A QRD-LSL Smoothing Algorithm for Adaptive Equalization

Jenq-Tay Yuan

Department of Electronic Engineering, Fu Jen Catholic University, 24205 Taipei, Taiwan, R.O.C.

email: yuan@mars.ee.fju.edu.tw

Abstract — **This paper introduces order-recursive QRD-based least-squares lattice (QRD-LSL) smoothers for adaptive equalization. The QRD-LSL smoothers use past, present and future received data sequence to estimate the present value of the transmitted data sequence. Except for an overall delay needed for physical realization, QRD-LSL smoothers can substantially outperform conventional QRD-LSL filters in adaptive equalization while retaining all the advantages of an order-recursive structure.**

I. INTRODUCTION

One of the most important families of fast algorithms in *order-recursive* adaptive filtering are those of the conventional recursive *least-squares lattice (LSL)* algorithms [1]-[5][16]. Unfortunately, these fast algorithms tend to suffer from some form of numerical instability due to finite-precision effects. The *QR-decomposition (QRD)* technique is, in general, well-conditioned and numerically stable [4][6][8]. This is because all operations involved are the numerically well-conditioned orthogonal Givens rotations that are applied directly to the input data matrix rather than the associated covariance matrix. Furthermore, a useful property of the QRD technique is that, upon solving an N^{th} order filtering problem, the solutions to all lower order problems are obtained as a by-product [9][10]. This property is also found in the least-squares (LS) *lattice filters*. The lattice filter generally exhibits more robust characteristics when operated in numerically uncertain environments than the transversal filter. Accordingly, it is possible to solve the LSL problems using the QRD technique. Indeed, the LSL algorithm for adaptive filtering based on the QRD is endowed with a highly desirable set of features that include robustness to round-off error, superior numerical properties, modularity, and a high level of computational efficiency [4]. Proudler et al. [11][12] developed for the first time a QRD-based recursive least-squares algorithm using a lattice structure. Regalia and Bellanger [10] introduced a fast least squares algorithm, which is a hybrid between a QR and a lattice algorithm. Haykin [4] presented a development of the *QRD-based least-squares lattice (QRD-LSL)* filtering algorithm that is based on a hybridization of Proudler et al [11] and Regalia and Bellanger [10]. Extensive computer simulations have shown that the QRD-LSL algorithm has excellent numerical properties [7][12][13].

Smoothing differs from filtering in that not only the "subspace of past and present observations" but the "subspace of *future* observations" are taken into account in estimating the present desired signal. The smoothing process is known to be more accurate than the filtering process since the former is more "complete" than the latter in terms of the available information at a certain time step [14]. A smoother can be realized by introducing a suitable delay into any filter which makes the filter "noncausal" in the sense that a linear combination of the present, past and future observations can be used to estimate the present desired signal. However, once delay is introduced into a QRD-LSL filter, the order-recursive property no longer holds. Higher order noncausal filters, or smoothers, cannot be built from lower-order ones simply by adding more lattice stages as more "*future*" observations are used to estimate the present desired signal. A LS orthogonal basis theorem was thus developed in [5] for the design of *order-recursive LSL smoothers*. However, the recursive LSL smoothing algorithm derived in [5] requiring the inversion of a correlation matrix may suffer from some form of numerical instability and inaccuracy due to its numerical sensitivity to limited

0-7803-5284-X/99/$10.00

arithmetic precision. This paper extends the well-conditioned and numerically stable QRD-LSL filtering algorithm [12][4] to QRD-LSL smoothing algorithm while retaining all the desirable features of the former. The QRD-LSL smoothing algorithm is order-recursive in the sense that one can always increase the smoother order by adding more "past" stages (which correspond to more past observations used) as well as more "future" stages (which correspond to more future observations used) while the "old" part of the smoother still remains optimal. This property is particularly useful when there is no prior knowledge as to what the final value of the smoother order should be. The total computational complexity of the proposed algorithm scales linearly with the smoothing order, N. Except for an overall delay needed for physical realization, the order-recursive QRD-LSL smoothers implemented by using the QRD-LSL smoothing algorithm can substantially outperform the QRD-LSL filters. Computer simulation results show that the rate of convergence of the QRD-LSL smoothers developed in this paper is faster than that of the QRD-LSL filters with an appropriate delay. The latter are not order-recursive as more future observations are used to estimate the present desired signal.

II. FIR SMOOTHERS AND LS ORTHOGONAL BASIS SET

Consider the direct-form realization of an N^{th} order FIR least-squares smoother. The desired sequence x(i) is estimated from its current, p past, and f "future" observations y(i) (the data sequence), for i = 1,2,...,n. The length of the observations, n, is variable. The order, N=p+f. We will refer to any N^{th} order smoother that uses p past and f future data values as a $(p,f)^{th}$ order smoother where N=p+f is assumed implicitly. The estimation error is

$$e_{p,f}(i) = x(i) - \hat{x}_{p,f}(i)$$
$$= x(i) - \mathbf{h}^T_{p,f}(n-f)\mathbf{y}_{N+1}(i+f),\ 1 - f \le i \le n - f, \qquad (1)$$

where

$$\mathbf{y}^T_{N+1}(i+f) = [y(i+f), y(i+f-1), \ldots, y(i-p)] \qquad (2)$$

and

$$\mathbf{h}^T_{p,f}(n-f) = [h_{(p,f),f}(n-f), \ldots, h_{(p,f),1}(n-f),$$
$$h_{(p,f),0}(n-f), h_{(p,f),-1}(n-f), \ldots, h_{(p,f),-p}(n-f)]. \qquad (3)$$

The vector $\mathbf{h}_{p,f}(n-f)$ contains the fixed coefficients of the $(p,f)^{th}$ order FIR smoother and will be chosen, at time n-f, for least-squares estimation error over the time interval $1 - f \le i \le n - f$ with the prewindowing condition on the data. Equation (1) can be written in matrix form as

$$\mathbf{e}_{p,f}(n-f) = \mathbf{x}(n-f) - \mathbf{Y}_{N+1}(n)\,\mathbf{h}_{p,f}(n-f), \qquad (4)$$

where

$$\mathbf{x}^T(n-f) = [x(1-f), x(2-f), \ldots, x(n-f)], \qquad (5)$$
$$\mathbf{e}^T_{p,f}(n-f) = [e_{p,f}(1-f), e_{p,f}(2-f), \ldots, e_{p,f}(n-f)], \qquad (6)$$

$$\mathbf{Y}_{N+1}(n) = \begin{bmatrix} y(1) & 0 & & 0 \\ y(2) & y(1) & - & 0 \\ y(3) & y(2) & & 0 \\ \vdots & \vdots & & \vdots \\ y(n) & y(n-1) & \cdots & y(n-N) \end{bmatrix}. \qquad (7)$$

We use { } to represent each row of elements of matrix $\mathbf{Y}_{N+1}(n)$ as the linear subspace spanned by the columns of matrix $\mathbf{Y}_{N+1}(n)$. For example, the bottom row of elements of matrix $\mathbf{Y}_{N+1}(n)$ constitutes the linear subspace $\mathbf{Y}_{p,f}(n-f)$ = {y(n-N),y(n-N-1),...,y(n-1),y(n)}, which may be viewed as the subspace of the current observation (i.e., y(n-f)), p past observations (i.e., y(n-f-1),..., y(n-N)), and f future observations (i.e., y(n-f+1),...,y(n)). The optimum coefficients in (3) can be chosen by minimizing the squared Euclidean norm of the error vector $\|\Lambda^{1/2}(n)\mathbf{e}_{p,f}(n-f)\|^2$, where $\Lambda(n) = \text{diag}[\lambda^{n-1}, \lambda^{n-2}, \ldots, 1]$ is the n-by-n exponential weighting matrix in which $0 << \lambda \le 1$.

Since a smoother uses both past and future observations to estimate the desired sequence, we need to consider the problem of increasing order $N \to N + 1$ by increasing either f or p by one when developing an order-update recursion for the estimation error. This order-update recursion can be accomplished by embedding an $(N+1)^{st}$ order prediction lattice into a LSL smoother of order (p+1, f) or (p, f+1). It was first shown in [5] that when a sequence of p past and f future observations are considered, appropriate combinations of f delayed forward prediction errors and p delayed backward prediction errors form $C_N^f = \frac{N!}{(N-f)!\,f!}$ sets of orthogonal bases. The LS orthogonality among all the elements within each of these orthogonal bases has been referred to as the LS orthogonal basis theorem [5]. The LS orthogonal basis theorem is reproduced here for convenience:

There are C_N^f possible sets of (p+f+1) orthogonal bases directly accessible from an N^{th} order prediction error lattice that can be embedded into a LSL smoother of order (p,f). The following conditions must be satisfied for a set of f+p+1 prediction errors to form an orthogonal basis: (a) There are f forward and p backward prediction errors in the set; (b) the order of the forward and backward prediction errors corresponds to the total number of future and past observations used so far; (c) whenever a forward prediction error is used, all previous forward and backward prediction errors already in the set are delayed by one time unit.

For example, there are $C_3^1 = 3$ permissible sets of orthogonal bases that can be used in a LSL smoother of order (2,1). Each of the three orthogonal basis sets provides an orthogonal basis for the linear subspace $\mathbf{Y}_{2,1}(n-1) = \{y(n-3), y(n-2), y(n-1), y(n)\}$. The basis sets are

$\mathbf{e}_{2,1_{FBB}}^T(n-1) = [y(n-1), e_1^F(n), e_2^B(n), e_3^B(n)]$,

$\mathbf{e}_{2,1_{BFB}}^T(n-1) = [y(n-1), e_1^B(n-1), e_2^F(n), e_3^B(n)]$, and

$\mathbf{e}_{2,1_{BBF}}^T(n-1) = [y(n-1), e_1^B(n-1), e_2^B(n-1), e_3^F(n)]$, where $e_i^F(n)$ and $e_i^B(n)$ denote the forward and backward prediction errors of order i, respectively. The three basis sets are identified by the sequences FBB, BFB, and BBF of forward (F) and backward (B) prediction errors signified respectively in the subscript of the three above basis vectors. Each of the orthogonal basis sets can be obtained from the data y(n-3),y(n-2),y(n-1),y(n) by a Gram-Schmidt procedure. For example, consider the basis $\mathbf{e}_{2,1_{BFB}}^T(n-1) = [y(n-1), e_1^B(n-1), e_2^F(n), e_3^B(n)]$:

$$\begin{bmatrix} y(n-1) \\ e_1^B(n-1) \\ e_2^F(n) \\ e_3^B(n) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ c_{11}(n) & 1 & 0 & 0 \\ a_{21}(n) & a_{22}(n) & 1 & 0 \\ c_{32}(n) & c_{31}(n) & c_{33}(n) & 1 \end{bmatrix} \begin{bmatrix} y(n-1) \\ y(n-2) \\ y(n) \\ y(n-3) \end{bmatrix}, \tag{8}$$

where $a_{Ni}(n)$ and $c_{Ni}(n)$, i = 1, 2, ..., N, are N^{th} order forward and backward prediction coefficients, respectively. In this paper, we consider only the BFBFBF... sequence, among all C_N^f permissible sequences, for the implementation of a QRD-LSL smoother of an arbitrary order (p,f). For example, for a $(7,4)^{th}$ order smoother, we only consider the BFBFBFBFBBB sequence. Other sequences can be similarly obtained.

III. A MODIFIED QR-DECOMPOSITION FOR SMOOTHING

In this section we modify the well known QR-decomposition based on the LS orthogonal basis theorem so that it will be suited for implementing the order-recursive QRD-LSL smoothers. It can be shown that an n-by-n orthogonal matrix $\overline{\mathbf{Q}(n)}$ can always be constructed from one of the C_N^f orthonormal basis sets each of which provides an orthonormal basis for the linear subspace $\mathbf{Y}_{p,f}(n-f) = \{y(n-N), y(n-N+1), ..., y(n)\}$ such that [15] (due to space limitation, details are omitted here)

$$\overline{\mathbf{Q}(n)}\mathbf{Y}_{N+1}(n) = \begin{bmatrix} \mathbf{Q}_{p,f}^T(n) \\ \mathbf{S}^T(n) \end{bmatrix} \mathbf{Y}_{N+1}(n) = \begin{bmatrix} \mathbf{R}_{p,f}(n) \\ \mathbf{0} \end{bmatrix} \tag{9}$$

where $\mathbf{Q}_{p,f}^T(n)$ contains the first (N+1) rows of $\overline{\mathbf{Q}(n)}$, while $\mathbf{S}^T(n)$ contains the remaining rows. Since C_N^f possible sequences can be used, the matrix $\mathbf{R}_{p,f}(n)$ in (9) can display C_N^f different forms all of which, however, contain one (f+1)-by-(f+1) left-hand lower triangular matrix and one (p+1)-by-(p+1) right-hand upper triangular matrix. We indicate, in particular, that $\mathbf{R}_{p,f_{BFBFBF...}}(n)$ using the BFBFBF... sequence with p = f can be shown to be

$$\mathbf{R}_{p,f_{BFBFBF...}}(n) = \begin{bmatrix} F_N^{1/2}(n) & 0 & \cdots & 0 & \times & \cdots & \times \\ \times & \ddots & F_4^{1/2}(n-f+2) \ddots & \vdots & \vdots & \ddots & \vdots \\ \vdots & \ddots & F_2^{1/2}(n-f+1) & 0 & \times & \cdots & \times \\ \times & \cdots & \times & B_0^{1/2}(n-f) & \times & \cdots & \times \\ \times & \cdots & \times & 0 & B_1^{1/2}(n-f) & \ddots & \vdots \\ \vdots & \ddots & \vdots & \vdots & \ddots & B_3^{1/2}(n-f+1) \ddots & \times \\ \times & \cdots & \times & 0 & \cdots & 0 & B_{N-1}^{1/2}(n-1) \end{bmatrix}, \tag{10}$$

where the symbol $\times$ in (10) denotes either a zero or a nonzero element whose value is not of direct interest. Notice that unlike the upper triangular form we have in the conventional QRD in which the diagonal elements give the square roots of backward prediction error energies of all orders owing to the use of current and past observations only to estimate the present desired signal [4][10]. In matrix $\mathbf{R}_{p,f_{BFBFBF...}}(n)$, however, the first f diagonal elements give the square roots of forward prediction error energies corresponding to the f future observations y(n),...,y(n-f+1) and the remaining (p+1) diagonal elements give the square roots of backward prediction error energies corresponding to the current observation y(n-f) and the p past observations y(n-f-1),...,y(n-N). The order and the time delay associated with these prediction error energies on the main diagonal are obtained in accordance with the conditions stated in the LS orthogonal basis theorem. We will refer to the result in (9) as the *modified QRD for smoothing* and refer to the form of the lower and upper triangular matrices in $\mathbf{R}_{p,f}(n)$ as the *LU triangular form*. The modified QR-decomposition provides a theoretical basis for the development of the QRD-LSL smoothers (see Section IV). For the special case in which (p,f) = (N,0) (i.e., none of the future observations is considered), each column of $\mathbf{Q}_{p,f}(n)$ can be shown to be connected with backward prediction errors. Therefore the LU triangular form in the matrix $\mathbf{R}_{p,f}(n)$ in (9) is reduced to matrix $\mathbf{R}_{N,0}(n)$ which contains $C_N^0 = 1$ unique (N+1)-by-(N+1) upper triangular matrix whose diagonal elements give the square roots of backward prediction error energies of all orders. This is exactly the case of the conventional QRD [4][10].

IV. ORDER-RECURSIVE QRD-LSL SMOOTHERS

To develop the order-recursive QRD-LSL smoothers, we suppose that at time n-1, the (n-1)-by-(N+1) weighted data matrix $\Lambda^{1/2}(n-1)\mathbf{Y}_{N+1}(n-1)$ (see (7)) has already been reduced to the LU triangular form by the (n-1)-by-(n-1) orthogonal matrix, $\overline{\mathbf{Q}(n-1)}$. We thus have

$$\overline{\mathbf{Q}(n-1)}\Lambda^{1/2}(n-1)\mathbf{Y}_{N+1}(n-1) = \begin{bmatrix} \mathbf{R}_{p,f}(n-1) \\ \mathbf{O} \end{bmatrix} \tag{11}$$

where $\mathbf{R}_{p,f}(n-1)$ is an LU triangular matrix, and $\mathbf{O}$ is the (n-N-2)-by-(N+1) null matrix. With the solution at time (n-1) and the new observations for time n, we have

$$\begin{bmatrix} \overline{\mathbf{Q}(n-1)} & \mathbf{o} \\ \mathbf{o}^T & 1 \end{bmatrix} \Lambda^{1/2}(n)\mathbf{Y}_{N+1}(n)$$

$$= \begin{bmatrix} \lambda^{1/2}\mathbf{R}_{p,f}(n-1) \\ \mathbf{o} \\ y(n), y(n-1),..., y(n-f+1), y(n-f), y(n-f-1),..., y(n-N) \end{bmatrix}$$

$$= \mathbf{R}'_{p,f}(n) \tag{12}$$

where $\mathbf{o}$ in the matrix at the top of (12) is the (n-1)-by-1 zero vector. To complete the LU triangularization, we define the orthogonal matrix, $\widehat{\mathbf{Q}}_{p,f}(n)$, that is used to annihilate the new observations at the bottom row of $\mathbf{R}'_{p,f}(n)$ such that

$$\widehat{\mathbf{Q}}_{p,f}(n)\mathbf{R}'_{p,f}(n) = \begin{bmatrix} \mathbf{R}_{p,f}(n) \\ \mathbf{o} \\ \mathbf{o}^T \end{bmatrix}, \tag{13}$$

where $\mathbf{o}^T$ is the 1-by-(N+1) null vector and the (N+1)-by-(N+1) matrix $\mathbf{R}_{p,f}(n)$ corresponds to a complete LU triangular portion of the n-by-(N+1) weighted data matrix $\Lambda^{1/2}(n)\mathbf{Y}_{N+1}(n)$. The diagonal matrix $\widehat{\mathbf{Q}}_{p,f}(n)$ can be formed as the product of (N+1) Givens rotations as

$$\widehat{\mathbf{Q}}_{p,f}(n) = \mathbf{Q}_{0,0}(n)\mathbf{Q}_{1,0}(n)\mathbf{Q}_{1,1}(n)\mathbf{Q}_{2,1}(n)\mathbf{Q}_{2,2}(n)\ldots\mathbf{Q}_{p,f}(n)$$

for the BFBFBF... sequence which is one of the C_N^f possible sequences for a $(p,f)^{th}$ order smoother. Note that unlike the conventional QRD, the sequences of Givens rotations are applied by first annihilating y(n) in $\mathbf{R}'_{p,f}(n)$ in (12), and then successively annihilating the resulting bottom row by moving to the right in N steps. In the modified QRD, however, the current observation y(n-f) is first being annihilated by the Givens rotation $\mathbf{Q}_{0,0}(n)$ and then the annihilations proceed bidirectionally leftwards (for an "F") and rightwards (for a "B") in accordance with a particular sequence chosen (e.g., BFBFBF... in this case) until the leftmost observation, y(n), and the rightmost observation, y(n-N), are both annihilated to transform $\mathbf{R}'_{p,f}(n)$ into the LU triangular matrix as shown in (13). In other words, the order of performing the (N+1) Givens rotations in the combined effect of $\widehat{\mathbf{Q}}_{p,f}(n)$ (i.e., $\mathbf{Q}_{0,0}(n) \rightarrow \mathbf{Q}_{1,0}(n) \rightarrow \mathbf{Q}_{1,1}(n) \rightarrow \mathbf{Q}_{2,1}(n) \rightarrow \mathbf{Q}_{2,2}(n) \rightarrow \ldots \rightarrow \mathbf{Q}_{p,f}(n)$) must be consistent with the chosen sequence BFBFBF... until all the elements at the bottom row of $\mathbf{R}'_{p,f}(n)$ are annihilated to preserve the LU triangular form of $\mathbf{R}_{p,f}(n)$. We thus can obtain a recursion that relates the updated value of the n-by-n orthogonal matrix, $\overline{\mathbf{Q}(n)}$, to the old value of the orthogonal matrix, $\overline{\mathbf{Q}(n-1)}$:

$$\overline{\mathbf{Q}(n)} = \hat{\mathbf{Q}}_{p,f}(n)\begin{bmatrix} \overline{\mathbf{Q}(n-1)} & \mathbf{o} \\ \mathbf{o}^T & 1 \end{bmatrix} \quad (14)$$

To obtain the optimum value of $\mathbf{h}_{p,f}(n-f)$, we use $\overline{\mathbf{Q}(n)}$ to rotate (4) into

$$\begin{bmatrix} \mathbf{Q}^T_{p,f}(n) \\ \mathbf{S}^T(n) \end{bmatrix} \Lambda^{1/2}(n)\mathbf{e}_{p,f}(n-f) = \begin{bmatrix} \mathbf{Q}^T_{p,f}(n) \\ \mathbf{S}^T(n) \end{bmatrix} \Lambda^{1/2}(n)\mathbf{x}(n-f)$$

$$- \begin{bmatrix} \mathbf{Q}^T_{p,f}(n) \\ \mathbf{S}^T(n) \end{bmatrix} \Lambda^{1/2}(n)\mathbf{Y}_{N+1}(\mathbf{n})\,\mathbf{h}_{p,f}(n-f)$$

$$= \begin{bmatrix} \mathbf{p}_{p,f}(n-f) - \mathbf{R}_{p,f}(n)\mathbf{h}_{p,f}(n-f) \\ \mathbf{v}_{p,f}(n-f) \end{bmatrix} \quad (15)$$

The optimal weight vector containing the optimum coefficients of the (p,f)th order FIR smoother can be obtained from

$$\mathbf{p}_{p,f}(n-f) = \mathbf{R}_{p,f}(n)\mathbf{h}_{p,f}(n-f) \quad (16)$$

Note that since $\mathbf{R}_{p,f}(n)$ is LU triangular, the optimum coefficients may readily be solved by back substitution. This choice of $\mathbf{h}_{p,f}(n-f)$ yields a minimum value of the sum of weighted error squares

$$E_{p,f_{min}}(n) = \min_h \|\Lambda^{1/2}(n)\mathbf{e}_{p,f}(n-f)\|^2 = \|\mathbf{v}_{p,f}(n-f)\|^2. \quad (17)$$

Both vectors $\mathbf{p}_{p,f}(n-f)$ and $\mathbf{v}_{p,f}(n-f)$ in (15) can be computed recursively by using the same diagonal matrix, $\hat{\mathbf{Q}}_{p,f}(n)$,

$$\begin{bmatrix} \mathbf{p}_{p,f}(n-f) \\ \mathbf{v}_{p,f}(n-f) \end{bmatrix} = \hat{\mathbf{Q}}_{p,f}(n)\begin{bmatrix} \lambda^{1/2}\mathbf{p}_{p,f}(n-f-1) \\ \lambda^{1/2}\mathbf{v}_{p,f}(n-f-1) \\ x(n-f) \end{bmatrix} = \begin{bmatrix} \mathbf{p}_{p,f}(n-f) \\ \lambda^{1/2}\mathbf{v}_{p,f}(n-f-1) \\ \varepsilon_{p,f}(n-f) \end{bmatrix} \quad (18)$$

where $\varepsilon_{p,f}(n-f)$ can be referred to as the angle-normalized joint-process estimation error [4] of x(n-f) by using the current observation, f future observations and p past observations. The elements of vector $\mathbf{p}^T_{p,f}(n-f) = [\rho^F_{p,f}(n),\ldots,\rho^F_{1,1}(n-f+1),\rho^B_{0,0}(n-f),\rho^F_{1,0}(n-f),\ldots,\rho^F_{p,f-1}(n-1)]$ are referred to as the joint-process auxiliary parameters and play a key role in implementing a (p,f)th order-recursive QRD-LSL smoother. From (15), the auxiliary parameters can be shown to be the coefficients of the projection of the desired signal vector $\mathbf{x}(n-f)$ onto the orthonormal basis set provided by the rows of matrix $\mathbf{Q}^T_{p,f}(n)$. To develop recursions for computing the auxiliary parameters and angle-normalized joint-process estimation error for a QRD-LSL smoother, we consider first the problem of increasing order N → N+1 by using one additional *future* observation: (p, f) → (p, f+1). Since one additional future observation, y(n+1), is used, one can append the column vector $\mathbf{y}(n+1) = [y(1), y(2),\ldots, y(n+1)]^T$ to the left of the data matrix $\mathbf{Y}_{N+1}(n)$ and complete the top row with zeros to obtain matrix $\mathbf{Y}_{N+2}(n+1)$

$$\mathbf{Y}_{N+2}(n+1) = \begin{bmatrix} y(n+1) & \mathbf{o}^T \\ & \mathbf{Y}_{N+1}(n) \end{bmatrix} \quad (19)$$

Note that from (4) and (15), the rotation of the desired signal vector $\mathbf{x}(n-f)$ by the orthogonal matrix $\overline{\mathbf{Q}(n)}$ defined in (9) and the LU triangularization of $\mathbf{Y}_{N+1}(n)$ by the same matrix $\overline{\mathbf{Q}(n)}$ are exactly what is required in the solution of the (p,f)th order adaptive smoothing problem at time (n-f). Hence $\mathbf{x}(n-f)$ can also be appended to the right of $\mathbf{Y}_{N+1}(n)$ to generate the composite matrix:

$$\begin{bmatrix} y(n+1) & \mathbf{o}^T & 0 \\ & \mathbf{Y}_{N+1}(n) & \mathbf{x}(n-f) \end{bmatrix}.$$

Transforming the weighted data matrix $\Lambda^{1/2}(n)\mathbf{Y}_{N+1}(n)$ into a LU triangular matrix $\mathbf{R}_{p,f}(n)$ by using $\overline{\mathbf{Q}(n)}$, we have

$$\begin{bmatrix} 1 & \\ & \mathbf{Q}^T_{p,f}(n) \\ & \mathbf{S}^T(n) \end{bmatrix} \Lambda^{1/2}(n+1)\begin{bmatrix} y(n+1) & \mathbf{o}^T & 0 \\ & \mathbf{Y}_{N+1}(n) & \mathbf{x}(n-f) \end{bmatrix}$$

$$= \begin{bmatrix} \lambda^{n/2}y(1) & \mathbf{o}^T & 0 \\ \mathbf{p}^F(n+1) & \mathbf{R}_{p,f}(n) & \mathbf{p}_{p,f}(n-f) \\ \mathbf{v}^F(n+1) & \mathbf{O} & \mathbf{v}_{p,f}(n-f) \end{bmatrix} \quad (20)$$

where $\mathbf{p}^F(n+1) = [\mathbf{o}\ \mathbf{Q}^T_{p,f}(n)]\Lambda^{1/2}(n+1)\mathbf{y}(n+1)$ and $\mathbf{v}^F(n+1) = [\mathbf{o}\ \mathbf{S}^T(n)]\Lambda^{1/2}(n+1)\mathbf{y}(n+1)$ are (N+1)-by-1 vector and (n-N-1)-by-1 vector respectively. One can then rotate all the elements of vector $\mathbf{v}^F(n+1)$ into the (1,1)th element of the right-hand side of (20) by an orthogonal matrix $\mathbf{Q}_v(n+1)$

$$\mathbf{Q}_v(n)\begin{bmatrix} \lambda^{(n-1)/2}y(1) & \mathbf{o}^T & 0 \\ \mathbf{p}^F(n) & \mathbf{R}_{p,f}(\mathbf{n}-1) & \mathbf{p}_{p,f}(n-f-1) \\ \mathbf{v}^F(n) & \mathbf{O} & \mathbf{v}_{p,f}(n-f-1) \end{bmatrix}$$

$$=\begin{bmatrix} F_{N+1}^{1/2}(n) & \mathbf{o}^T & \rho_{p,f+1}^F(n) \\ \mathbf{p}^F(n) & \mathbf{R}_{p,f}(n-1) & \mathbf{p}_{p,f}(n-f-1) \\ \mathbf{o} & \mathbf{O} & \mathbf{v}_{p,f}(n-f) \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{p,f+1}(n) & \begin{matrix}\rho_{p,f+1}^F(n) \\ \mathbf{p}_{p,f}(n-f-1)\end{matrix} \\ \mathbf{O} & \mathbf{v}_{p,f}(n-f) \end{bmatrix} \quad (21)$$

to obtain the weighted forward prediction error energy $F_{N+1}(n) = \lambda^{(n-1)}y^2(1) + \|\mathbf{v}^F(\mathbf{n})\|^2$, where the matrix $\mathbf{R}_{p,f+1}(n)$ is LU triangular and it contains one (f+2)-by-(f+2) left-hand lower triangular matrix and one (p+1)-by-(p+1) right-hand upper triangular matrix. The auxiliary parameter, $\rho_{p,f+1}^F(n)$, in (21) is the first element of the order-updated vector $\mathbf{p}_{p,f+1}(n-f-1)$ (see (24)). Note that since one additional *future* observation has been used, one unit time delay has been introduced in (21) and the dimension of the lower triangular matrix in $\mathbf{R}_{p,f+1}(n)$ increases by one.

Now at time n+1, the new input y(n+1) becomes available for processing. Using the right-hand side of (21), we obtain

$$\begin{bmatrix} \lambda^{1/2}F_{N+1}^{1/2}(n) & \mathbf{o}^T & \lambda^{1/2}\rho_{p,f+1}^F(n) \\ \lambda^{1/2}\mathbf{p}^F(n) & \lambda^{1/2}\mathbf{R}_{p,f}(n-1) & \lambda^{1/2}\mathbf{p}_{p,f}(n-f-1) \\ \mathbf{o} & \mathbf{O} & \lambda^{1/2}\mathbf{v}_{p,f}(n-f) \\ & \mathbf{y}_{N+2}^T(n+1) & x(n-f) \end{bmatrix} = \mathbf{A}(n), \quad (22)$$

where $\mathbf{y}_{N+2}^T(n+1)$ = [y(n+1),...,y(n-f+1),y(n-f),y(n-f-1),...,y(n-N)]. Through a sequence of (N+1) Givens rotations applied to the above matrix in which the current observation y(n-f) is first being annihilated and then the annihilations proceed bidirectionally leftwards (for an "F") and rightwards (for a "B") in accordance with a particular sequence chosen until all the elements of the bottom row except for the $(n+1,1)^{th}$ element and the $(n+1,N+3)^{th}$ element have been annihilated by the orthogonal matrix $\hat{\mathbf{Q}}_{p,f}(n+1)$ such that

$$\hat{\mathbf{Q}}_{p,f}(n+1)\mathbf{A}(n) = \begin{bmatrix} \lambda^{1/2}F_{N+1}^{1/2}(n) & \mathbf{o}^T & \lambda^{1/2}\rho_{p,f+1}^F(n) \\ \mathbf{p}^F(n+1) & \mathbf{R}_{p,f}(n) & \mathbf{p}_{p,f}(n-f) \\ \mathbf{o} & \mathbf{O} & \lambda^{1/2}\mathbf{v}_{p,f}(n-f) \\ e_{N+1}^F(n+1)\ 0\ 0 & \cdots & 0\ \ \varepsilon_{p,f}(n-f) \end{bmatrix}$$

$$= \mathbf{B}(n) \quad (23)$$

where $\varepsilon_{p,f}(n-f)$ denotes the angle-normalized estimation error of x(n-f) by using the current observation, f future and p past observations, and $e_{N+1}^F(n+1)$ is the angle-normalized forward prediction error of order (N+1) at time (n+1) [4].

Finally, the LU triangular form can be accomplished by applying a single Givens rotation $\mathbf{Q}_{p,f+1}(n+1)$ that annihilates the nonzero element $e_{N+1}^F(n+1)$ in $\mathbf{B}(n)$ by rotating against the element $\lambda^{1/2}F_{N+1}^{1/2}(n)$ such that

$$\mathbf{Q}_{p,f+1}(n+1)\mathbf{B}(n) = \begin{bmatrix} F_{N+1}^{1/2}(n+1) & \mathbf{o}^T & \rho_{p,f+1}^F(n+1) \\ \mathbf{p}^F(n+1) & \mathbf{R}_{p,f}(n) & \mathbf{p}_{p,f}(n-f) \\ \mathbf{o} & \mathbf{O} & \lambda^{1/2}\mathbf{v}_{p,f}(n-f) \\ 0\ 0\ 0 & \cdots & 0\ \ \varepsilon_{p,f+1}(n-f) \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{p,f+1}(n+1) & \mathbf{p}_{p,f+1}(n-f) \\ \mathbf{O} & \mathbf{v}_{p,f+1}(n-f) \end{bmatrix} \quad (24)$$

where

$$\mathbf{Q}_{p,f+1}(n+1) = \begin{bmatrix} c_{f,N+1}(n+1) & & s_{f,N+1}(n+1) \\ & \mathbf{I}_{n-1} & \\ -s_{f,N+1}(n+1) & & c_{f,N+1}(n+1) \end{bmatrix}$$

The cosine-sine pair of rotation parameters of the Givens rotation $\mathbf{Q}_{p,f+1}(n+1)$ can be expressed as

$$c_{f,N+1}(n+1) = \frac{\lambda^{1/2}F_{N+1}^{1/2}(n)}{F_{N+1}^{1/2}(n+1)} \quad (25)$$

$$s_{f,N+1}(n+1) = \frac{e_{N+1}^F(n+1)}{F_{N+1}^{1/2}(n+1)} \quad (26)$$

where

$$F_{N+1}(n+1) = \lambda F_{N+1}(n) + \left(e_{N+1}^F(n+1)\right)^2 \quad (27)$$

which results from the use of the *forward* linear prediction since one additional *future* observation is used. By introducing one unit delay, an order-recursion for the angle-normalized estimation error by using one additional future observation can be obtained by (24) as

$$\varepsilon_{p,f+1}(n-f-1) = c_{f,N+1}(n)\varepsilon_{p,f}(n-f-1) - \lambda^{1/2}s_{f,N+1}(n)\rho_{p,f+1}^F(n-1) \quad (28)$$

$$\rho_{p,f+1}^F(n) = \lambda^{1/2}c_{f,N+1}(n)\rho_{p,f+1}^F(n-1) + s_{f,N+1}(n)\varepsilon_{p,f}(n-f-1) \quad (29)$$

Note that the computations of the cosine-sine pair of rotation parameters $c_{f,N+1}(n+1)$ and $s_{f,N+1}(n+1)$ described in (25) and (26), respectively, are the same as those of the rotations parameters computed in the *forward* linear prediction. Hence, a certain amount of computations needed for the computation of $\varepsilon_{p,f+1}(n-f-1)$ in (28) is saved. The a posteriori estimation error can then be computed by [17]

$$e_{p,f+1}(n-f-1) = \gamma_{p,f+1}^{1/2}(n)\varepsilon_{p,f+1}(n-f-1) \quad (30)$$

where

$$\gamma_{p,f+1}^{1/2}(n) = c_{f,N+1}(n)\gamma_{p,f}^{1/2}(n) \quad (31)$$

is the square root of the likelihood variable [2]. It can be similarly shown that an order-update recursion for the angle-normalized estimation error by using one additional *past* observation can be obtained to be

$$\varepsilon_{p+1,f}(n-f) = c_{b,N+1}(n)\varepsilon_{p,f}(n-f) - \lambda^{1/2}s_{b,N+1}(n)\rho_{p+1,f}^{B}(n-1) \quad (32)$$

$$\rho_{p+1,f}^{B}(n) = \lambda^{1/2}c_{b,N+1}(n)\rho_{p+1,f}^{B}(n-1) + s_{b,N+1}(n)\varepsilon_{p,f}(n-f) \quad (33)$$

where

$$c_{b,N+1}(n) = \frac{\lambda^{1/2}B_{N+1}^{1/2}(n-1)}{B_{N+1}^{1/2}(n)} \quad (34)$$

$$s_{b,N+1}(n) = \frac{e_{N+1}^{B}(n)}{B_{N+1}^{1/2}(n)} \quad (35)$$

and

$$B_{N+1}(n) = \lambda B_{N+1}(n-1) + \left(e_{N+1}^{B}(n)\right)^2 \quad (36)$$

Note that $e_{N+1}^{B}(n)$ is the angle-normalized backward prediction error of order (N+1). The a posteriori estimation error can also be computed by

$$e_{p+1,f}(n-f) = \gamma_{p+1,f}^{1/2}(n)\varepsilon_{p+1,f}(n-f) \quad (37)$$

where

$$\gamma_{p+1,f}^{1/2}(n) = c_{b,N+1}(n)\gamma_{p,f}^{1/2}(n) \quad (38)$$

Equations (28), (29), (30), (31), (32), (33), (37), and (38) constitute the *QRD-LSL smoothing algorithm.* A signal-flow graph of the QRD-LSL smoothing algorithm showing the $(2,2)^{th}$ order QRD-LSL smoother using the sequence BFBF is shown in Figure 1.

V. COMPUTER SIMULATIONS ON ADAPTIVE EQUALIZATION

In this section, we present results of computer simulations of adaptive equalization of a linear channel having unknown distortion. The simulation closely follow that of [16] and [4]. A polar form pseudo-random signal, x(n), is applied to a channel having unit pulse response:

$$h_n = \begin{cases} \frac{1}{2}\left[1 + \cos\left(\frac{2\pi}{W}(n-2)\right)\right], & n=1,2,3 \\ 0, & \text{otherwise} \end{cases} \quad (39)$$

The observation, y(n), is the sum of the channel output and an independent white Gaussian noise with variance 0.001. The adaptive equalizer attempts to correct the distortion produced by the channel and the additive noise. We compared the performances of three equalizers, each having order N=10 (11 taps). Equalizer #1 was a 10^{th} order QRD-LSL filter. Equalizer #2 was a 10^{th} order QRD-LSL filter with 5 units time delay (i.e., 5 "future" observations were used) of the type described in [16]. Equalizer #2 would have possessed the order-recursive property were it not for the 5 units of delay. As noted earlier, once delay is introduced into a QRD-LSL filter, the order-recursive property is lost. Equalizer #3 was a $(5,5)^{th}$ order QRD-LSL smoother possessing the order-recursive property of the type described in this paper. Of the $C_{10}^{5}=252$ possible realizations of a $(5,5)^{th}$ order QRD-LSL smoother, we used the sequencing BFBFBFBFBF. The parameter W in (39) was set equal to 3.5 to provide for eigenvalue spreads S = 46.82.

The learning curves for the three equalizers are shown in Figure 2. Each learning curve was obtained by ensemble-averaging the squared value of the a posteriori error over 200 independent trials of the experiment. It can be seen from the plots that the steady-state mean squared error of noncausal filters including the smoother and the filter with delay is about 15 dB less than that of a causal filter. It can also be seen that the rate of convergence of the $(5,5)^{th}$ order QRD-LSL smoother is a little faster than that of the 10^{th} order filter with delay. Additional realizations including the sequencing BBBBBFFFFF and the sequencing FFFFFBBBBB were tried. The simulation results revealed that the alternating sequence BFBFBFBFBF (or FBFBFBFBFB) and equalizer #2 displayed the fastest and slowest initial transient performance respectively compared to other sequences of the $(5,5)^{th}$-order smoother although their differences were not large. We conjecture that this is because signal autocorrelation functions are typically monotonically decreasing. And the two alternating sequences which use the present observation first immediately followed by the use of its nearest neighboring observations to estimate the present desired signal make earlier use of the correlation between the desired signal and the corresponding observations than other sequences. On the other hand, the realization of equalizer #2 (a 10^{th}-order filter with 5 units time delay) that actually corresponds to the BBBBBBBBBB sequence with 5 units time delay introduced to the desired signal x(n) makes the latest use of the correlation among all possible sequences.

VI. CONCLUSIONS

This work extends the QRD-LSL algorithm from filtering to smoothing with identical computational cost. All the desirable features found in the QRD-LSL filtering algorithm are also shared by the QRD-LSL smoothing algorithm with a finite delay but with a significant reduction in minimum mean square error. Computer simulations show that except for an overall delay needed for physical realization, QRD-LSL smoothers can substantially outperform conventional QRD-LSL filters in adaptive equalization while retaining all the advantages of an order-recursive structure.

REFERENCES

[1] M. Morf and D. T. Lee, "Recursive Least Squares Ladder Forms for Fast Parameter Tracking," Proc. 1978 IEEE Conf. Decision and Control, SanDiego, Calif., pp. 1362-1367.

[2] D. T. Lee, M. Morf, and B. Friedlander, "Recursive Least-Squares Ladder Estimation Algorithms," IEEE Trans. on Circuits and Systems, Vol. CAS-28, pp. 467-481, 1981.

[3] H. Lev-Ari, T. Kailath, and J. Cioffi, "Least-Squares Adaptive Lattice and Transeveral Filters: A Unified Geometric Theory," IEEE Trans. Information Theory, Vol. IT-30, pp.222-236, 1984.

[4] S. Haykin, "Adaptive Filter Theory," Englewood Cliffs, New Jersey: Prentice-Hall Inc., 1996, 3rd ed.

[5] J. T. Yuan, and John A. Stuller, "Least Squares Order-Recursive Lattice Smoothers," IEEE Trans. on Signal Processing, Vol 43, No.5, May 1995, pp.1058 - 1067.

[6] C. R. Ward, P. J. Hargrave, and J. G. McWhirter, "A novel algorithm and architecture for adaptive digital beamforming," IEEE Trans. on Antennas and Propagation, Vol. AP-34, No. 3, March 1986, pp. 338-346.

[7] M. D. Levin, and C. F. N. Cowan, "The Performance of Eight Recursive Least Squares Adaptive Filtering Algorithms in a Limited Precision Environment," 1994 Proc. European Signal Process. Conf., Edinburgh, Scotland, pp. 1261-1264.

[8] I. K. Proudler, J. G. McWhirter and T. J. Shepherd, "QRD-based lattice filter algorithms," Proc. SPIE, Vol. 1152 Advanced Algorithms and Architecture for Signal Processing IV, 1989.

[9] I. K. Proudler, J. G. McWhirter, and T. J. Shepherd, "The QRD-based Least Squares Lattice Algorithm for Wide-band Beamforming and Adaptive Filtering," Proc. ISSPA-90, Brisbane, Australia, 27th-31st August 1990.

[10] P. A. Regalia, M. G. Bellanger, "On the Duality Between Fast QR Methods and Lattice Methods in Least Squares Adaptive Filtering," IEEE Trans. on Signal Processing, Vol 39, No.4, April 1991, pp.879-891.

[11] I. K. Proudler, J. G. McWhirter and T. J. Shepherd, "QRD-Based Lattice Filter Algorithms," SPIE, 1989, San Diego, Calif.

[12] I. K. Proudler, J. G. McWhirter and T. J. Shepherd, "Computationally efficient QR decomposition approach to least squares adaptive filtering," IEE Proceedings-F, Vol. 138, No. 4, August 1991.

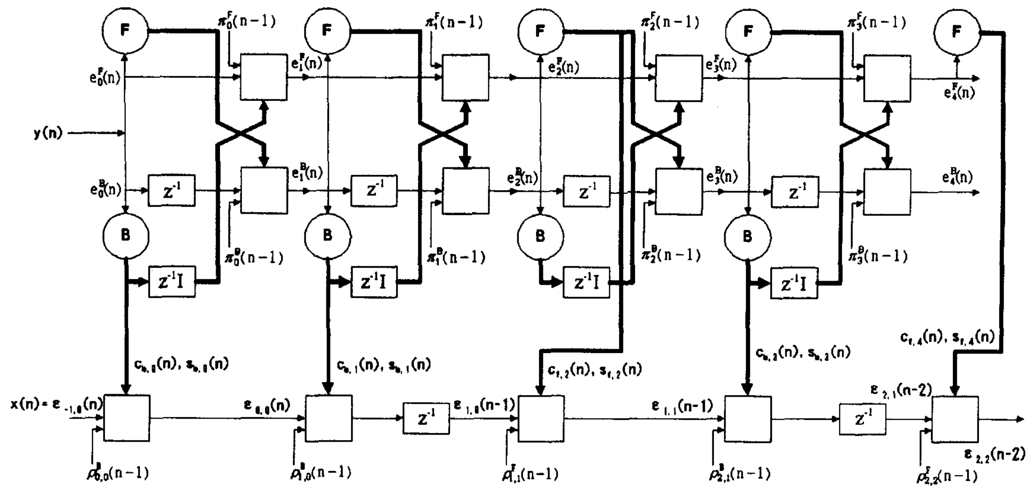
[13] B. Yang, and J. F. Bohme, "Rotation-based RLS Algorithms: Unified Derivations, Numerical Properties and Parallel Implementations," IEEE Trans. Signal Process., Vol. 40, pp. 1151-1167, 1992.

[14] Peter Strobach, "Linear Prediction Theory - A Mathematical Basis for Adaptive Systems," Springer - Verlag, 1990.

[15] J. T. Yuan, "QR-Decomposition for Noncausal Filtering – Part I: A Modified QR-Decomposition," Proceedings of IASTED International Conference on Modelling and Simulation - Pittsburgh, May 14-17, 1997, pp. 349 - 353.

[16] E. H. Satorius, and J. D. Pack, "Applications of Least Squares Lattice Algorithms to Adaptive Equalization," IEEE Trans. on Communications, Vol. COM-29, NO.2, Feb. 1981, pp.136-142.

[17] J. G. McWhirter, "Recursive least squares minimization using a systolic array," Proc. SPIE Int. Soc. Opt. Eng.. vol. 431, pp. 105-112, 1983.



(F): The computation of the forward cosine-sine pair of rotation parameters $c_{f,m}(n)$ and $s_{f,m}(n)$

(B): The computation of the backward cosine-sine pair of rotation parameters $c_{b,m}(n)$ and $s_{b,m}(n)$

Figure 1. Signal-flow graph of the (2,2)-th order QRD-LSL smoother using the sequence BFBF

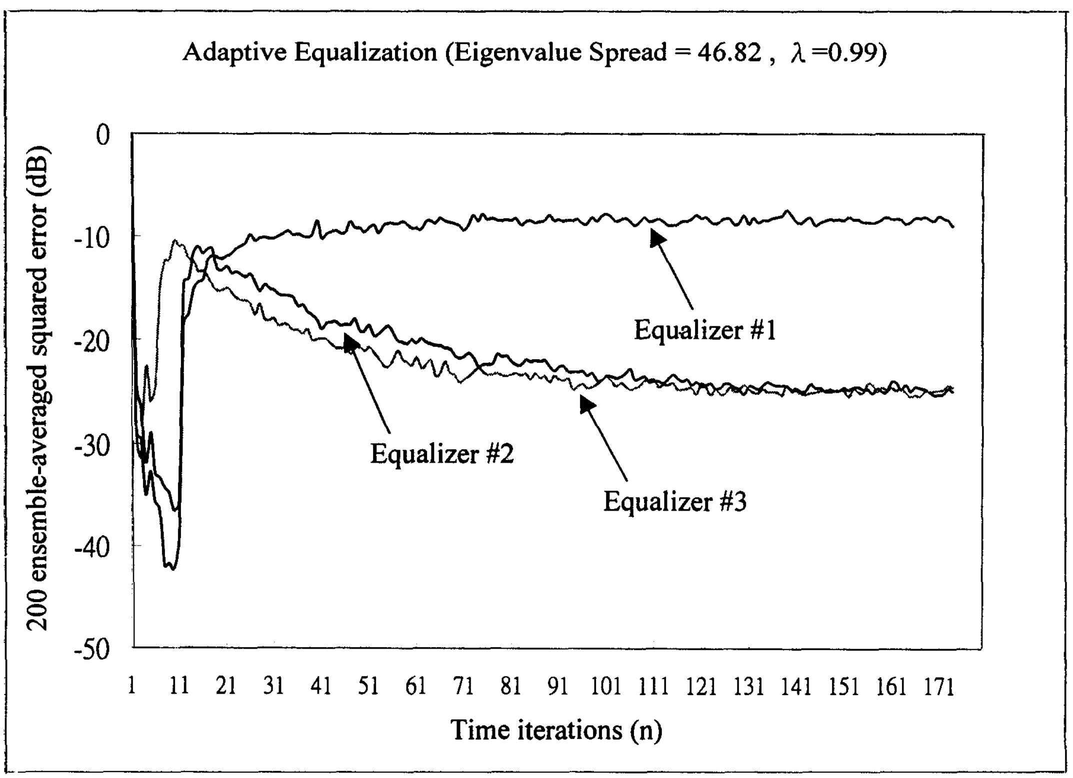


Figure 2. Learning curves for the three equalizers (Eigenvalue spread = 46.82)