QUATORZIEME COLLOQUE GRETSI - JUAN-LES-PINS - DU 13 AU 16 SEPTEMBRE 1993

UN ALGORITHME DES MOINDRES CARRÉS RÉCURSIF RAPIDE: LE FSU RLS

Dirk T. M. Slock **Karim Maouche**

Institut EURECOM, 2229 route des Crêtes, B.P. 193, 06904, Sophia Antipolis Cedex, FRANCE

RÉSUMÉ

Nous présentons un nouvel algorithme des moindres carrés récursif rapide. Cet algorithme présente un intérêt certain pour l'adaptation de filtres très longs comme ceux utilisés dans les problèmes d'annulation d'écho acoustique. L'idée de départ est d'utiliser l'algorithme RLS avec une mise à jour "sous-échantillonnée" du filtre. Dans cet algorithme (le SU RLS) le gain de Kalman et la variable de vraisemblance sont des matrices qui ont des rangs de déplacement faibles. Ces quantités sont alors représentées et mises à jour par le biais de leurs générateurs, sous forme de sommes de produits de matrices de Toeplitz triangulaires et le produit de l'une de ces quantités avec un vecteur peut alors être calculé en utilisant la transformée de Fourier rapide.

ABSTRACT

We derive a new fast algorithm for Recursive Least-Squares (RLS) adaptive filtering. This algorithm is especially suited for adapting very long filters such as in the acoustic echo cancellation problem. The starting point is to introduce subsampled updating (SU) in the RLS algorithm. In the SU RLS algorithm, the Kalman gain and the likelihood variable are matrices. Due to the shift invariance of the adaptive FIR filtering problem, these matrices exhibit a low displacement rank. This leads to a representation of these quantities in terms of sums of products of triangular Toeplitz matrices. Finally, the product of these Toeplitz matrices with a vector can be computed efficiently by using the Fast Fourier Transform (FFT).

1 INTRODUCTION

Pour adapter un filtre RIF sous forme transverse (FTF) [1], [2] ou en treillis (FLA/FQR) [3], les algorithmes récursifs rapides des moindres carrés exploitent la propriété dite de "shift invariance" inhérente au problème du filtrage adaptatif. Avec ces algorithmes adaptatifs, la complexité passe de $\mathcal{O}(N^2)$ pour le RLS à $\mathcal{O}(N)$ par échantillon (N est la longueur du filtre RIF). Pour réduire davantage la complexité de ces algorithmes, il faut réduire la fréquence d'adaptation du filtre. La mise à jour du filtre est alors dite sous-échantillonnée avec un rapport de sous-échantillonnage donné L. Deux stratégies apparaissent: La première consiste à faire un traitement par blocs en résolvant les équations normales à chaque L échantillons, c'est l'algorithme BRLS (Block RLS) [4]. La deuxième alternative consiste à utiliser la même stratégie que le RLS en calculant le nouveau filtre et d'autres quantités auxiliaires à partir des données disponibles L échantillons auparavant. Cette démarche amène à l'algorithme SU RLS (Subsampled Updating RLS). Dans ce qui suit, nous nous intéresserons plus particulièrement à une version rapide du SU RLS (FSU RLS) en utilisant la notion de structure de déplacement ainsi que la technique de convolution rapide par utilisation de la transformée de Fourier rapide (FFT). En fait, la stratégie du BRLS s'avère plus intéressante pour les grandes valeurs de L (comparables à N) alors que le FSU RLS est plus intéressant pour des valeurs intermédiaires.

2 L'ALGORITHME SU RLS

Afin de fixer les notations, nous commençons par rappeler l'algorithme des moindres carrés récursif (RLS).

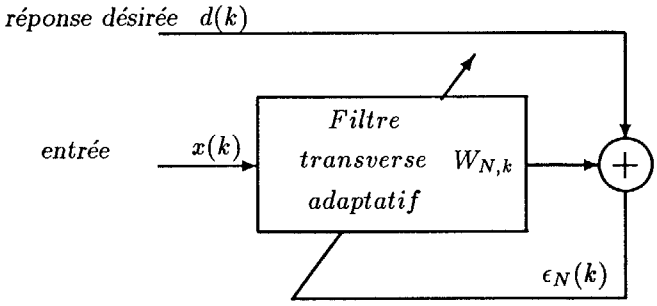

Figure 1: Système de filtrage RIF adaptatif.

2.1 L'algorithme RLS

Le filtre transverse adaptatif $W_{N,k}$ forme une combinaison linéaire des N échantillons consécutifs du signal d'entrée $\{x(i-n), n = 0, \ldots, N-1\}$ pour approximer l'opposé du signal désiré $d(i)$. Le signal d'erreur résultant est donné par (voir Fig. 1)

$$\epsilon_N(i|k) \;=\; d(i) + W_{N,k}\, X_N(i) \;=\; d(i) + \sum_{n=0}^{N-1} W_{N,k}^{n+1}\, x(i-n) \tag{1}$$

où $X_N(i) = \left[x^H(i)\, x^H(i-1) \cdots x^H(i-N+1)\right]^H$ est le vecteur de régression et ^{H} l'opérateur de transposition et conjugaison complexe. Dans l'algorithme RLS, les N coefficients du filtre $W_{N,k} = \left[W_{N,k}^1 \cdots W_{N,k}^N\right]$ sont adaptés de manière à minimiser récursivement le critère des moindres carrés suivant

$$\xi_N(k) = \min_{W_N} \left\{ \sum_{i=1}^{k} \lambda^{k-i} \left\| d(i) + W_N\, X_N(i) \right\|^2 \right\} \quad (2)$$

où $\lambda \in (0,1]$ est le facteur d'oubli exponentiel.
La minimisation de ce critère donne

$$W_{N,k} = -P_{N,k}^H R_{N,k}^{-1} \quad (3)$$

où

$$\begin{aligned} R_{N,k} &= \sum_{i=1}^{k} \lambda^{k-i} X_N(i) X_N^H(i) \\ &= \lambda R_{N,k-1} + X_N(k) X_N^H(k) \\ P_{N,k} &= \sum_{i=1}^{k} \lambda^{k-i} X_N(i) d^H(i) \\ &= \lambda P_{N,k-1} + X_N(k) d^H(k) \end{aligned} \quad (4)$$

sont respectivement la matrice d'autocorrélation et le vecteur d'intercorrélation estimés.
En remplacant les récurrences pour $R_{N,k}$ et $P_{N,k}$ (4) dans (3) et en utilisant le lemme d'inversion matriciel pour $R_{N,k}^{-1}$, on obtient l'algorithme des moindres carrés récursif:

$$\widetilde{C}_{N,k} = -X_N^H(k) \lambda^{-1} R_{N,k-1}^{-1} \quad (5)$$

$$\gamma_N^{-1}(k) = 1 - \widetilde{C}_{N,k} X_N(k) \quad (6)$$

$$R_{N,k}^{-1} = \lambda^{-1} R_{N,k-1}^{-1} - \widetilde{C}_{N,k}^H \gamma_N(k) \widetilde{C}_{N,k} \quad (7)$$

$$\epsilon_N^p(k) = \epsilon_N(k|k-1) = d(k) + W_{N,k-1} X_N(k) \quad (8)$$

$$\epsilon_N(k) = \epsilon_N(k|k) = \epsilon_N^p(k)\, \gamma_N(k) \quad (9)$$

$$W_{N,k} = W_{N,k-1} + \epsilon_N(k) \widetilde{C}_{N,k} \quad (10)$$

où $\epsilon_N^p(k)$ et $\epsilon_N(k)$ sont les signaux d'erreurs à priori et à postériori. Ces erreurs sont reliées par la variable de vraisemblance $\gamma_N(k)$ (9) et $\widetilde{C}_{N,k}$ est le gain de Kalman.

2.2 L'algorithme SU RLS

Il est possible d'obtenir un algorithme équivalent au RLS en mettant à jour le filtre, non pas à chaque échantillon du signal d'entrée mais à chaque L échantillons. Dans ce qui suit, nous supposerons pour des raisons de commodité que L est une puissance de 2 et que $M = \frac{N+1}{L}$ est un entier. Posons

$$d_{L,k} = \begin{bmatrix} d^H(k-L+1) \\ \vdots \\ d^H(k) \end{bmatrix}, \; x_{L,k} = \begin{bmatrix} x^H(k-L+1) \\ \vdots \\ x^H(k) \end{bmatrix}, \quad (11)$$

$$X_{N,L,k} = \begin{bmatrix} X_N^H(k-L+1) \\ \vdots \\ X_N^H(k) \end{bmatrix} = \left[x_{L,k} \cdots x_{L,k-N+1}\right]. \quad (12)$$

La matrice des données $X_{N,L,k}$ est construite de manière à ce qu'elle ait une structure de Toeplitz.
On peut alors réécrire (4) sous la forme

$$\begin{aligned} R_{N,k} &= \lambda^L R_{N,k-L} + X_{N,L,k}^H \Lambda_L X_{N,L,k} \\ P_{N,k} &= \lambda^L P_{N,k-L} + X_{N,L,k}^H \Lambda_L d_{L,k} \end{aligned} \quad (13)$$

avec $\Lambda_N = \operatorname{diag}\left\{\lambda^{N-1}, \ldots, \lambda, 1\right\}$.
En remplaçant dans (3) $P_{N,k}$ et $R_{N,k}$ par leurs expressions données en (13), on obtient l'algorithme SU RLS

$$\underline{\widetilde{C}}_{N,k} = -X_{N,L,k} \lambda^{-L} R_{N,k-L}^{-1} \quad (14)$$

$$\underline{\gamma}_N^{-1}(k) = \Lambda_L^{-1} - X_{N,L,k} \underline{\widetilde{C}}_{N,k}^H \quad (15)$$

$$R_{N,k}^{-1} = \lambda^{-L} R_{N,k-L}^{-1} - \underline{\widetilde{C}}_{N,k}^H \underline{\gamma}_N(k) \underline{\widetilde{C}}_{N,k} \quad (16)$$

$$\epsilon_{N,L,k}^p = d_{L,k} + X_{N,L,k} W_{N,k-L}^H \quad (17)$$

$$\underline{\gamma}_N^{-1}(k)\, \epsilon_{N,L,k} = \epsilon_{N,L,k}^p \quad (18)$$

$$W_{N,k} = W_{N,k-L} + \epsilon_{N,L,k}^H \underline{\widetilde{C}}_{N,k} \quad (19)$$

où $\epsilon_{N,L,k}^p$ est le vecteur des erreurs à priori:

$$\epsilon_{N,L,k}^p = \begin{bmatrix} \epsilon_N^H(k-L+1|k-L) \\ \vdots \\ \epsilon_N^H(k|k-L) \end{bmatrix} \quad (20)$$

et $\epsilon_{N,L,k}$ est le vecteur des erreurs à postériori:

$$\epsilon_{N,L,k} = \begin{bmatrix} \epsilon_N^H(k-L+1|k) \\ \vdots \\ \epsilon_N^H(k|k) \end{bmatrix}. \quad (21)$$

Cet algorithme est équivalent au RLS quand $L = 1$. Il présente la même complexité par échantillon. Le gain de Kalman $\underline{\widetilde{C}}_{N,k}$ ainsi que la variable de vraisemblance $\underline{\gamma}_N(k)$ sont des matrices respectivement $L \times N$ et $L \times L$.

2.3 Équivalence des algorithmes RLS et SU RLS

Les erreurs de filtrage du SU RLS ne sont pas des erreurs de prédiction à un pas mais à plusieurs pas. Ceci est dû au fait que le filtre $W_{N,k}$ est adapté à chaque L échantillons. Il est cependant possible de calculer les erreurs à priori du RLS à partir de ceux du SU RLS.
Soit $\underline{\epsilon}_{N,k}^p$ le vecteur des erreurs à priori du RLS entre les instants $k-L+1$ et k

$$\underline{\epsilon}_{N,k}^p = \begin{bmatrix} \epsilon_N^H(k-L+1|k-L) \\ \vdots \\ \epsilon_N^H(k|k-1) \end{bmatrix} \quad (22)$$

et soit $D_{N,L,k}$ la matrice diagonale suivante:

$$D_{N,L,k} = \operatorname{diag}\left\{\lambda^{L-1} \gamma_N(k-L+1), \ldots, \gamma_N(k)\right\} \quad (23)$$

alors, on peut montrer que

$$\epsilon_{N,L,k}^{p\,H} \underline{\gamma}_N(k) \epsilon_{N,L,k}^p = \underline{\epsilon}_{N,k}^{p\,H} D_{N,L,k} \underline{\epsilon}_{N,k}^p. \quad (24)$$

Considérons la décomposition UDL de $\underline{\gamma}_N(k)$

$$\underline{\gamma}_N(k) = U_{N,L,k}\, D_{N,L,k}\, U_{N,L,k}^H \quad (25)$$

$U_{N,L,k}$ est triangulaire supérieure avec une diagonale unité et la matrice diagonale $D_{N,L,k}$ est celle définie dans (23). En remplacant (25) dans (24), on obtient

$$U^H_{N,L,k}\epsilon^p_{N,L,k} = \underline{\epsilon}^p_{N,k} \cdot \quad (26)$$

Cette relation permet de calculer les erreurs à priori $\underline{\epsilon}^p_{N,k}$ de l'algorithme RLS à partir des erreurs à priori $\epsilon^p_{N,L,k}$ du SU RLS. De la même façon, on montre que le gain de Kalman du SU RLS est lié à celui du RLS par la relation

$$U^H_{N,L,k}\underline{\widetilde{C}}_{N,k} = \Lambda_L^{-1}\begin{bmatrix}\widetilde{C}_{N,k-L+1}\\ \vdots \\ \widetilde{C}_{N,k}\end{bmatrix} . \quad (27)$$

En particulier, soit $u_{N,L,k}$ la dernière colonne de $U_{N,L,k}$. Alors (27) donne

$$u^H_{N,L,k}\,\underline{\widetilde{C}}_{N,k} = \widetilde{C}_{N,k} . \quad (28)$$

3 STRUCTURES DE DÉPLACEMENT POUR LE SU RLS

La structure de déplacement d'une matrice R est:

$$\nabla_\lambda R = R - \lambda Z R Z^H = \sum_{i=1}^{\delta} u_i v_i^H \quad (29)$$

où δ, le rang de $R-\lambda Z R Z^H$ est appelé le rang de déplacement et Z est la matrice de déplacement inférieure (1 sur la première diagonale supérieure et 0 ailleurs). La résolution de cette équation de Lyapunov donne la représentation suivante de R:

$$R = \nabla_\lambda^{-1}\left(\sum_{i=1}^{\delta} u_i v_i^H\right) = \sum_{i=1}^{\delta}\mathcal{L}(u_i)\,\widetilde{\Lambda}\,\mathcal{L}^H(v_i) \quad (30)$$

où $\widetilde{\Lambda} = \text{diag}\{1, \lambda, \lambda^2, \ldots\}$ et $\mathcal{L}(u)$ est une matrice de Toeplitz triangulaire inférieure avec u comme première colonne. Cette représentation sera utilisée pour $\underline{\widetilde{C}}_{N,k}$ et $\underline{\gamma}_N^{-1}(k)$ afin de réduire la complexité du SU RLS.

3.1 Structure de déplacement du gain de Kalman

Soient $A_{N,k}$ et $B_{N,k}$ les filtres de prédiction respectivement direct et rétrograde et $\alpha_N(k)$ et $\beta_N(k)$ les variances des erreurs de prédiction correspondantes (voir [1]) et soient $e^p_{N,L,k}$ et $r^p_{N,L,k}$ les vecteurs des erreurs de prédiction respectivement avant et arrière

$$e^p_{N,L,k} = X_{N+1,L,k}\,A^H_{N,k-L} \quad (31)$$
$$r^p_{N,L,k} = X_{N+1,L,k}\,B^H_{N,k-L} . \quad (32)$$

La structure de déplacement du gain de Kalman est

$$\begin{aligned}\nabla_\lambda\left[\underline{\widetilde{C}}_{N,k}\ \ 0\right] &= -e^p_{N,L,k}\lambda^{-L}\alpha_N^{-1}(k-L)A_{N,k-L}\\ &\quad + r^p_{N,L,k}\lambda^{-L}\beta_N^{-1}(k-L)B_{N,k-L}\\ &\quad -(\eta_{N,L,k}-u_{L,1})\,\lambda^{-L+1}\gamma_N(k-L)\left[0\ \ \widetilde{C}_{N,k-L}\right]\end{aligned} \quad (33)$$

avec

$$\eta_{N,L,k} = X_{N+1,L,k}\left[0\ \ \widetilde{C}_{N,k-L}\right]^H \quad (34)$$
$$u_{L,1} = [\,1\,0\,0\cdots 0\,]^H \quad (L\times 1) . \quad (35)$$

3.2 Structure de déplacement de la variable de vraisemblance

La structure de déplacement de $\underline{\gamma}_N^{-1}(k)$ est donnée par

$$\nabla_\lambda\,\underline{\gamma}_N^{-1}(k) = \nabla_\lambda\,\Lambda_L^{-1} + \lambda^{-L}\,\nabla_\lambda\left(X_{N,L,k}R^{-1}_{N,k-L}X^H_{N,L,k}\right) . \quad (36)$$

Elle est égale à

$$\begin{aligned}\nabla_\lambda\,\underline{\gamma}_N^{-1}(k) &= e^p_{N,L,k}\lambda^{-L}\alpha_N^{-1}(k-L)e^{p\,H}_{N,L,k}\\ &\quad -r^p_{N,L,k}\lambda^{-L}\beta_N^{-1}(k-L)r^{p\,H}_{N,L,k}\\ &\quad +(\eta_{N,L,k}-u_{L,1})\,\lambda^{-L+1}\gamma_N(k-L)\,(\eta_{N,L,k}-u_{L,1})^H .\end{aligned} \quad (37)$$

Le gain de Kalman ainsi que la variable de vraisemblance ont des rangs de déplacement égaux à 3. Ces matrices ont une structure proche de celle de Toeplitz et peuvent être remplacées par leurs représentations comme dans (30). Pour la mise à jour de $\underline{\widetilde{C}}_{N,k}$ et $\underline{\gamma}_N^{-1}(k)$ (14), (15), il suffit maintenant de mettre à jour les filtres $A_{N,k}$, $B_{N,k}$ et $\left[0\ \ \widetilde{C}_{N,k}\right]$, et de calculer $e^p_{N,L,k}$, $r^p_{N,L,k}$ et $\eta_{N,L,k}$ à partir de leurs définitions (31), (32) et (34), en utilisant les données disponibles à l'instant k. L'apparition dans ces nouvelles représentations de matrices de Toeplitz va permettre en plus de réduire la complexité des calculs. En effet, le produit du gain de Kalman par le vecteur d'erreurs à postériori pourra être calculé à l'aide de la technique de convolution rapide Overlap-save (utilisation de la transformée de Fourier rapide [5]). De plus, l'inversion de la variable de vraisemblance ainsi que sa décomposition UDL pourra être faite avec l'algorithme de Schur généralisé.

4 L'ALGORITHME FSU RLS

L'équation (19) peut être mise sous la forme

$$[W_{N,k}\ \ 0] = [W_{N,k-L}\ \ 0] + \epsilon^H_{N,L,k}\left[\underline{\widetilde{C}}_{N,k}\ \ 0\right] . \quad (38)$$

Si on remplace $\left[\underline{\widetilde{C}}_{N,k}\ \ 0\right]$ par sa représentation en fonction de ses générateurs via (33) et (30), (38) prendra la forme suivante

$$[W_{N,k}\ \ 0] = [W_{N,k-L}\ \ 0] + \epsilon^H_{N,L,k}\sum_{i=1}^{3} d_i\mathcal{T}^i_{L,L}\,\widetilde{\Lambda}_L\,\mathcal{G}^i_{L,N+1} \quad (39)$$

où d_i, $\mathcal{T}^i_{L,L}$ et $\mathcal{G}^i_{L,N+1}$ sont respectivement des scalaires, des matrices de Toeplitz $L\times L$ triangulaires inférieures et des matrices de Toeplitz $L\times(N+1)$ triangulaires supérieures. Le dernier terme de (39) peut alors être calculé de la manière suivante:
Pour $i = 1,2,3$:

- multiplier d_i par $\epsilon_{N,L,k}$ pour obtenir $p^1_{L,i} = \epsilon^H_{N,L,k}d_i$.
- utiliser la méthode Overlap-save pour calculer le produit $p^2_{L,i} = p^1_{L,i}\mathcal{T}^i_{L,L}$.
- multiplier $p^2_{L,i}$ avec la matrice diagonale $\widetilde{\Lambda}_L$. Ce qui donne un vecteur $1\times L$: $p^3_{L,i} = p^2_{L,i}\widetilde{\Lambda}_L$.

- utiliser la méthode Overlap-save pour calculer le produit $p^4_{L,i} = p^3_{L,i} G^i_{L,N+1}$ en $\frac{N+1}{L}$ portions de taille L.

Finalement, sommer les trois vecteurs $p^4_{L,i}$ et appliquer la transformée de Fourier inverse à la somme $\sum_{i=1}^{3} p^4_{L,i}$.

Le gain de Kalman RLS sera mis à jour en utilisant (28) et la représentation de $\underline{\widetilde{C}}_{N,k}$ en fonction de ses générateurs. Les filtres de prédiction seront mis à jour de la même façon que le filtre $W_{N,k}$. Une légère complication se présente néanmoins pour la réactualisation du filtre de prédiction direct. Les détails peuvent être trouvés dans [6].
L'algorithme FSU RLS résultant est donné dans la table ci-dessous.

Algorithme FSU RLS

$$\epsilon^p_{N,L,k} = d_{L,k} + X_{N+1,L,k} \begin{bmatrix} W_{N,k-L} & 0 \end{bmatrix}^H$$
$$r^p_{N,L,k} = X_{N+1,L,k} \; B^H_{N,k-L}$$
$$\eta_{N,L,k} = X_{N+1,L,k} \begin{bmatrix} 0 & \widetilde{C}_{N,k-L} \end{bmatrix}^H$$
$$\begin{bmatrix} e^H_N(k-L+1) \\ e^p_{N,L,k+1} \end{bmatrix} = \begin{bmatrix} X_{N+1,L,k} \, A^H_{N,k-L+1} \\ X^H_{N+1}(k+1) \, A^H_{N,k-L+1} \end{bmatrix}$$
$$e^p_{N,L,k} = \begin{bmatrix} e^H_N(k-L+1) \\ \left(e^p_{N,L,k+1}\right)_{1:L-1} \end{bmatrix} - \eta_{N,L,k} \, e^H_N(k-L+1)$$
$$\begin{aligned} \underline{\gamma}^{-1}_N(k) = \nabla^{-1}_\lambda \Big\{ & e^p_{N,L,k} \lambda^{-L} \alpha^{-1}_N(k-L) e^{p\,H}_{N,L,k} \\ & - r^p_{N,L,k} \lambda^{-L} \beta^{-1}_N(k-L) r^{p\,H}_{N,L,k} \\ & + (\eta_{N,L,k} - u_{L,1}) \, \lambda^{-L+1} \gamma_N(k-L) \, (\eta_{N,L,k} - u_{L,1})^H \Big\} \end{aligned}$$
$$\underline{\gamma}^{-1}_N(k) = L_{N,L,k} \, G_{N,L,k} \, L^H_{N,L,k}$$
$$L^H u_{N,L,k} = u_{L,L}$$
$$\underline{\gamma}^{-1}_N(k) \; \epsilon_{N,L,k} = \epsilon^p_{N,L,k}$$
$$\underline{\gamma}^{-1}_N(k) \; r_{N,L,k} = r^p_{N,L,k}$$
$$\underline{\gamma}^{-1}_N(k) \; e_{N,L,k+1} = e^p_{N,L,k+1}$$
$$\begin{aligned} \begin{bmatrix} \underline{\widetilde{C}}_{N,k} & 0 \end{bmatrix} = \nabla^{-1}_\lambda \Big\{ & -e^p_{N,L,k} \lambda^{-L} \alpha^{-1}_N(k-L) A^H_{N,k-L} \\ & + r^p_{N,L,k} \lambda^{-L} \beta^{-1}_N(k-L) B^H_{N,k-L} \\ & - (\eta_{N,L,k} - u_{L,1}) \lambda^{-L+1} \gamma_N(k-L) \begin{bmatrix} 0 & \widetilde{C}_{N,k-L} \end{bmatrix} \Big\} \end{aligned}$$
$$B_{N,k} = B_{N,k-L} + r^H_{N,L,k} \begin{bmatrix} \underline{\widetilde{C}}_{N,k} & 0 \end{bmatrix}$$
$$\beta_N(k) = \lambda^L \beta_N(k-L) + r^H_{N,L,k} r^p_{N,L,k}$$
$$\begin{bmatrix} \widetilde{C}_{N,k} & 0 \end{bmatrix} = u^H_{N,L,k} \begin{bmatrix} \underline{\widetilde{C}}_{N,k} & 0 \end{bmatrix}$$
$$\gamma^{-1}_N(k) = (G_{N,L,k})_{L,L}$$
$$A_{N,k+1} = A_{N,k-L+1} + e^H_{N,L,k+1} \begin{bmatrix} \underline{\widetilde{C}}_{N,k} & 0 \end{bmatrix} Z^H_{N+1}$$
$$e^p_N(k+1) = e^{p\,H}_{N,L,k+1} u_{N,L,k}$$
$$e_N(k+1) = e^p_N(k+1) \gamma_N(k)$$
$$A_{N,k} = A_{N,k+1} - e_N(k+1) \begin{bmatrix} 0 & \widetilde{C}_{N,k} \end{bmatrix}$$
$$\alpha_N(k+1) = \lambda^L \alpha_N(k-L+1) + e^H_{N,L,k+1} e^p_{N,L,k+1}$$
$$\alpha_N(k) = \lambda^{-1} \left(\alpha_N(k+1) - e_N(k+1) e^{p\,H}_N(k+1) \right)$$
$$\begin{bmatrix} W_{N,k} & 0 \end{bmatrix} = \begin{bmatrix} W_{N,k-L} & 0 \end{bmatrix} + \epsilon^H_{N,L,k} \begin{bmatrix} \underline{\widetilde{C}}_{N,k} & 0 \end{bmatrix}$$

5 Conclusions

La complexité du FSU RLS est $\mathcal{O}(8\frac{N+1}{L}\frac{FFT(2L)}{L} + 34\frac{N}{L} + 5.5L)$ opérations par échantillon où $FFT(m)$ représente le nombre d'opérations nécessaire pour réaliser une FFT d'une séquence de longueur m. Ceci est particulièrement intéressant pour les filtres de grande taille. Par exemple lorsque $(N, L) = (4095, 256), (8191, 256)$ et avec une FFT split radix ($FFT(2m) = mlog_2(2m)$ multiplications réelles pour des signaux réels) la complexité est respectivement $0.8N$ et $0.6N$ par échantillon, complexité à comparer avec celle du LMS ($2N$) et celle du FTF qui est $7N$ et qui est actuellement l'algorithme des moindres carrés le plus rapide. Le prix à payer est cependant, l'introduction d'un retard de $\mathcal{O}(L)$ échantillons dans le traitement.
Les simulations ont montré que l'algorithme est conforme à la théorie. Cependant, celui-ci présente des problèmes d'instabilité numérique dûs à la propagation des erreurs d'arrondi. Un moyen simple de le stabiliser consisterait à rafraîchir les générateurs à intervalles réguliers en utilisant par exemple le BRLS [4]. La stabilisation du FSU RLS est néanmoins envisageable et constitue l'objet des recherches à venir.

Références

[1] J.M. Cioffi and T. Kailath. "Fast, recursive least squares transversal filters for adaptive filtering". *IEEE Trans. on ASSP*, ASSP-32(2):304–337, April 1984.

[2] D.T.M. Slock and T. Kailath. "Numerically Stable Fast Transversal Filters for Recursive Least-Squares Adaptive Filtering". *IEEE Trans. Signal Proc.*, ASSP-39(1):92–114, Jan. 1991.

[3] D.T.M. Slock. "Reconciling Fast RLS Lattice and QR Algorithms". In *Proc. ICASSP 90 Conf.*, pages 1591–1594, Albuquerque, NM, April 3–6 1990.

[4] X.-H. Yu and Z.-Y. He. "Efficient Block Implementation of Exact Sequential Least-Squares Problems". *IEEE Trans. Acoust., Speech and Signal Proc.*, ASSP-36:392–399, March 1988.

[5] M. Vetterli. "Fast Algorithms for Signal Processing". In M. Kunt, editor, *Techniques modernes de traitement numérique des signaux*. Presses Polytechniques et Universitaires Romandes, Lausanne, Switzerland, 1991. ISBN 2-88074-207-2.

[6] D.T.M Slock and Karim Maouche. "The Fast Subsampled-Updating Recursive Least-Squares (FSU RLS) Algorithm for Adaptive Filtering Based on Displacement Structure and the FFT". Research Report N° 92-001, Institut Eurécom.