MULTICHANNEL FAST QR-DECOMPOSITION RLS ALGORITHMS WITH EXPLICIT WEIGHT EXTRACTION

Mobien Shoaib,[1] *Stefan Werner*,[1] *José A. Apolinário Jr.*,[2] *and Timo I. Laakso*[1]

[1]**Helsinki University of Technology**
Signal Processing Laboratory Espoo, Finland
email:{mobien.shoaib, stefan.werner, timo.laakso}@tkk.fi

[2]**Instituto Militar de Engenharia**
Depto. Engenharia Elétrica
Rio de Janeiro, Brazil
apolin@ieee.org

ABSTRACT

Multichannel fast QR decomposition recursive least-squares (MC-FQRD-RLS) algorithms are well known for their good numerical properties and low computational complexity. However, these algorithms have been restricted to problems seeking an estimate of the output error signal. This is because their transversal weights are embedded in the algorithm variables and are not explicitly available. In this paper we present a novel technique that can extract the filter weights associated with the MC-FQRD-RLS algorithm at any time instant. As a consequence, the range of applications is extended to include problems where explicit knowledge of the filter weights is required. The proposed weight extraction technique is used to identify the beampattern of a broadband adaptive beamformer implemented with an MC-FQRD-RLS algorithm. The results confirm that the extracted coefficients of the MC-FQRD-RLS algorithm are identical to those obtained by any RLS algorithm such as the inverse QRD-RLS algorithm.

1. INTRODUCTION

Multichannel adaptive signal processing can be found in various applications such as broadband beamforming, equalization, stereophonic echo cancellation, and speech enhancement [1].

When considering multichannel adaptive implementations, it is often possible to directly apply standard single-channel algorithms to the multichannel problem, e.g., the numerically stable and fast converging QR decomposition RLS algorithm (QRD-RLS). Even though such a solution would provide fast convergence, it may be computationally too complex due to a large number of coefficients. In order to obtain a computationally efficient solution, RLS-type algorithms specially tailored for the multichannel setup are a good option. Two types of multichannel algorithms have been proposed in the literature: 1) *Block-type algorithms* where the channels are processed simultaneously, and; 2) *Sequential algorithms* that process each channel individually [1].

In this paper, we focus on block-type *multichannel fast QR decomposition RLS* (MC-FQRD-RLS) algorithms, which enable parallel implementation due to the joint processing of channels. The MC-FQRD-RLS algorithms exhibit RLS like convergence and numerical robustness at a lower complexity than the single-channel QRD-RLS. The main idea of MC-FQRD-RLS algorithms is to exploit the underlying time-shift structure of the input-signal vector of each channel in order to replace matrix update equations with vector update equations. By doing so, the computational complexity can be reduced from $\mathcal{O}(P^2)$ of the standard QRD-RLS implementation to $\mathcal{O}(M^2P)$ of the block MC-FQRD-RLS algorithms, where P is the total number of filter coefficients and M is the number of channels.

The main disadvantage of MC-FQRD-RLS algorithms is the fact that the weight vector associated with the underlying weighted least-squares problem is embedded in the internal algorithm variables. Furthermore, they do not directly provide the variables allowing for a straightforward computation of the weight vector, as is the case with the conventional QRD-RLS algorithm, where a back-substitution procedure can be used to compute the coefficients. Therefore, the applications are limited to output error based settings (e.g., noise or echo cancellation), or to those requiring a decision-feedback estimate of the training signal (e.g., adaptive beamformer operating in decision-directed mode). The absence of weights in MC-FQRD-RLS algorithms makes the problem of system identification non-trivial. For example, the beampattern (spatial response) is not available in an adaptive beamformer implementation using an MC-FQRD-RLS algorithm.

This paper addresses the problem of identifying the weight vector from the internal variables of the block MC-FQRD-RLS algorithm. This *weight extraction* problem was solved for the single-channel FQRD-RLS algorithms in [2]. The main results, summarized by two lemmas, provide us with an algorithm that allows us at any time instant during adaptation to sequentially extract the columns of the Cholesky factor embedded in the MC-FQRD-RLS algorithm. From the Cholesky factor we can obtain the true weights of the underlying LS solution by reusing the known MC-FQRD-RLS variables. We emphasize that the proposed method relies on the knowledge of only vector updates present in the MC-FQRD-RLS algorithms, as opposed to the matrix-embedded structure of the conventional QRD-RLS described in [3]. The problem of parameter identification has been addressed in [4] using the duality between the single channel FQRD-RLS algorithm in [4, 5] to a normalized lattice structure. The relation between the results of this paper that are related to the identification of transversal filter weights and those of the multichannel extension of the lattice parameter identification in [4] is currently under investigation.

In the following we present the basic principles of the block MC-FQRD-RLS algorithm. Thereafter, the weight extraction (WE) algorithm is derived. Simulation results are followed by conclusions.

2. THE MULTICHANNEL FAST QR-DECOMPOSITION ALGORITHM

This section presents the basic concepts of MC-QRD-RLS algorithms. Two versions of the MC-FQRD-RLS algorithms are reviewed to aid the explanation of the weight extraction technique.

2.1 Basic concepts of QR decomposition algorithms

Consider the multichannel adaptive filter setup in Fig. 1 with M channels and N filter coefficients per channel, i.e., a total of $P = MN$ coefficients. The MC-QRD-RLS algorithm minimizes the following cost function with respect to $\mathbf{w}_P(k)$

$$\xi(k) = \sum_{i=0}^{k} \lambda^{k-i} |d^*(i) - \mathbf{x}_P^{\mathrm{H}}(i)\mathbf{w}_P(k)|^2 = \|\mathbf{e}^*(k)\|^2 \qquad (1)$$

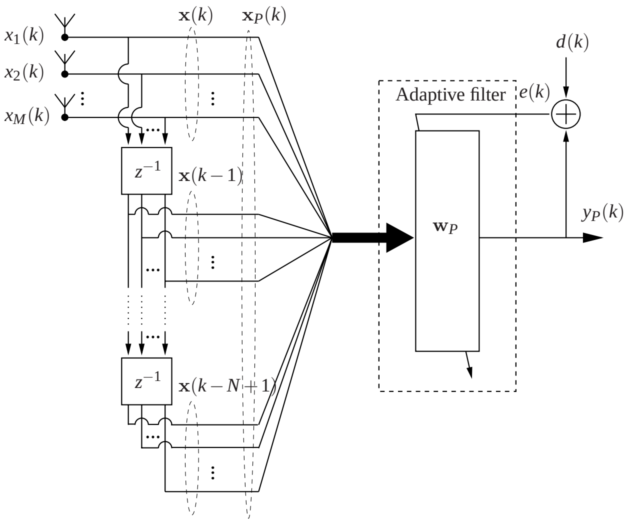

Figure 1: Multichannel adaptive filter setup.

where λ is the forgetting factor, $*$ denotes the conjugate, and $\mathbf{e}(k) \in \mathbb{C}^{(k+1)\times 1}$ is the *a posteriori* error vector given as

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

where $\mathbf{d}(k) \in \mathbb{C}^{(k+1)\times 1}$ is the desired signal vector, $\mathbf{w}_P(k) \in \mathbb{C}^{P\times 1}$ is the multichannel coefficient vector, $\mathbf{x}_P(k) \in \mathbb{C}^{P\times 1}$ is the multichannel input vector

$$\mathbf{x}_P(k) = \begin{bmatrix} \mathbf{x}^{\mathrm{T}}(k) & \mathbf{x}^{\mathrm{T}}(k-1) & \dots & \mathbf{x}^{\mathrm{T}}(k-N+1) \end{bmatrix}^{\mathrm{T}} \quad (3)$$

and

$$\mathbf{x}(k) = \begin{bmatrix} x_1(k) & x_2(k) & \dots & x_M(k) \end{bmatrix}^{\mathrm{T}} \quad (4)$$

is the $M \times 1$ input vector. The QRD-RLS algorithm uses an orthogonal rotation matrix $\mathbf{Q}_N(k) \in \mathbb{C}^{(k+1)\times(k+1)}$ to triangularize matrix $\mathbf{X}_P(k)$ as [6]

$$\begin{bmatrix} \mathbf{0}_{(k+1-P)\times P} \\ \mathbf{U}_P(k) \end{bmatrix} = \mathbf{Q}_P(k)\mathbf{X}_P(k) \quad (5)$$

where $\mathbf{U}_P(k) \in \mathbb{C}^{P\times P}$ is the Cholesky factor of the deterministic autocorrelation matrix $\mathbf{R}_P(k) = \mathbf{X}_P^{\mathrm{H}}(k)\mathbf{X}_P(k)$.

Pre-multiplying (2) with $\mathbf{Q}_P(k)$ gives

$$\mathbf{Q}_P(k)\mathbf{e}^*(k) = \begin{bmatrix} \mathbf{e}_{q1}(k) \\ \mathbf{e}_{q2}(k) \end{bmatrix} = \begin{bmatrix} \mathbf{d}_{q1}(k) \\ \mathbf{d}_{q2}(k) \end{bmatrix} - \begin{bmatrix} \mathbf{0}_{(k+1-P)\times P} \\ \mathbf{U}_P(k) \end{bmatrix} \mathbf{w}_P(k) \quad (6)$$

We emphasize that $\mathbf{d}_{q1}(k)$ and $\mathbf{d}_{q2}(k)$ are partitions of vector $\mathbf{d}^*(k)$ after rotation, similarly $\mathbf{e}_{q1}(k)$ and $\mathbf{e}_{q2}(k)$ are partitions of vector $\mathbf{e}^*(k)$ after rotation. The cost function in (1) is minimized by choosing $\mathbf{w}_P(k)$ such that $\mathbf{d}_{q2}(k) - \mathbf{U}_P(k)\mathbf{w}_P(k)$ is zero, i.e.,

$$\mathbf{w}_P(k) = \mathbf{U}_P^{-1}(k)\mathbf{d}_{q2}(k) \quad (7)$$

The QRD-RLS algorithm updates vector $\mathbf{d}_{q2}(k)$ and matrix $\mathbf{U}_P(k)$ as follows [6]

$$\begin{bmatrix} e_{q1}(k) \\ \mathbf{d}_{q2}(k) \end{bmatrix} = \mathbf{Q}_\theta(k) \begin{bmatrix} d^*(k) \\ \lambda^{1/2}\mathbf{d}_{q2}(k-1) \end{bmatrix} \quad (8)$$

$$\begin{bmatrix} \mathbf{0}_{1\times P} \\ \mathbf{U}_P(k) \end{bmatrix} = \mathbf{Q}_\theta(k) \begin{bmatrix} \mathbf{x}_P^{\mathrm{H}}(k) \\ \lambda^{1/2}\mathbf{U}_P(k-1) \end{bmatrix} \quad (9)$$

where $\mathbf{Q}_\theta(k) \in \mathbb{C}^{(P+1)\times(P+1)}$ is a sequence of Givens rotation matrices which annihilates the input vector $\mathbf{x}(k)$ in (9) and can be partitioned as [7]

$$\mathbf{Q}_\theta(k) = \begin{bmatrix} \gamma(k) & \mathbf{g}_P^{\mathrm{H}}(k) \\ \mathbf{f}_P(k) & \mathbf{E}_P(k) \end{bmatrix} \quad (10)$$

The QRD-RLS algorithm is complete with the definition of the *a priori* error value $e(k) = e_{q1}^*(k)/\gamma(k)$ where $\gamma(k)$ is a scalar element in matrix $\mathbf{Q}_\theta(k)$, see (10).

2.2 Block MC-FQRD-RLS algorithm based on backward prediction error update

MC-FQRD-RLS algorithms update either the *a priori* or the *a posteriori* backward prediction error vector [9]. Therefore we have two versions of the backward prediction error update based MC-FQRD-RLS algorithm, referred to as MC-FQR_PRI_B and MC-FQR_POS_B. The MC-FQR_PRI_B algorithm updates vector $\mathbf{a}_P(k)$ defined as

$$\mathbf{a}_P(k) = \lambda^{-1/2}\mathbf{U}_P^{-\mathrm{H}}(k-1)\mathbf{x}_P(k) = -\mathbf{g}_P(k)/\gamma(k) \quad (11)$$

while the MC-FQR_POS_B algorithm updates vector $\mathbf{f}_P(k)$ given by

$$\mathbf{f}_P(k) = \mathbf{U}_P^{-\mathrm{H}}(k)\mathbf{x}_P(k) \quad (12)$$

The basic idea of the MC-FQRD based algorithms is to replace the update for matrix $\mathbf{U}_P^{-\mathrm{H}}(k)$ in (9) with an update equation for a vector, i.e., either $\mathbf{a}_P(k)$ or $\mathbf{f}_P(k)$. The extended Cholesky matrix $\mathbf{U}_{P+1}(k) \in \mathbb{C}^{(P+M)\times(P+M)}$ is defined as the Cholesky factor of the extended input matrix $\mathbf{X}_{P+1}(k) \in \mathbb{C}^{(k+1)\times(P+M)}$ as

$$\begin{bmatrix} \mathbf{0}_{(k+1-P-M)\times(P+M)} \\ \mathbf{U}_{P+1}(k-1) \end{bmatrix} = \mathbf{Q}_{P+1}(k-1)\mathbf{X}_{P+1}(k-1) \quad (13)$$

where $\mathbf{X}_{P+1}(k)$ is constructed by appending a column to the right of $\mathbf{X}_P(k-1)$ consisting of corresponding past input values. Note that the forward and backward prediction equations can be specified using $\mathbf{X}_{P+1}(k-1)$. Therefore, the triangularization of $\mathbf{X}_{P+1}(k-1)$ can be approached from either a forward or a backward prediction perspective. The extended Cholesky matrix is therefore written as [8]

$$\mathbf{U}_{P+1}^{-\mathrm{H}}(k-1) = \mathbf{Q}'_{\theta f}(k-1)\times \begin{bmatrix} \mathbf{0}_{P\times M} & \mathbf{U}_P^{-\mathrm{H}}(k-2) \\ [\mathbf{E}'_f(k-1)]^{-\mathrm{H}} & -[\mathbf{E}'_f(k-1)]^{-\mathrm{H}}\mathbf{D}_{fq2}^{\mathrm{H}}(k-1)\mathbf{U}_P^{-\mathrm{H}}(k-2) \end{bmatrix} \quad (14)$$

where $\mathbf{E}'_f(k)$ and $\mathbf{D}_{fq2}(k)$ are the rotated forward prediction error and desired signal matrices, respectively. By post-multiplying (14) with the extended multichannel input data vector $\mathbf{x}_{P+1}(k-1)$ and $\lambda^{-1/2}$ we obtain the update equation for vector $\mathbf{a}_P(k)$

$$\mathbf{a}_{P+1}(k) = \lambda^{-1/2}\mathbf{Q}'_{\theta f}(k-1)\begin{bmatrix} \mathbf{a}_P(k-1) \\ \mathbf{r}(k) \end{bmatrix} \quad (15)$$

where $\mathbf{r}(k) = \lambda^{-1/2}[\mathbf{E}'_f(k-1)]^{-\mathrm{H}}\tilde{\mathbf{e}}_f(k)$ and $\tilde{\mathbf{e}}_f(k) = \gamma(k-1)\tilde{\mathbf{e}}_{fq1}(k)$. If we instead post-multiply (14), evaluated at time instant k, with the vector $\mathbf{x}_{P+1}(k-1)$, we get the update equation for $\mathbf{f}_P(k)$

$$\mathbf{f}_{P+1}(k) = \mathbf{Q}'_{\theta f}(k)\begin{bmatrix} \mathbf{f}_P(k-1) \\ \mathbf{p}(k) \end{bmatrix} \quad (16)$$

where $\mathbf{p}(k) = [\mathbf{E}'_f(k)]^{-\mathrm{H}}\tilde{\mathbf{e}}_f(k)$ and $\tilde{\mathbf{e}}_f(k) = \gamma(k-1)\tilde{\mathbf{e}}_{fq1}(k)$.

Table 1: The FQR_PRI_B algorithm based on backward prediction errors: $P = MN$ is the total number of coefficients, M is the number of channels, and N is the number of coefficients per channel.

for each k

{ Obtaining $\mathbf{D}_{fq2}(k)$:

$$\begin{bmatrix} \tilde{\mathbf{e}}^{\mathrm{H}}_{fq1}(k) \\ \mathbf{D}_{fq2}(k) \end{bmatrix} = \mathbf{Q}_\theta(k-1) \begin{bmatrix} \mathbf{x}^{\mathrm{H}}(k) \\ \lambda^{1/2}\mathbf{D}_{fq2}(k-1) \end{bmatrix}$$

Obtaining $\mathbf{E}'_f(k)$:

$$\begin{bmatrix} \mathbf{0}_{1\times M} \\ \mathbf{E}'_f(k) \end{bmatrix} = \bar{\mathbf{Q}}_f(k) \begin{bmatrix} \tilde{\mathbf{e}}^{\mathrm{H}}_{fq1}(k) \\ \lambda^{1/2}\mathbf{E}'_f(k-1) \end{bmatrix}$$

Obtaining $\mathbf{r}(k)$:

$$\begin{bmatrix} \xi \\ \mathbf{0}_{M\times 1} \end{bmatrix} = \bar{\mathbf{Q}}_f(k) \begin{bmatrix} 1/\gamma(k) \\ -\mathbf{r}(k) \end{bmatrix}$$

Obtaining $\mathbf{a}_P(k)$:

$$\mathbf{a}_{P+1}(k) = \lambda^{-1/2}\tilde{\mathbf{Q}}_{\theta f}(k-1) \begin{bmatrix} \mathbf{a}_P(k-1) \\ \mathbf{r}(k) \end{bmatrix}$$

Obtaining $\tilde{\mathbf{Q}}_{\theta f}(k)$:

$$\begin{bmatrix} \mathbf{0}_{P\times M} \\ \mathbf{E}^{(0)}_f(k) \end{bmatrix} = \tilde{\mathbf{Q}}_{\theta f}(k) \begin{bmatrix} \mathbf{D}_{fq2}(k) \\ \mathbf{E}'_f(k) \end{bmatrix}$$

Obtaining $\mathbf{Q}_\theta(k)$:

$$\begin{bmatrix} 1/\gamma(k) \\ \mathbf{0} \end{bmatrix} = \mathbf{Q}_\theta(k) \begin{bmatrix} 1 \\ -\mathbf{a}_P(k) \end{bmatrix}$$

Joint Process Estimation:

$$\begin{bmatrix} e_{q1}(k) \\ \mathbf{d}_{q2}(k) \end{bmatrix} = \mathbf{Q}_\theta(k) \begin{bmatrix} d^*(k) \\ \lambda^{1/2}\mathbf{d}_{q2}(k-1) \end{bmatrix}$$

$e(k) = e^*_{q1}(k)/\gamma(k)$

}

From the updated vector $\mathbf{a}_P(k)$, the update equation for the rotation matrix $\mathbf{Q}_\theta(k)$ is obtained as

$$\begin{bmatrix} 1/\gamma(k) \\ \mathbf{0}_{P\times 1} \end{bmatrix} = \mathbf{Q}_\theta(k) \begin{bmatrix} 1 \\ -\mathbf{a}_P(k) \end{bmatrix} \tag{17}$$

Similarly, the rotation matrix is also obtained from the updated vector $\mathbf{f}_P(k)$

$$\begin{bmatrix} 1 \\ \mathbf{0}_{P\times 1} \end{bmatrix} = \mathbf{Q}^{\mathrm{T}}_\theta(k) \begin{bmatrix} \gamma(k) \\ \mathbf{f}_P(k) \end{bmatrix} \tag{18}$$

In order to avoid the matrix inversion associated with $\mathbf{r}(k)$ and $\mathbf{p}(k)$ in (15) and (16), respectively, we can use [9]

$$\begin{bmatrix} \xi \\ \mathbf{0}_{M\times 1} \end{bmatrix} = \bar{\mathbf{Q}}_f(k) \begin{bmatrix} 1/\gamma(k) \\ -\mathbf{r}(k) \end{bmatrix} \tag{19}$$

and

$$\begin{bmatrix} \xi \\ \mathbf{p}(k) \end{bmatrix} = \bar{\mathbf{Q}}_f(k) \begin{bmatrix} \gamma(k) \\ \mathbf{0}_{M\times 1} \end{bmatrix} \tag{20}$$

where ξ is an *a priori* unknown variable (dummy variable). The MC-FQR_PRI_B algorithm is summarized in Table 1. For details on the MC-FQR_POS_B implementation, see [9]. As can be seen from Table 1, the weight vector $\mathbf{w}_P(k)$ is not available. The next section presents an algorithm that extracts the weight vector values of both algorithms.

3. THE MULTICHANNEL WEIGHT EXTRACTION

The weight extraction method presented in this section is a multichannel extension of the method given in [2]. The method computes the coefficient values in a serial manner.

Consider the *a priori* output of the multichannel adaptive filter $y_P(k)$ given by

$$y^*_P(k) = \mathbf{w}^{\mathrm{H}}_P(k-1)\mathbf{x}_P(k) = \mathbf{d}^{\mathrm{H}}_{q2}(k-1)\mathbf{U}^{-\mathrm{H}}_P(k-1)\mathbf{x}_P(k) \tag{21}$$

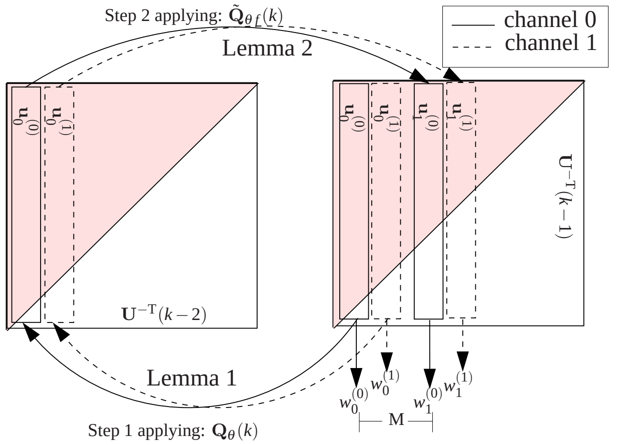

Figure 2: The procedure for updating $\mathbf{u}^n_i(k-1)$ for weight extraction in MCFQRD-RLS *a priori* algorithm. The number of channels $M = 2$. Note that the indices of some variables have been omitted.

Let us define a $P\times 1$ impulse vector δ_i with a "1" at the ith position ($1 \le i \le P$). Note that vector $\mathbf{x}_P(k)$ in (3) comprises input vectors from M channels. As can be seen from (3) and Fig. 1, the elements corresponding to one channel are placed at every Mth position in vector $\mathbf{x}_P(k)$. The jth element of the weight vector for the ith channel is given as

$$\begin{aligned} w^*_{P,i+jM}(k) &= \mathbf{w}^{\mathrm{H}}_P(k-1)\delta_{i+jM} \\ &= \mathbf{d}^{\mathrm{H}}_{q2}(k-1)\mathbf{U}^{-\mathrm{H}}_P(k-1)\delta_{i+jM} \\ &= \mathbf{d}^{\mathrm{H}}_{q2}(k-1)\mathbf{u}_{i+jM}(k-1) \end{aligned} \tag{22}$$

where $\mathbf{u}_{i+jM}(k-1)$ is the $(i+jM)$th column of $\mathbf{U}^{-\mathrm{H}}_P(k-1)$. It can be seen from (22) that the elements of coefficient vector $\mathbf{w}_P(k)$ can be computed if all the column vectors of $\mathbf{U}^{-\mathrm{H}}_P(k-1)$ are known. Using the following two lemmas we show how all the column vectors $\mathbf{u}_i(k-1)$ can be obtained in a serial manner. The main result is that the column vector $\mathbf{u}_i(k-1)$ can be obtained from $\mathbf{u}_{i-1}(k-1)$. Lemma 1 was derived in [2] and is included here for sake of clarity.

Lemma 1. *Let $\mathbf{u}^{\mathrm{T}}_i(k) = [u_{i,0}(k) \quad \ldots \quad u_{i,P-1}(k)]^{\mathrm{T}} \in \mathbb{C}^{P\times 1}$ denote the ith column of the upper triangular matrix $\mathbf{U}^{-\mathrm{H}}_P(k) \in \mathbb{C}^{P\times P}$. Given $\mathbf{Q}_\theta(k-1) \in \mathbb{C}^{(P+1)\times(P+1)}$ from Table 1, then $\mathbf{u}_i(k-2)$ can be obtained from $\mathbf{u}_i(k-1)$ using the relation below*

$$\begin{bmatrix} 0 \\ \lambda^{-1/2}\mathbf{u}_i(k-2) \end{bmatrix} = \mathbf{Q}^{\mathrm{H}}_\theta(k-1) \begin{bmatrix} z_i \\ \mathbf{u}_i(k-1) \end{bmatrix},\ i = 0,\ldots,N-1 \tag{23}$$

where $z_i = -\mathbf{f}^{\mathrm{H}}_N(k-1)\mathbf{u}_i(k-1)/\gamma(k-1)$.

Lemma 2. *Let $\mathbf{u}_i(k) = [u_{i,0}(k) \quad \ldots \quad u_{i,P-1}(k)]^{\mathrm{T}} \in \mathbb{C}^{P\times 1}$ denote the ith column of the upper triangular matrix $\mathbf{U}^{-\mathrm{H}}_P(k-1) \in \mathbb{C}^{P\times P}$. Given $\tilde{\mathbf{Q}}_{\theta f}(k) \in \mathbb{C}^{(P+1)\times(P+1)}$ from Table 1, then $\mathbf{u}_{i+jM}(k-1)$ can be obtained from $\mathbf{u}_{i+(j-1)M}(k-2)$ using the following relation*

$$\begin{bmatrix} \tilde{\mathbf{r}}'(k) \\ \mathbf{u}_{i+jM}(k-1) \end{bmatrix} = \tilde{\mathbf{Q}}_{\theta f}(k-1) \begin{bmatrix} \mathbf{u}_{i+(j-1)M}(k-2) \\ \tilde{\mathbf{r}}_{i+(j-1)M}(k) \end{bmatrix}, \tag{24}$$

where $\tilde{\mathbf{r}}_i(k) = -[\mathbf{E}'_f(k)]^{-\mathrm{H}}\mathbf{D}^{\mathrm{H}}_{fq2}(k)\mathbf{u}_i(k-1)$. Also for $j = 1,\ldots,M$, $\mathbf{u}_{-j}(k-2) = \mathbf{0}_{P\times 1}$ and $\tilde{\mathbf{r}}_{-j}(k) = \mathbf{e}_{f,-j}(k)$, where $\mathbf{e}_{f,-j}(k)$ is the jth column of $-[\mathbf{E}'_f(k)]^{-\mathrm{H}}$.

Assuming vector $\mathbf{u}_{i+(j-1)M}(k-1)$ to be known, Lemmas 1 and 2 can be used first to compute vector $\mathbf{u}_{i+(j-1)M}(k-2)$ and

Table 2: "Weight Extraction" algorithm: M is the number of channels, N is the number of coefficients per channel, $P = MN$ is the total number of coefficients.

$$
\begin{array}{l}
\tilde{\mathbf{r}}_l(k) = \mathbf{e}_{f,l}(k) \text{ for } l = -M, \ldots, -1 \\
\mathbf{u}_l(k-2) = \mathbf{0}_{P\times 1} \text{ for } l = -M, \ldots, -1 \\
\mathbf{E}_I = [\mathbf{E}'_f(k-1)]^{-\mathrm{H}} \\
\text{for each } i = 0 : N-1 \\
\;\text{for each } j = 0 : M-1 \\
\{ \\
\quad \text{Compute } \mathbf{u}_i(k-1) \\
\quad \begin{bmatrix} \tilde{\mathbf{r}}'(k) \\ \mathbf{u}_{i+jM}(k-1) \end{bmatrix} = \tilde{\mathbf{Q}}_{\theta f}(k-1) \begin{bmatrix} \mathbf{u}_{i+(j-1)M}(k-2) \\ \tilde{\mathbf{r}}_{i+(j-1)M}(k) \end{bmatrix} \\
\quad \text{Compute } z_{i+jM}(k) \\
\quad z_{i+jM}(k) = \frac{\mathbf{f}_P^{\mathrm{H}}(k)\mathbf{u}_{i+jM}(k-1)}{\gamma(k)} \\
\quad \text{Compute } \mathbf{u}_i(k-2) \\
\quad \begin{bmatrix} 0 \\ \lambda^{-1/2}\mathbf{u}_{i+jM}(k-2) \end{bmatrix} = \mathbf{Q}_\theta^{\mathrm{H}}(k-1) \begin{bmatrix} z_{i+jM}(k) \\ \mathbf{u}_{i+jM}(k-1) \end{bmatrix} \\
\quad \text{Compute } \tilde{\mathbf{r}}_{i+jM}(k) \\
\quad \tilde{\mathbf{r}}_{i+jM}(k) = -\mathbf{E}_I \mathbf{D}_{fq2}^{\mathrm{H}}(k-1)\mathbf{u}_{i+jM}(k-2) \\
\quad \text{Compute the coefficients} \\
\quad w_{i,j}(k-1) = \mathbf{u}_{i+jM}^{\mathrm{H}}(k-1)\mathbf{d}_{q2}(k-1) \\
\}
\end{array}
$$

then $\mathbf{u}_{i+jM}(k-1)$, respectively. Therefore all the column vectors corresponding to the ith channel are obtained by iterating through all the possible values of j. Consequently, we obtain all the weights for the ith channel. Note that in order to obtain the column vector $\mathbf{u}_{i+jM}(k-1)$ corresponding to a particular channel, we need to initialize (24) given in Lemma 2 properly, which means choosing the appropriate column of matrix $[\mathbf{E}'_f(k-1)]^{-\mathrm{H}}$. A schematic for obtaining the column vectors is given in Figure 2. It is shown that starting from $\mathbf{u}_0(k-1)$, first the column $\mathbf{u}_0(k-2)$ is obtained using Lemma 1 and then $\mathbf{u}_M(k-1)$ using Lemma 2. Both of them correspond to channel 1. These column vectors can be used to compute the weight coefficients $w_0(k-1)$ and $w_M(k-1)$ respectively. The other channels from 2 to M are treated in the same way to compute the corresponding weight coefficients. There are a total of P weight coefficients, so that we need P iterations to compute the whole coefficient vector. The multichannel weight extraction algorithm is summarized in Table 2.

The number of operations required to completely extract all the coefficients is given in Table 3. For comparison, the computational costs of the MC-FQRD-RLS algorithm based on *a priori* backward prediction errors and the inverse QRD-RLS algorithm are given.

4. SIMULATIONS

This section investigates the equivalence of the weights obtained using weight extraction and inverse QRD-RLS algorithm in two applications. First the beampattern identification of a broadband beamformer is considered. The second application is a multichannel system-identification application.

4.1 Broadband beamforming

A uniform linear array with $M = 4$ antenna elements with spacing equal to half wavelength is used in a system with $K = 4$ signals, one being the desired signal with direction of arrival 0^o and the rest are interference signals with direction of arrivals -35^o, 45^o, and 50^o respectively. The number of coefficients per channel is $N = 6$. The SNR for the interfering signals was set to 40 dB and 5 dB for the desired signal. The inverse QRD-RLS and MC-FQRD-RLS algorithms are used for adapting the beamformer. The Weight Extraction algorithm is used to extract the weights of the MC-FQRD-RLS algorithm. The beampatterns for both algorithms after 4000 iterations are shown in Fig. 3. It can be observed that both algorithms give the same solution for the beam pattern. Most importantly the solution using a MC-FQRD-RLS algorithm followed by weight extraction leads to a solution with much lower overall complexity.

4.2 System identification

The multichannel system consists of $M = 3$ channels and $N = 6$ taps per channel. The SNR is 30 dB. The MC-FQRD-RLS algorithm was used to identify the system. After convergence the weight extraction algorithm was run to compute the filter weights. In order to verify how close the weights are to the true ones, an IQRD-RLS algorithm was used to identify the same system. The difference of weights from both the algorithms after 4000 iterations is seen to be approximately -300 dB as shown in Fig. 4. This is within the numerical accuracy of the software used in simulation (MATLAB).

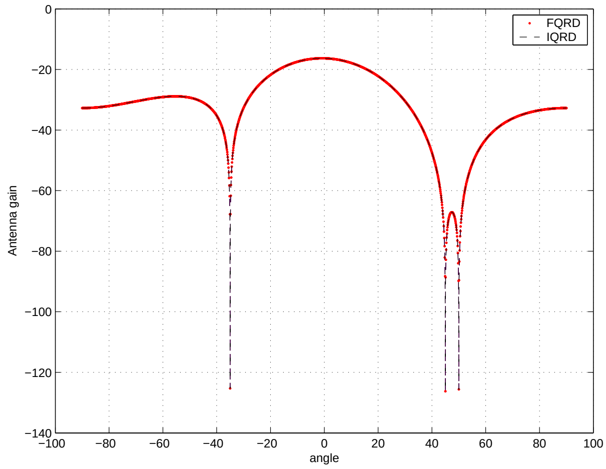

Figure 3: Comparison of beam pattern obtained with the IQRD-RLS algorithm and the MC-FQRD-RLS algorithm with weight extraction.

5. CONCLUSIONS

This paper showed how to reuse the internal variables of the MC-FQRD-RLS algorithms to extract the weights in a serial manner. The presented technique enables new applications of the MC-FQRD-RLS algorithms which are different to the standard output-error type applications. The new weight extraction technique was used in a beamforming and in a system identification setup to make the weight vector explicitly available. The results were compared with those using a design based on the inverse QRD-RLS algorithm. It was verified that identical results are obtained using the proposed design method at a much lower computational cost.

6. APPENDIX

Proof of Lemma 1:
The update equation for $\mathbf{U}_P^{-\mathrm{H}}(k-2)$ in the inverse QRD-RLS algorithm is given by

$$
\begin{bmatrix} \mathbf{z}^{\mathrm{H}}(k-1) \\ \mathbf{U}_P^{-\mathrm{H}}(k-1) \end{bmatrix} = \mathbf{Q}_\theta(k-1) \begin{bmatrix} \mathbf{0}_{1\times P}^{\mathrm{T}} \\ \lambda^{-1/2}\mathbf{U}_P^{-\mathrm{H}}(k-2) \end{bmatrix} \tag{25}
$$

where $\mathbf{z}(k-1) = \gamma^{-1}(k-1)\mathbf{f}_P^{\mathrm{H}}(k-1)\mathbf{U}_P^{-\mathrm{H}}(k-1)$. Pre-multiplying both sides with $\mathbf{Q}_\theta^{\mathrm{T}}(k-1)$ and considering each column we get

$$
\begin{bmatrix} 0 \\ \lambda^{-1/2}\mathbf{u}_i(k-2) \end{bmatrix} = \mathbf{Q}_\theta^{\mathrm{H}}(k-1) \begin{bmatrix} z_i(k-1) \\ \mathbf{u}_i(k-1) \end{bmatrix} \tag{26}
$$

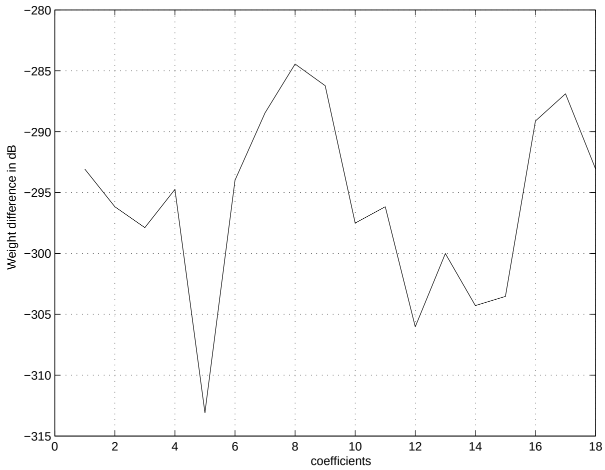

Figure 4: Comparison of difference of the weights obtained with the IQRD-RLS algorithm and the MC-FQRD-RLS algorithm with weight extraction.

where $z_i(k-1)$ is the ith element of vector $\mathbf{z}(k)$

$$z_i(k-1) = -\mathbf{f}_P^{\mathrm{H}}(k-1)\mathbf{u}_i(k-1)/\gamma(k-1) \tag{27}$$

and the elements of vector $\mathbf{f}_P(k-1)$ and $\gamma(k-1)$ are obtained from the rotation matrix $\mathbf{Q}_\theta(k-1)$ as

$$\begin{bmatrix}\gamma(k-1)\\ \mathbf{f}(k-1)\end{bmatrix} = \mathbf{Q}_\theta(k-1)\begin{bmatrix}1\\ \mathbf{0}_{P\times 1}\end{bmatrix} \tag{28}$$

Equation (28) needs only to be evaluated once at the beginning of the weight extraction procedure.

Proof of Lemma 2:
The update equation for $\mathbf{a}_P(k)$ and $\mathbf{f}_P(k)$ is given by (15) and (16) respectively. If same rotation matrix $\tilde{\mathbf{Q}}_{\theta f}$ is considered we get

$$\mathbf{a}_{P+1}(k) = \tilde{\mathbf{Q}}_{\theta f}(k-1)\begin{bmatrix}\mathbf{a}_P(k-1)\\ \mathbf{r}(k)\end{bmatrix} \tag{29}$$

$$\mathbf{f}_{P+1}(k-1) = \tilde{\mathbf{Q}}_{\theta f}(k-1)\begin{bmatrix}\mathbf{f}_P(k-2)\\ \mathbf{p}(k-1)\end{bmatrix} \tag{30}$$

where $\mathbf{r}(k) = \lambda^{1/2}[\mathbf{E}'_f(k-1)]^{-\mathrm{H}}\tilde{\mathbf{e}}'_f(k)$, $\mathbf{p}(k-1) = \lambda^{-1/2}\mathbf{r}(k)$, and

$$\tilde{\mathbf{e}}'^{*}_f(k) = \mathbf{x}^*(k) - \mathbf{W}_{Pf}^{\mathrm{H}}(k-1)\mathbf{x}_P(k-1) \tag{31}$$

with $\mathbf{W}_{Pf}(k) = \mathbf{U}_P^{-1}(k)\mathbf{D}_{fq2}(k)$. Using Equation (31), the definition of $\mathbf{a}_P(k)$, and removing vectors related to input signal $x(k)$, the following relation is obtained from Equation (15)

$$\begin{bmatrix}[-\mathbf{E}_{bq1}(k)]^{-\mathrm{H}}\mathbf{D}_{bq2}^{\mathrm{H}}(k)\mathbf{U}_P^{-\mathrm{H}}(k-1) & [\mathbf{E}_{bq1}(k)]^{-\mathrm{H}}\\ \mathbf{U}_P^{-\mathrm{H}}(k) & \mathbf{0}_{P\times M}\end{bmatrix}$$
$$= \mathbf{Q}'_{\theta f}(k)\begin{bmatrix}\mathbf{0}_{P\times M} & \mathbf{U}_P^{-\mathrm{H}}(k-1)\\ [\mathbf{E}'_f(k)]^{-\mathrm{H}} & -[\mathbf{E}'_f(k)]^{-\mathrm{H}}\mathbf{D}_{fq2}^{\mathrm{H}}(k)\mathbf{U}_P^{-\mathrm{H}}(k-1)\end{bmatrix} \tag{32}$$

Note that we can also reach (32) from (30). Considering the partition of matrix $\mathbf{U}_P^{-\mathrm{H}}(k-1)$ into its column vectors $\mathbf{u}_i(k-1)$, the column version of (32) becomes

$$\begin{bmatrix}\tilde{\mathbf{r}}'(k)\\ \mathbf{u}_{i-1+M}(k)\end{bmatrix} = \mathbf{Q}'_{\theta f}(k)\begin{bmatrix}\mathbf{u}_{i-1}(k-1)\\ \tilde{\mathbf{r}}_{i-1}(k)\end{bmatrix} \tag{33}$$

where $\tilde{\mathbf{r}}_{i-1}(k) = -[\mathbf{E}'_f(k)]^{-\mathrm{H}}\mathbf{D}_{fq2}^{\mathrm{H}}(k)\mathbf{u}_{i-1}(k-1)$. From (32), the first M columns correspond to initialization. In (33) we have $\mathbf{u}_{-j}(k-2) = \mathbf{0}_{NM\times 1}$ and $\tilde{\mathbf{r}}_{-j}(k) = \mathbf{e}_{f,-j}(k)$, where $\mathbf{e}_{f,-j}(k)$ is the jth column of $-[\mathbf{E}'_f(k)]^{-\mathrm{H}}$.

Table 3: Operations required for weight extraction (WE): M is the number of channels, N is the number of coefficients per channel, $P = MN$ is the total number of coefficients.

ALG.	MULT	DIV	SQRT
MCFQR	$4PM^2 + 11PM$ $+9P + 5.5M^2$ $+7.5M + 1$	$PM + P$ $+1.5M^2 + M$ $+1$	$PM + P$ $+M$
WE (per weight i)	$(5PM + 5P$ $+M^2)i$	M	0
WE (total)	$5P^2M + 5P^2$ $+PM^2$	M	0
IQRD-RLS	$3P^2 + 2P + 1$	$2P$	P

Acknowledgment

This work was partially funded by the Academy of Finland, Smart and Novel Radios (SMARAD) Center of Excellence, CNPq, CAPES, and FAPERJ.

REFERENCES

[1] A. A. Rontogiannis, S. Theodoridis, “New fast inverse QR least squares adaptive algorithms,” *International Conference on Acoustics, Speech, and Signal Processing. (ICASSP’1995)*, Detroit, MI, USA, pp. 1412–1415, 1995.

[2] M. Shoaib, S. Werner, J. A. Apolinário Jr., T. I. Laakso, “Solution to the weight extraction problem in fast QRD-RLS algorithms,” *International Conference on Acoustics, Speech, and Signal Processing. (ICASSP’2006)*, Toulouse, France, 2006.

[3] C. R. Ward *et al.*, “Application of a systolic array to adaptive beamforming,” *IEE Proceedings*, vol. 131, pp. 638–645, 1984.

[4] Phillip A. Regalia and Maurice G. Bellanger, “On the duality between fast QR methods and lattice methods in least squares adaptive filtering,” *IEEE Trans. Signal Processing*, vol. 39, no. 4, pp. 876–891, Apr. 1991.

[5] J. A. Apolinário Jr., M. G. Siqueira, and P. S. R. Diniz, “The 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, 2003.

[6] P. S. R. Diniz, *Adaptive Filtering: Algorithms and Practical Implementation*. Kluwer Academic Press USA, 1997.

[7] J. A. Apolinário Jr., P. S. R. Diniz, “A new fast QR algorithm based on a priori errors,” *IEEE Signal Processing Lett.*, vol. 4, no. 11, pp. 307–309, Nov. 1997.

[8] A. A. Rontogiannis, S. Theodoridis, “Multichannel Fast QRD-LS Adaptive Filtering: New Technique and Algorithms,” *IEEE Trans. Signal Processing*, Vol. 46, No. 11, pp. 2862–2876, 1998.

[9] C.A. Medina, J.A. Apolinário Jr., M.G. Siqueira, “A unified framework for multichannel fast QRD-LS adaptive filters based on backward prediction errors,” *Midwest Symposium on Circuits and Systems. (MWSCAS’2002)*, vol. 3, pp. 668–671, Aug. 2002.