

A Lattice version of the Multichannel Fast QRD Algorithm Based on *a posteriori* backward errors

Antônio L. L. Ramos and José A. Apolinário Jr. *

Instituto Militar de Engenharia, **IME – DE/3**
Praça General Tibúrcio, 80
22290-270, Rio de Janeiro, RJ, Brazil

Abstract. Fast QR decomposition (QRD) RLS algorithms based on backward prediction errors are well known for their good numerical behavior and their low complexity when compared to similar algorithms with forward error update. Although the basic matrix expressions are similar, their application to multiple channel input signals generate more complex equations. This paper presents a lattice version of the multichannel fast QRD algorithm based on *a posteriori* backward errors updating. This new algorithm comprises scalar operations only; its modularity and pipelinability favors its systolic array implementation.

1 Introduction

Digital processing of multichannel signals using adaptive filters has recently found a variety of new applications including color image processing, multi-spectral remote sensing imagery, biomedicine, channel equalization, stereophonic echo cancellation, multidimensional signal processing, Volterra –type nonlinear system identification, and speech enhancement [1]. This increased number of applications has spawned a renewed interest in efficient multichannel algorithms. One class of algorithms, known as multichannel fast QR decomposition least-squares adaptive algorithms based on backward error updating, has become an attractive option because of fast convergence properties and reduced computational complexity.

In the case of one single channel, a unified formulation for Fast QRD-LS algorithms is available in [2]. In this paper, a new algorithm based on the *a posteriori* backward error updating is developed, using an approach similar to the one used in [3]. Our starting point is the block multichannel algorithm based on the *a posteriori* backward error updating presented in [4]. It is well known that, due to its blocking characteristic, the algorithm of [4] deals with matrix inversions which are potential sources of instability. To overcome this, a new expression is derived; moreover, with the help of order recursive implementations, the resulting new lattice–type algorithm is introduced. This new algorithm uses scalar operations only.

This paper is organized as follows. In Section 2, we review the basic matrix equations of the Multichannel Fast QR Decomposition Least Squares (**MCFQRD-LS**) algorithms based on backward prediction errors. Section 3

* The authors thank CNPq and CAPES for partial funding of this paper.

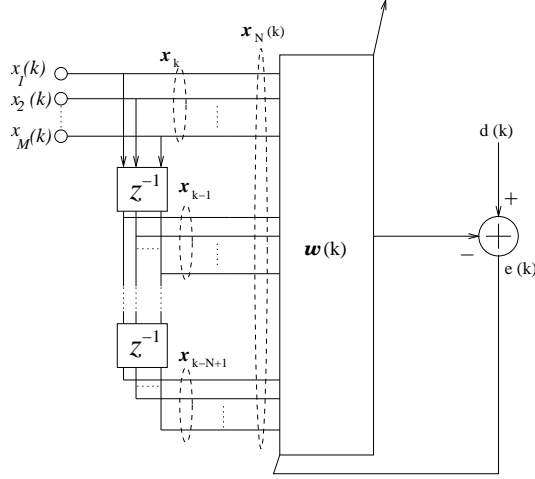


Fig. 1. Multichannel Adaptive Filter.

addresses the Multichannel Fast QRD based on the updating of the *a posteriori* error vector (**MCFQRD_POS_B**). In Section 4, the Lattice version of the **MCFQRD_POS_B** algorithm is presented. Simulation results and conclusions are Summarized in Sections 5 and 6, respectively.

2 The Multichannel Fast QRD Algorithm Based on Backward Prediction Errors Updating

The objective function to be minimized according to the least-squares (LS) algorithm is defined as

$$\xi_{LS}(k) = \sum_{i=0}^k \lambda^{k-i} e^2(i) = \mathbf{e}^T(k) \mathbf{e}(k) \quad (1)$$

where $\mathbf{e}(k) = [e(k) \quad \lambda^{1/2}e(k-1) \quad \dots \quad \lambda^{k/2}e(0)]^T$ is an error vector and may be represented as follows.

$$\mathbf{e}(k) = \begin{bmatrix} d(k) \\ \lambda^{1/2}d(k-1) \\ \vdots \\ \lambda^{k/2}d(0) \end{bmatrix} - \begin{bmatrix} \mathbf{x}_N^T(k) \\ \lambda^{1/2}\mathbf{x}_N^T(k-1) \\ \vdots \\ \lambda^{k/2}\mathbf{x}_N^T(0) \end{bmatrix} \mathbf{w}_N(k) = \mathbf{d}(k) - \mathbf{X}_N(k) \mathbf{w}_N(k) \quad (2)$$

where, as seen in Figure 1,

$$\mathbf{x}_N^T(k) = [\mathbf{x}_k^T \quad \mathbf{x}_{k-1}^T \quad \dots \quad \mathbf{x}_{k-N+1}^T] \quad (3)$$

and $\mathbf{x}_k^T = [x_1(k) \quad x_2(k) \quad \dots \quad x_M(k)]$ is the input vector at time instant k . Note that N is defined here as the order or the number of filter coefficients per channel, M is the number of input channels, and $\mathbf{w}_N(k)$ is the $MN \times 1$ coefficient vector at time instant k .

If $\mathbf{U}_N(k)$ is the Cholesky factor of the $(k+1) \times M$ input data matrix $\mathbf{X}_N(k)$, obtained through the Givens rotation matrix $\mathbf{Q}_N(k)$, then

$$\mathbf{e}_q(k) = \mathbf{Q}_N(k)\mathbf{e}(k) = \begin{bmatrix} \mathbf{e}_{q1}(k) \\ \mathbf{e}_{q2}(k) \end{bmatrix} = \begin{bmatrix} \mathbf{d}_{q1}(k) \\ \mathbf{d}_{q2}(k) \end{bmatrix} - \begin{bmatrix} \mathbf{0} \\ \mathbf{U}_N(k) \end{bmatrix} \mathbf{w}_N(k) \quad (4)$$

From the definition of the forward prediction problem in a multichannel scenario, we can write matrix $\mathbf{X}_{N+1}(k)$ as follows

$$\mathbf{X}_{N+1}(k) = \left[\begin{array}{c|c} \mathbf{D}_f(k) & \mathbf{X}_N(k-1) \\ \hline \mathbf{0}_{(M-1) \times (MN+M)} & \mathbf{0}^T \end{array} \right] \quad (5)$$

where $\mathbf{D}_f(k) = [\mathbf{x}_k \ \lambda^{1/2}\mathbf{x}_{k-1} \cdots \lambda^{k/2}\mathbf{x}_0]^T$ is the $(k+1) \times M$ forward reference signal and the subscript $N+1$ corresponds to the $(N+1)$ th order problem.

The triangularization process of $\mathbf{X}_{N+1}(k)$ leading to $\mathbf{U}_{N+1}(k)$ is performed applying Givens rotations to (5) as follows.

$$\left[\begin{array}{cc} \mathbf{Q}(k-1) & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{M \times M} \end{array} \right] \left[\begin{array}{c|c} \mathbf{D}_f(k) & \mathbf{X}_N(k-1) \\ \hline \mathbf{0}_{(M-1) \times (MN+M)} & \mathbf{0}^T \end{array} \right] = \left[\begin{array}{cc} \mathbf{E}_{fq1}(k) & \mathbf{0} \\ \mathbf{D}_{fq2}(k) & \mathbf{U}_N(k-1) \\ \lambda^{1/2}\mathbf{x}_0^T & \mathbf{0}^T \\ \mathbf{0}_{(M-1) \times (MN+M)} & \end{array} \right] \quad (6)$$

Now, premultiplying (6) by a series of Givens rotations $\mathbf{Q}_f(k)$, that zeroes its first $k - MN$ rows, and by $\mathbf{Q}'_f(k)$ that completes the triangularization process, we have

$$\mathbf{U}_{N+1}(k) = \mathbf{Q}'_{\theta f}(k) \left[\begin{array}{cc} \mathbf{D}_{fq2}(k) & \mathbf{U}_N(k-1) \\ \mathbf{E}_f(k) & \mathbf{0} \end{array} \right] \quad (7)$$

after removing the resulting *null* sections. In (7), $\mathbf{Q}'_{\theta f}(k)$ is a fixed order matrix obtained from $\mathbf{Q}'_f(k)$. It is clear that the $(MN+M) \times (MN+M)$ matrix $\mathbf{Q}'_{\theta f}(k)$ contains the Givens rotations that annihilates $\mathbf{D}_{fq2}(k)$ against the diagonal of $\mathbf{E}_f(k)$, the $M \times M$ Cholesky factor of the forward error covariance matrix.

Based on (7), it is possible to obtain

$$[\mathbf{U}_{N+1}(k)]^{-1} = \left[\begin{array}{cc} \mathbf{0} & \mathbf{E}_f^{-1}(k) \\ \mathbf{U}_N^{-1}(k-1) & -\mathbf{U}_N^{-1}(k-1)\mathbf{D}_{fq2}(k)\mathbf{E}_f^{-1}(k) \end{array} \right] \mathbf{Q}'_{\theta f}(k). \quad (8)$$

The result in (8) will be used in the next section to derive an expression for the updating of the *a posteriori* backward prediction error vector. Also from (7) we can write

$$\left[\begin{array}{c} \mathbf{0} \\ \mathbf{E}_f^0(k) \end{array} \right] = \mathbf{Q}'_{\theta f}(k+1) \left[\begin{array}{c} \mathbf{D}_{fq2}(k) \\ \mathbf{E}_f(k) \end{array} \right] \quad (9)$$

where $\mathbf{E}_f^0(k)$ corresponds to the zero order forward error covariance matrix.

Algebraic manipulations based on Givens rotations applied to (6), generate the following equation using fixed order matrix $\mathbf{Q}_\theta(k)$ obtained from $\mathbf{Q}_N(k)$.

$$\left[\begin{array}{c} \tilde{\mathbf{e}}_{fq1}^T(k+1) \\ \mathbf{D}_{fq2}(k+1) \end{array} \right] = \mathbf{Q}_\theta(k) \left[\begin{array}{c} \mathbf{x}_{k+1}^T \\ \lambda^{1/2}\mathbf{D}_{fq2}(k) \end{array} \right] \quad (10)$$

where $\tilde{\mathbf{e}}_{fq_1}^T(k+1)$ is the first row of $\mathbf{E}_{fq_1}^T(k+1)$.

Similarly, from (6) it is possible to obtain

$$\begin{bmatrix} \mathbf{0}^T \\ \mathbf{E}_f(k+1) \end{bmatrix} = \overline{\mathbf{Q}}_f(k+1) \begin{bmatrix} \tilde{\mathbf{e}}_{fq_1}^T(k+1) \\ \lambda^{1/2} \mathbf{E}_f(k) \end{bmatrix} \quad (11)$$

where $\overline{\mathbf{Q}}_f(k+1)$ is a fixed order matrix of the orthogonal matrix $\mathbf{Q}_f(k+1)$, responsible for annihilate $\tilde{\mathbf{e}}_{fq_1}^T(k+1)$ against $\lambda^{1/2} \mathbf{E}_f(k)$.

Finally, the joint process estimation is performed by the following expressions [1]

$$\begin{bmatrix} \mathbf{e}_{q_1}(k+1) \\ \mathbf{d}_{q_2}(k+1) \end{bmatrix} = \mathbf{Q}_\theta(k+1) \begin{bmatrix} d(k+1) \\ \lambda^{1/2} \mathbf{d}_{q_2}(k) \end{bmatrix} \quad (12)$$

$$e'(k) = e_{q_1}(k)/\gamma(k) = e(k)/\gamma^2(k) \quad (13)$$

Remark: The expression needed to obtain $\mathbf{Q}_\theta(k)$ depends on the type of error to be updated (*a priori* or *a posteriori*) and is provided in the next section for the *a posteriori* case.

3 The Multichannel Fast QRD_POS_B Algorithm

Regardless the type of triangularization applied to $\mathbf{X}_N(k)$ to generate $\mathbf{U}_N(k)$, matrix $\mathbf{Q}_\theta(k)$ can be partitioned as

$$\mathbf{Q}_\theta(k) = \begin{bmatrix} \gamma(k) & \mathbf{g}_N^T(k) \\ \mathbf{f}_N(k) & \mathbf{E}_N(k) \end{bmatrix} \quad (14)$$

where $\gamma(k) = \prod_{i=0}^{MN} \cos \theta_i(k)$; $\mathbf{f}_N(k)$, $\mathbf{g}_N(k)$, and $\mathbf{E}_N(k)$ have more complicated expressions and depend on the type of triangularization used (in our case, lower triangularization, corresponding to backward prediction errors).

From (6) and (14), we can write

$$\begin{bmatrix} \mathbf{0}^T \\ \mathbf{U}_N(k) \end{bmatrix} = \mathbf{Q}_\theta(k) \begin{bmatrix} \mathbf{x}_N^T(k) \\ \lambda^{1/2} \mathbf{U}_N(k-1) \end{bmatrix} = \begin{bmatrix} \gamma(k) & \mathbf{g}_N^T(k) \\ \mathbf{f}_N(k) & \mathbf{E}_N(k) \end{bmatrix} \begin{bmatrix} \mathbf{x}_N^T(k) \\ \lambda^{1/2} \mathbf{U}_N(k-1) \end{bmatrix} \quad (15)$$

We know that $\mathbf{Q}_\theta(k)$ is orthogonal. Hence,

$$\mathbf{I}_{N+1} = \mathbf{Q}_\theta(k) \mathbf{Q}_\theta^T(k) = \begin{bmatrix} \gamma(k) & \mathbf{g}_N^T(k) \\ \mathbf{f}_N(k) & \mathbf{E}_N(k) \end{bmatrix} \begin{bmatrix} \gamma(k) & \mathbf{f}_N^T(k) \\ \mathbf{g}_N(k) & \mathbf{E}_N^T(k) \end{bmatrix} \quad (16)$$

From (15) and (16), we can obtain the two following relations:

$$\mathbf{f}_N(k) \mathbf{x}_N^T(k) + \lambda^{1/2} \mathbf{E}_N(k) \mathbf{U}_N(k-1) = \mathbf{U}_N(k) \quad (17)$$

and

$$\gamma(k) \mathbf{f}_N(k) + \mathbf{E}_N(k) \mathbf{g}_N(k) = \mathbf{0} \quad (18)$$

From (15), we can observe that $\mathbf{U}_N(k)$ is Cholesky factor of $[\mathbf{x}_N(k) \quad \lambda^{1/2}\mathbf{U}_N^T(k-1)]^T$. Hence, we can premultiply (15) by its transpose to have

$$\mathbf{U}_N^T(k)\mathbf{U}_N(k) = \mathbf{x}_N(k)\mathbf{x}_N^T(k) + \lambda\mathbf{U}_N^T(k-1)\mathbf{U}_N(k-1) \quad (19)$$

Premultiplying (17) by $\mathbf{U}_N^T(k)$ and comparing to (19) we can write

$$\mathbf{f}_N(k) = \mathbf{U}_N^{-T}(k)\mathbf{x}_N(k) \quad (20)$$

$$\mathbf{E}_N(k) = \lambda^{1/2}\mathbf{U}_N^{-T}(k-1)\mathbf{U}_N^T(k-1) \quad (21)$$

$\mathbf{f}_N(k)$ is referred to as the *a posteriori* backward error vector and it is our particular vector of interest in the fast QRD-RLS algorithms based on *a posteriori* backward error updating. By substituting (20) and (21) in (18), it is possible to obtain

$$\mathbf{g}_N(k) = -\gamma(k)\mathbf{U}_N^{-T}(k-1)\mathbf{x}_N(k)/\sqrt{\lambda} \quad (22)$$

which is the quantity of interest in the Fast QRD-RLS algorithms based on the *a priori* backward error updating.

From (20), it is straightforward to see that

$$\mathbf{f}_{N+1}(k+1) = \mathbf{U}_{N+1}^{-T}(k+1)\mathbf{x}_{N+1}(k+1) \quad (23)$$

Now, combining (8) and the expression above, we obtain [5]

$$\mathbf{f}_{N+1}(k+1) = \mathbf{Q}'_{\theta f}(k+1) \begin{bmatrix} \mathbf{f}_N(k) \\ \mathbf{p}(k+1) \end{bmatrix} \quad (24)$$

where

$$\mathbf{p}(k+1) = \mathbf{E}_f^{-T}(k+1)\tilde{\mathbf{e}}_f(k+1) \quad (25)$$

with $\tilde{\mathbf{e}}_f(k+1)$ –the first line of the multichannel forward error, transposed– being the *a posteriori* forward error vector. Also from [5], we have the following expression, similar to its single dimension counterpart, to update $\mathbf{Q}_\theta(k)$.

$$\mathbf{Q}_\theta(k+1) \begin{bmatrix} 1 \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \gamma(k+1) \\ \mathbf{f}_N(k+1) \end{bmatrix} \quad (26)$$

The expression in (25) requires a matrix inversion operation which can be numerically unstable, leading to stability problems. For calculating $\mathbf{p}(k+1)$ in a simpler manner, it is easily shown from (11) that the following equation can be used instead of (25):

$$\overline{\mathbf{Q}}_f(k+1) \begin{bmatrix} \gamma(k) \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} * \\ \mathbf{p}(k+1) \end{bmatrix} \quad (27)$$

Proof. From (11), it is clear that $\mathbf{E}_f(k+1)$ is the Cholesky factor of $[\tilde{\mathbf{e}}_{fq1} \quad \lambda^{1/2}\mathbf{E}_f^T(k)]^T$ [6]. Consequently, we can write

$$\begin{aligned} \mathbf{E}_f^T(k+1)\mathbf{E}_f(k+1) &= \begin{bmatrix} \tilde{\mathbf{e}}_{fq1}^T(k+1) \\ \lambda^{1/2}\mathbf{E}_f^T(k) \end{bmatrix}^T \begin{bmatrix} \tilde{\mathbf{e}}_{fq1}^T(k+1) \\ \lambda^{1/2}\mathbf{E}_f(k) \end{bmatrix} \\ &= \tilde{\mathbf{e}}_{fq1}(k+1)\tilde{\mathbf{e}}_{fq1}^T(k+1) + \lambda\mathbf{E}_f^T(k)\mathbf{E}_f(k) \end{aligned} \quad (28)$$

The above equation is the product form of (11). Now, premultiplying and post multiplying (28) by $\mathbf{E}_f^{-T}(k+1)\gamma^2(k)$ and $\mathbf{E}_f^{-1}(k+1)$, respectively, after some algebraic manipulations, yields

$$\gamma^2(k)\mathbf{I} = \mathbf{p}(k+1)\mathbf{p}^T(k+1) + \mathbf{\Psi} \quad (29)$$

where $\mathbf{\Psi} = \lambda\gamma^2(k)\mathbf{E}_f^{-T}(k+1)\mathbf{E}_f^T(k)\mathbf{E}_f(k)\mathbf{E}_f^{-1}(k+1)$.

Finally, premultiplying and post multiplying (29) by $\mathbf{p}^T(k+1)$ and $\mathbf{p}(k+1)$, respectively, it simplifies to

$$\gamma^2(k) = \mathbf{p}^T(k+1)\mathbf{p}(k+1) + \frac{\mathbf{p}^T(k+1)\mathbf{\Psi}\mathbf{p}(k+1)}{\mathbf{p}^T(k+1)\mathbf{p}(k+1)} = \mathbf{p}^T(k+1)\mathbf{p}(k+1) + *^2 \quad (30)$$

With farther manipulation, it can be shown that the quantity represented by the *asterisk* is known prior to the computation of $\mathbf{p}(k+1)$; this knowledge, however, is useless for that purpose because it suffices to know $\gamma(k)$ and $\overline{\mathbf{Q}}_f(k+1)$ to obtain $\mathbf{p}(k+1)$ in a simple manner.

The expression in (30) is clearly a Cholesky product. Hence, there must exist an orthogonal matrix \mathbf{Q} such that

$$\begin{bmatrix} \gamma(k) \\ \mathbf{0} \end{bmatrix} = \mathbf{Q} \begin{bmatrix} * \\ \mathbf{p}(k+1) \end{bmatrix} \quad (31)$$

Now, recalling our starting point in (11), we figure out that \mathbf{Q} is related to $\overline{\mathbf{Q}}_f(k+1)$. Moreover, from the knowledge of the internal structure of $\overline{\mathbf{Q}}_f(k+1)$, we finally realize that $\mathbf{Q} = \overline{\mathbf{Q}}_f^T(k+1)$ satisfies (31) leading to (27), which concludes the proof. \square

4 The New LATTICE Multichannel Fast QRD_POS_B algorithm

Because of the blocking nature of the input vector used to derive the equations of the algorithm presented in the previous section, the quantities $\mathbf{D}_{fq2}(k)$, $\mathbf{d}_{q2}(k)$, and $\mathbf{f}_N(k)$ can be split up into N blocks from top to bottom. For the matrix $\mathbf{D}_{fq2}(k)$ we have

$$\mathbf{D}_{fq2}(k) = \begin{bmatrix} \mathbf{D}_{fq2}^{(1)}(k) \\ \vdots \\ \mathbf{D}_{fq2}^{(N)}(k) \end{bmatrix} \quad (32)$$

where $\mathbf{D}_{fq2}^{(i)}(k)$ has dimensions $M \times M$. In light of this assumption, (9) can be rewritten as

$$\begin{bmatrix} \mathbf{0}_{M(N-i-1) \times M} \\ \mathbf{0}_{M(i-1) \times M} \\ \mathbf{E}_f^{(i-1)}(k+1) \end{bmatrix} = \mathbf{Q}'_{\theta f}{}^{(N-i+1)}(k+1) \begin{bmatrix} \mathbf{0}_{M(N-i) \times M} \\ \mathbf{D}_{fq2}^{(N-i+1)}(k) \\ \mathbf{0}_{M(i-1) \times M} \\ \mathbf{E}_f^{(i)}(k+1) \end{bmatrix} \quad (33)$$

Table 1. The MCFQRD_POS_B Equations.

<p><i>Initializations:</i> $\mathbf{f}_N(0) = 0; \quad \mathbf{D}_{fq2}(0) = 0$ $\mathbf{d}_{q2}(0) = 0; \quad \mathbf{E}_f(0) = \mathbf{I}$ All <i>cosines</i> = 1, and all <i>sines</i> = 0; For each k, do { 1. Obtaining $\mathbf{D}_{fq2}(k+1)$ and $\tilde{\mathbf{e}}_{fq1}(k+1)$ $\begin{bmatrix} \tilde{\mathbf{e}}_{fq1}^T(k+1) \\ \mathbf{D}_{fq2}(k+1) \end{bmatrix} = \mathbf{Q}_\theta(k) \begin{bmatrix} \mathbf{x}_{k+1}^T \\ \lambda^{1/2} \mathbf{D}_{fq2}(k) \end{bmatrix}$ 2. Obtaining $\mathbf{E}_f(k+1)$ $\begin{bmatrix} \mathbf{0}^T \\ \mathbf{E}_f(k+1) \end{bmatrix} = \overline{\mathbf{Q}}_f(k+1) \begin{bmatrix} \tilde{\mathbf{e}}_{fq1}^T(k+1) \\ \lambda^{1/2} \mathbf{E}_f(k) \end{bmatrix}$ 3. Obtaining $\mathbf{p}(k+1)$ $\begin{bmatrix} * \\ \mathbf{p}(k+1) \end{bmatrix} = \overline{\mathbf{Q}}_f(k+1) \begin{bmatrix} \gamma(k) \\ \mathbf{0} \end{bmatrix}$ 4. Obtaining $\mathbf{Q}'_{\theta f}(k+1)$ $\begin{bmatrix} \mathbf{0} \\ \mathbf{E}_f^0(k+1) \end{bmatrix} = \mathbf{Q}'_{\theta f}(k+1) \begin{bmatrix} \mathbf{D}_{fq2}(k+1) \\ \mathbf{E}_f(k+1) \end{bmatrix}$ 5. Obtaining $\mathbf{f}_N(k+1)$ $\mathbf{f}_{N+1}(k+1) = \mathbf{Q}'_{\theta f}(k+1) \begin{bmatrix} \mathbf{f}_N(k) \\ \mathbf{p}(k+1) \end{bmatrix}$ 6. Obtaining $\mathbf{Q}_\theta(k+1)$ and $\gamma(k+1)$ $\mathbf{Q}_\theta(k+1) \begin{bmatrix} 1 \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \gamma(k+1) \\ \mathbf{f}_N(k+1) \end{bmatrix}$ 7. Joint Estimation $\begin{bmatrix} \mathbf{e}_{q1}(k+1) \\ \mathbf{d}_{q2}(k+1) \end{bmatrix} = \mathbf{Q}_\theta(k+1) \begin{bmatrix} d(k+1) \\ \lambda^{1/2} \mathbf{d}_{q2}(k) \end{bmatrix}$ 8. Obtaining the <i>a priori</i> error $e'(k+1) = e_{q1}(k+1)/\gamma(k+1)$ } }</p>
--

for $i = N, N-1, \dots, 1$, which means a backward execution. From the previous equation, it is easy to see that

$$\mathbf{Q}'_{\theta f}(k+1) = \mathbf{Q}'_{\theta f}^{(N)}(k+1) \mathbf{Q}'_{\theta f}^{(N-1)}(k+1) \cdots \mathbf{Q}'_{\theta f}^{(1)}(k+1).$$

Nevertheless, (33) suggests that it can also be performed in a forward manner, that is, for $i = 1, 2, \dots, N$. This property is the key to derive the lattice version of the algorithm. Now, recalling that $\mathbf{Q}'_{\theta f}(k)$ is used to update $\mathbf{f}_N(k)$, we can rewrite (24) as

$$\begin{bmatrix} \mathbf{0}_{M(N-i) \times M} \\ \mathbf{f}^{(N-i+1)}(k+1) \\ \mathbf{0}_{M(i-1) \times M} \\ \mathbf{p}_{i-1}(k+1) \end{bmatrix} = \mathbf{Q}'_{\theta f}^{(N-i+1)}(k+1) \begin{bmatrix} \mathbf{0}_{M(N-i) \times M} \\ \mathbf{f}^{(N-i+2)}(k) \\ \mathbf{0}_{M(i-1) \times M} \\ \mathbf{p}_i(k+1) \end{bmatrix} \quad (34)$$

for $i = 1, 2, \dots, N$. Note that we are taking into account the forward option for both (33) and (34).

From the last two equations, we realize that steps 4 and 5 of the algorithm in Table 1 can now be carried out in a forward manner. The rotation angles $\mathbf{Q}_\theta^{(i)}(k+1)$ are obtained through

$$\mathbf{Q}_\theta^{(i)}(k+1) \begin{bmatrix} \gamma_{i-1}(k+1) \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \gamma_i(k+1) \\ \mathbf{f}^{(N-i+2)}(k+1) \end{bmatrix} \quad (35)$$

and the joint estimation is performed according to

$$\begin{bmatrix} \mathbf{e}_{q_1}^{(i)}(k+1) \\ \mathbf{d}_{q_2}^{(N-i+1)}(k+1) \end{bmatrix} = \mathbf{Q}_\theta^{(N-i+1)}(k+1) \begin{bmatrix} \mathbf{e}_{q_1}^{(i)}(k+1) \\ \lambda^{1/2} \mathbf{d}_{q_2}^{(N-i+1)}(k) \end{bmatrix} \quad (36)$$

In order to adequate the equations of steps 1 to 3 of the algorithm as in Table 1 to this formulation, it suffices to observe that they can be easily split up into $M \times M$ blocks that will be executed recursively as shown in Table 2. It is worth mentioning that, for the sake of simplification due to space constraints, we have used matrix notation in the single loop operations as shown in Table 2. However, when implementing these equations, it is straightforward to reduce the simple Givens rotations matrices into scalar operations.

5 Simulations Results

In this section we perform an evaluation of the Multichannel Fast QRD-RLS in an adaptive beamforming scenario. Although this kind of application requires the use of constrained algorithms, structures like the *Generalized Sidelobe Canceller* (GSC) [7], used here, or the Householder structure [8], make possible the use of unconstrained algorithms to solve constrained problems.

In our adaptive beamforming experiment, we have used a linear array of 7 sensors with a look-direction set to 0° and three jammers with incident angles corresponding to -25° , 45° , and 50° . The signal-to-noise ratio (SNR) was set to $0dB$ and a jammer-to-noise ratio (JNR) of $30dB$ was used. The forgetting factor (λ) was set to 0.98.

The MSE converging paths (identical) are presented in Figure 2 for both MCFQR_POS_B (introduced here in its lattice version) and the MCFQR_PRI_B of [3]. Both algorithms are of $O(NM^3)$ computational complexity. Nevertheless our proposed algorithm saves $2NM$ multiplications and $2NM$ divisions in steps 3 and 6 when compared with its equivalent counterpart of [3]. After 10 independent runs of a considerably large number of samples (6×10^6), we have observed no sign of divergence, as expected for algorithms of the QRD-LS family.

6 Conclusions

In this paper we have introduced the Lattice version of the Multichannel Fast QRD-RLS algorithm based on the *a posteriori* backward error updating. Its order recursiveness and stability are very attractive features and it can be used in a wide range of applications, many of them in the field of telecommunications.

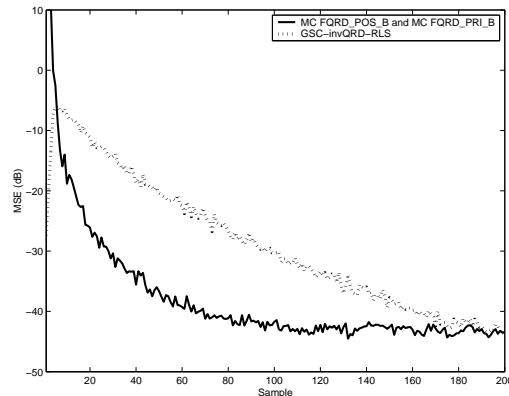


Fig. 2. Convergence in a beamforming scenario using GSC.

Although the new algorithm introduced here presents the same converge properties as the one of [3], it is worth mentioning that it saves computational load that makes it particularly attractive as N and M increase.

References

1. N. Kalouptsidis and S. Theodoridis, *Adaptive Systems Identification and Signal Processing Algorithms*, Englewood Cliffs, NJ: Prentice Hall, 1993.
2. J. A. Apolinário Jr., M. G. Siqueira, and P. S. R. Diniz, "Fast QR Algorithms Based on Backward Prediction Errors: A New Implementation and Its Finite Precision Performance," *Birkhäuser Circuits, Systems, and Signal Processing*, vol. 22, no. 4, pp. 335–349, July/August 2003.
3. A. A. Rontogiannis and S. Theodoridis, "Multichannel fast QRD-LS adaptive filtering: New technique and algorithms," *IEEE Transactions on Signal Processing*, vol. 46, pp. 2862–2876, November 1998.
4. C. A. Medina S., J. A. Apolinário Jr., and M. G. Siqueira, "A unified framework for multichannel fast QRD-LS adaptive filters based on backward prediction errors," *IEEE Midwest Symposium on Circuits and Systems*, vol. 3, pp. 668–671, USA, August 2002.
5. J. A. Apolinário Jr., *New algorithms of adaptive filtering: LMS with data-reusing and fast RLS based on QR decomposition*, D.Sc. Thesis, COPPE/Federal University of Rio de Janeiro, Rio de Janeiro, Brazil, 1998.
6. G. H. Golub and C. F. Van Loan, *Matrix Computations*, Baltimore: The Johns Hopkins University Press, 1983.
7. L. J. Griffiths and C. W. Jim, "An alternative approach to linearly constrained adaptive beamforming," *IEEE Transactions on Antennas and Propagation*, vol. AP-30, pp. 27–34, January 1982.
8. M. L. R. de Campos, S. Werner, J. A. Apolinário Jr., and T. I. Laakso, "Constrained adaptation algorithms employing Housholder transformation," *IEEE Transactions on Signal Processing*, vol. 50, no. 9, pp. 2187–2195, September 2002.

Table 2. The Lattice MCFQRD_POS_B Equations.

<p><i>Initializations:</i></p> <p>$\mathbf{f}_N(0) = 0;$ $\mathbf{D}_{fq2}(0) = 0;$ $\gamma_0(0) = 1;$ $\mathbf{d}_{q2}(0) = 0;$ $\mathbf{E}_f^i(0) = \mu \mathbf{I},$ $\mu = \text{small number}$ All <i>cosines</i> = 1, and all <i>sines</i> = 0; For each k, do { $\tilde{\mathbf{e}}_{fq1}^{(0)T}(k+1) = \mathbf{x}_{k+1}^T$ A. Obtaining $\mathbf{E}_f^{(0)}(k+1)$ and $\overline{\mathbf{Q}}_f^{(0)}(k+1)$ $\begin{bmatrix} \mathbf{0}^T \\ \mathbf{E}_f^{(0)}(k+1) \end{bmatrix} = \overline{\mathbf{Q}}_f^{(0)}(k+1) \begin{bmatrix} \tilde{\mathbf{e}}_{fq1}^{(0)T}(k+1) \\ \lambda^{1/2} \mathbf{E}_f^{(0)}(k) \end{bmatrix}$ B. Obtaining $\mathbf{p}_0(k+1)$ $\begin{bmatrix} * \\ \mathbf{p}_0(k+1) \end{bmatrix} = \overline{\mathbf{Q}}_f^{(0)}(k+1) \begin{bmatrix} \gamma_0(k) \\ \mathbf{0} \end{bmatrix}$ $\mathbf{f}^{(N+1)}(k+1) = \mathbf{p}_0(k+1);$ $\gamma_0(k+1) = 1;$ $e_{q1}(k+1) = d(k+1);$ for $i = 1 : N$ { 1. Obtaining $\mathbf{D}_{fq2}^{(N-i+1)}(k+1)$ and $\mathbf{e}_{fq1}^{(i)}(k+1)$ $\begin{bmatrix} \tilde{\mathbf{e}}_{fq1}^{(i)T}(k+1) \\ \mathbf{D}_{fq2}^{(N-i+1)}(k+1) \end{bmatrix} = \mathbf{Q}_{\theta}^{(i)}(k) \begin{bmatrix} \tilde{\mathbf{e}}_{fq1}^{(i-1)T}(k+1) \\ \lambda^{1/2} \mathbf{D}_{fq2}^{(N-i+1)}(k) \end{bmatrix}$ 2. Obtaining $\mathbf{E}_f^{(i)}(k+1)$ $\begin{bmatrix} \mathbf{0}^T \\ \mathbf{E}_f^{(i)}(k+1) \end{bmatrix} = \overline{\mathbf{Q}}_f^{(i)}(k+1) \begin{bmatrix} \tilde{\mathbf{e}}_{fq1}^{(i)T}(k+1) \\ \lambda^{1/2} \mathbf{E}_f^{(i)}(k) \end{bmatrix}$ 3. Obtaining $\mathbf{p}_i(k+1)$ $\begin{bmatrix} * \\ \mathbf{p}_i(k+1) \end{bmatrix} = \overline{\mathbf{Q}}_f^{(i)}(k+1) \begin{bmatrix} \gamma_i(k) \\ \mathbf{0} \end{bmatrix}$ 4. Obtaining $\mathbf{Q}'_{\theta f}{}^{(N-i+1)}(k+1)$ $\begin{bmatrix} \mathbf{0}_{M(N-i-1) \times M} \\ \mathbf{0}_{M(i-1) \times M} \\ \mathbf{E}_f^{(i-1)}(k+1) \end{bmatrix} = \mathbf{Q}'_{\theta f}{}^{(N-i+1)}(k+1) \begin{bmatrix} \mathbf{0}_{M(N-i) \times M} \\ \mathbf{D}_{fq2}^{(N-i+1)}(k) \\ \mathbf{0}_{M(i-1) \times M} \\ \mathbf{E}_f^{(i)}(k+1) \end{bmatrix}$ 5. Obtaining $\mathbf{f}^{(N-i+1)}(k+1)$ $\begin{bmatrix} \mathbf{0}_{M(N-i) \times M} \\ \mathbf{f}^{(N-i+1)}(k+1) \\ \mathbf{0}_{M(i-1) \times M} \\ \mathbf{p}_{i-1}(k+1) \end{bmatrix} = \mathbf{Q}'_{\theta f}{}^{(N-i+1)}(k+1) \begin{bmatrix} \mathbf{0}_{M(N-i) \times M} \\ \mathbf{f}^{(N-i+2)}(k) \\ \mathbf{0}_{M(i-1) \times M} \\ \mathbf{p}_i(k+1) \end{bmatrix}$ 6. Obtaining $\mathbf{Q}_{\theta}^{(i)}(k+1)$ and $\gamma_i(k+1)$ $\mathbf{Q}_{\theta}^{(i)}(k+1) \begin{bmatrix} \gamma_{i-1}(k+1) \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \gamma_i(k+1) \\ \mathbf{f}^{(N-i+2)}(k+1) \end{bmatrix}$ 7. Joint Estimation $\begin{bmatrix} \mathbf{e}_{q1}^{(i)}(k+1) \\ \mathbf{d}_{q2}^{(N-i+1)}(k+1) \end{bmatrix} = \mathbf{Q}_{\theta}^{(i)}(k+1) \begin{bmatrix} \mathbf{e}_{q1}^{(i-1)}(k+1) \\ \lambda^{1/2} \mathbf{d}_{q2}^{(N-i+1)}(k) \end{bmatrix}$ } 8. Obtaining the <i>a priori</i> error $e'(k+1) = e_{q1}(k+1)/\gamma(k+1)$ } }</p>
