A Fast Multichannel Adaptive Filtering Algorithm

Mounir Bhouri *Mamadou Mboup* *Madeleine Bonnet*
UFR Mathématiques et Informatique - Université René Descartes-Paris 5
45, rue des Saints-Pères, 75270 Paris cedex 06, France
email: bhom@math-info.univ-paris5.fr

ABSTRACT

In this paper we present a new QR decomposition (QRD) based fast multichannel adaptive filtering algorithm. This algorithm is based on a block reduction technique which leads to a substantial reduction of complexity compared to other QRD-based algorithms. It has a complexity of $O(N)$ where N is the sum of the channels orders which may be different. It, also, has good numerical properties and is amenable to systolic architecture. Simulation shows a better robustness than the QRD-RLS in the context of highly intercorrelated channels inputs.

1 Introduction

Multichannel adaptive algorithms find application in many area such as multipath equalization, adaptive antennas and stereophonic echo cancellation. Fast multichannel algorithms, which extend monochannel fast QRD-based adaptive algorithms, have been developed for both equal and unequal channel orders (see [7][8][1][6]). Here, we introduce a new QRD-based approach to multichannel adaptive algorithm, that permits a substantial reduction of complexity. This approach is based on an iterative block transform that separates the channels treatment. Thus, the processing in each channel taken separately (by annulling the other channels inputs) corresponds exactly to the processing in a Fast QRD-RLS single channel algorithm [4]. This paper begins with the presentation of the QRD-RLS multichannel algorithm, then we derive our fast adaptive algorithm.

2 Multichannel Algorithm Framework

We consider a p-channel adaptive algorithm, where the desired response $d(n)$ is modelled as a sum of p filters of orders N_i $(i = 1,\ldots,p)$ driven by inputs $x_i(n)$ $(i = 1,\ldots,p)$. The adaptive filter parameter vectors are $\mathbf{w}_i(n)$ $(i = 1,\ldots,p)$. We define the N-dimension $(N = N_1 + \ldots + N_p)$ vectors: $\mathbf{x}^T(n) = \left(\mathbf{x}_1^T(n)\cdots\mathbf{x}_p^T(n)\right)$ where $\mathbf{x}_i^T(n) = (x_i(n)\cdots x_i(n-N_i+1))$, and $\mathbf{w}^T(n) = \left(\mathbf{w}_1^T(n)\cdots\mathbf{w}_p^T(n)\right)$. Then the filter error output is

$$\begin{aligned} e(n) &= d(n) - \sum_{i=1}^{p} \mathbf{x}_i^T(n)\,\mathbf{w}_i(n) \\ &= d(n) - \mathbf{x}^T(n)\,\mathbf{w}(n) \end{aligned}$$

In the multichannel QRD-RLS algorithm, the filter coefficient vector $\mathbf{w}(n)$ is identified by solving the least squares problem $\min_{\mathbf{w}(n)} J(n)$ where $J(n) = \sum_{k=0}^{n} \lambda^{n-k}\left[d(k) - \mathbf{x}^T(k)\,\mathbf{w}(n)\right]^2$ and λ is the forgetting factor.

If we denote

$$\mathbf{e}(n) = \mathbf{d}(n) - \mathbf{X}(n)\,\mathbf{w}(n) \tag{1}$$

with

$$\begin{aligned} \mathbf{X}^T(n) &= (\mathbf{x}(n)\cdots\mathbf{x}(0)) \\ \mathbf{d}^T(n) &= (d(n)\cdots d(0)) \end{aligned}$$

and let $\mathbf{\Lambda}(n) = \mathrm{diag}(1,\lambda,\ldots,\lambda^n)$, then we have $J(n) = \|\mathbf{\Lambda}(n)\,\mathbf{e}(n)\|^2 = \left\|\mathbf{Q}^T(n)\,\mathbf{\Lambda}(n)\,\mathbf{e}(n)\right\|^2$ where $\mathbf{Q}(n)$ is an orthogonal matrix such that $\mathbf{Q}^T(n)\,\mathbf{\Lambda}(n)\,\mathbf{X}(n) = \begin{pmatrix} \bigcirc \\ \mathbf{R}(n) \end{pmatrix}$ and $\mathbf{R}(n)$ is a $(N \times N)$ upper triangular matrix. In the following, symbol $\bigcirc$ will indicates any adequate zero matrix.

Consequently, applying $\mathbf{Q}^T(n)\,\mathbf{\Lambda}(n)$ to both sides of equation (1) gives

$$\begin{pmatrix} \bar{\mathbf{e}}^e(n) \\ \bar{\mathbf{e}}^w(n) \end{pmatrix} = \begin{pmatrix} \bar{\mathbf{d}}^e(n) \\ \bar{\mathbf{d}}^w(n) \end{pmatrix} - \begin{pmatrix} \bigcirc \\ \mathbf{R}(n) \end{pmatrix} \mathbf{w}(n)$$

from which we deduce the optimal filter $\mathbf{w}(n)$ of the QRD-RLS algorithm as the solution of

$$\mathbf{R}(n)\,\mathbf{w}(n) = \bar{\mathbf{d}}^w(n) \tag{2}$$

The input data matrix can be expressed recursively $\mathbf{\Lambda}(n)\,\mathbf{X}(n) = \begin{pmatrix} \mathbf{x}^T(n) \\ \lambda\mathbf{\Lambda}(n-1)\,\mathbf{X}(n-1) \end{pmatrix}$, so the matrix $\mathbf{R}(n)$ can be updated recursively.

This is done via an updation scheme of the $(N \times N)$ upper triangular matrix $\mathbf{R}(n)$ and the N-dimension vector $\bar{\mathbf{d}}^{w}(n)$:

$$\mathbf{Q}^{v}(n)\begin{pmatrix} \mathbf{x}^{T}(n) & d(n) \\ \lambda \mathbf{R}(n) & \lambda \bar{\mathbf{d}}^{w}(n) \end{pmatrix} = \begin{pmatrix} \mathbf{0} & \bar{d}_{1}^{e}(n) \\ \mathbf{R}(n+1) & \bar{\mathbf{d}}^{w}(n+1) \end{pmatrix} \tag{3}$$

where $\mathbf{Q}^{v}(n)$ is an orthogonal $((N+1)\times(N+1))$ matrix.

We introduce in the next section an iterative transformation which permits to decouple the channels in order to apply a fast QRD-based algorithmic scheme to each one.

3 Fast Multichannel Algorithm

If we partition $\mathbf{R}(n)$ in p diagonal upper triangular blocks $\mathbf{R}_{k}(n), k=1,\ldots,p$ of sizes $(N_{k}\times N_{k})$, and $p-1$ extra-diagonal blocks $\mathbf{B}_{k,p}(n), k=1,\ldots,p-1$ of sizes $(N_{k}\times(N_{k+1}+\cdots+N_{p}))$ [3]

$$\mathbf{R}(n) = \begin{pmatrix} \mathbf{R}_{1}(n) & & \bigcirc \\ & \ddots & \\ \bigcirc & & \mathbf{R}_{p}(n) \end{pmatrix} + \begin{pmatrix} \bigcirc & \boxed{\mathbf{B}_{1p}(n)} \\ & \vdots \\ & \boxed{\mathbf{B}_{p-1,p}(n)} \\ \bigcirc & \bigcirc \end{pmatrix}.$$

which can also be represented by

$$\begin{aligned} \mathbf{R}^{T}(n) &= \begin{pmatrix} \mathbf{C}_{1}^{T}(n) & \cdots & \mathbf{C}_{p}^{T}(n) \end{pmatrix} \\ \mathbf{C}_{k}(n) &= \begin{pmatrix} \bigcirc & \mathbf{R}_{k}(n) & \mathbf{B}_{k,p}(n) \end{pmatrix},\ k=1,\ldots,p-1 \\ \mathbf{C}_{p}(n) &= \begin{pmatrix} \bigcirc & \mathbf{R}_{p}(n) \end{pmatrix}. \end{aligned}$$

Equation (2) then leads to p subsystems

$$\mathbf{C}_{k}(n)\mathbf{w}_{k,p}(n) = \bar{\mathbf{d}}_{k}^{w}(n), (k=1,\ldots,p) \tag{4}$$

for an adequate partitioning of $\bar{\mathbf{d}}^{w}(n)$ into $\bar{\mathbf{d}}_{k}^{w}(n)$ and with $\mathbf{w}_{k,p}(n) = \left(\mathbf{w}_{k}^{T}(n)\cdots\mathbf{w}_{p}^{T}(n)\right)$. Next let us denote $\mathbf{R}_{k}^{J}(n) = \mathbf{J}.\mathbf{R}_{k}(n)$, $\mathbf{B}_{k,p}^{J}(n) = \mathbf{J}.\mathbf{B}_{k,p}(n)$ and $\bar{\mathbf{d}}_{k}^{wJ}(n) = \mathbf{J}.\bar{\mathbf{d}}_{k}^{w}(n)$ with $\mathbf{J}$ being an up-down permutation. Equation (4) is equivalent to

$$\mathbf{R}_{k}^{J}(n)\mathbf{w}_{k}(n) = \bar{\mathbf{d}}_{k}^{w\prime}(n), \quad k=1,\ldots,p$$

with

$$\begin{aligned} \bar{\mathbf{d}}_{k}^{w\prime}(n) &= \bar{\mathbf{d}}_{k}^{wJ}(n) - \mathbf{B}_{k,p}^{J}(n)\mathbf{w}_{k+1,p}(n) \\ \text{for } k &= 1,\ldots,p-1 \\ \bar{\mathbf{d}}_{p}^{w\prime}(n) &= \bar{\mathbf{d}}_{p}^{wJ}(n) \end{aligned} \tag{5}$$

Thus we have introduced an iterative transformation of both $\left(\mathbf{R}(n), \bar{\mathbf{d}}^{w}(n)\right)$ into $\left(\mathbf{R}_{k}^{J}(n), \bar{\mathbf{d}}_{k}^{w\prime}(n)\right)_{1\le k\le p}$, The above quantities are updated by (see [2])

$$\begin{aligned} &\mathbf{Q}_{k}^{v}(n)\begin{pmatrix} \mathbf{x}_{k}^{T}(n) \\ \lambda \mathbf{R}_{k}^{J}(n-1) \end{pmatrix} = \begin{pmatrix} \bigcirc \\ \mathbf{R}_{k}^{J}(n) \end{pmatrix} \\ &\quad \text{for } k=1,\ldots,p. \\ &\mathbf{Q}_{k}^{v}(n)\begin{pmatrix} \mathbf{z}_{k}^{T}(n) & \alpha_{k}(n-1) \\ \bigcirc & \lambda \bar{\mathbf{d}}_{k}^{w\prime}(n-1) \end{pmatrix} = \begin{pmatrix} \mathbf{z}_{k}^{\prime T}(n) & \alpha_{k}^{\prime}(n) \\ \mathbf{B}_{k,p}^{J}(n) & \bar{\mathbf{d}}_{k}^{wJ}(n) \end{pmatrix} \\ &\bar{\mathbf{d}}_{k}^{wJ}(n) - \mathbf{B}_{k,p}^{J}(n)\mathbf{w}(n) = \bar{\mathbf{d}}_{k}^{w\prime}(n) \\ &\quad \text{for } k=1,\ldots,p-1 \text{ , and} \\ &\mathbf{Q}_{p}^{v}(n)\begin{pmatrix} \alpha_{p}(n-1) \\ \lambda \bar{\mathbf{d}}_{k}^{w\prime}(n-1) \end{pmatrix} = \begin{pmatrix} \alpha_{p}^{\prime}(n) \\ \bar{\mathbf{d}}_{k}^{wJ}(n) \end{pmatrix} \\ &\bar{\mathbf{d}}_{p}^{wJ}(n) = \bar{\mathbf{d}}_{p}^{w\prime}(n) \end{aligned} \tag{6}$$

where $\mathbf{z}_{k}^{T}(n) = \begin{pmatrix} \mathbf{x}_{k+1}^{T}(n) & \cdots & \mathbf{x}_{p}^{T}(n) \end{pmatrix}$.

In the next subsections, we derive two equivalent triangularization schemes of an augmented input data matrix. Which leads to the update equations for the fast $O(N)$ algorithm [5]. In the following, we are concerned with the k^{th} channel (for $k=1,\ldots,p$).

3.1 Backward linear prediction

We consider the backward linear prediction problem of order N_{k} ; it consists on estimating the desired backward input $d_{k}^{b}(n) \triangleq x_{k}(n-N_{k})$ using $\mathbf{x}_{k}(n)$. The error of the backward prediction is

$$e_{k}^{b}(n) = d_{k}^{b}(n) - \mathbf{x}_{k}^{T}(n)\mathbf{w}_{k}^{b}(n)$$

where $\mathbf{w}_{k}^{b}(n)$ is the transversal backward prediction coefficient vector, selected so as to minimize $\left\|\mathbf{\Lambda}(n)\mathbf{e}_{k}^{b}(n)\right\|^{2}$ with

$$\mathbf{e}_{k}^{b}(n) = \mathbf{d}_{k}^{b}(n) - \mathbf{X}_{k,N_{k}}(n)\mathbf{w}_{k}^{b}(n)$$

where $\mathbf{d}_{k}^{bT}(n) = \begin{pmatrix} d_{k}^{b}(n) & \cdots & d_{k}^{b}(0) \end{pmatrix}$ and $\mathbf{X}_{k,N_{k}}^{T}(n) = \begin{pmatrix} \mathbf{x}_{k}(n) & \cdots & \mathbf{x}_{k}(0) \end{pmatrix}$.

Then, by applying an orthogonal matrix $\mathbf{Q}_{k,N_{k}}(n)$ to both sides of the weighted equation, we obtain

$$\mathbf{Q}_{k,N_{k}}(n)\mathbf{\Lambda}(n)\mathbf{e}_{k}^{b}(n) = \begin{pmatrix} \bar{\mathbf{d}}_{k}^{be}(n) \\ \bar{\mathbf{d}}_{k}^{bw}(n) \end{pmatrix} - \begin{pmatrix} \bigcirc \\ \mathbf{R}_{k}^{J}(n) \end{pmatrix}\mathbf{w}_{k}^{b}(n) \tag{7}$$

where $\bar{\mathbf{d}}_{k}^{be}(n)$ is a $(n-N_{k})$-vector, $\bar{\mathbf{d}}_{k}^{bw}(n)$ is a N_{k}-vector.

We construct an augmented input $(n\times(N_{k}+1))$-matrix $\mathbf{X}_{k,N_{k}+1}(n) = \begin{pmatrix} \mathbf{X}_{k,N_{k}}(n) & \mathbf{d}_{k}^{b}(n) \end{pmatrix}$. Then applying $\mathbf{Q}_{k,N_{k}}(n)$ to $\mathbf{X}_{k,N_{k}+1}(n)$ with (7) gives

$$\mathbf{Q}_{k,N_{k}}(n)\mathbf{X}_{k,N_{k}+1}(n) = \begin{pmatrix} \bigcirc & \bar{\mathbf{d}}_{k}^{be}(n) \\ \mathbf{R}_{k}^{J}(n) & \bar{\mathbf{d}}_{k}^{bw}(n) \end{pmatrix}$$

Again, we apply $\mathbf{Q}_{k,N_{k}}(n)$ to the pinning n-vector $\sigma(n) = \begin{pmatrix} 1 & 0 & \cdots & 0 \end{pmatrix}$

$$\mathbf{\Sigma}_{k,N_{k}}(n) = \mathbf{Q}_{k,N_{k}}(n)\sigma(n) = \begin{pmatrix} \gamma_{k,N_{k}}(n) \\ \mathbf{0} \\ \mathbf{g}_{k,N_{k}}(n) \end{pmatrix} \tag{8}$$

where $\mathbf{g}_{k,N_{k}}(n)$ is a N_{k}-vector.

If we denote $\bar{d}^{be}_{k,1}(n)$ the first element of $\bar{\mathbf{d}}^{be}_k(n)$, the backward prediction error is equal to

$$e^b_k(n) = \gamma_{k,N_k}(n)\,\bar{d}^{be}_{k,1}(n)$$

Finally, a supplementary orthogonal transformation $\mathbf{Q}^b_k(n)$ is required to achieve the triangularization of $\mathbf{X}_{k,N_k+1}(n)$

$$\mathbf{Q}^b_k(n)\begin{pmatrix} \bigcirc & \bar{\mathbf{d}}^{be}_k(n) \\ \mathbf{R}^J_k(n) & \bar{\mathbf{d}}^{bw}_k(n) \end{pmatrix} = \begin{pmatrix} \bigcirc & \bigcirc \\ \bigcirc & \varepsilon^b_k(n) \\ \mathbf{R}^J_k(n) & \bar{\mathbf{d}}^{bw}_k(n) \end{pmatrix}$$

Transforming in the same manner $\sigma(n)$ leads to (see [1])

$$\mathbf{\Sigma}_{k,N_k+1}(n) = \mathbf{Q}_{k,N_k+1}(n)\,\sigma(n) = \begin{pmatrix} \gamma_{k,N_k+1}(n) \\ \mathbf{0} \\ \mathbf{g}_{k,N_k+1}(n) \end{pmatrix}$$

with $\mathbf{g}_{k,N_k+1}(n) = \begin{pmatrix} g_{k,N_k+1}(n) \\ \mathbf{g}_{k,N_k}(n) \end{pmatrix}$

3.2 Forward linear prediction

Similarly, we consider the forward linear prediction problem of order N_k ; it consists on estimating the desired forward input $d^f_k(n) \triangleq x_k(n)$ using $\mathbf{x}_k(n-1)$. The error of the forward prediction is

$$e^f_k(n) = d^f_k(n) - \mathbf{x}^T_k(n-1)\,\mathbf{w}^f_k(n)$$

where $\mathbf{w}^b_k(n)$ is the transversal forward prediction coefficient vector, selected so as to minimize $\left\|\mathbf{\Lambda}(n)\,\mathbf{e}^f_k(n)\right\|^2$ with

$$\mathbf{e}^f_k(n) = \mathbf{d}^f_k(n) - \mathbf{X}_{k,N_k}(n-1)\,\mathbf{w}^f_k(n)$$

where $\mathbf{d}^{fT}_k(n) = \begin{pmatrix} d^f_k(n) & \cdots & d^f_k(0) \end{pmatrix}$.

This leads to another expression of $\mathbf{X}_{k,N_k+1}(n) = \begin{pmatrix} \mathbf{d}^f_k(n) & \bar{\mathbf{X}}_{k,N_k}(n-1) \end{pmatrix}$ with $\bar{\mathbf{X}}^T_{k,N_k}(n-1) = \begin{pmatrix} \mathbf{X}^T_{k,N_k}(n-1) & \mathbf{0} \end{pmatrix}$

Then, by applying the orthogonal matrix $\mathbf{Q}_{k,N_k}(n-1)$ of equation (7) to both sides of the weighted equation, we obtain

$$\begin{pmatrix} \mathbf{Q}_{k,N_k}(n-1) & \mathbf{0} \\ \mathbf{0} & 1 \end{pmatrix}\mathbf{\Lambda}(n)\,\mathbf{e}^f_k(n) \tag{9}$$

$$= \begin{pmatrix} \bar{\mathbf{d}}^{fe}_k(n) \\ \bar{\mathbf{d}}^{fw}_k(n) \\ \lambda^n d^f_k(0) \end{pmatrix} - \begin{pmatrix} \bigcirc \\ \mathbf{R}^J_k(n-1) \\ \mathbf{0} \end{pmatrix}\mathbf{w}^f_k(n)$$

where $\bar{\mathbf{d}}^{fe}_k(n)$ is a $(n-N_k-1)$-vector and $\bar{\mathbf{d}}^{fw}_k(n)$ is a N_k-vector.

If we denote $\bar{d}^{fe}_{k,1}(n)$ the first element of $\bar{\mathbf{d}}^{fe}_k(n)$, the forward prediction error is equal to

$$e^f_k(n) = \gamma_k(n)\,\bar{d}^{fe}_{k,1}(n)$$

After the application of $\mathbf{Q}_{k,N_k}(n-1)$ in equation (9), the triangularization of $\mathbf{X}_{k,N_k+1}(n)$ continues by applying an orthogonal transformation $\mathbf{Q}^{fe}_k(n)$ to annihilate $\bar{\mathbf{d}}^{fe}_k(n)$

$$\mathbf{Q}^{fe}_k(n)\begin{pmatrix} \bar{\mathbf{d}}^{fe}_k(n) & \bigcirc \\ \bar{\mathbf{d}}^{fw}_k(n) & \mathbf{R}^J_k(n-1) \\ \lambda^n d^f_k(0) & \mathbf{0} \end{pmatrix} = \begin{pmatrix} \bigcirc & \bigcirc \\ \bar{\mathbf{d}}^{fw}_k(n) & \mathbf{R}^J_k(n-1) \\ \varepsilon^f_k(n) & \mathbf{0} \end{pmatrix}$$

Then a supplementary orthogonal transformation $\mathbf{Q}^{fw}_k(n)$ is required to achieve the triangularization of $\mathbf{X}_{k,N_k+1}(n)$

$$\mathbf{Q}^{fw}_k(n)\begin{pmatrix} \bigcirc & \bigcirc \\ \bar{\mathbf{d}}^{fw}_k(n) & \mathbf{R}^J_k(n-1) \\ \varepsilon^f_k(n) & \mathbf{0} \end{pmatrix} = \begin{pmatrix} \bigcirc \\ \bar{\mathbf{R}}^J_k(n) \end{pmatrix}$$

The same scheme operated on the pinning vector $\sigma(n+1)$ gives another expression of $\mathbf{\Sigma}_{k,N_k+1}(n)$

$$\begin{pmatrix} \gamma_{k,N_k+1}(n) \\ \mathbf{0} \\ \mathbf{g}_{k,N_k+1}(n) \end{pmatrix} = \mathbf{Q}^{fw}_k(n)\begin{pmatrix} \gamma_{k,N_k+1}(n) \\ \mathbf{0} \\ \mathbf{g}_{k,N_k}(n-1) \\ \delta_k(n) \end{pmatrix}$$

Thus we can efficiently update $\gamma_{k,N_k}(n)$ and $\mathbf{g}_{k,N_k}(n)$.

3.3 The filtering section

Let us define the channel k filter output $y_k(n) = \mathbf{x}^T_k(n-1)\,\mathbf{w}^f_k(n)$, and the partial filter output $y_{k,p}(n) = \sum_{i=k}^{p} y_i(n)$, it follows that the filter output error

$$e(n) = d(n) - y_{1,p}(n)$$

From equation (6), we can show that

$$\bar{\mathbf{d}}^{w\prime}_k(n) = \bar{\mathbf{d}}^w_k(n) - y_{k+1,p}(n)\,\mathbf{g}_{k,N_k}(n) \tag{10}$$

which require only $O(N_k)$ complexity. The partial filter output $y_{k,p}(n)$ is computed recursively in the backward direction ($k = p \to 1$) using the partial filter error expression $\gamma_{k,N_k}(n)\,\alpha_k(n) = \alpha_k(n-1) - y_{k,p}(n)$.

The algorithm summary is presented below using the reduced matrix $\mathbf{Q}^v_{k,N_k}(n), \mathbf{Q}^{fev}_k(n)$ and $\mathbf{Q}^{fwv}_k(n)$, rather than the original long matrix involved in the previous section. It is subdivided in two main loops on channels :

- The first one (steps 1-8) is done forwardly; it concerns the updation for the backward-forward prediction problem and the first part of the filtering section (steps 7-8). Like other sequential algorithms [7][6], this loop requires $O(N)$ operations.

- The second one (steps 9-10) is done backwardly, it achieves the second part of the filtering section as well as the block transform eq.(5). This second loop requires only $O(N)$ operations since the block transform can be efficiently computed with the same complexity eq.(10).

The overall algorithm is summarized by

for $n = 0 : L$
$\alpha_1(n-1) = d(n)$
for $k = 1 : p$

1- $\mathbf{Q}^{v}_{k,N_k}(n-1)\begin{pmatrix} x_k(n) \\ \lambda\bar{\mathbf{d}}^{fw}_k(n-1)\end{pmatrix} = \begin{pmatrix} \bar{d}^{fe}_{k,1}(n) \\ \bar{\mathbf{d}}^{fw}_k(n)\end{pmatrix}$

2- $\mathbf{Q}^{fev}_k(n)\begin{pmatrix} \bar{d}^{fe}_{k,1}(n) \\ \lambda\varepsilon^{f}_k(n-1)\end{pmatrix} = \begin{pmatrix} 0 \\ \varepsilon^{f}_k(n)\end{pmatrix}$

3- $\mathbf{Q}^{fev}_k(n)\begin{pmatrix} \gamma_{k,N_k}(n-1) \\ 0\end{pmatrix} = \begin{pmatrix} \gamma_{k,N_k+1}(n) \\ \delta_k(n)\end{pmatrix}$

4- $\mathbf{Q}^{fwv}_k(n)\begin{pmatrix} \bar{\mathbf{d}}^{fw}_k(n) \\ \varepsilon^{f}_k(n)\end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \varepsilon^{b}_k(n)\end{pmatrix}$

5- $\mathbf{Q}^{fwv}_k(n)\begin{pmatrix} \mathbf{g}_{k,N_k}(n-1) \\ \delta_k(n)\end{pmatrix} = \mathbf{g}_{k,N_k+1}(n)$

6- $\mathbf{Q}^{v}_{k,N_k}(n)\begin{pmatrix} 1 \\ \mathbf{0}\end{pmatrix} = \begin{pmatrix} \gamma_{k,N_k}(n) \\ \mathbf{g}_{k,N_k}(n)\end{pmatrix}$

7- $\mathbf{Q}^{v}_{k,N_k}(n)\begin{pmatrix} \alpha_k(n-1) \\ \lambda\bar{\mathbf{d}}^{w\prime}_k(n-1)\end{pmatrix} = \begin{pmatrix} \alpha'_k(n) \\ \bar{\mathbf{d}}^{wJ}_k(n)\end{pmatrix}$

8- $\gamma_{k,N_k}(n)\,\alpha_{k+1}(n-1) = \alpha'_k(n)$

end k

for $k = p : -1 : 1$

9- $\begin{pmatrix} \alpha_k(n) \\ \bar{\mathbf{d}}^{w\prime}_k(n)\end{pmatrix} = \begin{pmatrix} \alpha'_k(n) \\ \bar{\mathbf{d}}^{wJ}_k(n)\end{pmatrix} - \begin{pmatrix} \gamma_{k,N_k}(n) \\ \mathbf{g}_{k,N_k}(n)\end{pmatrix} y_{k+1,p}(n)$

10- $y_{k,p}(n) = \alpha_k(n-1) - \gamma_{k,N_k}(n)\,\alpha_k(n)$

end k

$e(n) = d(n) - y_{1,p}(n)$

end n

4 Simulation

In this section we present a simulation of our algorithm in the context of highly intercorrelated inputs x_k, this case is encountered in some real applications: for example the stereophonic acoustic echo cancellation problem. We consider a two-channel filter of orders 36 and 32. These filters are driven by two highly intercorrelated inputs, $x_1(n)$ and $x_2(n)$ respectively. The result of simulations are averaged over 100 Monte Carlo runs. Figure 1 compares our algorithm with the multichannel QRD-RLS algorithm, when the channels switch characteristics abruptly at iteration $n = 400$. It shows a better convergence of our algorithm.

5 Conclusion

We have introduced a new algorithm to multichannel QRD-based adaptive filtering. Besides, the good numerical properties of the algorithm and its low complexity $O(N)$, our algorithm gives better result than standard QRD-RLS multichannel adaptive filter for highly intercorrelated input channels.

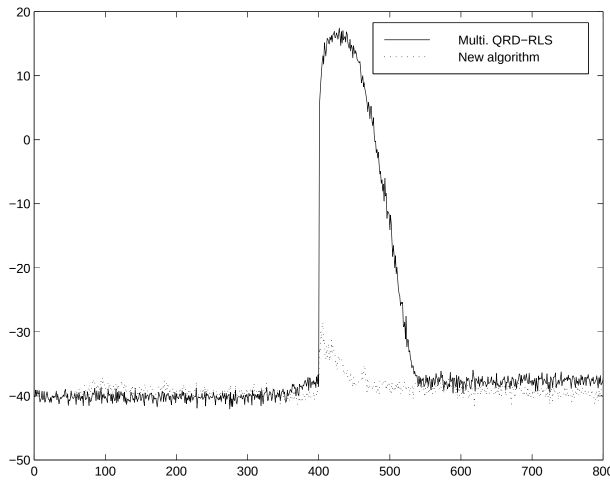

Figure 1: MSE vs. time for the Multichannel QRD-RLS and the New algorithm, with w=0.95

References

[1] M. Bellanger and P. Regalia. The FLS-QR algorithm for adaptive filtering : The case of multichannel signals. *Signal Processing*, 22:115–126, 1991.

[2] M. Bhouri. A new fast approach to multichannel QRD-based adaptive filtering. in preparation, 1998.

[3] M. Bhouri, M. Bonnet, and M. Mboup. A new QRD-based block adaptive algorithm. In *Proc. ICASSP*, pages 1497–1500, May 1998.

[4] J. Cioffi. A fast QR frequency-domain RLS adaptive filter. In *Proc. ICASSP*, pages 407–410, Apr. 1987.

[5] J. Cioffi. The fast adaptive ROTOR's RLS algorithm. *IEEE Trans. ASSP*, 38:631–653, 1990.

[6] A. Rontogiannis and S. Theodoridis. New multichannel fast QRD-LS adaptive algorithms. In *Proc. DSP97*, pages 45–48, July 1997.

[7] M. Syed. QRD-based fast RLS multichannel adaptive algorithms. In *Proc. ICASSP*, pages 1837–1840, Apr. 1991.

[8] M. Syed and V. Mathews. QR-decomposition based algorithms for adaptive volterra filtering. *IEEE Trans. CAS*, I-40:372–382, 1993.