MULTICHANNEL FAST QR-DECOMPOSITION RLS ALGORITHMS WITH EXPLICIT WEIGHT EXTRACTION

Mobien Shoaib,1 Stefan Werner,1 José A. Apolinário Jr.,2 and Timo I. Laakso1

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

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

1. INTRODUCTION

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.

2. THE MULTICHANNEL FAST QR-DECOMPOSITION ALGORITHM

This section presents the basic principles of the block MC-FQRD-RLS algorithm. Thereafter, the weight extraction (WE) algorithm is derived. Simulation results are followed by conclusions.

2.1 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 $w_p(k)$

$$\xi(k) = \sum_{i=0}^{k} 2^{k-i}|d(i) - x_i^H|\sum_{p} w_p(k) |^2 = \|e^*(k)\|^2$$

(1)
We emphasize that after rotation, similarly, $e(k)$ is the multichannel coefficient vector, $x(k)$ is the a posteriori error vector given as

$$ e^*(k) = \begin{bmatrix} d^*(k) \\ \lambda^{-1/2} d^*(k-1) \\ \vdots \\ \lambda^{k/2} d^*(0) \end{bmatrix} = d^*(k) - U_p(k)w_p(k) $$

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

$$ x_p(k) = \begin{bmatrix} x^T(k) & x^T(k-1) & \ldots & x^T(k-N+1) \end{bmatrix}^T $$

and $x(k) = [x_1(k) \ x_2(k) \ \ldots \ x_M(k)]^T$ is the $M \times 1$ input vector. The QRD-RLS algorithm uses an orthogonal rotation matrix $Q_p(k)$ in $\mathbb{C}^{(k+1) \times (k+1)}$ to triangularize matrix $X_p(k)$ as

$$ \begin{bmatrix} 0_{(k+1-P) \times P} & X_p(k) \end{bmatrix} = Q_p(k)X_p(k) $$

where $U_p(k) \in \mathbb{C}^{P \times P}$ is the Cholesky factor of the deterministic autocorrelation matrix $R_p(k) = X_p(k)^H X_p(k)$. Pre-multiplying (2) with $Q_p(k)$ gives

$$ Q_p(k)e^*(k) = \begin{bmatrix} e_{q1}(k) \\ e_{q2}(k) \end{bmatrix} = \begin{bmatrix} d_{q1}(k) \\ d_{q2}(k) \end{bmatrix} = \begin{bmatrix} 0_{(k+1-P) \times P} & U_p(k) \end{bmatrix}w_p(k) $$

We emphasize that $d_{q1}(k)$ and $d_{q2}(k)$ are partitions of vector $d^*(k)$ after rotation, similarly $e_{q1}(k)$ and $e_{q2}(k)$ are partitions of vector $e^*(k)$ after rotation. The cost function in (1) is minimized by choosing $w_p(k)$ such that $d_{q2}(k) - U_p(k)w_p(k)$ is zero, i.e.,

$$ w_p(k) = U_p^{-1}(k)d_{q2}(k) $$

The QRD-RLS algorithm updates vector $d_{q2}(k)$ and matrix $U_p(k)$ as follows [6]

$$ \begin{bmatrix} e_{q1}(k) \\ e_{q2}(k) \end{bmatrix} = Q_p(k) \begin{bmatrix} d^*(k) \\ \lambda^{1/2} d_{q2}(k-1) \end{bmatrix} $$

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

$$ Q_p(k) = \begin{bmatrix} \gamma(k) & P_p(k) \\ p(k) & \gamma(k) \end{bmatrix} $$

The QRD-RLS algorithm is complete with the definition of the a priori error value $e(k) = e_{q2}(k)/\gamma(k)$ where $\gamma(k)$ is a scalar element in matrix $Q_p(k)$, 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-FQRD_PRI and MC-FQRD_POS. The MC-FQRD_PRI algorithm updates vector $a_P(k)$ defined as

$$ a_P(k) = \lambda^{-1/2}U_p^{-H}(k-1)x_p(k) = -g_p(k)/\gamma(k) $$

while the MC-FQRD_POS algorithm updates vector $f_p(k)$ given by

$$ f_p(k) = U_p^{-H}(k)x_p(k) $$

The basic idea of the MC-FQRD based algorithms is to replace the update for matrix $U_p^{-H}(k)$ in (9) with an update equation for a vector, i.e., either $a_P(k)$ or $f_p(k)$. The extended Cholesky matrix $U_{p+1}(k) \in \mathbb{C}^{(P+M) \times (P+M)}$ is defined as the Cholesky factor of the extended input matrix $X_{p+1}(k) \in \mathbb{C}^{(k+1) \times (P+M)}$

$$ \begin{bmatrix} 0_{(k+1-P-M) \times (P+M)} & X_{p+1}(k-1) \end{bmatrix} = Q_{p+1}(k-1)X_{p+1}(k-1) $$

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

$$ U_p^{-H}(k-1) = Q_{p+1}(k-1) \times \begin{bmatrix} 0_{P \times M} & U_{p+1}(k-1) \end{bmatrix} + [E_{f}(k-1)]^H - [E_{f}(k-1)]^H D_{f,l}(k-1)U_p^{-H}(k-2) $$

where $E_{f}(k)$ and $D_{f,l}(k)$ are the rotated forward prediction error and desired signal matrices, respectively. By post-multiplying (14) with the extended multichannel input data vector $x_{p+1}(k-1)$ and $\lambda^{-1/2}$ we obtain the update equation for vector $a_P(k)$

$$ a_{p+1}(k) = \lambda^{-1/2}Q_{p+1}(k-1) \begin{bmatrix} a_P(k-1) \\ r(k) \end{bmatrix} $$

where $r(k) = \lambda^{-1/2}[E_{f}(k-1)]^H \tilde{e}_f(k)$ and $\tilde{e}_f(k) = \gamma(k-1)\tilde{e}_{f,l}(k)$. If we instead post-multiply (14), evaluated at time instant $k$, with the vector $x_{p+1}(k-1)$, we get the update equation for $f_p(k)$

$$ f_{p+1}(k) = Q_{p+1}(k) \begin{bmatrix} f_p(k-1) \\ p(k) \end{bmatrix} $$

where $p(k) = [E_{f}(k)]^H \tilde{e}_f(k)$ and $\tilde{e}_f(k) = \gamma(k-1)\tilde{e}_{f,l}(k)$.
Table 1: The FQRPRLB 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.

<table>
<thead>
<tr>
<th>Step 1 applying: $Q_{0f}(k)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 2 applying: $Q_{0f}(k)$</td>
</tr>
</tbody>
</table>

Figure 2: The procedure for updating $u_i^c(k-1)$ for weight extraction in MFQRD-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 $\delta_i$ with a “1” at the $i$th position ($1 \leq i \leq P$). Note that vector $x_p(k)$ 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 $M$th position in vector $x_p(k)$. The $j$th element of the weight vector for the $i$th channel is given as

$$w_{i+jM}^p(k) = w_{i+1}^p(k-1)\delta_i + M \cdots \delta_{jM}$$

where $u_{i+jM}(k-1)$ is the $(i + jM)$th column of $U_p^{-1}(k-1)$.

**Lemma 1.** Let $u_i^T(k) = [u_i(0) \ldots u_i(k-1)]^T \in \mathbb{C}^{p \times 1}$ denote the $i$th column of the upper triangular matrix $U_p^H(k) \in \mathbb{C}^{p \times p}$. Given $Q_{0f}(k-1) \in \mathbb{C}^{(P+1) \times (P+1)}$ from Table 1, then $u_i(k-2)$ can be obtained from $u_i(k-1)$ using the relation below

$$u_i(k-2) = [0 \cdots 0 \lambda_{i+1}^{-1/2}u_i(k-1)]^T z_i, i = 0, \ldots, N - 1$$

where $z_i = -\delta_i^H(k-1)u_i(k-1)/\gamma(k)$. 

**Lemma 2.** Let $u_i^T(k) = [u_i(0) \ldots u_i(k-1)]^T \in \mathbb{C}^{p \times 1}$ denote the $i$th column of the upper triangular matrix $U_p^H(k-1) \in \mathbb{C}^{p \times p}$. Given $Q_{0f}(k) \in \mathbb{C}^{(P+1) \times (P+1)}$ from Table 1, then $u_i(k-1)$ can be obtained from $u_{i+jM}(k-2)$ using the following relation

$$u_i(k-1) = F_i(k-1)^{-1}u_{i+jM}(k-2), \quad j = 1, \ldots, M$$

where $F_i(k) = -[E_i(k)]^{-1}\tilde{D}_{i+1}^H(k)$. Also for $j = 1$ and $j = M$, $u_i(k-1) = 0_{P \times 1}$ and $F_i(k) = \varepsilon_i$, where $\varepsilon_i$ is the $i$th column of $-E_i(k)^{-1}$. 

Assuming vector $u_i(j-1)M(k-1)$ to be known, Lemmas 1 and 2 can be used first to compute vector $u_{i+jM}(k-2)$ and

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

$$y_p(k) = w_{p}^T(k-1)x_p(k) = d_{q2}^H(k-1)U_p^{-H}(k-1)x_p(k)$$

From the updated vector $a_p(k)$, the update equation for the rotation matrix $Q_{0f}(k)$ is obtained as

$$Q_{0f}(k) = Q_{0f}(k-1)\left[1 - a_p(k)^{-1}\right]$$

Similarly, the rotation matrix is also obtained from the updated vector $f_p(k)$

$$Q_{0f}(k) = Q_{0f}(k-1)\left[1 - a_p(k)^{-1}\right]$$

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

$$Q_{0f}(k) = Q_{0f}(k-1)\left[1 - a_p(k)^{-1}\right]$$

and

$$Q_{0f}(k) = Q_{0f}(k-1)\left[1 - a_p(k)^{-1}\right]$$

where $a_p$ is an a priori unknown variable (dummy variable). The MFQRD-RLS algorithm is summarized in Table 1. For details on the MFQRD-RLS implementation, see [9]. As can be seen from Table 1, the weight vector $w_p(k)$ is not available. The next section presents an algorithm that extracts the weight vector values of both algorithms.
Lemma 1 and then interference signals with direction of arrivals \( w \) the weight coefficients \( a \) priori coefficients is given in Table 3. For comparison, the computational summarized in Table 2.

The multichannel weight extraction algorithm is considered. The second application is a multichannel prediction errors and the inverse QRD-RLS algorithm are given.

For the appropriate column of matrix \( P \)


taking the total number of coefficients.

The number of operations required to completely extract all the

for each \( i \)

\( \gamma(k - 1) = Q_{\theta}^{-1}(k - 1)H(k - 2)u_{i+1}\) for \( i = 0; \ldots, M - 1 \)

Compute \( u_i(k - 1) \)

\( \tilde{z}_{i+1}^j M(k) \)

\( u_{i+1}^j M(k - 2) \)

Compute \( f_{i+1}^j M(k) \)

\( P_{\theta}^j(k) \)

\( \theta_{\gamma}^j(k - 1) = Q_{\theta}^{-1}(k - 1)H(k - 1)u_i(k - 1) \)

Compute \( z_{i+1}^j M(k) \)

\( f_{i+1}^j M(k) = E_{\gamma}D_{\gamma}^j(k - 1)u_{i+1}^j M(k - 2) \)

Compute the coefficients \( w^j_{i+1}(k - 1) = u_{i+1}^j M(k - 1)d_{\gamma}^j(k - 1) \)

then \( u_{i+1}^j M(k - 1), \) respectively. Therefore all the column vectors corresponding to the \( i \)th channel are obtained by iterating through all the possible values of \( j \). Consequently, we obtain all the weights for the \( i \)th channel. Note that in order to obtain the column vector \( u_{i+1}^j M(k - 1) \) corresponding to a particular channel, we need to initialize (24) given in Lemma 2 properly, which means choosing the appropriate column of matrix \( E_{\gamma}^{-1}(k - 1)H^{-1} \). A schematic for obtaining the column vectors is given in Figure 2. It is shown that starting from \( u_0(k - 1), \) first the column \( u_0(k - 2) \) is obtained using Lemma 1 and then \( u_{i+1}^j M(k - 1) \) using Lemma 2. Both of them correspond to channel 1. These column vectors can be used to compute the weight coefficients \( w^j_{i+1}(k - 1) \) and \( w^j_{i+1}(k - 1) \) 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 matrix. 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 \( \theta_d \) and the rest are interference signals with direction of arrivals \( \theta_i = 35^\circ, 45^\circ, \) and \( 50^\circ \) 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).

### 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 \( U_p^{-1}(k - 2) \) in the inverse QRD-RLS algorithm is given by

\[
\begin{bmatrix}
\tilde{z}(k - 1) \\
U_p^{-1}(k - 1)
\end{bmatrix} = Q_{\theta}(k - 1) \begin{bmatrix}
\gamma \cdot U_p^{-1}(k - 2) \\
\gamma
\end{bmatrix}
\]

(25)

where \( \tilde{z}(k - 1) = y^{-1}(k - 1)E_{\gamma}D_{\gamma}(k - 1)U_p^{-1}(k - 1) \). Pre-multiplying both sides with \( Q_{\theta}(k - 1) \) and considering each column we get

\[
\begin{bmatrix}
\tilde{z}(k - 1) \\
U_p^{-1}(k - 1)
\end{bmatrix} = Q_{\theta}(k - 1) \begin{bmatrix}
\gamma \cdot U_p^{-1}(k - 2) \\
\gamma
\end{bmatrix}
\]

(26)
where $z_i(k-1)$ is the $i$th element of vector $z(k)$

$$z_i(k-1) = -\hat{f}_i^H(k-1)u_i(k-1)/\gamma(k-1)$$

(27)

and the elements of vector $f(k-1)$ and $\gamma(k-1)$ are obtained from the rotation matrix $Q_\theta(k-1)$ as

$$\begin{bmatrix} \gamma(k-1) \\ f(k-1) \end{bmatrix} = Q_\theta(k-1) \begin{bmatrix} 1 \\ 0_{P-1} \end{bmatrix}$$

(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 $a_p(k)$ and $f_p(k)$ is given by (15) and (16) respectively. If same rotation matrix $Q_{\theta_f}$ is considered we get

$$a_{p+1}(k) = Q_{\theta_f}(k-1) \begin{bmatrix} a_p(k-1) \\ r(k) \end{bmatrix}$$

(29)

$$f_{p+1}(k-1) = Q_{\theta_f}(k-1) \begin{bmatrix} f_p(k-2) \\ p(k-1) \end{bmatrix}$$

(30)

where $r(k) = \lambda^{1/2}(E_f^H(k-1)-\gamma^2\hat{e}_f^H(k), p(k-1) = \lambda^{-1/2}r(k)$, and

$$\hat{e}_f^H(k) = x^H(k) - W_{f_f}(k-1)x_p(k-1)$$

(31)

with $W_{f_f}(k) = U_p^{-1}(k)D_{fq2}(k)$. Using Equation (31), the definition of $a_p(k)$, and removing vectors related to input signal $x(k)$, the following relation is obtained from Equation (15)

$$[-E_{pq1}(k)]^{-1}D_{pq2}^H(k)U_p^H(k-1) - [E_{pq1}(k)]^{-1}U_p^H(k-1)$$

$$= Q_{\theta_f}(k) \begin{bmatrix} 0_{P-M} \\ U_p^H(k-1) \end{bmatrix} \begin{bmatrix} \gamma(k-1) \\ \hat{e}_f^H(k) \end{bmatrix}^{-1}U_p^H(k-1)$$

(32)

Note that we can also reach (32) from (30). Considering the partition of matrix $U_p^H(k-1)$ into its column vectors $u_i(k-1)$, the column version of (32) becomes

$$\begin{bmatrix} \hat{f}_i^H(k) \\ u_{i-1+1}(k-1) \end{bmatrix} = Q_{\theta_f}(k) \begin{bmatrix} u_{i-1+1}(k-1) \\ \hat{f}_{i-1}^H(k-1) \end{bmatrix}$$

(33)

where $\hat{r}_{i-1}(k) = -[E_{f_i}^H(k)]^{-1}D_{fq2}^H(k)u_{i-1}(k-1)$. From (32), the first $M$ columns correspond to initialization. In (33) we have $u_{i-1}(k-2) = 0_{NM+1}$ and $\hat{r}_{i-1}(k) = \epsilon_{f_i-1}(k)$, where $\epsilon_{f_i-1}(k)$ is the $i$th column of $-[E_{f_i}^H(k)]^{-1}$.

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.

<table>
<thead>
<tr>
<th>ALG</th>
<th>MULT</th>
<th>DIV</th>
<th>SQRT</th>
</tr>
</thead>
<tbody>
<tr>
<td>MCFQR</td>
<td>4PM+11PM</td>
<td>PM+P</td>
<td>PM+P</td>
</tr>
<tr>
<td></td>
<td>9P+5.5M²</td>
<td>+1.5M²+1</td>
<td>+M</td>
</tr>
<tr>
<td>(per weight $i$)</td>
<td>5P²+5P²</td>
<td>$M$</td>
<td>0</td>
</tr>
<tr>
<td>(total)</td>
<td>5P²+5P²</td>
<td>$M$</td>
<td>0</td>
</tr>
<tr>
<td>IQRD-RLS</td>
<td>3P²+2P+1</td>
<td>2P</td>
<td>$P$</td>
</tr>
</tbody>
</table>

Acknowledgment

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

REFERENCES


