# Reduced Complexity Solution for Weight Extraction in QRD-LSL Algorithms

Mobien Shoaib, *Student Member, IEEE*, Stefan Werner, *Senior Member, IEEE*, and José A. Apolinário Jr., *Senior Member, IEEE*

***Abstract*—QR-decomposition-based least-squares lattice (QRD-LSL) algorithms do not provide the transversal weight vector in explicit form. These weights can be computed from the variables of the QRD-LSL algorithm using the Levinson–Durbin (LD) recursion. If the prediction coefficients do not vary over time, a reduced complexity but approximate solution can be obtained. Nonetheless, this approximate solution requires algorithm convergence and infinite memory support (forgetting factor equal to one). To obtain the exact weights at any time instant and for any choice of the forgetting factor, the computational complexity of the true LD recursion increases by an order of magnitude. In this letter, we show that an exact solution can be obtained with a reduced computational complexity and without any added restriction. Simulation results show that the solutions obtained using the proposed method and the exact LD recursion are the same up to the precision used, whereas the weights from the approximate method always deviate from the true solution.**

***Index Terms*—Adaptive filtering, adaptive systems.**

## I. Introduction

LEAST-SQUARES lattice (LSL)-based algorithms provide a good alternative to the recursive least-squares (RLS) algorithm. These algorithms are stable in finite precision [1], and their computational complexity is $\mathcal{O}(N)$, with $N$ being the number of coefficients, thus making them attractive for practical applications. An example of such algorithm is the (angle-normalized) QRD-LSL algorithm [1].

It is known that the QRD-LSL recursions do not explicitly provide the transversal weight vector. If an application does not require weight coefficients at each iteration, e.g., in system identification, a weight extraction mechanism can be used in tandem with the QRD-LSL algorithm at the particular iteration of interest, thus saving computational cost. In order to identify the exact weights, the least-squares version of the Levinson–Durbin (LD) recursion may be used [1], [2] for which the computational cost is $\mathcal{O}(N^3)$. However, if infinite memory support and algorithm convergence are assumed, the computational complexity may be decreased by an order of magnitude. This is because the backward prediction weights have converged and the "solution pyramid" of the LD recursion reduces to the one in [3]. Since these assumptions do not usually hold in practice, only an approximate solution is obtained which may differ significantly from the true solution (even after convergence).

In this letter, we show how to extract the exact transversal weights associated with the QRD-LSL algorithm using the least-squares LD with a computational complexity of $\mathcal{O}(N^2)$, i.e., significantly reducing the computational cost. In our solution, the backward prediction coefficients of all orders are efficiently computed by exploiting their relation to the Cholesky factor of the input-data matrix. The elements of each column in the inverse Cholesky factor are recursively computed using the LD recursion in conjunction with a sequence of Givens rotations. Thereafter, the transversal weights can be obtained as in [1]. The proposed method is different than the serial weight identification technique proposed for the fast QRD-RLS algorithm in [4] (or its multichannel version in [5]), where two Givens rotations matrices were used to get the rows of the inverse Cholesky factor. For the purpose of comparison, a system identification setup is considered and the weights obtained from the proposed method and the approximate LD recursion are compared to those obtained with the exact Levinson–Durbin algorithm. Also, the computational complexity of the proposed method is addressed.

## II. Basic Equations for the Algorithms

In this section, we provide the basic concepts of the QRD-RLS and the QRD-LSL algorithms to aid the explanation of the proposed weight extraction technique. The notation used with the QRD-LSL is adopted from [1], where a detailed description and pseudo-code implementation can be found.

### A. Basic Concepts of QR Decomposition Algorithms

The RLS algorithm minimizes the following cost function:

$$\xi_N(k) = \sum_{i=0}^{k} \lambda^{k-i} |d^*(i) - \mathbf{x}_N^{\mathrm{H}}(i)\mathbf{w}_N(k)|^2 = \|\boldsymbol{\varepsilon}_N^*(k)\|^2 \quad (1)$$

where $\lambda$ is the forgetting factor, $[.]^*$ denotes complex conjugate, $d(i)$ is the $i$th desired signal value, $\mathbf{x}_N(i) \in \mathbb{C}^{N\times 1}$ is the input vector at the $i$th instant, $\mathbf{w}_N(i) \in \mathbb{C}^{N\times 1}$ is the coefficient vector, and $\boldsymbol{\varepsilon}_N(k) \in \mathbb{C}^{(k+1)\times 1}$ is the *a posteriori* error vector

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

where $\mathbf{d}_N(k) \in \mathbb{C}^{(k+1)\times 1}$ is the desired signal vector, and $\mathbf{X}_N(k) \in \mathbb{C}^{(k+1)\times N}$ is the input data matrix. Note that the forgetting factor is incorporated into the definition of $\mathbf{d}_N(k)$ and $\mathbf{X}_N(k)$. The QRD-RLS algorithm uses an orthogonal rotation matrix $\mathbf{Q}_N(k)$ to triangularize matrix $\mathbf{X}_N(k)$ [1] as

$$\begin{bmatrix} \mathbf{0}_{N\times(k+1-N)} \\ \mathbf{U}_N(k) \end{bmatrix} = \mathbf{Q}_N(k)\mathbf{X}_N(k) \quad (3)$$

Manuscript received August 2, 2007; revised November 16, 2007. This work was supported in part by the Academy of Finland, Smart and Novel Radios (SMARAD) Center of Excellence, and in part by Brazilian agencies CAPES and CNPq. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Ljubisa Stankovic.

M. Shoaib and S. Werner are with the Signal Processing Laboratory, Helsinki University of Technology, FIN-02015 HUT Helsinki, Finland (e-mail: mobien@ieee.org; stefan.werner@tkk.fi).

J. A. Apolinário, Jr. is with the Department of Electrical Engineering, Instituto Militar de Engenharia, 22290-270 Rio de Janeiro, RJ, Brazil (e-mail: apolin@ieee.org).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/LSP.2008.917023

where $\mathbf{U}_N(k) \in \mathbb{C}^{N\times N}$ is the Cholesky factor of $\mathbf{R}_N(k) = \mathbf{X}_N^{\mathrm{H}}(k)\mathbf{X}_N(k)$; $[.]^{\mathrm{H}}$ denotes Hermitian. Pre-multiplying (2) with $\mathbf{Q}_N(k)$, $\mathbf{Q}_N(k)\boldsymbol{\varepsilon}_N^*(k)$ gives

$$\mathbf{e}_q(k) = \begin{bmatrix}\mathbf{e}_{q1}(k)\\ \mathbf{e}_{q2}(k)\end{bmatrix} = \begin{bmatrix}\mathbf{d}_{q1}(k)\\ \mathbf{d}_{q2}(k)\end{bmatrix} - \begin{bmatrix}\mathbf{0}_{(k+1-N)\times N}\\ \mathbf{U}_N(k)\end{bmatrix}\mathbf{w}_N(k). \tag{4}$$

If $\mathbf{d}_{q2}(k) - \mathbf{U}_N(k)\mathbf{w}_N(k)$ is zero, the cost function in (1) is minimized and

$$\mathbf{w}_N(k) = \mathbf{U}_N^{-1}(k)\mathbf{d}_{q2}(k). \tag{5}$$

The QRD-RLS update of $\mathbf{d}_{q2}(k)$ and $\mathbf{U}(k)$ is given by

$$\begin{bmatrix}e_{q1}(k)\\ \mathbf{d}_{q2}(k)\end{bmatrix} = \mathbf{Q}_{\theta N}(k)\begin{bmatrix}d^*(k)\\ \lambda^{1/2}\mathbf{d}_{q2}(k-1)\end{bmatrix} \tag{6}$$

$$\begin{bmatrix}\mathbf{0}_{1\times N}\\ \mathbf{U}_N(k)\end{bmatrix} = \mathbf{Q}_{\theta N}(k)\begin{bmatrix}\mathbf{x}_N^{\mathrm{H}}(k)\\ \lambda^{1/2}\mathbf{U}_N(k-1)\end{bmatrix} \tag{7}$$

where $\mathbf{Q}_{\theta N}(k) \in \mathbb{C}^{(N+1)\times(N+1)}$ is a sequence of Givens rotation matrices which annihilates the input vector $\mathbf{x}_N(k)$ in (7) and can be partitioned as [6]

$$\mathbf{Q}_{\theta N}(k) = \begin{bmatrix}\gamma_N(k) & \mathbf{g}_N^{\mathrm{H}}(k)\\ \mathbf{f}_N(k) & \mathbf{E}_N(k)\end{bmatrix}. \tag{8}$$

The QRD-RLS algorithm is complete with the definition of the *a priori* error value $e^*(k) = d^*(k) - \mathbf{x}^{\mathrm{H}}(k)\mathbf{w}_N(k-1) = e_{q1}(k)/\gamma_N(k)$, where $\gamma_N(k)$ is a scalar found in matrix $\mathbf{Q}_{\theta N}(k)$; see (8). An alternative relation used in the inverse QRD-RLS algorithm is [7]

$$\begin{bmatrix}\mathbf{z}^{\mathrm{H}}(k)\\ \mathbf{U}_N^{-\mathrm{H}}(k)\end{bmatrix} = \mathbf{Q}_{\theta N}(k)\begin{bmatrix}\mathbf{0}_{1\times N}\\ \lambda^{-1/2}\mathbf{U}_N^{-\mathrm{H}}(k-1)\end{bmatrix} \tag{9}$$

where $\mathbf{z}(k) = -\mathbf{U}^{-1}(k)\mathbf{f}(k)/\gamma(k)$ and $[.]^{-\mathrm{H}}$ means $\{[.]^{\mathrm{H}}\}^{-1}$. The above relation will be used in the weight extraction method presented in Section III.

### B. QRD-LSL Algorithm

The main idea in the QRD-LSL algorithm is to use order-recursive equations to find an efficient output-error-based adaptive filtering algorithm.

*1) Normalized Output Error Order-Recursive Equations:* The output-error vector in (2), considered for filter order $i$, is normalized by $\gamma_i^{1/2}(k)$ and rewritten in order recursive form as [1]

$$\boldsymbol{\epsilon}_i(k) = \boldsymbol{\epsilon}_{i-1}(k) - \kappa_i^*(k)\boldsymbol{\epsilon}_{bi-1}(k) \tag{10}$$

where the order update is from $i-1$ to $i$, $\boldsymbol{\epsilon}_{bi-1}(k) \in \mathbb{C}^{(k+1)\times 1}$ is the normalized backward prediction error vector, and

$$\kappa_i(k) = \frac{\boldsymbol{\epsilon}_{i-1}^{\mathrm{H}}(k)\boldsymbol{\epsilon}_{bi-1}(k)}{\boldsymbol{\epsilon}_{bi-1}^{\mathrm{H}}(k)\boldsymbol{\epsilon}_{bi-1}(k)} = \frac{p_{i-1}(k)}{\|\boldsymbol{\epsilon}_{bi-1}(k)\|} \tag{11}$$

is the regression coefficient responsible for the order update where $p_{i-1}(k)$ corresponds to the scaled value of $\kappa_i(k)$.

The normalized version of error vector in (2) is achieved for $i = N$; however, it is only after (10) has been computed for $1 \leq i < N$. Instead of the normalized output-error vector $\boldsymbol{\epsilon}_{i-1}(k)$, we are actually interested in computing the current value of the normalized output-error (scalar) $\epsilon_{i-1}(k)$. According to [1], the $i$th order update for $\epsilon_{i-1}(k)$ is written as

$$\begin{bmatrix}\sqrt{\lambda}\|\boldsymbol{\epsilon}_{bi-1}(k-1)\| & \epsilon_{bi-1}(k)\\ \sqrt{\lambda}p_{i-1}^*(k-1) & \epsilon_{i-1}(k)\end{bmatrix}\mathbf{Q}_{\theta bi-1}(k) = \begin{bmatrix}\|\boldsymbol{\epsilon}_{bi-1}(k)\| & 0\\ p_{i-1}^*(k) & \epsilon_i(k)\end{bmatrix} \tag{12}$$

where $\mathbf{Q}_{\theta bi-1}(k) \in \mathbb{C}^{2\times 2}$ is a Givens rotation matrix responsible for the $i$th order update of $\epsilon_i(k)$ with values of sine and cosine given by $\cos[\theta_{bi-1}(k)] = \lambda^{1/2}\|\boldsymbol{\epsilon}_{bi-1}(k-1)\|/\|\boldsymbol{\epsilon}_{bi-1}(k)\|$ and $\sin[\theta_{bi-1}(k)] = \epsilon_{bi-1}^*(k)/\|\boldsymbol{\epsilon}_{bi-1}(k)\|$.

*2) Normalized Backward Prediction Order-Recursive Equations:* The order-recursive equation for the normalized backward prediction error vector is written as

$$\boldsymbol{\epsilon}_{bi}(k) = \boldsymbol{\epsilon}_{bi-1}(k-1) + \kappa_{bi}^*(k)\boldsymbol{\epsilon}_{fi-1}(k) \tag{13}$$

where $\kappa_{bi}(k)$ is the backward reflection coefficient and $\boldsymbol{\epsilon}_{fi-1}(k) \in \mathbb{C}^{(k+1)\times 1}$. Note that (13) provides the update for $\boldsymbol{\epsilon}_{bi}(k)$ which is needed in (10). The backward reflection coefficient $\kappa_{bi}(k)$ is computed similarly to regression coefficient in Section II-BI, i.e.,

$$\kappa_{bi}(k) = \frac{\boldsymbol{\epsilon}_{bi-1}^{\mathrm{H}}(k-1)\boldsymbol{\epsilon}_{fi-1}(k)}{\boldsymbol{\epsilon}_{fi}^{\mathrm{H}}(k)\boldsymbol{\epsilon}_{fi-1}(k)} = \frac{p_{b,i-1}(k)}{\|\epsilon_{fi-1}(k)\|}. \tag{14}$$

Similarly, the update of the scalar $\epsilon_{bi-1}(k)$ is provided using Given rotation matrices as [1]

$$\begin{bmatrix}\sqrt{\lambda}\|\boldsymbol{\epsilon}_{fi-1}(k-1)\| & \epsilon_{fi-1}(k)\\ \sqrt{\lambda}p_{bi-1}^*(k-1) & \epsilon_{bi-1}(k-1)\end{bmatrix}\mathbf{Q}_{\theta fi-1}(k) = \begin{bmatrix}\|\boldsymbol{\epsilon}_{fi-1}(k-1)\| & 0\\ p_{bi-1}^*(k) & \epsilon_{bi}(k)\end{bmatrix} \tag{15}$$

where $\mathbf{Q}_{\theta fi-1}(k) \in \mathbb{C}^{2\times 2}$ is a Givens rotation matrix responsible for the $i$th order update of $\epsilon_{bi}(k)$ with sine and cosine values given by $\cos[\theta_{fi-1}(k)] = \lambda^{1/2}\|\boldsymbol{\epsilon}_{fi-1}(k-1)\|/\|\boldsymbol{\epsilon}_{fi-1}(k)\|$, $\sin[\theta_{fi-1}(k)] = \epsilon_{fi-1}^*(k)/\|\boldsymbol{\epsilon}_{fi-1}(k)\|$.

*3) Normalized Forward Prediction Order-Recursive Equations:* The order-recursive equation for normalized forward prediction error vector, which is s crucial for computing (13), is given by

$$\boldsymbol{\epsilon}_{fi}(k) = \boldsymbol{\epsilon}_{fi-1}(k) + \kappa_{fi-1}^*(k)\boldsymbol{\epsilon}_{bi-1}(k-1). \tag{16}$$

The update of $\epsilon_{fi}(k)$ is similar to $\epsilon_{bi-1}(k)$ in (15), i.e.,

$$\begin{bmatrix}\sqrt{\lambda}\|\boldsymbol{\epsilon}_{bi-1}(k-2)\| & \epsilon_{bi-1}(k-1)\\ \sqrt{\lambda}p_{fi-1}^*(k-1) & \epsilon_{fi-1}(k)\\ 0 & \sqrt{\gamma_{i-1}(k-1)}\end{bmatrix}\mathbf{Q}_{\theta bi-1}(k-1) = \begin{bmatrix}\|\boldsymbol{\epsilon}_{bi-1}(k-1)\| & 0\\ p_{fi-1}^*(k) & \varepsilon_{fi}(k)\\ \frac{\epsilon_{bi-1}(K-1)}{\|\boldsymbol{\epsilon}_{bi-1}(k-1)\|} & \sqrt{\gamma_i(k-1)}\end{bmatrix} \tag{17}$$

where $\mathbf{Q}_{\theta bi-1}(k-1)$ is the Givens rotation matrix used for the update of $\epsilon_{fi}(k)$, and $\gamma_{i-1}(k)$ is the $(i-1)$th-order conversion factor which corresponds to the scalar in (8) for $i = N+1$.

## III. Weight Extraction for QRD-LSL

In this section, we briefly describe the conventional weight extraction method using the Levinson–Durbin recursions. We point out reasonable approximations leading to a simplified solution and discuss the restrictions they impose on the method. Thereafter, we present the details of the reduced complexity solution for exact weight identification.

### A. Conventional Method

The weights of the RLS algorithm are related to the auto-correlation matrix $\mathbf{R}_N(k)$ and cross-correlation vector $\mathbf{p}_N(k)$ as $\mathbf{w}_N(k) = \mathbf{R}_N^{-1}(k)\mathbf{p}_N(k)$. The relation for the QRD-RLS algorithm is given in (5). Another relationship involves the backward prediction weight vectors $\mathbf{w}_{bi}(k) \in \mathbb{C}^{i\times 1}$ given in form of the backward prediction error equation

$$\boldsymbol{\epsilon}_{bi}(k) = \mathbf{X}_{i+1}(k)\begin{bmatrix} -\mathbf{w}_{bi}(k) \\ 1 \end{bmatrix} = \mathbf{X}_{i+1}(k)\mathbf{c}_{bi}(k) \qquad (18)$$

and the regression coefficients $\kappa_i(k)$ defined in (11) [1]. Equation (18) is obtained directly from (5) by applying the definition of the Cholesky matrix $\mathbf{U}_N(k)$ and making use of [2]

$$\mathbf{U}_N(k) = \mathbf{D}(k)\mathbf{L}^{-1}(k) \qquad (19)$$

where $\mathbf{D}(k) = \text{diag}\{\|\boldsymbol{\epsilon}_{bN-1}(k)\|, \ldots, \|\boldsymbol{\epsilon}_{b0}(k)\|\}$. Therefore

$$\mathbf{w}_N(k) = \mathbf{L}(k)\boldsymbol{\kappa}(k) \qquad (20)$$

with

$$\mathbf{L}(k) = \begin{bmatrix} & & -\mathbf{w}_{bi}(k) & & \\ -\mathbf{w}_{bN-1}(k) & & & & 1 \\ & \cdots & 1 & \cdots & \\ 1 & & & & \mathbf{0}_{N-1\times 1} \\ & & \mathbf{0}_{N-i-2} & & \end{bmatrix} \qquad (21)$$

and $\boldsymbol{\kappa}^{\mathrm{T}}(k) = [\kappa_{N-1}(k) \quad \kappa_{N-2}(k) \quad \ldots \quad \kappa_0(k)]$.

The scaled version of the elements of vector $\boldsymbol{\kappa}(k)$ is available as variable $p_i(k)$ defined in (11). Henceforth, we need to find a method to compute the backward prediction transversal weights in $\mathbf{L}(k)$. The LD recursion uses the reflection coefficients $\kappa_{fi}(k)$ and $\kappa_{bi}(k)$ to compute the backward prediction weights, for $i = 1, \ldots, N-1$, as follows [1]:

$$\mathbf{c}_{fi+1}(k) = \begin{bmatrix} \mathbf{c}_{fi}(k) \\ 0 \end{bmatrix} + \kappa_{fi+1}(k)\begin{bmatrix} 0 \\ \mathbf{c}_{bi}(k-1) \end{bmatrix}$$
$$\mathbf{c}_{bi+1}(k) = \begin{bmatrix} 0 \\ \mathbf{c}_{bi}(k-1) \end{bmatrix} + \kappa_{bi+1}(k)\begin{bmatrix} \mathbf{c}_{fi}(k) \\ 0 \end{bmatrix} \qquad (22)$$

where $\mathbf{c}_{fi}^{\mathrm{T}}(k) = [1 \quad -\mathbf{w}_{fi}^{\mathrm{T}}(k)]$, and $\mathbf{c}_{bi}^{\mathrm{T}}(k) = [-\mathbf{w}_{bi}^{\mathrm{T}}(k) \quad 1]$ are vectors containing the forward and backward prediction transversal weights, respectively. The computational complexity of this LD recursion is $\mathcal{O}(N^3)$. The reason is that the LD recursion in (22) requires the knowledge of the vector at previous time instant, $\mathbf{w}_{bi}(k-1)$, for computing the order updated vector at current time instant, i.e., $\mathbf{w}_{bi+1}(k)$. That is

$$\mathbf{w}_{b0}(k-N) \rightarrow \mathbf{w}_{b1}(k-N+1)\cdots \rightarrow \mathbf{w}_{bN}(k) \qquad (23)$$

where $\rightarrow$ denotes that there exists a relation from left to right. However, if we can assume that $\mathbf{w}_{bN}(k-1) = \mathbf{w}_{bN}(k)$, the complexity is reduced to $\mathcal{O}(N^2)$. *This assumption holds to be fairly accurate only after that convergence has taken place and for values of the forgetting factor close to 1, i.e., infinite memory support.* Therefore, the $\mathcal{O}(N^2)$ realization of (22) is only an approximation of the weight vector in (20).

### B. Proposed Weight Extraction Method

The weight extraction technique proposed here can be invoked at any iteration of the QRD-LSL algorithm. The main computational load associated with the Levinson–Durbin of the previous section is due to the construction of the matrix $\mathbf{L}(k)$, for which the chain in (23) must be followed. Using Lemma 1 below, we show how the weights $\mathbf{w}_{bi}(k-1)$ can be obtained from $\mathbf{w}_{bi}(k)$ using variables available in the QRD-LSL algorithm. This results in a great reduction of complexity as $\mathbf{L}(k)$ can be obtained from $N^2$ recursions in (22) as compared with $N^3$ recursions using the conventional method.

*Lemma 1:* Let $\mathbf{u}_{r,i}(k) \in \mathbb{C}^{1\times N}$ denote the $i$th row of the upper triangular matrix $\mathbf{U}_N^{-\mathrm{H}}(k) \in \mathbb{C}^{N\times N}$. Given $\mathbf{Q}_{\theta bi}(k) \in \mathbb{C}^{2\times 2}$ from (17), then $\mathbf{w}_{bi}(k-1)$ can be obtained from $\mathbf{w}_{bi}(k)$ using the relations

$$\mathbf{u}_{r,i}(k) = \frac{[-\mathbf{w}_{bi}^{\mathrm{T}}(k) \quad 1 \quad \mathbf{0}_{N-i-1}]}{\|\boldsymbol{\epsilon}_{bi}(k)\|} \qquad (24)$$

and

$$\begin{bmatrix} \mathbf{z}_{i-1}^{\mathrm{H}}(k) \\ \lambda^{-1/2}\mathbf{u}_{r,i}(k-1) \end{bmatrix} = \mathbf{Q}_{\theta bi}^{\mathrm{H}}(k)\begin{bmatrix} \mathbf{z}_i^{\mathrm{H}}(k) \\ \mathbf{u}_{r,i}(k) \end{bmatrix} \qquad (25)$$

where $i$ is 0 to $N-1$ and $\mathbf{z}_i^{\mathrm{H}}(k) = \mathbf{z}_{i-1}^{\mathrm{H}}(k)/\cos\theta_{bi}(k) - \mathbf{u}_{r,i}(k)\sin^*\theta_{bi}(k)/\cos\theta_{bi}(k)$.

The proof of Lemma 1 is in the Appendix . In order to build matrix $\mathbf{L}(k)$, the recursions in (22) are initialized as the Levinson–Durbin approach [1] to get $\mathbf{w}_{b1}(k)$. Lemma 1 is then invoked to obtain $\mathbf{w}_{b1}(k-1)$. By induction, we conclude that all columns of matrix $\mathbf{L}(k)$ can be obtained. Finally, the transversal weights $\mathbf{w}_N(k)$ are obtained from (20). The regression coefficient vector $\boldsymbol{\kappa}(k)$ is computed from (11). The implementation details of the proposed method are given in Table I. Note that the divisions $[\cos\theta_{bi}(k)]^{-1}$ in Table I are avoided as the variable $\|\boldsymbol{\epsilon}_{bi}(k-1)\|^{-1}$ is readily available from the QRD-LSL algorithm at each $k$. Therefore, $[\cos\theta_{bi}(k)]^{-1} = \lambda^{-1/2}\|\boldsymbol{\epsilon}_{bi-1}(k)\|/\|\boldsymbol{\epsilon}_{bi-1}(k-1)\|$ can be computed using two multiplications. Division by zero is not a problem since $\|\boldsymbol{\epsilon}_{bi-1}(k)\| \approx \|\boldsymbol{\epsilon}_{bi-1}(k-1)\|$ as $k$ increases. For the pseudo-code of the QRD-LSL, see [1]. The number of operations required to extract all coefficients using the exact and approximate Levinson–Durbin (LD) approach along with the proposed approach are given as follows:

- —LD exact: $(2N^3 - 4N^2 + 2N)/3$ multiplications and 0 divisions;
- —LD approx.: $N^2 - N$ multiplications and 0 divisions;
- —Proposed: $4N^2 + 5N$ multiplications and 0 divisions.

## IV. Simulations

In this section, we show that the performance of the weight extraction using the approximate LD recursion, i.e., the one that leads to the "solution pyramid" in [3], is always worse than the exact LD recursion. Moreover, the proposed method and the exact LD recursion show the same results at all time instances up to the simulation precision. We consider both finite precision and infinite precision results. A small regularization factor $\delta = 10^{-6}$ was chosen for the initialization of the QRD-LSL algorithm according to pseudocode given in [1, p. 666]. We consider a system identification setup where the input signal sequence is generated as a complex fifth-order AR process. The plant is an FIR filter with $N = 10$ randomly generated complex-valued taps, and the SNR was set to 30 dB. A single simulation run (3000 iterations) is considered, and the

TABLE I
WEIGHT EXTRACTION ALGORITHM

```
Initialization:
At any index $k$ of the QRD-LSL algorithm.
$\mathbf{c}_{bi}(k) = \begin{bmatrix}\mathbf{0}_{1\times i-1} & 1\end{bmatrix}^{\mathrm{T}}$
$\mathbf{c}_{fi}(k) = \begin{bmatrix}1 & \mathbf{0}_{1\times i-1}\end{bmatrix}^{\mathrm{T}}$
$\mathbf{z}_N(k) = \begin{bmatrix}\frac{-\sin\theta_{b0}(k)}{\|\boldsymbol{\epsilon}_{b0}(k)\|\cos\theta_{b0}(k)} & \mathbf{0}_{1\times N}\end{bmatrix}$
For any vector $\mathbf{v}$, $v^{(i)}$ defines the $i$th element
--------------------------------------------------------------
for each $0 \le i \le N-1${
    Update $\mathbf{c}_{fi}(k)$ and $\mathbf{c}_{bi}(k)$ using (22)
    $\mathbf{u}_{r,i+1}(k) = \frac{\mathbf{c}_{bi+1}(k)}{\|\boldsymbol{\epsilon}_{bi+1}(k)\|}$
    for each $1 \le j \le i+1${
        $u^{(j)}_{r,i+1}(k-1) = \frac{u^{(j)}_{r,i+1}(k)}{\cos\theta_{bi}(k)} - \frac{z^{(j-1)}_N(k)\sin\theta_{bi}(k)}{\cos\theta_{bi}(k)}$
        $z^j_N(k) = \frac{z^{(j-1)}_N(k)}{\cos\theta_{bi}(k)} - \frac{u^{(j)}_{r,i+1}(k)\sin\theta_{bi}(k)}{\cos\theta_{bi}(k)}$
    }
    $\mathbf{c}_{bi+1}(k-1) = \lambda^{1/2}\|\boldsymbol{\epsilon}_{bi}(k-1)\|\mathbf{u}_{r,i+1}(k-1)$
}
$\mathbf{w}_N(k) = \begin{bmatrix}\mathbf{c}_{b0}(k) & \mathbf{c}_{b1}(k) & \dots & \mathbf{c}_{bN-1}(k)\end{bmatrix}\boldsymbol{\kappa}(k)$
```

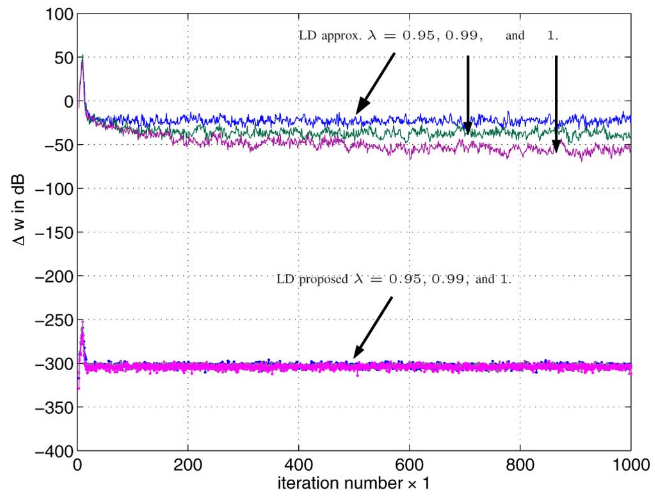

Fig. 1. Comparison of weight extraction techniques in infinite precision.

transversal weights obtained by the proposed method and the approximate LD recursions, at every step, are compared with those obtained by employing the exact LD recursion in (22). Fig. 1 shows the evolution of the coefficient error, defined as $\Delta w = (1/N)\|\mathbf{w}(k) - \mathbf{w}_{opt}(k)\|^2$, where $\mathbf{w}_{opt}(k)$ is the coefficient vector obtained with the exact LD recursion, and $\mathbf{w}(k)$ is the vector obtained with either the approximate LD or the proposed method. We see that the approximate LD method provides a good (but not exact) approximation only under restrictive assumptions of convergence and infinite memory support, while the proposed method is identical to the exact LD. The small deviation in the transient is due to the regularization. Fig. 2 shows the finite precision results of the average coefficient error, i.e., $\Delta\bar{w}(k) = 10^{-3}\sum_{k=2001}^{3000}\Delta w(k)$. We see that the proposed method behaves well in finite precision environment, approaching the quantization limit. Moreover, an average of ten independent runs of $10^5$ samples each was carried out. The case of 8-mantissa bits showed no sign of divergence.

## V. CONCLUSIONS

This letter showed how to use the variables of the QRD-LSL algorithm to compute the exact transversal weights in an efficient manner. The presented technique is an order of magnitude lower in complexity than a currently known exact method employing the Levinson–Durbin recursion. The proposed method has similar complexity to the solution obtained with the conventional Levinson–Durbin recursion assuming a stationary environment. Computer simulations showed that the results of the proposed method were identical to those of the Levinson–Durbin method in infinite precision and within the quantization limits of the finite precision environment.

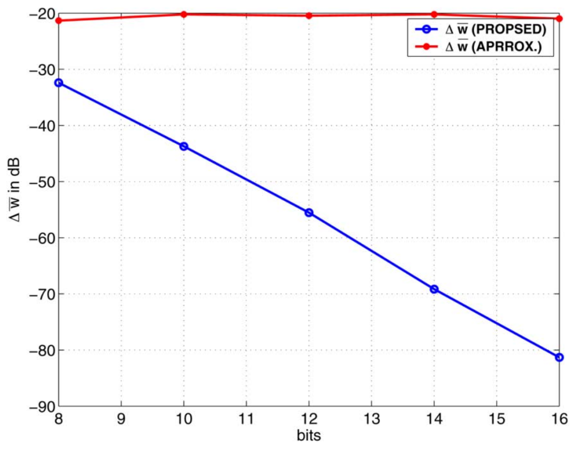

Fig. 2. Comparison of weight extraction techniques in a finite precision environment; 8–16 mantissa bits (forgetting factor $\lambda = 0.95$).

## APPENDIX

*Proof for Lemma 1:* The relation in (24) is obtained from (19) and (21). It is known that $\mathbf{Q}_{\theta N}(k) = \prod_{i=N-1}^{0}\mathbf{Q}_{\theta i}(k)$. The sine and cosine of rotation matrix $\mathbf{Q}_{\theta i}(k)$ and $\mathbf{Q}_{\theta bi}(k)$ are identical [2], i.e., $\mathbf{Q}_{\theta bi}(k)$ is the $2\times 2$ compact form of $\mathbf{Q}_{\theta i}(k)$. From (9), we see that matrix $\mathbf{Q}_{\theta i}(k)$ only applies on the first and the $(N-i)$th row of the right-hand-side matrix. In other words, row $(N-i)$ is available after this operation. Therefore, compact form using $\mathbf{Q}_{\theta bi}(k)$ is written as

$$\begin{bmatrix}\mathbf{z}_i^{\mathrm{H}}(k)\\ \mathbf{u}_{r,i}(k)\end{bmatrix} = \begin{bmatrix}\cos\theta_{bi}(k) & -\sin^*\theta_{bi}(k)\\ \sin\theta_{bi}(k) & \cos\theta_{bi}(k)\end{bmatrix}\begin{bmatrix}\mathbf{z}_{i-1}^{\mathrm{H}}(k)\\ \lambda^{-1/2}\mathbf{u}_{r,i}(k-1)\end{bmatrix}. \tag{26}$$

Due to the upper triangular structure of $\mathbf{U}_N^{-\mathrm{H}}(k)$, only $i$ entries of $\mathbf{z}_i(k)$ will be filled after this step. We can recursively compute $\mathbf{z}_i(k)$ by solving the expression (26) (starting from $i = 0$): $\mathbf{z}_i^{\mathrm{H}}(k) = (\mathbf{z}_{i-1}^{\mathrm{H}}(k) - \mathbf{u}_{r,i}(k)\sin^*\theta_{bi}(k))/\cos\theta_{bi}(k)$, where $\mathbf{u}_{r,i}(k)$, $\sin\theta_{bi}(k)$, $\cos\theta_{bi}(k)$, and $\mathbf{z}_{i-1}(k)$ are known.

## REFERENCES

[1] S. Haykin, *Adaptive Filter Theory*, 3rd ed. Englewood Cliffs, NJ: Prentice-Hall, 1996.
[2] P. A. Regalia and M. G. Bellanger, "On the duality between fast QR methods and lattice methods in least squares adaptive filtering," *IEEE Trans. Signal Process.*, vol. 39, no. 4, pp. 876–891, Apr. 1991.
[3] P. Strobach, "New forms of Levinson and Schur algorithms," *IEEE Signal Process. Mag.*, vol. 8, no. 1, pp. 12–36, Jan. 1991.
[4] M. Shoaib, S. Werner, J. A. Apolinário Jr., and T. I. Laakso, "Solution to the weight extraction problem in FQRD-RLS algorithms," in *Proc. ICASSP'2006*, Toulouse, France, 2006.
[5] M. Shoaib, S. Werner, J. A. Apolinário Jr., and T. I. Laakso, "Multi-channel fast QR-decomposition RLS algorithms with explicit weight extraction," in *Proc. EUSIPCO'2006*, Florence, Italy, 2006.
[6] J. A. Apolinário Jr. and P. S. R. Diniz, "A new fast QR algorithm based on *a priori* errors," *IEEE Signal Process. Lett.*, vol. 4, no. 11, pp. 307–309, Nov. 1997.
[7] S. T. Alexander and A. L. Ghirnikar, "A method for recursive least squares filtering based upon an inverse QR decomposition," *IEEE Trans. Signal Process.*, vol. 41, no. 1, pp. 20–30, Jan. 1993.