NEW MULTICHANNEL FAST QRD-LS ADAPTIVE ALGORITHMS

Athanasios A. Rontogiannis and Sergios Theodoridis
Department of Informatics
Division of Communications and Signal Processing
University of Athens
GR-157 71 Zografou, GREECE
e-mail : {tronto,stheodor}@di.uoa.gr

ABSTRACT

Two new fast multichannel QR decomposition (QRD) least squares (LS) adaptive algorithms are presented in this paper. Both algorithms deal with the general case of channels with different orders, comprise scalar operations only and are based on numerically robust orthogonal Givens rotations. The first algorithm is a block type scheme which processes all channels jointly. The second algorithm processes each channel separately and offers substantially reduced computational complexity compared to previously derived multichannel fast QRD schemes. This is demonstrated in the context of Volterra filtering.

1 Introduction

Multichannel LS adaptive algorithms [1] find wide applications in diverse areas such as channel equalization, stereophonic echo cancellation, Volterra type nonlinear system identification, to name but a few. Among the various issues, characterizing the performance of an algorithm, those of computational complexity, and numerical robustness are of particular importance, in most applications. Especially the need for numerically robust schemes has led to the development of a class of algorithms based on the QR decomposition of the input data matrix.

Multichannel fast QRD algorithms, which spring from the corresponding single channel fast QRD schemes, have already been developed [3]-[5]. Both the cases of equal [3]-[4] and unequal [5] channel orders have been treated. Especially in [5] a novel channel partitioning technique is introduced, which makes possible the manipulation of channels of different orders. Based on this technique a block as well as a channel decomposition based multichannel QRD algorithm are described.

In this paper, a novel approach for deriving multichannel fast QRD algorithms is introduced. The methodology is based on the efficient time update of a particular vector quantity, which provides all the necessary for the LS error update, rotation parameters. A direct consequence of this approach, is that explicit backward steps are essentially alleviated, a fact which simplifies the derivation procedure. In the new block type scheme the basic vector quantity is updated in one forward step and its computational complexity is similar to that of the block algorithm of [5][1]. Based on the new block scheme, a new channel decomposition algorithm is derived. The time update of the basic vector quantity is now accomplished with l forward steps where l is the number of channels. Compared to [5], this algorithm offers significantly reduced computational complexity in terms of both multiplications/divisions and square roots.

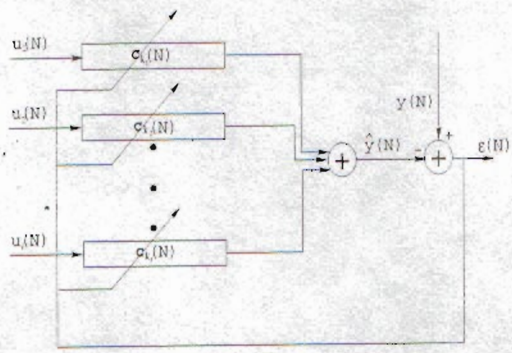

Figure 1: The multichannel model

2 Formulation of the problem

Figure 1 illustrates the typical l input - single output channel problem. We seek to determine the $k \times 1$ coefficients' vector $\mathbf{c}(N)$ to satisfy the following LS optimization scheme

$$\min_{\mathbf{c}(N)} \sum_{n=1}^{N} \lambda^{N-n}[y(n) - \mathbf{c}^T(N)\mathbf{u}(n)]^2 \qquad (1)$$

λ stands for the usual forgetting factor, $\mathbf{c}^T(N) = [\mathbf{c}_{k_1}^T(N) \ldots \mathbf{c}_{k_l}^T(N)]$, $\mathbf{u}^T(n) = [\mathbf{u}_{k_1}^T(n) \ldots \mathbf{u}_{k_l}^T(n)]$ is the $k \times 1$ input data vector and $y(n)$ is the desired

[1] if formulated for the case of general channel orders

0-7803-4137-6/97/$10.00©1977 IEEE

response at time n. $k_1, \ldots, k_l$ denote the channel orders and $k = \sum_{i=1}^{l} k_i$. The input-output information can be used to form the following data matrix

$$[\mathbf{y}(N)|U(N)] = \Lambda(N) \begin{bmatrix} y(1) & \mathbf{u}^T(1) \\ y(2) & \mathbf{u}^T(2) \\ \vdots & \vdots \\ y(N) & \mathbf{u}^T(N) \end{bmatrix} \quad (2)$$

where $\Lambda(N) = diag[\lambda^{\frac{N-1}{2}}, \lambda^{\frac{N-2}{2}}, \ldots, 1]$. If $Q(N)$ stands for the orthogonal matrix which converts $U(N)$ into the $k \times k$ upper triangular form $\tilde{R}(N)$ then

$$Q(N)[\mathbf{y}(N)|U(N)] = \begin{bmatrix} \mathbf{p}(N) & \tilde{R}(N) \\ \mathbf{v}(N) & \bigcirc \end{bmatrix} \quad (3)$$

where $\mathbf{p}(N) \in \mathcal{R}^{k\times 1}$ and $\mathbf{v}(N) \in \mathcal{R}^{(N-k)\times 1}$. It is obvious from (1),(2) and (3) that $\mathbf{c}(N)$ is given by

$$\tilde{R}(N)\mathbf{c}(N) = \mathbf{p}(N) \quad (4)$$

In a time varying environment, time update of $\tilde{R}(N)$ and $\mathbf{p}(N)$ is required. It turns out that all necessary quantities for the update of these matrices can be obtained from the manipulation of the following vector term

$$\mathbf{g}(N+1) = \lambda^{-1/2}\tilde{R}^{-T}(N)\mathbf{u}(N+1) \quad (5)$$

Indeed, if $Q(N+1)$ is a sequence of k Givens rotations which annul the elements of $-\mathbf{g}(N+1)$ as follows

$$Q(N+1)\begin{bmatrix} -\mathbf{g}(N+1) \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ \delta(N+1) \end{bmatrix} \quad (6)$$

then this $Q(N+1)$ updates $\tilde{R}(N)$ as well as $\mathbf{p}(N)$ according to the expression [1]

$$Q(N+1)\begin{bmatrix} \lambda^{1/2}\mathbf{p}(N) \\ y(N+1) \end{bmatrix} = \begin{bmatrix} \mathbf{p}(N+1) \\ \tilde{e}(N+1) \end{bmatrix} \quad (7)$$

The quantity $\delta(N+1)$ in (6) relates the angle normalized error $\tilde{e}(N+1)$ to the a priori error $e(N+1)$ as follows [2]

$$e(N+1) = \delta(N+1)\tilde{e}(N+1) \quad (8)$$

In the following sections we present two new multichannel fast QRD adaptive algorithms which achieve the efficient update of $e(N+1)$.

3 New block algorithm

Without loss of generality we assume that $k_1 \geq k_2 \geq \cdots \geq k_l$. The samples of the input data vector $\mathbf{u}(N)$ are properly permuted and the vector $\mathbf{u}_k(N)$ results. Specifically, we choose the $k_1 - k_2$ most recent samples of the first channel to be the leading elements of $\mathbf{u}_k(N)$, followed by $k_2 - k_3$ pairs of samples of the first and second channel, followed by $k_3 - k_4$ triples of samples of the first three channels etc. followed by k_l l-ples of samples of all channels. The position of the first (most recent) sample of the i-th channel is then given by [8]

$$m_i = \sum_{r=1}^{i-1} r(k_r - k_{r+1}) + i \qquad i = 1, 2, \ldots, l$$

We now define the input data vector $\mathbf{u}_{k+l}^T(N+1) = [u_1(N+1)\ldots u_l(N+1)\ \mathbf{u}_k^T(N)]S$, where S is a permutation matrix which moves $u_i(N+1)$ to the m_i-th position. It can then be verified that $\mathbf{u}_{k+l}^T(N+1) = [\mathbf{u}_k^T(N+1)\ u(N-k_1+1)\ldots u(N-k_l+1)]$, that is, the first k elements of $\mathbf{u}_{k+l}^T(N+1)$ coincide with the input data vector of the next time instant. From $\mathbf{u}_k$, $\mathbf{u}_{k+l}$ the corresponding input data matrices U_k, U_{k+l} can be defined, as in (2). If $\tilde{R}_k, \tilde{R}_{k+l}$, stand for the Cholesky factors of these matrices, the respective vectors $\mathbf{g}_k, \mathbf{g}_{k+l}$ can be obtained according to (5). The new block algorithm is based on the update of $\mathbf{g}_k$ according to the following scheme

$$\mathbf{g}_k(N) \rightarrow \mathbf{g}_{k+l}(N+1) \rightarrow \mathbf{g}_k(N+1) \quad (9)$$

From the above definitions it is not difficult to show that [8]

$$\mathbf{g}_{k+l}(N+1) = \begin{bmatrix} \mathbf{g}_k(N+1) \\ \mathbf{g}^{(k)}(N+1) \end{bmatrix} \quad (10)$$

that is, the first k elements of $\mathbf{g}_{k+l}(N+1)$ provide the vector $\mathbf{g}_k$ of the next time instant. Moreover, by developing the relation between $\tilde{R}_k(N-1)$ and $\tilde{R}_{k+l}(N)$ [8], vector $\mathbf{g}_{k+l}(N+1)$ can be computed from $\mathbf{g}_k(N)$ as follows

$$\mathbf{g}_{k+l}(N+1) = S^T Q_k^f(N) \begin{bmatrix} \mathbf{r}_k(N+1) \\ \mathbf{g}_k(N) \end{bmatrix}$$

$\mathbf{r}_k(N+1)$ is a $l \times 1$ normalized forward error vector and $Q_k^f(N)$ is a sequence of $\sum_{i=1}^{l}(k+i-m_i)$ Givens rotations, as explained later on.

The new block multichannel fast QRD algorithm is shown in figure 2. In this figure, the $k \times l$ matrix $P_k^f(N)$ is defined from the equation $\tilde{R}_k(N-1)\tilde{A}_k^f(N) = P_k^f(N)$, where $\tilde{A}_k^f(N)$ is the LS solution of the multichannel forward problem. Furthermore, $\mathbf{u}_{N+1}^T = [u_1(N+1)\ldots u_l(N+1)]$ and $\tilde{\mathbf{c}}_k^f(N+1)$ stands for the angle normalized forward error vector. $\tilde{A}_k^f(N)$ denotes the Cholesky factor of the forward error covariance matrix and $\tilde{Q}_k(N+1)$ is a sequence of l Givens rotations which successively nullify the elements of $\tilde{\mathbf{c}}_k^f(N+1)$ with respect to the diagonal elements of $\tilde{A}_k^f(N)$. Note that the rotations of $Q_k^f(N+1)$ are generated from the successive annihilation of the last $k+i-m_i$ elements of the i-th

1. $\hat{Q}_k(N)\begin{bmatrix} \lambda^{1/2}P^f_k(N) \\ \mathbf{u}_{N+1} \end{bmatrix} = \begin{bmatrix} P^f_k(N+1) \\ (\hat{\mathbf{e}}^f_k(N+1))^T \end{bmatrix}$

2. $\hat{Q}_k(N+1)\begin{bmatrix} \lambda^{1/2}\tilde{\Lambda}^f_k(N) \\ (\hat{\mathbf{e}}^f_k(N+1))^T \end{bmatrix} = \begin{bmatrix} \tilde{\Lambda}^f_k(N+1) \\ \mathbf{0}^T \end{bmatrix}$

3. $\hat{Q}_k(N+1)\begin{bmatrix} -\mathbf{r}_k(N+1) \\ \delta_k(N) \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ * \end{bmatrix}$

4. $S^T\hat{Q}^f_k(N)\begin{bmatrix} \mathbf{r}_k(N+1) \\ \mathbf{g}_k(N) \end{bmatrix} = \begin{bmatrix} \mathbf{g}_k(N+1) \\ \mathbf{g}^{(k)}(N+1) \end{bmatrix}$

5. $\hat{Q}^f_k(N+1)\begin{bmatrix} \tilde{\Lambda}^f_k(N+1) \\ P^f_k(N+1) \end{bmatrix} = \begin{bmatrix} * \\ * \\ \bigcirc \end{bmatrix}$

6. $\hat{Q}_k(N+1)\begin{bmatrix} -\mathbf{g}_k(N+1) \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \delta_k(N+1) \end{bmatrix}$

7. $\hat{Q}_k(N+1)\begin{bmatrix} \lambda^{1/2}\mathbf{p}_k(N) \\ y(N+1) \end{bmatrix} = \begin{bmatrix} \mathbf{p}_k(N+1) \\ \hat{c}_k(N+1) \end{bmatrix}$

8. $c_k(N+1) = \delta_k(N+1)\hat{c}_k(N+1)$

Initialization
$\mathbf{g}_k(0) = \mathbf{0}$, $\mathbf{p}_k(0) = \mathbf{0}$, $P^f_k(0) = \bigcirc$
$\delta_k(0) = 1$, $\tilde{\Lambda}_k(0) = \mu I$, ($\mu$: small constant)
$\hat{Q}_k(0) = I$, $\hat{Q}^f_k(0) = I$

Figure 2: The new block multichannel algorithm

column of $P^f_k(N+1)$ (for $i = 1,\ldots,l$) with respect to the diagonal elements of $\tilde{\Lambda}^f_k(N+1)$. The $k \times 1$ vector $\mathbf{p}_k(N)$ is defined from $\tilde{R}_k(N)\mathbf{c}_k(N) = \mathbf{p}_k(N)$, where $\mathbf{c}_k(N)$ results from $\mathbf{c}(N)$ after its elements have been permuted as described at the beginning of this section. Finally, $e_k(N+1) \equiv e(N+1)$.
The new algorithm of figure 2 is based exclusively on orthogonal Givens rotations. Despite its block nature, the algorithm comprises scalar operations only. Due to step 5 of figure 2, the proposed scheme is of $O(kl^2)$ computational complexity, similar to that of the block multichannel algorithm of [5]. Reduction of the computational burden by an order of magnitude is achieved with the channel decomposition based algorithm of the next section.

4 New channel decomposition algorithm

The computational complexity can be reduced if the first step in (9) is decomposed in l forward steps. This is accomplished by defining the following input data vectors: $\mathbf{u}^T_{k+1}(N+1)= [u_1(N+1)\,\mathbf{u}^T_k(N)]$ and $\mathbf{u}^T_{k+i}(N+1)= [u_i(N+1)\,\mathbf{u}^T_{k+i-1}(N+1)]S_i$ for $i = 2,3,\ldots,l$. S_i is a permutation matrix which moves $u_i(N+1)$ to the m_i-th position after left shifting the first $m_i - 1$ elements of $\mathbf{u}^T_{k+i-1}(N+1)$. The following input data matrices can now be defined for $i = 0,1,\ldots,l$

$$U_{k+i}(N) = \Lambda(N)\begin{bmatrix} \mathbf{u}^T_{k+i}(1) \\ \mathbf{u}^T_{k+i}(2) \\ \vdots \\ \mathbf{u}^T_{k+i}(N) \end{bmatrix} \qquad (11)$$

If $\tilde{R}_{k+i}(N)$ stands for the Cholesky factor of $U_{k+i}(N)$, the corresponding vectors $\mathbf{g}_{k+i}(N+1)$ can be expressed as

$$\mathbf{g}_{k+i}(N+1) = \lambda^{-1/2}\tilde{R}^{-T}_{k+i}(N)\mathbf{u}_{k+i}(N+1) \qquad (12)$$

Time update of $\mathbf{g}_k(N)$ can now be realized according to the following scheme

$$\mathbf{g}_k(N) \rightarrow \mathbf{g}_{k+1}(N+1) \rightarrow \cdots \rightarrow \mathbf{g}_{k+l}(N+1)$$

It is clear that because of (10) explicit backward steps are essentially avoided and thus our methodology only requires forward steps.
From the above vector definitions and the relations between successive Cholesky factors for $i = 1,\ldots,l$ [6], we obtain the following expressions

$$\mathbf{g}_{k+1}(N+1) = Q^{f(1)}_k(N)\begin{bmatrix} r^{(1)}_k(N+1) \\ \mathbf{g}_k(N) \end{bmatrix}$$

$$\mathbf{g}_{k+i}(N+1) = S^T_i Q^{f(i)}_{k+i-1}(N)\begin{bmatrix} r^{(i)}_{k+i-1}(N+1) \\ \mathbf{g}_{k+i-1}(N+1) \end{bmatrix}$$

$\hat{Q}^{f(i)}_{k+i-1}(N)$ is a sequence of $(k+i) - m_i$ Givens rotations that nullify the last $(k+i) - m_i$ elements of a rotated vector $\mathbf{p}^{(i)}_{k+i-1}(N)$ with respect to the energy $\tilde{a}^{(i)}_{k+i-1}(N)$ in a bottom-up procedure. The time update of $\mathbf{p}^{(i)}_{k+i-1}$ is realized through the application of a Givens matrix $Q^{(i-1)}_{k+i-1}$, whose rotations have been computed in the previous forward step as follows

$$\hat{Q}^{(i-1)}_{k+i-1}(N)\begin{bmatrix} -\mathbf{g}_{k+i-1}(N) \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ \delta^{(i-1)}_{k+i-1}(N-1) \end{bmatrix}$$

Especially the rotations produced at the l-th forward step are used in the filtering part of the algorithm as well as in the first forward step of the next time instant.
The new algorithm is shown in figure 3. Since k rotation parameters are essentially used in the filtering section, each forward procedure can be restricted to k elementary steps (instead of $k+i$). In figure 3, $\theta^{(i-1)}_j(N)$, $j = 1,2,\ldots,k$ stand for the first k rotation angles of $Q^{(i-1)}_{k+i-1}(N)$ while $\phi^{(i)}_j(N+1)$, $j = m_i, m_i+1,\ldots,k$ are the last $k - m_i + 1$ rotation angles of $Q^{f(i)}_{k+i-1}(N+1)$. $p^{(i)}_j(N)$ is the j-th element of $\mathbf{p}^{(i)}_{k+i-1}(N)$ and $g^{(i)}_{j-1}(N+1)$ denotes the j-th element of $\mathbf{g}_{k+i}(N+1)$. In order to maintain a unified notation the following conventions are adopted

$\delta_0^{(i)}(N) = 1$; $\bar{e}_0(N+1) = y(N+1)$;
for $i = 1 : l$,
  $\bar{z}_0^{(i)}(N+1) = u_i(N+1)$;
  for $j = 1 : k$, (Step 1)
    $p_j^{(i)}(N+1) = \lambda^{1/2}\cos[\theta_j^{(i-1)}(N)]p_j^{(i)}(N) + \sin[\theta_j^{(i-1)}(N)]\bar{e}_{j-1}^{(i)}(N+1)$;
    $\bar{e}_j^{(i)}(N+1) = \cos[\theta_j^{(i-1)}(N)]\bar{e}_{j-1}^{(i)}(N+1) - \lambda^{1/2}\sin[\theta_j^{(i-1)}(N)]p_j^{(i)}(N)$;
  end;
  $r_k^{(i)}(N+1) = \frac{\bar{e}_k^{(i)}(N+1)\delta_k^{(i-1)}(N)}{\sqrt{\lambda}\bar{a}_k^{(i)}(N)}$;
  for $j = k : -1 : m_i$, (Step 2)
    $r_{j-1}^{(i)}(N+1) = \cos[\phi_j^{(i)}(N)]r_j^{(i)}(N+1) - \sin[\phi_j^{(i)}(N)]g_{j-1}^{(i-1)}(N+1)$;
    $g_j^{(i)}(N+1) = -\sin[\phi_j^{(i)}(N)]r_j^{(i)}(N+1) + \cos[\phi_j^{(i)}(N)]g_{j-1}^{(i-1)}(N+1)$;
  end;
  $g_{m_i-1}^{(i)}(N+1) = r_{m_i-1}^{(i)}(N+1)$;
  $\bar{a}_k^{(i)}(N+1) = \sqrt{(\sqrt{\lambda}\bar{a}_k^{(i)}(N))^2 + (\bar{e}_k^{(i)}(N+1))^2}$;
  for $j = k : -1 : m_i$, (Step 3)
    $\bar{a}_{j-1}^{(i)}(N+1) = \sqrt{(\bar{a}_j^{(i)}(N))^2 + (p_j^{(i)}(N+1))^2}$;
    $\cos[\phi_j^{(i)}(N+1)] = \frac{\bar{a}_j^{(i)}(N+1)}{\bar{a}_{j-1}^{(i)}(N+1)}$;
    $\sin[\phi_j^{(i)}(N+1)] = \frac{p_j^{(i)}(N+1)}{\bar{a}_{j-1}^{(i)}(N+1)}$;
  end;
  for $j = m_i : k$, (Step 4)
    $\delta_j^{(i)}(N) = \sqrt{(\delta_{j-1}^{(i)}(N))^2 + (g_{j-1}^{(i)}(N+1))^2}$;
    $\cos[\theta_j^{(i)}(N)] = \frac{\delta_{j-1}^{(i)}(N)}{\delta_j^{(i)}(N)}$;
    $\sin[\theta_j^{(i)}(N)] = \frac{g_{j-1}^{(i)}(N+1)}{\delta_j^{(i)}(N)}$;
  end;
end; { i-loop }
for $j = 1 : k$, (Step 5)
  $p_j(N+1) = \lambda^{1/2}\cos[\theta_j^{(0)}(N+1)]p_j(N) + \sin[\theta_j^{(0)}(N+1)]\bar{e}_{j-1}(N+1)$;
  $\bar{e}_j(N+1) = \cos[\theta_j^{(0)}(N+1)]\bar{e}_{j-1}(N+1) - \lambda^{1/2}\sin[\theta_j^{(0)}(N+1)]p_j(N)$;
end;
$e_k(N+1) = \delta_k^{(0)}(N+1)\bar{e}_k(N+1)$;

Initialization
$p_j(0) = 0$, $g_j^{(i)}(0) = 0$, $\cos[\theta_j^{(0)}(0)] = 1$, $j = 1, 2, \ldots, k$
$p_j^{(i)}(0) = 0$, $\cos[\phi_j^{(i)}(0)] = 1$, $i = 1, 2, \ldots, l$, $j = 1, 2, \ldots, m_i$
$\delta_k^{(0)}(0) = 1$, $\bar{a}_k^{(i)}(0) = \mu$, $i = 1, 2, \ldots, l$

Figure 3: The new channel decomposition algorithm

for the l-th forward step : $\theta_j^{(l)}(N) = \theta_j^{(0)}(N+1)$, $\delta_j^{(l)}(N) = \delta_j^{(0)}(N+1)$ and $g_{j-1}^{(l)}(N) = g_{j-1}^{(0)}(N+1)$, $j = 1, 2, \ldots, k$.
The multichannel algorithm of figure 3 is based exclusively on orthogonal Givens rotations. As a result, its numerical performance is expected to be favorable. The new algorithm offers reduced computational complexity if compared to the channel decomposition based algorithm of [5]. The computational requirements of the two algorithms for the second order Volterra problem treated in [5], are shown in table 1. We observe that the new algorithm offers significant computational savings in terms of both the number of multiplications/divisions (15% less) and the number of square roots (20% less). This improvement can be even greater for a different application (in the higher order Volterra case, for instance). Moreover, it can be shown that a modification of the algorithm of figure 3 leads to a scheme which can be implemented on a circular systolic architecture [7] where the a priori error is produced every l clock cycles. Note that there is not such a possibility for

Alg.	Mults/Divs	Sqrt.'s
[5]	$7.6L^3 + 38.5L^2 + O(L)$	$\frac{5}{6}L^3 + 2.7L^2 + O(L)$
New	$6.5L^3 + 21.5L^2 + O(L)$	$\frac{2}{3}L^3 + 1.5L^2 + O(L)$

Table 1: Comparison of complexities

corresponding algorithm of [5].
It can be shown that the fast QRD aglorithms of [3]-[5] are obtained if one adopts the vector

$$\mathbf{q}_k(N) = \bar{R}_k^{-T}(N)\mathbf{u}_k(N)$$

in place of $\mathbf{g}_k$. *Then, a similar methodology can also be followed for the derivation of these algorithms. In other words, our technique provides a unified framework for the development of multichannel fast QRD algorithms.*

5 Conclusion

Two new multichannel fast QRD algorithms are described in this paper. The algorithms are based on numerically robust orthogonal Givens rotations and consist of scalar operations only. Moreover, the second algorithm offers substantially reduced computational complexity compared to a known algorithm of the same category.

References

[1] N. Kalouptsidis and S. Theodoridis Eds., *Adaptive system identification and signal processing algorithms*, Engelwood Cliffs, NJ : Prentice Hall, 1993.

[2] A.A. Rontogiannis, S. Theodoridis, "New fast inverse QR least squares adaptive algorithms", *Proc. IEEE ICASSP*, pp. 1412-1415, Detroit, MI, May 1995.

[3] M.G. Bellanger, P.A. Regalia, "The FLS-QR algorithm for adaptive filtering : the case of multichannel signals", *Signal Processing* 22, pp. 115-126, 1991.

[4] M.A. Syed, "QRD-based fast RLS multichannel adaptive algorithms", *Proc. IEEE ICASSP*, pp. 1837-1840, Toronto 1991.

[5] M.A. Syed, V.J. Mathews, "QR-decomposition based algorithms for adaptive Volterra filtering", *IEEE Trans. CAS-I-40*, no.6, pp. 372-382, June 1993.

[6] A.A. Rontogiannis, S. Theodoridis, "Multichannel fast QRD-LS adaprive filtering: New technique and algorithms", under review, *IEEE Trans. on SP*.

[7] A.A. Rontogiannis, S. Theodoridis, "A highly parallel multichannel fast QRD-LS adaptive algorithm, *Proc. Eusipco 96*, pp. 1381-1384, Trieste, Sept. 1996.

[8] A.A. Rontogiannis, "New recursive LS algorithms based on orthogonal transformations", Ph.D. Dissertation, Feb. 1997.