The Generalized Sliding Window Recursive Least-Squares (GSW RLS) Algorithm

Karim MAOUCHE† and Dirk T. M. SLOCK††

†, †† Institut EURECOM, 2229 route des Crêtes
B.P. 193, 06904, Sophia Antipolis Cedex, FRANCE
† *Tel/Fax: +33 93002632/ 93002627, E-Mail: maouche@eurecom.fr*
†† *Tel/Fax: +33 93002606/ 93002627, E-Mail: slock@eurecom.fr*

Abstract. In this paper, we derive a new RLS algorithm: the Generalized Sliding Window RLS (GSW RLS) algorithm that has a better tracking ability than the SWC RLS algorithm. This algorithm uses a generalized window which consists of an exponential window for the L' most recent data and the same but attenuated exponential window for the rest of the data. We give a theoritical proof that the use of this window leads to a better compromise between the Excess Mean Squared Errors due to estimation noise and lag noise. Furthermore, after providing a fast version of the GSW RLS algorithm, namely the GSW FTF algorithm, we apply the Subsampled-Upadating technique to derive the FSU GSW FTF algorithm, a doubly-fast version of the GSW RLS algorithm.

1 Introduction

Tracking ability is a desired feature in adaptive filtering, especially when the system to identify can vary quickly as in the case of acoustic echo channels. In the actual state of the art, two Finite Impulse Response (FIR) adaptive filtering algorithms are known for their good tracking ability among the large set of FIR adaptive filtering algorithms: the Affine Projection (AP) algorithm and the Recursive Least-Squares algorithm with a Sliding Rectangular Window (SWC RLS). The AP algorithm has a better tracking ability than the SWC RLS algorithm. It solves recursively an underdetermined system of linear equations while the SWC RLS algorithm solves recursively an overdetermined system of linear equations. The SWC RLS algorithm exhibits a better tracking ability than the classical RLS algorithm with an exponential window (WRLS). This is explained by the fact that the rectangular window allows to forget the past more abruptly than the exponential window does.

In this paper, we propose the GSW RLS algorithm, a new RLS algorithm that generalizes the WRLS and SWC RLS algorithms and shows a better tracking ability than the SWC RLS algorithm. This algorithm uses a generalized window which consists of the superposition of an exponential window for the L' most recent data and the same but attenuated exponential window for the rest of the data. The improvement of tracking comes from the fact that the length L' of the first window can be much smaller than the length of the rectangular window of the SWC RLS algorithm. Moreover, we prove theoretically that the use of this window leads to a better compromise between the Excess Mean Squared Errors (EMSE) due to estimation noise and lag noise. The GSW RLS algorithm turns out to have the same structure as the SWC RLS algorithm. Its computational complexity is $\mathcal{O}(N^2)$, the same as the SWC RLS algorithm. Nevertheless, the exploitation of a certain shift-invariance property that is inherent to the adaptive filtering problem allows the derivation of fast version and leads to the GSW Fast Transversal Filter (GSW FTF) algorithm whose computational complexity is $14N$ ($16N$ for the stabilized version), i.e., the same complexity as the SWC FTF algorithm. Furthermore, by applying the Subsampled-Updating technique to the GSW FTF algorithm, we derive a very fast version of the GSW RLS algorithm called the FSU GSW RLS algorithm, a mathematically equivalent algorithm to the GSW RLS algorithm with a low computational complexity when dealing with long FIR filters.

2 The GSW RLS Algorithm

An adaptive transversal filter $W_{N,k}$ combines linearly N consecutive input samples $\{x(i-n), n = 0, \ldots, N-1\}$ to approximate (the negative of) the desired-response signal $d(i)$. The resulting error signal is given by

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

where $X_N(i) = \left[x^H(i)\ x^H(i-1) \cdots x^H(i-N+1)\right]^H$ is the input data vector and superscript ^{H} denotes Hermitian (complex conjugate) transpose. In the WRLS algorithm, the set of N transversal filter coefficients $W_{N,k} = \left[W_{N,k}^1 \cdots W_{N,k}^N\right]$ are adapted so as to minimize recursively the following LS criterion

$$\xi_N(k) \quad = \quad \sum_{i=1}^{k} \lambda^{k-i}\, \|\epsilon_N(i|k)\|^2 \quad , \quad (2)$$

where $\lambda \in (0, 1]$ is the exponential weighting factor, $\|v\|_\Lambda^2 = v\Lambda v^H$, $\|.\| = \|.\|_I$. On the other hand, the SWC RLS algorithm minimizes recursively the following criterion:

$$\xi_{N,L'}(k) = \sum_{i=k-L'+1}^{k} \|\epsilon_N(i|k)\|^2 \quad , \quad (3)$$

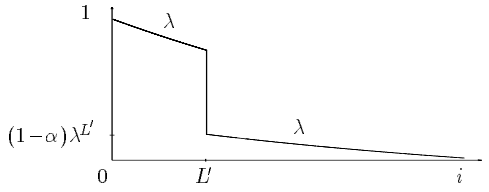

Figure 1: The generalized window.

where L', the length of the sliding window, must be greater than the filter length: $L' > N$, in which case, the associated covariance matrix is invertible. Compared to the WRLS algorithm, the SWC RLS algorithm exhibits better tracking. In order to improve the tracking of the SWC RLS algorithm, one should use a rectangular window whose length is smaller than the filter length. Doing this renders the problem singular. Hence, in order to regularize the problem, we add to the rectangular window of length L', an attenuated exponential window. This leads to minimizing the criterion over more data and weighting the past data less heavily than the most L' recent samples (see Fig. 1). Compared to the rectangular window of the SWC RLS algorithm, the new window introduces two degrees of freedom, the attenuation and the forgetting factor of the exponential tail. Furthermore, it has appeared that the derivation of a recursive algorithm for the new window requires the use of an exponential window for the L' most recent samples instead of the rectangular window and with the same forgetting factor as the attenuated exponential window. Hence, consider the following criterion:

$$\xi_{N,L'}(k)=\sum_{i=1}^{k-L'} w_{k-i}\,\|\epsilon_N(i|k)\|^2\,,\; w_i=\begin{cases}\lambda^i & 0\le i\le L'\\ (1-\alpha)\lambda^i & i>L'\end{cases}\,. \tag{4}$$

The new criterion generalizes the WRLS and SWC RLS criteria since the WRLS criterion (2) is obtained from (4) by setting $\alpha = 0$ and the SWC RLS criterion (4) is the one given by the generalized criterion when $\alpha = 1$ and $\lambda = 1$. Let $W_{N,L',k}$ be the adaptive RLS filter provided by such a window, the minimization of (4) leads to the following

$$W_{N,L',k} = -P^H_{N,L',k}\; R^{-1}_{N,L',k}\quad, \tag{5}$$

where

$$\begin{aligned} R_{N,L',k} &= (1-\alpha)\sum_{i=1}^{k-L'}\lambda^{k-i}X_N(i)X_N^H(i)+\sum_{i=k-L'+1}^{k}\lambda^{k-i}X_N(i)X_N^H(i)\\ P_{N,L',k} &= (1-\alpha)\sum_{i=1}^{k-L'}\lambda^{k-i}X_N(i)d^H(i)+\sum_{i=k-L'}^{k}\lambda^{k-i}X_N(i)d^H(i)\,. \end{aligned} \tag{6}$$

The use of the same forgetting factor for the two windows allows the following recursions for the sample second order moments

$$\begin{aligned} R_{N,L'+1,k} &= \lambda R_{N,L',k-1}+X_N(k)X_N^H(k)\\ &= R_{N,L',k}-\alpha\lambda^{L'}X_N(k-L')X_N^H(k-L') \end{aligned} \tag{7}$$

$$\begin{aligned} P_{N,L'+1,k} &= \lambda P_{N,L',k-1}+X_N(k)d^H(k)\\ &= P_{N,L',k}-\alpha\lambda^{L'}X_N(k-L')d^H(k-L')\quad. \end{aligned} \tag{8}$$

Hence, we can derive the new algorithm by applying the strategy for the usual WRLS algorithm twice. The first step will be devoted to the time and order update $(k-1,L')\rightarrow(k,L'+1)$, which is analogous to the update of the usual WRLS algorithm while the second step will be the order downdate $(k,L'+1)\rightarrow(k,L')$. The downdate scheme is obtained as follows: By using (8), one has

$$W_{N,L',k}R_{N,L',k}=W_{N,L'+1,k}R_{N,L'+1,k}-\alpha\lambda^{L'}d(k-L')X_N^H(k-L'). \tag{9}$$

Using (7) for $R_{N,L'+1,k}$ in term of $R_{N,L',k}$, we get

$$W_{N,L',k} = W_{N,L'+1,k}+\alpha\lambda^{L'}\nu_{N,L'+1}(k)\widetilde{D}_{N,L'+1,k}\quad, \tag{10}$$

where $\nu_{N,L'+1}(k) = d(k-L')+W_{N,L'+1,k}X_N(k-L')$ and $\widetilde{D}_{N,L'+1,k} = -X_N^H(k-L')R^{-1}_{N,L',k}$ are the a posteriori error signal and the a priori Kalman gain of the downdate part. Applying the MIL to (7) gives

$$R^{-1}_{N,L',k}=R^{-1}_{N,L'+1,k}-D^H_{N,L'+1,k}\delta^{-1}_{N,L'+1}(k)D_{N,L'+1,k}\,, \tag{11}$$

with $D_{N,L'+1,k} = -X_N^H(k-L')R^{-1}_{N,L'+1,k}$ and $\delta_{N,L'+1}(k) = \alpha^{-1}\lambda^{-L'}-D_{N,L'+1,k}X_N(k-L')$ are respectively the a posteriori Kalman gain and the likelihood variable associated with the downdate part. Now, it is straigthtforward to find that $\widetilde{D}_{N,L'+1,k}=\alpha^{-1}\lambda^{-L'}\delta^{-1}_{N,L'+1}(k)D_{N,L'+1,k}$ and that the a priori error is $\nu^p_{N,L'+1}(k)=d(k-L')+W_{N,L',k}X_N(k-L')=\alpha^{-1}\lambda^{-L'}\delta^{-1}_{N,L'+1}(k)\nu_{N,L'+1}(k)$. By associating the update part to the downdate part, we find the GSW RLS algorithm

$$\begin{aligned} \widetilde{C}_{N,L',k} &= -\lambda^{-1}X_N^H(k)R^{-1}_{N,L',k-1}\\ \gamma^{-1}_{N,L'}(k) &= 1-\widetilde{C}_{N,L',k}X_N(k)\\ \epsilon^p_{N,L'}(k) &= d(k)+W_{N,L',k-1}X_N(k)\\ \epsilon_{N,L'}(k) &= \epsilon^p_{N,L'}(k)\gamma_{N,L'}(k)\\ W_{N,L'+1,k} &= W_{N,L',k-1}+\widetilde{C}_{N,L',k}\epsilon_{N,L'}(k)\\ R^{-1}_{N,L'+1,k} &= \lambda^{-1}R^{-1}_{N,L',k-1}-\widetilde{C}^H_{N,L',k}\gamma_{N,L'}(k)\widetilde{C}_{N,L',k} \end{aligned} \tag{12}$$

$$\begin{aligned} D_{N,L'+1,k} &= -X_N^H(k-L')R^{-1}_{N,L'+1,k}\\ \delta_{N,L'+1}(k) &= \alpha^{-1}\lambda^{-L'}-D_{N,L'+1,k}X_N(k-L')\\ \nu_{N,L'+1}(k) &= d(k-L')+W_{N,L'+1,k}X_N(k-L')\\ \nu^p_{N,L'+1}(k) &= \alpha^{-1}\lambda^{-L'}\delta^{-1}_{N,L'+1}(k)\nu_{N,L'+1}(k)\\ W_{N,L',k} &= W_{N,L'+1,k}+\alpha\lambda^{L'}D_{N,L'+1,k}\nu^p_{N,L'+1}(k)\\ R^{-1}_{N,L'k} &= R^{-1}_{N,L'+1,k}-D^H_{N,L'+1,k}\delta^{-1}_{N,L'+1}(k)D_{N,L'+1,k}\,. \end{aligned}$$

The set of equations (12) constitute the complete time update of the algorithm. The algorithm is initialized with $R_{N,L',0}=\mu I$ where μ is a small scalar quantity. The GSW RLS algorithm shows a computational complexity of $O(N^2)$. One must notice that the GSW RLS algorithm has the same structure and complexity as the SWC RLS algorithm. Moreover, we prove in the next section that the use of this window leads to a better compromise between the EMSE due to estimation noise and lag noise.

3 Performance analysis

For the purpose of analysis, we consider the following classical identification model for the desired signal

$$d(k) = W^o_{N,k-1}X_N(k)+n(k)\quad, \tag{13}$$

where $n(k)$ is a centered Gaussian i.i.d. sequence with variance σ_n^2 $\left(n(k) \sim \mathcal{N}(0,\sigma_n^2)\right)$ and $W_{N,k}^o$ is the unknown filter that is time-varying according to a random walk

$$W_{N,k}^o = W_{N,k-1}^o + Z(k) \quad , \quad Z(k) \text{ i.i.d.} \sim \mathcal{N}(0,Q) \quad . \tag{14}$$

Considering a general window whose coefficients are denoted by w_i, one can show that the deviation filter $\widetilde{W}_{N,L',k} = W_{N,L',k} + W_{N,k}^o$ is given by

$$\widetilde{W}_{N,L',k} = \left(\sum_{i=0}^{\infty} w_i \left(\sum_{j=k-i}^{\infty} Z(j) \right) X_N(k-i) X_N(k-i)^H - \sum_{i=0}^{\infty} w_i\, n(k-i) X_N(k-i) \right) R_{N,L',k}^{-1} \quad .$$

Let COV_k be the covariance matrix of the deviation filter: $COV_k = \mathrm{E}\widetilde{W}_{N,L',k}\widetilde{W}_{N,L',k}^H$. Using the fact that $\mathrm{E}(.) = \mathrm{E}_X \mathrm{E}_{n,Z|X}(.)$ and assuming that k is big enough so that: $R_{N,L',k} \approx \mathrm{E}R_{N,L',k} = \left(\sum_{i=0}^{\infty} w_i \right) R$, we get after some manipulations

$$COV_k = \sigma_n^2 \left(\sum_{i=0}^{\infty} \widetilde{w}_i^2 \right) R^{-1} + \left(\sum_{i=0}^{\infty} \overline{w}_i^2 \right) Q \quad , \tag{15}$$

where

$$\widetilde{w}_i = w_i \left(\sum_{j=0}^{\infty} w_j \right)^{-1} \quad , \quad \overline{w}_i = 1 - \sum_{j=0}^{i-1} \widetilde{w}_j \tag{16}$$

We see from (15) that there are two contributions in COV_k: the first one is the contribution related to estimation noise while the second one is for lag noise and originates from the variation of the echo path response. Now, assuming statistical independence between $\widetilde{W}_{N,L',k}$ and $X_N(k-1)$, we can express the variance of the a priori error signal as

$$\begin{aligned} \mathrm{var}(\epsilon_N^p(k)) &= \sigma_n^2 + \mathrm{tr}\left(R\, COV_{k-1}\right) \qquad (17) \\ &= \sigma_n^2 + N \left(\sum_{j=0}^{\infty} \widetilde{w}_j^2 \right) \sigma_\eta^2 + \left(\sum_{j=0}^{\infty} \overline{w}_j^2 \right) \mathrm{tr}\left(RQ\right) \quad . \end{aligned}$$

Hence, we can compute the EMSE due to estimation noise for the WRLS, SWC RLS and GSW RLS algorithms:

$$\begin{array}{lcl} GSWRLS & : & \sigma_n^2 N \frac{1-\lambda}{1+\lambda} \frac{1+\alpha(\alpha-2)\lambda^{2L'}}{(1-\alpha\lambda^{L'})^2} \\ WRLS & : & \sigma_n^2 N \frac{1-\lambda}{1+\lambda} \\ SWCRLS & : & \sigma_n^2 \frac{N}{L'} \quad . \end{array} \tag{18}$$

The EMSE of WRLS can be recovered from the EMSE of the GSW RLS by letting $\alpha = 0$. The same thing holds for the SWC RLS algorithm for $\alpha = 1$ and $\lambda \to 1$. Taking $R = \sigma_x^2 I$ and $Q = \sigma_z^2 I$, leads to the following results for the EMSE due to lag noise

$$\begin{array}{lcl} GSWRLS & : & N\sigma_x^2\sigma_z^2 \frac{1-2\alpha\lambda^{L'}(1+\lambda-\lambda^{L'+1})+\alpha^2\lambda^{2L'}(1+L'-L'\lambda^2)}{(1-\lambda^2)(1-\alpha\lambda^{L'})^2} \\ WRLS & : & N\sigma_x^2\sigma_z^2 \frac{1}{1-\lambda^2} \\ SWCRLS & : & N\sigma_x^2\sigma_z^2 \frac{(L'+1)(2L'+1)}{6L'} \quad . \end{array} \tag{19}$$

Figure (2), shows curves that give lag noise misadjustment vs. estimation noise misadjustment for $N = 50, \sigma_x^2 = \sigma_z^2 = .01$ and $\sigma_n^2 = .1$. These curves show that for the same value of the estimation noise misadjustment, the GSW RLS algorithm has a lower lag noise misadjustment than the SWC RLS algorithm, so we can conclude that the GSW RLS algorithm is truly better than the SWC RLS algorithm. Exploitation of the shift-invariance property leads to the derivation of a fast version of the GSW RLS algorithm, namely the GSW FTF algorithm.

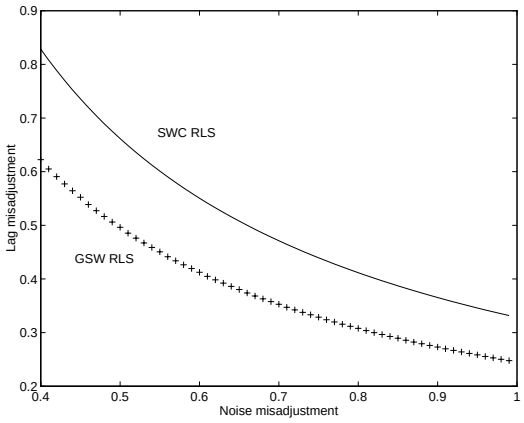

Figure 2: Comparison of SWC RLS and GSW RLS algorithms ($N = 50, \sigma_x^2 = \sigma_z^2 = .01, \sigma_n^2 = .1$).

4 The GSW FTF algorithm

The GSW FTF algorithm can be described in the following way, which emphasizes its rotational structure (see [1]) :

$$\begin{bmatrix} \begin{bmatrix} \widetilde{C}_{N,L',k} & 0 \end{bmatrix} \\ A_{N,L',k} \\ B_{N,L',k} \\ \begin{bmatrix} D_{N,L_p',k} & 0 \end{bmatrix} \\ \begin{bmatrix} W_{N,L',k} & 0 \end{bmatrix} \end{bmatrix} = \Theta_k \begin{bmatrix} \begin{bmatrix} 0 & \widetilde{C}_{N,L',k-1} \end{bmatrix} \\ A_{N,L',k-1} \\ B_{N,L',k-1} \\ \begin{bmatrix} 0 & D_{N,L_p',k-1} \end{bmatrix} \\ \begin{bmatrix} W_{N,L',k-1} & 0 \end{bmatrix} \end{bmatrix}$$

$$\begin{aligned} e_{N,L'}^p(k) &= A_{N,L',k-1} X_{N_p}(k) \\ e_{N,L_p'}(k) &= e_{N,L'}^p(k) \gamma_{N,L'}(k-1) \\ \gamma_{N_p,L'}^{-1}(k) &= \gamma_{N,L'}^{-1}(k-1) - \widetilde{C}_{N_p,L',k}^0 e_{N,L'}^p(k) \\ \gamma_{N,L'}^{-1}(k) &= \gamma_{N_p,L'}^{-1}(k) + \widetilde{C}_{N_p,L',k}^N r_{N,L'}^p(k) \\ r_N^p(k) &= -\lambda\beta_N(k-1)\widetilde{C}_{N_p,L',k}^{N\,H} \\ \alpha_{N,L_p'}^{-1}(k) &= \lambda^{-1}\, \alpha_{N,L'}^{-1}(k-1) - \widetilde{C}_{N_p,L',k}^{0\,H} \gamma_{N_p,L'}(k) \widetilde{C}_{N_p,L',k}^0 \\ r_{N,L_p'}(k) &= r_{N,L'}^p(k)\, \gamma_{N,L'}(k) \\ \beta_{N,L_p'}(k) &= \lambda\beta_{N,L'}(k-1) + r_{N,L'}^p(k)\, r_{N,L_p'}^H(k) \\ a_{N,L_p'}(k) &= A_{N,L_p',k} X_{N_p}(k-L'+1) \qquad (20) \\ a_{N,L'}^s(k) &= a_{N,L_p'}(k) \delta_{N,L_p'}^{-1}(k-1) \\ \alpha_{N,L'}(k) &= \alpha_{N,L_p'}(k) - \alpha\lambda^{L'} a_{N,L'}^s(k) a_{N,L'}^{s\,H}(k) \\ \delta_{N_p,L'}(k) &= \delta_{N,L'}(k-1) - D_{N_p,L_p',k}^0 a_{N,L_p'}(k) \end{aligned}$$

$$
\begin{aligned}
b_{N,L_p'}(k) &= -\beta_{N,L_p'}(k) D^{NH}_{N_p,L_p',k} \\
\delta_{N,L'}(k) &= \delta_{N_p,L'}(k) + D^{N}_{N_p,L_p',k} b_{N,L_p'}(k) \\
b^s_{N,L'}(k) &= b_{N,L_p'}(k) \delta^{-1}_{N,L_p'}(k) \\
\beta_{N,L'}(k) &= \beta_{N,L_p'}(k) - \alpha \lambda^{L'} b^s_{N,L'}(k) b^{sH}_{N,L'}(k) \quad ,
\end{aligned}
$$

where $L_p' = L'+1$ and $N_p = N+1$. $A_{N,L',k}$ and $B_{N,L',k}$ are the forward and backward prediction filters that are used in the update part, $e^p_{N,L'}(k)$ and $e_{N,L'}(k)$ are the a priori and a posteriori forward prediction errors, $r^p_{N,L'}(k)$ and $r_{N,L'}(k)$ are the a priori and a posteriori backward predition errors, $\widetilde{C}_{N_p,k} = \left[\widetilde{C}^0_{N_p,k} \cdots \widetilde{C}^N_{N_p,k} \right]$ and $\alpha_{N,L'}(k)$ and $\beta_{N,L'}(k)$ are the forward and backward prediction error variances. Θ_k is a 5×5 rotation matrix given by

$$
\Theta_k = \Theta^6_k \, \Theta^5_k \, \Theta^4_k \, \Theta^3_k \, \Theta^2_k \, \Theta^1_k \quad , \tag{21}
$$

where the 5×5 matrices Θ^i_k , $i = 1,2,3,4,5,6$ are

$$
\Theta^1_k = \begin{bmatrix} 1 & a & 0 & 0 & 0 \\ b & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad \Theta^2_k = \begin{bmatrix} 1 & 0 & c & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
\Theta^3_k = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ d & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ e & 0 & 0 & 0 & 1 \end{bmatrix} \quad \Theta^4_k = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & f & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & g & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{22}
$$

$$
\Theta^5_k = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & h & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad \Theta^6_k = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & i & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & j & 1 \end{bmatrix} \quad ,
$$

with $a = -e^{pH}_{N,L'}(k) \alpha^{-1}_{N,L'}(k)$, $b = e_{N,L_p'}(k)$, $c = -\widetilde{C}^N_{N,L',k}$, $d = r_{N,L_p'}(k)$, $e = \epsilon_{N,L'}(k)$, $f = -a^s_{N,L_p'}(k)$, $g = -a^H_{N,L_p'}(k) \alpha^{-1}_{N,L_p'}(k)$, $h = -D^N_{N,L_p',k}$, $i = -b^s_{N,L'}(k)$ and $j = \alpha \lambda^{L'} \nu_{N,L_p'}(k)$. The computational complexity of the GSW FTF algorithm is $14N$ operations, which is also the complexity of the SWC FTF algorithm. A numerically stabilized version of the GSW FTF algorithm is given in [2] and has a complexity of $16N$ operations.

5 The FSU GSW FTF Algorithm

In [3], [4] and [5], we have pursued an alternative way to reduce the complexity of adaptive filtering algorithms. The approach consists of subsampling the filter adaptation, i.e. the LS filter estimate is no longer provided every sample but every $L \geq 1$ samples (subsampling factor L). This strategy has led us to derive new algorithms that are the FSU FTF, FSU FNTF and FSU FAP algorithms which present a reduced complexity when dealing with long filters. Here, we apply this technique to the GSW FTF algorithm. Using the filter estimates at a certain time instant, we compute the filter outputs over the next L time instants. Using what we have called a GSW FTF-Schur algorithm, it is possible to compute the successive rotation matrices of the GSW FTF algorithm for the next L time instants. Applying the L rotation matrices to the filters vectors becomes an issue of multiplying polynomials, which can be efficiently carried out using the FFT. The complexity of the FSU GSW FTF is $\mathcal{O}((10\frac{N+1}{L}+26)\frac{FFT(2L)}{L}+50\frac{N}{L}+25L)$ operations per sample. This can be very interesting for long filters. For example, when $(N,L) = (4095,256)$, $(8191,256)$ and the FFT is done via the split radix ($FFT(2m) = mlog_2(2m)$ real multiplications for real signals) the multiplicative complexity is respectively $2.2N$ and $1.4N$ per sample. This should be compared to $14N$ for the GSW FTF algorithm and $2N$ for the LMS algorithm. The number of additions is somewhat higher. The cost we pay is a processing delay which is of the order of L samples. The subsampled updating technique turns out to be especially applicable in the case of very long filters such as occur in the acoustic echo cancellation problem. The computational gain it offers is obtained in exchange for some processing delay, as is typical of block processing.

6 Concluding Remarks

One major drawback of the SWC RLS algorithm is noise amplification that is due to the fact that the estimation of the covariance matrix of order N is done using a rectangular window of a relatively short length L' ($L' > N$). With such a window, the covariance matrix can be ill- conditionned if the input signal is not active in a significant portion of the window. The WRLS algorithm is less sensitive to noise amplification because of the larger memory of the exponential window. Due to its exponential tail, the generalized window reduces noise amplification. The AP algorithm also suffers from noise amplification and the use of the generalized window with the AP algorithm is the object of ongoing research.

REFERENCES

[1] D.T.M. Slock and T. Kailath "A modular prewindowing framework for covariance FTF RLS algorithms". *Signal Processing*, 28(1):47–61, 1992.

[2] K. Maouche and Dirk T.M. Slock. "The Generalized Sliding Window Recursive Least-Squares (GSW RLS) Algorithm". Technical Report RR N^{O} 95-021, Institut Eurécom, Sophia Antipolis, France, February 1995.

[3] D.T.M. Slock and K. Maouche. "The Fast Subsampled-Updating Fast Transversal Filter (FSU FTF) RLS Algorithm". *Annals of Telecommunications*, 49(7-8):407–413, 1994.

[4] K. Maouche and D.T.M. Slock. "The Fast Subsampled-Updating Fast Newton Transversal Filter (FSU FNTF) for Adapting Long FIR Filters". In *Proc. 28th Asilomar Conf. on Signals, Systems and Computers*, pages 1488–1492, Pacific Grove, CA, Oct. 31 - Nov. 2 1994.

[5] K. Maouche and Dirk T.M. Slock. "The Fast Subsampled-Updating Fast Affine Projection (FSU FAP) Algorithm". Technical Report RR N^{O} 95-018, Institut Eurécom, Sophia Antipolis, France, January 1995.