$$

$$E \left[ \varepsilon_p^{l \rightarrow p-1}(t) \varepsilon_p^{l \rightarrow p-1 T}(t) \right] = E \left[ x_p(t)x_p^T(t) \right] - 2 \operatorname{Re} \left( A_{p-1}^T(t)E \left[ X_{p-1}(t)x_p^T(t) \right] \right) + A_{p-1}^T(t)E \left[ X_{p-1}(t)X_{p-1}^T(t) \right]A_{p-1}(t)$$

On introduit alors la matrice  $R_{p-1}(t) = E \left[ X_{p-1}(t)X_{p-1}^T(t) \right]$  d'autocorrélation du vecteur d'entrée  $X_{p-1}(t)$  et le

vecteur  $S_{p-1}(t) = E \left[ X_{p-1}(t)x_p^T(t) \right]$  d'intercorrélacion entre le vecteur d'entrée  $X_{p-1}(t)$  et le signal désiré  $x_p(t)$ .

L'équation précédente devient alors :

$$E\left[\varepsilon_p^{l \rightarrow p-l}(t)\varepsilon_p^{l \rightarrow p-lT}(t)\right] = E\left[x_p(t)x_p^*(t)\right] - 2\operatorname{Re}\left(A_{p-l}^T(t)S_{p-l}(t)\right) + A_{p-l}^T(t)R_{p-l}(t)A_{p-l}(t)$$

On cherche le vecteur  $A_{p-l}(t)$  des coefficients du filtre qui minimise l'espérance carrée de l'erreur. En considérant que la matrice  $R_{p-l}(t)$  est de rang plein, il existe une solution unique qui s'obtient en annulant les dérivées de la forme quadratique  $E\left[\varepsilon_p^{l \rightarrow p-l}(t)\varepsilon_p^{l \rightarrow p-lT}(t)\right]$  par rapport aux coefficients du filtre  $A_{p-l}(t)$ .

On décompose alors le filtre  $A_{p-l}(t)$  en parties réelle et imaginaire :

$$A_{p-l}(t) = Ar_{p-l}(t) + jAi_{p-l}(t)$$

ainsi que le vecteur d'intercorrélation  $S_{p-l}(t)$

$$S_{p-l}(t) = Sr_{p-l}(t) + jSi_{p-l}(t)$$

D'où :

$$E\left[\varepsilon_{p-l}(t)\varepsilon_{p-l}^*(t)\right] = E\left[x_p(t)x_p^T(t)\right] - 2Ar_{p-l}^T Sr_{p-l} - 2Ai_{p-l}^T Si_{p-l} + A_{p-l}^T(n)R_{p-l}(n)A_{p-l}(n)$$

On considère alors la dérivation de cette expression par rapport à la  $k^{\text{ième}}$  composante  $a_k(t)$  du vecteur  $A_{p-l}(t)$ .

On décompose la composante  $a_k(t)$  en parties réelle et imaginaire :

$$a_k(t) = a_{k_r}(t) + ja_{k_i}(t)$$

et l'on distingue les dérivations par rapport à ces parties réelle et imaginaire.

**Remarque :**

Si  $x$  est une variable complexe  $x = a + jb$  avec  $a$  et  $b$  réels. On définit alors la dérivée par rapport à  $x$  par :

$$\frac{\partial}{\partial x} = \frac{1}{2}\left(\frac{\partial}{\partial a} - j\frac{\partial}{\partial b}\right)$$

$$\text{On a alors } \frac{\partial x}{\partial x} = 1 \text{ et } \frac{\partial x^*}{\partial x} = 0$$

On démontre que la dérivation de la forme bilinéaire  $A_{p-l}^T(t)R_{p-l}(t)A_{p-l}(t)$  par rapport aux composantes du vecteur  $A_{p-l}(n)$  donne les résultats suivants :

$$\frac{\partial A_{p-l}^T(t)R_{p-l}(t)A_{p-l}(t)}{\partial a_{k_r}(t)} = 2\operatorname{Re}\left(R_{p-l}(t)A_{p-l}(t)\right)_k, \text{ l'indice } k \text{ signifiant que l'on s'intéresse à la } k^{\text{ième}} \text{ composante}$$

du vecteur  $R_{p-l}(t)A_{p-l}(t)$ .

et :

$$\frac{\partial A_{p-1}^T(t) R_{p-1}(t) A_{p-1}(t)}{\partial a_{k_i}(t)} = 2 \operatorname{Im} \left( R_{p-1}(t) A_{p-1}(t) \right)_k$$

D'où finalement :

$$\frac{\partial E \left[ \varepsilon_{p-1}(t) \varepsilon_{p-1}^*(t) \right]}{\partial a_{k_r}(t)} = -2 \left( S r_{p-1} \right)_k + 2 \operatorname{Re} \left( R_{p-1}(t) A_{p-1}(t) \right)_k$$

et

$$\frac{\partial E \left[ \varepsilon_{p-1}(t) \varepsilon_{p-1}^*(t) \right]}{\partial a_{k_i}(t)} = -2 \left( S i_{p-1} \right)_k + 2 \operatorname{Im} \left( R_{p-1}(t) A_{p-1}(t) \right)_k$$

D'où en annulant la dérivée par rapport à  $a_k(t)$  :

$$-2 \left( S r_{p-1} \right)_k + 2 \operatorname{Re} \left( R_{p-1}(t) A_{p-1}(t) \right)_k = 0$$

$$-2 \left( S i_{p-1} \right)_k + 2 \operatorname{Im} \left( R_{p-1}(t) A_{p-1}(t) \right)_k = 0$$

En annulant toutes les dérivées partielles il vient donc :

$$-2 S r_{p-1} + 2 \operatorname{Re} \left( R_{p-1}(t) A_{p-1}(t) \right) = 0$$

$$-2 S i_{p-1} + 2 \operatorname{Im} \left( R_{p-1}(t) A_{p-1}(t) \right) = 0$$

D'où :

$$R_{p-1}(t) A_{p-1}(t) = S_{p-1}(t)$$

et :

$$\boxed{A_{p-1}(t) = R_{p-1}^{-1}(t) S_{p-1}(t)}$$

Cette solution portant le nom de solution de Wiener-Hopf.

### Introduction du facteur d'oubli

En considérant alors l'hypothèse ergodique vérifiée, c'est à dire en confondant les espérances mathématiques et les moyennes temporelles, et en introduisant un facteur d'oubli  $\lambda$  dans la moyenne temporelle, afin de munir l'algorithme de propriétés de poursuites, l'espérance mathématique s'écrit :

$$E[x(t)] = \sum_{i=0}^t \lambda^i x(t-i) \text{ avec } \lambda \text{ facteur d'oubli inférieur ou égal à un.}$$

Il est alors immédiat que:

$$R_{p-1}(t+I) = \lambda R_{p-1}(t) + X_{p-1}(t+I) X_{p-1}^T(t+I)$$

$$S_{p-1}(t+I) = \lambda S_{p-1}(t) + X_{p-1}(t+I) x_p^T(t+I)$$

Ainsi, à chaque instant  $t+1$ , il est possible, à partir du nouveau vecteur  $X_{p-1}(t+1)$  et du nouveau scalaire  $x_p(t+1)$ , de calculer la matrice  $R_{p-1}(t+1)$ , de l'inverser et de l'appliquer au vecteur  $S_{p-1}(t+1)$ . Le vecteur  $A_{p-1}(t+1)$  optimal au sens du minimum de l'erreur quadratique pondérée par un facteur d'oubli  $\lambda$  est alors mis à jour.

Le principal inconvénient de cette méthode réside dans l'inversion en temps réel de la matrice  $R_{p-1}(t+1)$ . Cette inversion demande un coût de calcul proportionnel à  $p^3$  qui peut rapidement s'avérer prohibitif lorsque la dimension  $p$  devient importante. Le but des algorithmes des moindres carrés qui vont suivre va être de faire diminuer ce coût de calcul. Pour cela, ils vont utiliser des propriétés récursives de la matrice  $R_{p-1}(t)$  et du vecteur  $S_{p-1}(t)$ . Ces récursions agiront, soit en fonction du temps  $n$ , soit en fonction de l'ordre  $p$ . Elles vont conduire aux algorithmes des moindres carrés récursifs (MCR) dont le coût de calcul est proportionnel à  $p^2$ . Enfin, dans les cas où les composantes  $x_i(t)$  du vecteur  $X_{p-1}(t)$  sont en fait les valeurs décalées dans le temps d'un même processus monodimensionnel  $x(t)$  l'utilisation de ces propriétés de décalage conduira aux algorithmes rapides dont le coût de calcul est proportionnel à  $p$ .

La résolution, à chaque instant  $t+1$ , de l'équation :

$$A_{p-1}(t+1) = R_{p-1}^{-1}(t+1)S_{p-1}(t+1)$$

n'utilise pas la résolution effectuée à l'instant précédent qui a permis de mettre à jour le vecteur  $A_{p-1}(t)$ .

Or la matrice  $R_{p-1}(t+1)$  a été modifiée, entre l'instant  $t$  et l'instant  $t+1$ , à travers l'équation :

$$R_{p-1}(t+1) = \lambda R_{p-1}(t) + X_{p-1}(t+1)X_{p-1}^T(t+1)$$

Il est alors légitime de chercher à exprimer  $A_{p-1}(t+1)$  en fonction de  $A_{p-1}(t)$ .

Ainsi, en écrivant le filtre optimal au temps  $t+1$  :

$$A_{p-1}(t+1) = R_{p-1}^{-1}(t+1)S_{p-1}(t+1)$$

et en remplaçant  $S_{p-1}(t+1)$  par son expression en fonction de  $S_{p-1}(t)$ , il vient:

$$A_{p-1}(t+1) = R_{p-1}^{-1}(t+1) \left[ \lambda S_{p-1}(t) + X_{p-1}(t+1)x_p^T(t+1) \right]$$

or  $S_{p-1}(t) = R_{p-1}(t)A_{p-1}(t)$  d'où:

$$A_{p-1}(t+1) = R_{p-1}^{-1}(t+1) \left[ \lambda R_{p-1}(t)A_{p-1}(t) + X_{p-1}(t+1)x_p^T(t+1) \right]$$

$$A_{p-1}(t+1) = R_{p-1}^{-1}(t+1) \left[ \left( R_{p-1}(t+1) - X_{p-1}(t+1)X_{p-1}^T(t+1) \right) A_{p-1}(t) + X_{p-1}(t+1)x_p^T(t+1) \right]$$

d'où finalement:

$$A_{p-1}(t+1) = A_{p-1}(t) + R_{p-1}^{-1}(t+1)X_{p-1}(t+1)\left(x_p^T(t+1) - X_{p-1}^T(t+1)A_{p-1}(t)\right)$$

l'expression  $x_p^T(t+1) - X_{p-1}^T(t+1)A_{p-1}(t)$  représente la conjuguée de l'erreur de prédiction a priori, faite en utilisant le vecteur  $A_{p-1}(t)$ , optimal au temps  $t$ , pour filtrer le vecteur  $X_{p-1}(t+1)$  disponible au temps  $t+1$ .

En écrivant:  $e_p^{l \rightarrow p-1}(t+1) = x_p(t+1) - A_{p-1}^T(t)X_{p-1}(t+1)$  cette erreur, l'équation précédente devient:

$$A_{p-1}(t+1) = A_{p-1}(t) + R_{p-1}^{-1}(t+1)X_{p-1}(t+1)e_p^{l \rightarrow p-1T}(t+1)$$

Cette expression, qui porte le nom d'équation normale, permet de mettre à jour le vecteur  $A_{p-1}(t+1)$  à partir du vecteur  $A_{p-1}(t)$ .

### 3.3 Algorithme LMS

Une première solution consiste à supposer que la matrice  $R_{p-1}^{-1}(t+1)$  est constante au cours du temps et égale à  $\delta I_{p-1}$  où  $I_{p-1}$  représente la matrice identité et  $\delta$  un scalaire.

Il s'agit là d'une approximation importante et le critère ne va pas être rigoureusement optimisé. Cependant l'algorithme qui va en découler est sans doute l'algorithme le plus simple à mettre en œuvre et celui dont le coût de calcul est le plus faible. C'est de loin l'algorithme qui a été jusqu'à aujourd'hui le plus utilisé dans les équipements industriels.

Avec cette hypothèse l'équation normale se simplifie et devient :

$$A_{p-1}(t+1) = A_{p-1}(t) + \delta X_{p-1}(t+1)e_p^{l \rightarrow p-1T}(t+1)$$

Le vecteur  $X_{p-1}(t+1)$  et le scalaire  $x_p(t+1)$  étant disponibles à l'instant  $t+1$ , il est aisé de calculer:

$$e_p^{l \rightarrow p-1}(t+1) = x_p(t+1) - A_{p-1}^T(t)X_{p-1}(t+1) \text{ et de mettre à jour le vecteur } A_{p-1}(t+1).$$

Cette mise à jour demande finalement un coût de calcul proportionnel à  $O(p-1)$ . Cet algorithme est appelé algorithme du gradient stochastique (LMS).

<b>Quantités obtenues au temps <math>n</math> et disponibles au temps <math>t+1</math> :</b>	
vecteur des coefficients du filtre :	$A_{p-1}(t)$
<b>Nouvelles données au temps <math>n+1</math> :</b>	
vecteur de données reçues :	$X_{p-1}(t+1)$
signal de référence :	$x_p(t+1)$
<b>Calcul du nouveau filtre au temps <math>n+1</math> :</b>	
erreur de prédiction avant :	$e_p^{l \rightarrow p-1}(t+1) = x_p(t+1) - A_{p-1}^T(t)X_{p-1}(t+1)$
mise à jour du filtre au temps $t+1$ :	$A_{p-1}(t+1) = A_{p-1}(t) + \delta e_p^{l \rightarrow p-1 T}(t+1)X_{p-1}(t+1)$

**- Algorithme LMS -**

Il est à rapprocher de l'algorithme classique du gradient déterministe où un vecteur  $A_{p-1}(t+1)$  est adapté en descendant le long de la dérivée d'un critère au moyen d'une formule du type :

$$A_{p-1}(t+1) = A_{p-1}(t) - \frac{\delta}{2} \nabla(\text{critère})$$

en considérant comme critère la minimisation de l'erreur quadratique a priori, cette équation devient :

$$A_{p-1}(t+1) = A_{p-1}(t) - \frac{\delta}{2} \nabla \left( E \left[ \left( x_p(t+1) - A_{p-1}^T(t)X_{p-1}(t+1) \right) \left( x_p(t+1) - A_{p-1}^T(t)X_{p-1}(t+1) \right)^T \right] \right)$$

d'où en développant la dérivée :

$$A_{p-1}(t+1) = A_{p-1}(t) + \frac{\delta}{2} 2 \left( E \left[ X_{p-1}(t+1)x_p^T(t+1) \right] - E \left[ X_{p-1}(t+1)X_{p-1}^T(t+1) \right] A_{p-1}(t) \right)$$

Le remplacement des espérances par les valeurs instantanées donne alors :

$$A_{p-1}(t+1) = A_{p-1}(t) + \delta X_{p-1}(t+1)e_p^{l \rightarrow p-1 T}(t+1)$$

l'algorithme est alors appelé algorithme du gradient stochastique et le scalaire  $\delta$  représente le pas de descente de l'algorithme.



**Conditions de convergence de l'algorithme :**

Dans une première approche on peut considérer que l'algorithme est convergent dès lors que l'espérance de la valeur absolue de l'erreur a posteriori :

$$E\left[\left|\varepsilon_p^{l \rightarrow p-l}(t+1)\right|\right] = E\left[\left|x_p(t+1) - X_{p-l}^T(t+1)A_{p-l}(t+1)\right|\right]$$

est inférieure à la valeur absolue a priori :

$$E\left[\left|e_p^{l \rightarrow p-l}(t+1)\right|\right] = E\left[\left|x_p(t+1) - X_{p-l}^T(t+1)A_{p-l}(t)\right|\right]$$

Or sachant que :

$$\varepsilon_p^{l \rightarrow p-l}(t+1) = x_p(t+1) - X_{p-l}^T(t+1)\left[A_{p-l}(t) + \delta X_{p-l}(t+1)e_p^{l \rightarrow p-l}(t+1)\right]$$

il vient :

$$\varepsilon_p^{l \rightarrow p-l}(t+1) = e_p^{l \rightarrow p-l}(t+1) - \delta X_{p-l}^T(t+1)X_{p-l}(t+1)e_p^{l \rightarrow p-l}(t+1)$$

ou encore :

$$\varepsilon_p^{l \rightarrow p-l}(t+1) = e_p^{l \rightarrow p-l}(t+1)\left(1 - \delta X_{p-l}^T(t+1)X_{p-l}(t+1)\right)$$

La condition de convergence s'écrit alors :

$$\begin{aligned} E\left[\left|\varepsilon_p^{l \rightarrow p-l}(t+1)\right|\right] &< E\left[\left|e_p^{l \rightarrow p-l}(t+1)\right|\right] \\ \Rightarrow \left|1 - \delta E\left[X_{p-l}^T(t+1)X_{p-l}(t+1)\right]\right| &< 1 \end{aligned}$$

d'où:

$$0 < \delta < \frac{2}{NP_x}$$

où  $P_x$  représente la puissance de la composante  $x_i(t)$  de puissance maximale du vecteur  $X_{p-l}(t)$

(Dans le cas où les composantes de ce vecteur sont les valeurs décalées dans le temps d'un processus monodimensionnel  $x(t)$ ,  $P_x$  représente la puissance du processus  $x(t)$ .)

Une autre approche pour démontrer la convergence de l'algorithme consiste à introduire le filtre optimal

$$A_0 = R_{p-l}^{-1}(t+1)S_{p-l}(t+1), \text{ et la variable } \varepsilon(t+1) = A_{p-l}(t+1) - A_0$$

On a alors :

$$\varepsilon(t+1) = A_{p-l}(t) - A_0 + \delta X_{p-l}(t+1)e_p^{l \rightarrow p-l}(t+1)$$

On introduit l'erreur de filtrage qui aurait été commise avec le filtre optimal :

$$e_0(t+1) = x_p(t+1) - A_0^T X_{p-l}(t+1)$$

D'où :

$$\begin{aligned}\varepsilon(t+1) &= A_{p-1}(t) - A_0 + \delta X_{p-1}(t+1) \left( x_p(t+1) - A_{p-1}^T(t) X_{p-1}(t+1) \right)^T \\ \varepsilon(t+1) &= \varepsilon(t) + \delta X_{p-1}(t+1) \left( e_0(t+1) + A_0^T X_{p-1}(t+1) - A_{p-1}^T(t) X_{p-1}(t+1) \right)^T \\ \varepsilon(t+1) &= \varepsilon(t) + \delta X_{p-1}(t+1) \left( e_0(t+1) - \varepsilon(t)^T X_{p-1}(t+1) \right)^T \\ \varepsilon(t+1) &= \left( I_{p-1} - \delta X_{p-1}(t+1) X_{p-1}^T(t+1) \right) \varepsilon(t) + \delta X_{p-1}(t+1) e_0^T(t+1)\end{aligned}$$

Par définition de  $e_0(t+1)$ , on a  $E \left[ X_{p-1}(t+1) e_0^T(t+1) \right] = 0$  car  $e_0(t+1)$  représente l'innovation de  $x_p(t)$  par rapport à la base de prédiction  $X_{p-1}(t+1)$ .

En prenant alors l'espérance mathématique de l'équation précédente, il vient :

$$E[\varepsilon(t+1)] = \left( I_{p-1} - \delta R_{p-1}(t+1) \right) E[\varepsilon(t)]$$

L'algorithme LMS convergera en moyenne si et seulement si  $E \left[ A_{p-1}(t) \right] \xrightarrow{t \rightarrow \infty} A_0$

donc si  $E[\varepsilon(t)] \xrightarrow{t \rightarrow \infty} 0$

Donc si  $0 < \delta < \frac{2}{\lambda_{\max}}$  où  $\lambda_{\max}$  représente la valeur propre maximale de la matrice  $R_{p-1}(t)$ .

### **Critère optimisé:**

Le fait d'avoir approché  $R_{p-1}(t)$  par  $\delta I_{p-1}$  a conduit à optimiser un critère différent de celui de l'erreur quadratique moyenne pondérée.

En considérant la solution obtenue:

$$A_{p-1}(t+1) = A_{p-1}(t) + \delta X_{p-1}(t+1) \left( x_p(t+1) - X_{p-1}^T(t+1) A_{p-1}(t) \right)^T$$

il vient:

$$A_{p-1}(t+1) = \left( I_{p-1} - \delta X_{p-1}(t+1) X_{p-1}^T(t+1) \right) A_{p-1}(t) + \delta X_{p-1}(t+1) x_p(t+1)$$

en se plaçant dans le cas où l'algorithme a convergé, il vient pour  $t \rightarrow \infty$  :

$$E \left[ A_{p-1}(t+1) \right] = E \left[ A_{p-1}(t) \right]$$

or :  $E[X_{p-1}(t+1)X_{p-1}^T(t+1)] = R_{p-1}(t+1)$  et  $E[X_{p-1}(t+1)x_p(t+1)] = S_{p-1}(t+1)$

d'où en prenant l'espérance de l'équation précédente :

$$E[A_{p-1}(t+1)] = E[A_{p-1}(t)] = E[A_{p-1}(t)] - \delta \left( R_{p-1}(t+1)E[A_{p-1}(t)] - S_{p-1}(t+1) \right)$$

d'où:

$$\text{pour } t \rightarrow \infty \quad E[A_{p-1}(t)] = R_{p-1}^{-1}(t+1)S_{p-1}(t+1)$$

La solution obtenue grâce à l'algorithme du gradient stochastique converge donc *en moyenne* vers la solution optimale. L'algorithme n'optimise donc pas un critère des moindres carrés exacts mais un critère des moindres carrés moyens.

### **Variantes de l'algorithme**

L'algorithme du gradient stochastique possède un grand nombre de variantes dont l'emploi dépend essentiellement de l'application envisagée. On peut citer par exemple:

L'algorithme de gradient normalisé (NLMS):

$$A_{p-1}(t+1) = A_{p-1}(t) + \frac{\delta X_{p-1}(t+1)e_p^{l \rightarrow p-1}(t+1)}{X_{p-1}^T(t+1)X_{p-1}(t+1)}$$

L'algorithme du gradient basé sur le signe de l'erreur (Error Sign):

$$A_{p-1}(t+1) = A_{p-1}(t) + \delta X_{p-1}(t+1) \operatorname{sgn}(e_p^{l \rightarrow p-1}(t+1))$$

L'algorithme du gradient basé sur le signe de la donnée (data sign):

$$A_{p-1}(t+1) = A_{p-1}(t) + \delta \operatorname{sgn}(X_{p-1}(t+1))e_p^{l \rightarrow p-1}(t+1)$$

L'algorithme du signe (sign algorithm):

$$A_{p-1}(t+1) = A_{p-1}(t) + \delta \operatorname{sgn}(X_{p-1}(t+1))e_p^{l \rightarrow p-1}(t+1)$$

### Conclusion

L'algorithme du gradient stochastique est un algorithme simple dont le coût de calcul est proportionnel à l'ordre  $N$  du filtre à identifier. A condition de respecter un pas d'adaptation suffisamment faible, cet algorithme est stable et optimise un critère des moindres carrés moyens. C'est encore aujourd'hui l'algorithme de filtrage adaptatif le plus employé dans les applications temps réel.

Le problème principal dans son emploi réside dans le choix du pas d'adaptation  $\delta$ . Un pas faible entraîne une convergence lente souvent incompatible des applications envisagées (égalisation en évansion de fréquence par exemple). Un pas trop fort va conduire, quant à lui, à des résultats imprécis. Il existe un grand nombre de méthodes qui tendent à faire varier ce pas  $\delta$  au cours de la phase de convergence. Cela peut avoir pour effet d'accélérer la convergence mais il faut rester circonspect vis à vis de ces méthodes car la convergence de l'algorithme n'est assurée que dans le cas du pas fixe.

### 3.4 Algorithme RLS

L'algorithme du gradient stochastique partait de l'approximation:  $R_{p-1}(t) = \delta I_{p-1}$ . Cette simplification a permis de passer d'un coût de calcul proportionnel à  $O(p^3)$  à un coût de calcul proportionnel à  $p$ . En évitant cette simplification, il est cependant possible de résoudre l'équation:

$$A_{p-1}(t+1) = R_{p-1}^{-1}(t+1)S_{p-1}(t+1)$$

avec un coût de calcul proportionnel à  $p^2$ .

En effet:  $R_{p-1}(t+1) = \lambda R_{p-1}(t) + X_{p-1}(t+1)X_{p-1}^T(t+1)$

En utilisant un lemme d'inversion matricielle rappelé ci-dessous :

$$\begin{aligned} &\text{si: } A = B + CDC^T \\ &\text{alors: } A^{-1} = B^{-1} - B^{-1}C[D + C^T B^{-1}C]^{-1}C^T B^{-1} \end{aligned}$$

il vient :

$$R_{p-1}^{-1}(t+1) = \frac{1}{\lambda} R_{p-1}^{-1}(t) - \frac{1}{\lambda} R_{p-1}^{-1}(t)X_{p-1}(t+1) \left[ I + X_{p-1}^T(t+1) \frac{R_{p-1}^{-1}(t)}{\lambda} X_{p-1}(t+1) \right]^{-1} X_{p-1}^T(t+1) \frac{R_{p-1}^{-1}(t)}{\lambda}$$

d'où:

$$R_{p-1}^{-1}(t+1) = \frac{1}{\lambda} \left( R_{p-1}^{-1}(t) - \frac{R_{p-1}^{-1}(t) X_{p-1}(t+1) X_{p-1}^T(t+1) R_{p-1}^{-1}(t)}{\lambda + X_{p-1}^T(t+1) R_{p-1}^{-1}(t) X_{p-1}(t+1)} \right)$$

En multipliant à gauche par  $X_{p-1}(t+1)$  et en factorisant, il vient:

$$G_{p-1}(t+1) = R_{p-1}^{-1}(t+1) X_{p-1}(t+1) = \frac{1}{\lambda} R_{p-1}^{-1}(t) X_{p-1}(t+1) \left( I - \frac{X_{p-1}^T(t+1) R_{p-1}^{-1}(t) X_{p-1}(t+1)}{\lambda + X_{p-1}^T(t+1) R_{p-1}^{-1}(t) X_{p-1}(t+1)} \right)$$

d'où:

$$G_{p-1}(t+1) = \frac{R_{p-1}^{-1}(t) X_{p-1}(t+1)}{\lambda + X_{p-1}^T(t+1) R_{p-1}^{-1}(t) X_{p-1}(t+1)}$$

d'où l'algorithme suivant :

<b>Quantités obtenues au temps <math>t</math> et disponibles au temps <math>t+1</math> :</b>	
inverse de la matrice d'autocorrélation :	$R_{p-1}^{-1}(t)$
vecteur des coefficients du filtre :	$A_{p-1}(t)$
<b>Nouvelles données au temps <math>t+1</math> :</b>	
vecteur de données reçues :	$X_{p-1}(t+1)$
signal de référence :	$x_p(t+1)$
<b>Calcul du nouveau filtre au temps <math>n+1</math> :</b>	
erreur de prédiction avant :	$e_p^{l \rightarrow p-1}(t+1) = x_p(t+1) - A_{p-1}^T(t) X_{p-1}(t+1)$
calcul du gain de Kalman :	$G_{p-1}(t+1) = \frac{R_{p-1}^{-1}(t) X_{p-1}(t+1)}{\lambda + X_{p-1}^T(t+1) R_{p-1}^{-1}(t) X_{p-1}(t+1)}$
mise à jour du filtre au temps $t+1$ :	$A_{p-1}(t+1) = A_{p-1}(t) + G_{p-1}(t+1) e_p^{l \rightarrow p-1}(t+1)$
propagation de l'inverse de $R_{p-1}(t+1)$ :	$R_{p-1}^{-1}(t+1) = \frac{1}{\lambda} \left( R_{p-1}^{-1}(t) - G_{p-1}(t+1) X_{p-1}^T(t+1) R_{p-1}^{-1}(t) \right)$

**- Algorithme RLS -**

Cet algorithme conduit à la mise à jour du vecteur  $A_{p-1}(t+1)$  avec un coût de calcul proportionnel à  $O(p^2)$ .

L'inversion de matrice a été évitée en propageant le gain de Kalman  $G_{p-1}(t+1)$  récursivement au cours du temps.

Pour  $t = 0$ , la matrice  $R_{p-1}^{-1}(0)$  est choisie égale à  $\frac{1}{\delta^2} I_{p-1}$  où  $\delta$  représente un scalaire petit devant 1.

Cet algorithme, à la différence de l'algorithme du gradient stochastique, optimise rigoureusement le critère des moindres carrés.

L'algorithme RLS s'applique aussi bien au cas d'un signal *multidimensionnel* :

$$X_{p-1}(t) = [x_1(t), x_2(t), \dots, x_{p-1}(t)]^T$$

qu'au cas d'un signal *monodimensionnel* dont les valeurs sont conservés avec un décalage temporel :

$$X_{p-1}(t) = [x(t), x(t-1), \dots, x(t-N+1)]^T$$

Cependant dans ce dernier cas, l'algorithme n'utilise pas cette propriété de décalage des variables. En utilisant cette propriété, il est possible d'optimiser l'algorithme et d'obtenir une version rapide (coût de calcul proportionnel à N).

### **3.4.1 Algorithme des moindres carrés récurrents rapides (Fast Kalman)**

La démonstration des algorithmes rapides: Fast Kalman, Faest et dérivés demandant un développement conséquent, ce paragraphe présente juste l'algorithme dont on trouvera la démonstration complète dans la référence [1].

( Attention au changement de variable par rapport aux paragraphes précédents, ici le filtre recherché est noté  $H_{p-1}(t)$  )

<b>Quantités disponibles au temps n:</b>	
vecteur des coefficients du filtre adaptatif :	$H_{p-1}(t)$
vecteur des coefficients de prédiction avant :	$A_{p-1}(t)$
vecteur des coefficients de prédiction arrière:	$B_{p-1}(t)$
vecteur de données à N éléments:	$X_{p-1}(t)$
gain d'adaptation (gain de Kalman):	$G_{p-1}(t)$
<b>Nouvelles données au temps n+1:</b>	
donnée reçue :	$x(t+1)$
signal de référence :	$x_p(t+1)$
<b>Calcul du nouveau gain d'adaptation:</b>	
erreur de prédiction avant a priori :	
	$e_a(t+1) = x(t+1) - A_{p-1}^T(t)X_{p-1}(t)$
mise à jour des coefficients de prédiction avant :	
	$A_{p-1}(t+1) = A_{p-1}(t) + G_{p-1}(t)e_a(t+1)$
erreur de prédiction avant a posteriori :	
	$\varepsilon_a(t+1) = x(t+1) - A_{p-1}^T(t+1)X_{p-1}(t)$
énergie de prédiction avant :	
	$E_a(t+1) = \lambda E_a(t) + e_a(t+1)\varepsilon_a(t+1)$
gain d'adaptation intermédiaire :	
	$G_I(t+1) = \begin{bmatrix} 0 \\ G_{p-1}(t) \end{bmatrix} + \frac{\varepsilon_a(t+1)}{E_a(t+1)} \begin{bmatrix} I \\ -A_{p-1}(t+1) \end{bmatrix} = \begin{bmatrix} M_{p-1}(t+1) \\ m(t+1) \end{bmatrix}$
erreur de prédiction arrière a priori :	
	$e_b(t+1) = x(t+1-p+1) - B_{p-1}^T(t)X_{p-1}(t+1)$
mise à jour du gain d'adaptation :	
	$G_{p-1}(t+1) = \frac{I}{I - m(t+1)e_b(t+1)} \begin{bmatrix} M_{p-1}(t+1) + m(t+1)B_{p-1}(t) \\ m(t+1) \end{bmatrix}$
mise à jour des coefficients de prédiction arrière :	
	$B_{p-1}(t+1) = B_{p-1}(t) + G_{p-1}(t+1)e_b(t+1)$
<b>Opération de filtrage:</b>	
calcul de l'erreur de sortie :	
	$e(t+1) = x_p(t+1) - H_{p-1}^T(t)X(t+1)$
mise à jour des coefficients :	
	$H_{p-1}(t+1) = H_{p-1}(t) + G_{p-1}(t+1)e(t+1)$

### 3.5 Algorithme du Treillis Spatial

#### 3.5.1 Principe

Le principe de l'algorithme du treillis spatial consiste à orthogonaliser la base de prédiction constituée par les signaux capteurs  $\{x_1(t), x_2(t), \dots, x_p(t)\}$ , puis à prédire un signal de référence  $y(t)$  à partir de cette base orthogonalisée. Pratiquement la prédiction de  $y(t)$  s'effectue suivant le même mécanisme que l'orthogonalisation des signaux capteurs. Il suffit pour cela de considérer que ce signal de référence est un signal capteur supplémentaire  $x_{p+1}(t) = y(t)$  et de le traiter comme les autres signaux capteurs.

L'algorithme propage un vecteur d'erreur :

$$\mathbf{E}_p(t) = L_p^{-1}(t) \cdot X_p(t) \quad (54.)$$

expression dans laquelle  $L_p(t)$  est obtenue par la décomposition  $LDU$  de l'estimation de la matrice d'autocorrélation  $R_p(t)$  du vecteur d'entrée  $X_p(t)$ .

$$R_p(t) = L_p(t) \cdot D_p(t) \cdot U_p(t) \quad (55.)$$

#### 3.5.2 Décomposition LDU

On note :

$$\text{Le vecteur d'entrée : } X_p(t) = \begin{pmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_p(t) \end{pmatrix}$$

L'estimation au temps  $t$  de la matrice d'autocorrélation du signal d'entrée :  $R_p(t) = \sum_{i=1}^t \lambda^{t-i} \cdot X_p(i) \cdot X_p^T(i)$

L'autocorrélation de  $x_{p+1}(t)$  :  $r_{p+1}^0(t) = \sum_{i=1}^t \lambda^{t-i} x_{p+1}(i) \cdot x_{p+1}^T(i)$

Le vecteur d'intercorrélation de la composante  $x_{p+1}(t)$  avec les composantes d'indice inférieur :

$$r_{p+1}^{1 \rightarrow p}(t) = \sum_{i=1}^t \lambda^{t-i} X_p(i) \cdot x_{p+1}^T(i)$$

#### Récursion en ordre

Ecrivons maintenant la décomposition  $LDU$  de la matrice  $R_{p+1}(t)$  en fonction de  $R_p(t)$ , afin de trouver  $L_{p+1}(t)$  et  $D_{p+1}(t)$  en fonction de  $L_p(t)$  et  $D_p(t)$



$$R_{p+1}(t) = L_{p+1}(t) \cdot D_{p+1}(t) \cdot L_{p+1}^T(t)$$

On considère alors les partitionnements suivants :

$$R_{p+1}(t) = \begin{bmatrix} R_p(t) & r_{p+1}^{l \rightarrow p}(t) \\ r_{p+1}^{l \rightarrow p^T}(t) & r_{p+1}^0(t) \end{bmatrix}, L_{p+1}(t) = \begin{bmatrix} L_p(t) & 0 \\ c_{p+1}^{l \rightarrow p}(t) & I \end{bmatrix}, D_{p+1}(t) = \begin{bmatrix} D_p(t) & 0 \\ 0 & \alpha_{p+1}(t) \end{bmatrix}$$

En identifiant, on obtient :

$$c_{p+1}^{l \rightarrow p}(t) = D_p^{-1}(t) \cdot L_p^{-1}(t) \cdot r_{p+1}^{l \rightarrow p}(t) \quad (56.)$$

et :

$$\alpha_{p+1}(t) = r_{p+1}^0 - c_{p+1}^{l \rightarrow p^T}(t) \cdot D_p(t) \cdot c_{p+1}^{l \rightarrow p}(t) \quad (57.)$$

### 3.5.3 Erreur a priori, erreur a posteriori

#### Erreur a priori

On définit l'erreur a priori par :

$$E_{p+1}(t) = L_{p+1}^{-1}(t-I) \cdot X_{p+1}(t) \quad (58.)$$

On considère alors les partitionnements suivants :

$$L_{p+1}^{-1}(t-I) = \begin{bmatrix} L_p^{-1}(t-I) & 0 \\ -c_{p+1}^{l \rightarrow p^T}(t-I) \cdot L_p^{-1}(t-I) & I \end{bmatrix}, E_{p+1}(t) = \begin{bmatrix} E_p(t) \\ e_{p+1}^{l \rightarrow p}(t) \end{bmatrix}, X_{p+1}(t) = \begin{bmatrix} X_p(t) \\ x_{p+1}(t) \end{bmatrix}$$

Par simple identification, on en déduit :

$$e_{p+1}^{l \rightarrow p}(t) = x_{p+1}(t) - c_{p+1}^{l \rightarrow p^T}(t-I) \cdot E_p(t) \quad (59.)$$

En utilisant  $c_{p+1}^{l \rightarrow p}(t-I) = D_p^{-1}(t-I) \cdot L_p^{-1}(t-I) \cdot r_{p+1}^{l \rightarrow p}(t-I)$ , il vient :

$$e_{p+1}^{l \rightarrow p}(t) = x_{p+1}(t) - r_{p+1}^{l \rightarrow p^T}(t-I) L_p^{-T}(t-I) D_p^{-1}(t-I) \cdot E_p(t) \quad (60.)$$

Et en remplaçant  $E_p(t)$  par  $L_p^{-1}(t-I) X_p(t)$ , il vient :

$$e_{p+1}^{l \rightarrow p}(t) = x_{p+1}(t) - r_{p+1}^{l \rightarrow p^T}(t-I) L_p^{-T}(t-I) D_p^{-1}(t-I) \cdot L_p^{-1}(t-I) X_p(t) \quad (61.)$$

Ou encore :

$$e_{p+1}^{l \rightarrow p}(t) = x_{p+1}(t) - r_{p+1}^{l \rightarrow p^T}(t-I) R_p^{-1}(t-I) X_p(t) \quad (62.)$$

Cette expression peut encore s'écrire :

$$e_{p+1}^{l \rightarrow p}(t) = x_{p+1}(t) - A_p^T(t-I) X_p(t) \quad (63.)$$

avec :  $A_p(t) = R_p^{-1}(t-I)r_{p+1}^{l \rightarrow p}(t-I)$

Il s'agit bien d'une erreur *a priori* car l'orthogonalisation des signaux capteurs est effectuée avant la remise à jour du filtre.

### **Erreur a posteriori**

On définit l'erreur a posteriori par :

$$\mathbf{E}_{p+1}(t) = L_{p+1}^{-1}(t)X_{p+1}(t) \quad (64.)$$

.En utilisant les partitionnements suivants :

$$L_{p+1}^{-1}(t) = \begin{bmatrix} L_p^{-1}(t) & 0 \\ -c_{p+1}^{l \rightarrow p^T}(t).L_p^{-1}(t) & I \end{bmatrix}, \mathbf{E}_{p+1}(t) = \begin{bmatrix} \mathbf{E}_p(t) \\ \varepsilon_{p+1}^{l \rightarrow p}(t) \end{bmatrix}, X_{p+1}(t) = \begin{bmatrix} X_p(t) \\ x_{p+1}(t) \end{bmatrix}$$

Par simple identification, on en déduit sur la dernière colonne

$$\varepsilon_{p+1}^{l \rightarrow p}(t) = x_{p+1}(t) - c_{p+1}^{l \rightarrow p^T}(t).\mathbf{E}_p(t) \quad (65.)$$

Il s'agit bien d'une erreur *a posteriori* car l'orthogonalisation des signaux capteurs est effectuée après la remise à jour du filtre.

### **3.5.4 Mise à jour des matrices L et D.**

On définit  $\nabla L_p(t)$  la matrice de passage de  $L_p(t-I)$  vers  $L_p(t)$

$$\nabla L_p(t) = L_p^{-1}(t-I).L_p(t) \quad (66.)$$

Comme les matrices  $L_p^{-1}(t-I)$  et  $L_p(t)$  sont triangulaires inférieures, avec des 1 sur la diagonale principale,  $\nabla L_p(t)$  l'est aussi.

On peut remarquer que  $\nabla L_p(t)$  est aussi la matrice de passage de l'erreur à posteriori vers l'erreur à priori. En effet, on a :

$$\nabla L_p(t).\mathbf{E}_p(t) = L_p^{-1}(t-I).L_p(t).L_p^{-1}(t).X_p(t) = E_p(t) \quad (67.)$$

### **Calcul de $\nabla L_p(t)$**

On part de l'équation de remise à jour de la matrice d'autocorrélation du signal d'ordre  $p$ .

$$R_p(t) = \lambda.R_p(t-I) + X_p(t).X_p^T(t) \quad (68.)$$

$$L_p(t).D_p(t).L_p^T(t) = \lambda.L_p(t-I).D_p(t-I).L_p^T(t-I) + X_p(t).X_p^T(t) \quad (69.)$$

En multipliant à gauche et à droite par respectivement  $L_p^{-1}(t-I)$  et  $L_p^{-T}(t-I)$ , on obtient :

$$\nabla L_p(t).D_p(t).\nabla L_p^T(t) = \lambda.D_p(t-I) + E_p(t).E_p^T(t) \quad (70.)$$

Comme  $\nabla L_p(t)$  est triangulaire inférieure avec des 1 sur sa diagonale principale, cette dernière équation montre, par unicité de la décomposition  $LDU$ , que  $\nabla L_p(t)$  s'obtient par la décomposition  $LDU$  de la matrice

$$\Gamma_p(t) = \lambda.D_p(t-I) + E_p(t).E_p^T(t) \quad (71.)$$

### Décomposition $LDU$ de $\Gamma_p(t)$

Pour ce faire, nous allons utiliser une récursion en ordre sur  $\Gamma_p(t)$  en utilisant l'égalité :

$$\Gamma_{p+1}(t) = \nabla L_{p+1}(t).D_{p+1}(t).\nabla L_{p+1}^T(t) = \lambda.D_{p+1}(t-I) + E_{p+1}(t).E_{p+1}^T(t) \quad (72.)$$

On note :

$$\nabla L_{p+1}(t) = \begin{bmatrix} \nabla L_p(t) & 0 \\ \nabla c_{p+1}^{l \rightarrow p^T}(t) & 1 \end{bmatrix} \quad (73.)$$

En développant l'équation précédente, il vient :

$$\Gamma_{p+1}(t) = \begin{bmatrix} \nabla L_p(t).D_p(t).\nabla L_p^T(t) & \nabla L_p(t).D_p(t).\nabla c_{p+1}^{l \rightarrow p^T}(t) \\ \nabla c_{p+1}^{l \rightarrow p^T}(t).D_p(t).\nabla L_p^T(t) & \nabla c_{p+1}^{l \rightarrow p^T}(t).D_p(t).\nabla c_{p+1}^{l \rightarrow p^T}(t) + \alpha_{p+1}(t) \end{bmatrix} \quad (74.)$$

$$\Gamma_{p+1}(t) = \begin{bmatrix} \Gamma_p(t) & E_p(t).e_{p+1}^{l \rightarrow p^T}(t) \\ e_{p+1}^{l \rightarrow p^T}(t).E_p^T(t) & \lambda.\alpha_{p+1}(t-I) + e_{p+1}^{l \rightarrow p^T}(t).e_{p+1}^{l \rightarrow p^T}(t) \end{bmatrix} \quad (75.)$$

Par simple identification, on tire après simplification :

$$\nabla c_{p+1}^{l \rightarrow p^T}(t) = D_p^{-1}(t).\mathbf{E}_p(t).e_{p+1}^{l \rightarrow p^T}(t) \quad (76.)$$

et

$$\alpha_{p+1}(t) = \lambda.\alpha_{p+1}(t-I) + e_{p+1}^{l \rightarrow p^T}(t).\left(1 - \mathbf{E}_p^T(t).D_p^{-1}(t).\mathbf{E}_p(t)\right).e_{p+1}^{l \rightarrow p^T}(t) \quad (77.)$$

### Equations de mise à jour de la matrice $D$

En posant

$$\gamma_p(t) = 1 - \mathbf{E}_p^T(t).D_p^{-1}(t).\mathbf{E}_p(t). \quad (78.)$$

on déduit l'équation de remise à jour de la matrice  $D_p(t)$

$$\alpha_{p+1}(t) = \lambda.\alpha_{p+1}(t-I) + \gamma_p(t).e_{p+1}^{l \rightarrow p^T}(t).e_{p+1}^{l \rightarrow p^T}(t) \quad (79.)$$

### Mise à jour de $\gamma_p(t)$

Reprenons l'équation

$$\nabla L_{p+1}(t). \mathbf{E}_{p+1}(t) = E_{p+1}(t) \quad (80.)$$

En développant la dernière ligne, il vient

$$e_{p+1}^{l \rightarrow p}(t) = \nabla c_{p+1}^{l \rightarrow p T}(t). \mathbf{E}_p(t) + \varepsilon_{p+1}^{l \rightarrow p}(t) \quad (81.)$$

$$e_{p+1}^{l \rightarrow p}(t) = e_{p+1}^{l \rightarrow p}(t). (\mathbf{E}_p^T(t). D_p^{-1}(t). \mathbf{E}_p(t)) + \varepsilon_{p+1}^{l \rightarrow p}(t) \quad (82.)$$

$$e_{p+1}^{l \rightarrow p}(t) = e_{p+1}^{l \rightarrow p}(t). (1 - \gamma_p(t)) + \varepsilon_{p+1}^{l \rightarrow p}(t) \quad (83.)$$

d'où la relation qui lie l'erreur a priori et l'erreur a posteriori

$$\varepsilon_{p+1}^{l \rightarrow p}(t) = e_{p+1}^{l \rightarrow p}(t). \gamma_p(t) \quad (84.)$$

### Récursion en ordre sur $\gamma_p(t)$

$$\gamma_{p+1}(t) = 1 - \mathbf{E}_{p+1}^T(t). D_{p+1}^{-1}(t). \mathbf{E}_{p+1}(t) \quad (85.)$$

$$\gamma_{p+1}(t) = 1 - \mathbf{E}_p^T(t). D_p^{-1}(t). \mathbf{E}_p(t) - \frac{\varepsilon_{p+1}^{l \rightarrow p T}(t). \varepsilon_{p+1}^{l \rightarrow p}(t)}{\alpha_{p+1}(t)} \quad (86.)$$

d'où l'équation de mise à jour de  $\gamma_p(t)$

$$\gamma_{p+1}(t) = \gamma_p(t) - \gamma_p^2(t) \frac{e_{p+1}^{l \rightarrow p T}(t). e_{p+1}^{l \rightarrow p}(t)}{\alpha_{p+1}(t)} \quad (87.)$$

### Calcul de $\nabla L_p^{-1}(t)$

De même que pour le calcul de  $\nabla L_p(t)$ , on part de l'équation de remise à jour de la matrice d'autocorrélation du signal d'ordre p, i.e. :

$$L_p(t). D_p(t). L_p^T(t) = \lambda. L_p(t-1). D_p(t-1). L_p^T(t-1) + X_p(t). X_p^T(t) \quad (88.)$$

On fait apparaître la matrice  $\nabla L_p^{-1}(t)$  en multipliant à gauche et à droite par respectivement  $L_p^{-1}(t)$  et  $L_p^{-T}(t)$ , on obtient :

$$D_p(t) = \lambda. \nabla L_p^{-1}(t). D_p(t-1). \nabla L_p^{-T}(t) + \mathbf{E}_p(t). \mathbf{E}_p^T(t) \quad (89.)$$

De la même façon que précédemment, cette équation montre que  $\nabla L_p^{-1}(t)$  s'obtient par la décomposition *LDU* de la matrice

$$\Lambda_p(t) = D_p(t) - \mathbf{E}_p(t). \mathbf{E}_p^T(t) \quad (90.)$$

### Décomposition *LDU* de $\Lambda_p(t)$

Pour ce faire, nous allons utiliser une récursion en ordre sur  $\Lambda_p(t)$  en utilisant l'égalité :

$$\Lambda_{p+1}(t) = \lambda \cdot \nabla L_{p+1}^{-1}(t) \cdot D_{p+1}(t-I) \cdot \nabla L_{p+1}^T(t) = D_{p+1}(t) - \mathbf{\epsilon}_{p+1}(t) \cdot \mathbf{\epsilon}_{p+1}^T(t) \quad (91.)$$

En vertu du lemme d'inversion matriciel partitionnée, on a :

$$\nabla L_{p+1}^{-1}(t) = \begin{bmatrix} \nabla L_p^{-1}(t) & 0 \\ -\nabla c_{p+1}^{l \rightarrow p^T}(t) \cdot \nabla L_p^{-1}(t) & I \end{bmatrix} \quad (92.)$$

$$\begin{bmatrix} \lambda \cdot \nabla L_p^{-1}(t) \cdot D_p(t-I) \cdot \nabla L_p^{-T}(t) & -\lambda \cdot \nabla L_p^{-1}(t) \cdot D_p(t-I) \cdot \nabla L_p^{-T}(t) \cdot \nabla c_{p+1}^{l \rightarrow p}(t) \\ -\lambda \cdot \nabla c_{p+1}^{l \rightarrow p^T}(t) \cdot \nabla L_p^{-1}(t) \cdot D_p(t-I) \cdot \nabla L_p^{-T}(t) & \lambda \cdot \alpha_{p+1}(t-I) + \lambda \cdot \nabla c_{p+1}^{l \rightarrow p^T}(t) \cdot \nabla L_p^{-1}(t) \cdot D_p(t-I) \cdot \nabla L_p^{-T}(t) \cdot \nabla c_{p+1}^{l \rightarrow p}(t) \end{bmatrix} \quad (93.)$$

En effectuant l'équation précédente, il vient :

$$\begin{bmatrix} \Lambda_p(t) & -\mathbf{\epsilon}_p(t) \cdot \mathbf{\epsilon}_{p+1}^{l \rightarrow p^T}(t) \\ -\mathbf{\epsilon}_{p+1}^{l \rightarrow p}(t) \cdot \mathbf{\epsilon}_p^T(t) & \alpha_{p+1}(t-I) - \mathbf{\epsilon}_{p+1}^{l \rightarrow p}(t) \cdot \mathbf{\epsilon}_{p+1}^{l \rightarrow p^T}(t) \end{bmatrix} = \Lambda_{p+1}(t) \quad (94.)$$

On en déduit l'équation suivante :

$$\nabla c_{p+1}^{l \rightarrow p}(t) = \lambda^{-1} \cdot \nabla L_p^T(t) \cdot D_p^{-1}(t-I) \cdot E_p(t) \cdot \mathbf{\epsilon}_{p+1}^{l \rightarrow p^T}(t) \quad (95.)$$

et une nouvelle équation de mise à jour pour la matrice  $D_p(t)$

$$\alpha_{p+1}(t) = \lambda \cdot \alpha_{p+1}(t-I) + \mathbf{\epsilon}_{p+1}^{l \rightarrow p}(t) \cdot \left( I + \lambda^{-1} \cdot E_p^T(t) \cdot D_p^{-1}(t-I) \cdot E_p(t) \right) \cdot \mathbf{\epsilon}_{p+1}^{l \rightarrow p^T}(t) \quad (96.)$$

Cette dernière équation pourrait être utilisée pour développer un algorithme basé sur l'erreur a priori. Nous ne développerons pas ces calculs, car ils conduisent à un algorithme dont le coût de calcul est plus important.

### Expression de $\nabla L_{p+1}^{-1}(t)$

$$\nabla L_{p+1}^{-1}(t) = \begin{bmatrix} \nabla L_p^{-1}(t) & 0 \\ -\lambda^{-1} \cdot \mathbf{\epsilon}_{p+1}^{l \rightarrow p} \cdot E_p^T(t) \cdot D_p^{-1}(t-I) & I \end{bmatrix} \quad (97.)$$

### Mise à jour de la matrice $L$

On cherche ici à trouver une relation entre le filtre  $c_{p+1}^{l \rightarrow p}(t)$  et  $c_{p+1}^{l \rightarrow p}(t-I)$

On part de l'équation :

$$c_{p+1}^{l \rightarrow p}(t) = D_p^{-1}(t) \cdot L_p^{-1}(t) \cdot r_{p+1}^{l \rightarrow p}(t)$$

L'équation de mise à jour de  $r_{p+1}^{l \rightarrow p}(t)$  s'écrit :

$$r_{p+1}^{l \rightarrow p}(t) = \lambda \cdot r_{p+1}^{l \rightarrow p}(t-1) + X_p(t) \cdot x_{p+1}^T(t) \quad (98.)$$

d'où

$$c_{p+1}^{l \rightarrow p}(t) = D_p^{-1}(t) \cdot L_p^{-1}(t) \cdot (\lambda \cdot r_{p+1}^{l \rightarrow p}(t-1) + X_p(t) \cdot x_{p+1}^T(t)) \quad (99.)$$

$$c_{p+1}^{l \rightarrow p}(t) = D_p^{-1}(t) \cdot \nabla L_p^{-1}(t) \cdot D_p(t-1) \cdot (\lambda \cdot D_p^{-1}(t-1) \cdot L_p^{-1}(t-1) \cdot r_{p+1}^{l \rightarrow p}(t-1)) + D_p^{-1}(t) \cdot \mathbf{E}_p(t) \cdot x_{p+1}^T(t) \quad (100.)$$

$$c_{p+1}^{l \rightarrow p}(t) = D_p^{-1}(t) \cdot (\lambda \cdot \nabla L_p^{-1}(t) \cdot D_p(t-1) \cdot c_{p+1}^{l \rightarrow p}(t-1) + \mathbf{E}_p(t) \cdot x_{p+1}^T(t)) \quad (101.)$$

L'équation précédente peut se réécrire, après calcul

$$\begin{bmatrix} c_{p+1}^{l \rightarrow p-1}(t) \\ c_{p+1}^p(t) \end{bmatrix} = \begin{bmatrix} D_{p-1}^{-1}(t) \cdot \{ \lambda \cdot \nabla L_{p-1}^{-1}(t) \cdot D_{p-1}(t-1) \cdot c_{p+1}^{l \rightarrow p-1}(t-1) + \mathbf{E}_{p-1}(t) \cdot x_{p+1}^T(t) \} \\ \lambda \cdot \frac{\alpha_p(t-1)}{\alpha_p(t)} \cdot c_{p+1}^p(t-1) + \frac{\mathbf{e}_p^{l \rightarrow p-1}(t)}{\alpha_p(t)} (x_{p+1}^T(t) - E_{p-1}^T(t) \cdot c_{p+1}^{l \rightarrow p-1}(t-1)) \end{bmatrix} \quad (102.)$$

A ce stade, on a donc trouvé une équation de mise à jour pour le coefficient  $c_{p+1}^p(t)$  de prédiction sur la  $p^{\text{ième}}$  composante du vecteur de sortie du signal d'entrée d'indice  $p+1$ . On peut en fait montrer que cette équation est valable quelque soit le signal d'entrée d'indice  $i$ , avec  $i \geq p+1$ .

On note  $c_i^p(t)$  le coefficient de prédiction sur la  $p^{\text{ième}}$  composante du signal de sortie (i.e. l'erreur a priori  $E(t)$ ) du signal d'entrée d'indice  $i$ , avec  $i > p$ . Ce coefficient représente également la  $p^{\text{ième}}$  composante de  $c_i^{l \rightarrow i-1}(t)$ .

$$L_{p+i}(t) = \begin{bmatrix} L_p(t) & 0 & & & \\ c_{p+1}^{l \rightarrow p^T}(t) & 1 & 0 & & \\ \dots & & 1 & & \\ c_i^{l \rightarrow p^T}(t) & c_i^{p+1}(t) & \dots & c_i^{i-1}(t) & 1 \end{bmatrix}$$

$$R_i(t) = \begin{bmatrix} R_p(t) & r_{p+1}^{l \rightarrow p}(t) & \dots & r_i^{l \rightarrow p}(t) \\ r_{p+1}^{l \rightarrow p^T}(t) & r_0^{p+1}(t) & \dots & r_i^{p+1^T}(t) \\ \dots & & \dots & \\ r_{p+i}^{l \rightarrow p^T}(t) & r_i^{p+1^T}(t) & \dots & r_i^{p+i-1^T}(t) & r_0^i(t) \end{bmatrix}$$

Le vecteur  $r_i^{l \rightarrow p}(t)$  n'est rien d'autre que le vecteur de corrélation de  $x_i(t)$  avec  $X_p(t)$

En effectuant  $L_i(t)D_i(t)L_i^T(t)$ , on trouve

$$R_{p+i}(t) = \begin{bmatrix} L_p(t)D_p(t)L_p^T(t) & & L_p(t).D_p(t).c_i^{l \rightarrow p}(t) \\ & \dots & \\ c_i^{l \rightarrow p T}(t).D_p(t).L_p^T(t) & & \end{bmatrix} \quad (103.)$$

par simple identification, on déduit :

$$c_{p+i}^{l \rightarrow p}(t) = D_p^{-1}(t).L_p^{-1}(t).r_i^{l \rightarrow p}(t) \quad (104.)$$

De plus, l'équation de mise à jour de  $r_i^{l \rightarrow p}(t)$  s'écrit

$$r_i^{l \rightarrow p}(t) = \lambda.r_i^{l \rightarrow p}(t-1) + X_p(t).x_i^T(t) \quad (105.)$$

Ces deux dernières équations montrent que l'on peut refaire exactement le même calcul en remplaçant  $c_{p+i}^{l \rightarrow p}(t)$  par  $c_i^{l \rightarrow p}(t)$ . On aboutit ainsi aux équations de mise à jours des  $c_{p+i}^p(t)$   $1 \leq i \leq N-p$ , où  $N$  est le nombre de signaux capteurs  $x_i(t)$ .

$$c_i^p(t) = \lambda \cdot \frac{\alpha_p(t-1)}{\alpha_p(t)} \cdot c_i^p(t-1) + \frac{1}{\alpha_p(t)} (x_i^T(t) - E_{p-1}^T(t).c_i^{l \rightarrow p-1}(t-1)).\epsilon_p^{l \rightarrow p-1}(t) \quad (106.)$$

On pose :

$$e_i^{l \rightarrow p-1}(t) = x_i(t) - c_i^{l \rightarrow p-1 T}(t-1).E_{p-1}(t). \quad (107.)$$

Cette dernière équation est également vraie pour  $i = p$ . Elle n'est alors rien d'autre que l'expression de l'erreur a priori

**Récursion en ordre sur  $e_i^{l \rightarrow p}(t)$**

$$e_i^{l \rightarrow p}(t) = x_i(t) - \begin{bmatrix} c_i^{l \rightarrow p-1 T}(t-1) & c_i^p T(t-1) \end{bmatrix} \cdot \begin{bmatrix} E_{p-1}(t) \\ e_p^{l \rightarrow p-1}(t) \end{bmatrix} \quad (108.)$$

En posant  $e_i^{l \rightarrow 0}(t) = x_i(t)$ , on tire une équation de récursion en ordre sur  $e_i^{l \rightarrow p}(t)$ , i.e.

$$e_i^{l \rightarrow p}(t) = e_i^{l \rightarrow p-1}(t) - c_i^p T(t-1).e_p^{l \rightarrow p-1}(t) \quad (109.)$$

pour  $1 \leq p \leq N$  et  $p+1 \leq i \leq N$

L'équation de remise à jour des coefficients du filtre s'écrit alors :

$$c_i^p(t) = \lambda \cdot \frac{\alpha_p(t-1)}{\alpha_p(t)} \cdot c_i^p(t-1) + \frac{1}{\alpha_p(t)} e_i^{l \rightarrow p-1T}(t) \cdot \varepsilon_p^{l \rightarrow p-1}(t) \quad (110.)$$

En utilisant le fait que  $\alpha_p(t) = \lambda \cdot \alpha_p(t-1) + \varepsilon_p^{l \rightarrow p-1T}(t) \cdot e_p^{l \rightarrow p-1}(t)$  et la récursion en ordre sur  $e_i^{l \rightarrow p}(t)$ , on aboutit à

$$c_i^p(t) = c_i^p(t-1) + \frac{\gamma_{p-1}(t)}{\alpha_p(t)} e_i^{l \rightarrow pT}(t) \cdot e_p^{l \rightarrow p-1}(t) \quad (111.)$$

### 3.5.5 Algorithme

Dans le cas d'un treillis servant à la prédiction d'une référence, si N est le nombre de capteurs, on note pour  $1 \leq p \leq N$ ,  $x_p(t)$  le signal sur le  $p^{\text{ième}}$  capteur et  $x_{N+1}(t)$  le signal de référence

Initialisation à t=0	$1 \leq p \leq N+1 \quad \alpha_p(t) = 0$ $1 \leq p \leq N+1 \text{ et } 1 \leq i \leq N+1 \quad c_i^p(t) = 0$
Initialisation à t	$\gamma_0(t) = 1$ $1 \leq p \leq N+1 \quad e_p^{l \rightarrow 0}(t) = x_p(t)$
	$1 \leq p \leq N+1 \left\{ \begin{array}{l} \alpha_p(t) = \lambda \cdot \alpha_p(t-1) + \gamma_{p-1}(t) \cdot e_p^{l \rightarrow p-1}(t) \cdot e_p^{l \rightarrow p-1T}(t) \\ \gamma_p(t) = \gamma_{p-1}(t) - \gamma_{p-1}^2(t) \frac{e_p^{l \rightarrow p-1T}(t) \cdot e_p^{l \rightarrow p-1}(t)}{\alpha_p(t)} \\ p+1 \leq i \leq N+1 \left\{ \begin{array}{l} e_i^{l \rightarrow p}(t) = e_i^{l \rightarrow p-1}(t) - c_i^{pT}(t) \cdot e_p^{l \rightarrow p-1}(t) \\ c_i^p(t) = c_i^p(t-1) + \frac{\gamma_{p-1}(t)}{\alpha_p(t)} e_i^{l \rightarrow p-1T}(t) \cdot e_p^{l \rightarrow p-1}(t) \end{array} \right. \end{array} \right.$



### 3.6 Conclusion

L'optimisation en temps réel d'un filtre adaptatif se heurte principalement à des problèmes de coût de calcul. Plusieurs cas sont à envisager:

- Si la puissance de calcul est importante et le débit des signaux n'est pas trop rapide, il est possible d'envisager une inversion directe de la matrice d'autocorrélation. Le coût de calcul est alors proportionnel à  $N^3$ .
- Si la puissance de calcul est plus limitée ou/et si le débit des signaux s'accélère, il est nécessaire de s'orienter vers des algorithmes dits lents dont le coût de calcul est proportionnel à  $N^2$ .
- Enfin dans le cas où la puissance est encore réduite et/ou le délai des signaux à traiter s'accélère encore, il faut s'orienter vers des algorithmes rapides dont le coût de calcul est proportionnel à  $N$ .

Parmi les algorithmes précédents, il existe principalement deux familles:

- les algorithmes directs: RLS, Fast Kalman,...
- les algorithmes en racine carrée: Treillis spatial, QR, treillis, FLSQR,...

Il a été observé que les problèmes d'instabilité étaient très courants dans la famille des algorithmes directs alors qu'ils étaient quasi inexistant dans la famille des algorithmes en racine carrée.

Enfin, il faut rappeler l'importance de l'algorithme du gradient stochastique qui, n'optimisant pas tout à fait le même critère que les algorithmes précédents, ne peut être rattaché à une famille particulière mais dont la simplicité, le faible coût de calcul et la robustesse ont fait pendant longtemps l'algorithme de filtrage adaptatif de référence.

# Annexe : Dérivation Forme Bilineaire

*Dérivation de  $A^T R A$  en réel*

$$u = (a_0 \quad \dots \quad a_{N-1}) \begin{pmatrix} r_{i,j} \\ \vdots \\ a_{N-1} \end{pmatrix}$$

$$u = (a_0 \quad \dots \quad a_{N-1}) \begin{pmatrix} \sum_j r_{k,j} a_j \end{pmatrix}$$

$$u = \sum_k a_k \left( \sum_j r_{k,j} a_j \right)$$

$$u = \sum_k \sum_j a_k a_j r_{k,j}$$

$$u = \sum_{k \neq p} \sum_j a_k a_j r_{k,j} + \sum_j a_p a_j r_{p,j}$$

$$u = \sum_{k \neq p} a_k a_p r_{k,p} + \sum_{k \neq p} \sum_{j \neq p} a_k a_j r_{k,j} + \sum_{j \neq p} a_p a_j r_{p,j} + a_p^2 r_{p,p}$$

$$\frac{\partial u}{\partial p} = \sum_{k \neq p} a_k r_{k,p} + \sum_{j \neq p} a_j r_{p,j} + 2a_p r_{p,p}$$

$R = R^T$  donc  $r_{k,p} = r_{p,k}$  d'où :

$$\frac{\partial u}{\partial a_p} = 2a_p r_{p,p} + 2 \sum_{j \neq p} a_j r_{p,j}$$

$$\begin{pmatrix} \vdots \\ \frac{\partial u}{\partial p} \\ \vdots \end{pmatrix} = 2 \begin{pmatrix} \dots & r_{p,j} & \dots \end{pmatrix} \begin{pmatrix} \vdots \\ \mathbf{a}_j \\ \vdots \end{pmatrix}$$

Donc :

$$\frac{\partial}{\partial A} A^T R A = 2 R A$$