QRD-BASED LSL INTERPOLATORS – PART II: A QRD-BASED LSL INTERPOLATION ALGORITHM

Ying-Wen Bai
Department of Electronic Engineering, Fu Jen Catholic University, 24205 Taipei, Taiwan, R.O.C.
email: bai@mars.ee.fju.edu.tw
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

In this paper, we derive a QRD-LSL interpolation algorithm that can be used to construct order-recursive QRD-LSL interpolators based on the exact decoupling property developed in a companion paper. QRD-LSL predictors are well known and use past data samples to predict the present data sample while the QRD-LSL interpolators use both past and future data samples to estimate the present data sample. Except for an overall delay needed for physical realization, QRD-LSL interpolators may achieve much better performance than that of the QRD-LSL predictors.

1. INTRODUCTION

It is widely known that the QRD-based LSL algorithm is endowed with a highly desirable set of features including a fast rate of convergence, excellent numerical properties, latticelike structure, modularity, and a high level of computational efficiency [1][2][3]. As a result, the performance of the QRD-LSL algorithm in a limited-precision environment is always superior to that of recursive LSL algorithms [3]. The recursive LSL interpolator was developed in [4][5] and can only be constructed by first solving the linear prediction coefficients of all orders. Consequently $O(N^2)$ operations are required, where N is the dimension of the interpolation filtering problem. More importantly, the effect of robustness to round-off error is largely diminished due to the use of the prediction coefficients [6]. In this paper, we develop a QRD-LSL interpolation algorithm that can be used to construct QRD-LSL interpolators based on the exact decoupling property introduced in [7]. The QRD-LSL interpolation algorithm requires only the computation of forward and backward prediction errors of all orders. The forward and backward prediction errors have a smaller dynamic range than the prediction coefficients and are directly accessible from a prediction error lattice. Consequently, the coefficients of the QRD-LSL interpolators are less sensitive to round-off noise than those of the recursive LSL interpolators developed in [4][5] and the computational load required by the QRD-LSL interpolators only needs O(N) operations.

In this paper, matrices and column vectors are set in uppercase boldface type and lowercase boldface type respectively, and scalars appear in plain text type. Dimensions of matrices and vectors appear as a subscript. As an example, $\mathbf{R}_{f\times p}(\mathbf{n})$ and $\mathbf{p}_f(\mathbf{n})$ denote the $f \times p$ matrix and $f \times 1$ vector respectively. $\mathbf{O}_{f\times p}$ and $\mathbf{o}_p$ denote a $f \times p$ null matrix and a column vector of p zeros respectively. The subscript of scalars represent the order. For example, $e^I_{p,f}(n)$ is the $(p,f)^{th}$ order interpolation error at time n.

2. MODIFIED QRD FOR INTERPOLATION

It can be shown that an n-by-n orthogonal matrix, $\overline{\mathbf{Q}(\mathbf{n})}$, can always be constructed from one of the C^f_N orthogonal basis sets of LSL interpolators [7] such that it rotates the data matrix $\mathbf{Y}_{N+1}(\mathbf{n})$ (see (3) of [7]) into the (N+1)-by-(N+1) matrix $\mathbf{R}_{p,f}(\mathbf{n})$, that is,

$$\overline{\mathbf{Q}(\mathbf{n})}\,\mathbf{Y}_{N+1}(\mathbf{n}) = \begin{bmatrix} \mathbf{R}_{p,f}(\mathbf{n}) \\ \mathbf{O}_{(n-N-1)\times(N+1)} \end{bmatrix}, \qquad (1)$$

where the matrix $\mathbf{R}_{p,f}(\mathbf{n})$ may be expressed as $\mathbf{R}_{p,f}(\mathbf{n})$ =

$$\begin{bmatrix} F_N^{1/2}(n,n-f) & 0 & \cdots & 0 & \times & \times & & \cdots & \times \\ \times & \ddots & \ddots & \vdots & \vdots & \vdots & & \ddots & \vdots \\ \vdots & \ddots & F_4^{1/2}(n-f+2,n-f) & 0 & & & & & \\ \times & \cdots & \times & F_2^{1/2}(n-f+1,n-f) & \times & \times & & \cdots & \times \\ 0 & \cdots & & 0 & I_{p,f}^{1/2}(n-f) & 0 & & \cdots & 0 \\ \times & \cdots & & \times & \times & B_1^{1/2}(n-f,n-f) & \times & \cdots & \times \\ & & & & & 0 & B_3^{1/2}(n-f+1,n-f) & \ddots & \vdots \\ \vdots & \ddots & & \vdots & \vdots & \vdots & \ddots & \ddots & \times \\ \times & \cdots & & \times & \times & 0 & \cdots & 0 & B_{N-1}^{1/2}(n-f,n-f) \end{bmatrix} \qquad (2)$$

by choosing the BFBFBF... sequence with p = f. The diagonal elements in matrix $\mathbf{R}_{p,f}(\mathbf{n})$ are defined as follows:

$$I_{p,f}^{1/2}(n-f) = \left(\sum_{i=1}^{n} \left(e^I_{p,f}(i-f)\right)^2\right)^{1/2},$$

$$F_m^{1/2}(n-j,n-f) = \left(\sum_{i=1}^{n} \left(e^F_m(i-j,i-f)\right)^2\right)^{1/2},$$

$B_m^{1/2}(n-j,n-f) = \left(\sum_{i=1}^{n} (e_m^B(i-j,i-f))^2\right)^{1/2}$. The symbol $\times$ in (2) denotes either a zero or a nonzero element whose value is not of direct interest. We refer to the result in (1) as the *modified QR-decomposition for interpolation* and refer to the form in $\mathbf{R}_{p,f}(\mathbf{n})$ as the *lower/upper (LU) triangular form* for a $(p,f)^{th}$ order interpolator based on the QRD. A derivation of (1) and (2) is not included in this paper due to space limitation. Since there are C_N^f possible sequences that can be used, the matrix $\mathbf{R}_{p,f}(\mathbf{n})$ can display C_N^f different forms all of which, however, contain one f-by-f lower triangular submatrix and one p-by-p upper triangular submatrix denoted by $\mathbf{L}_{f\times f}(\mathbf{n})$ and $\mathbf{U}_{p\times p}(\mathbf{n})$ respectively with zero elements filling the $(f+1)^{st}$ row except for the $(f+1,f+1)^{th}$ element as shown in (2). For example, when p = f = 2, we then have $\overline{\mathbf{Q}(\mathbf{n})}\mathbf{Y}_5(\mathbf{n}) = \begin{bmatrix} \mathbf{R}_{2,2}(\mathbf{n}) \\ \mathbf{O}_{(n-5)\times 5} \end{bmatrix}$, where $\mathbf{R}_{2,2}(\mathbf{n}) =$

$$\begin{bmatrix} F_4^{1/2}(n,n-2) & 0 & \times & 0 & 0 \\ \times & F_2^{1/2}(n-1,n-2) & \times & 0 & \times \\ 0 & 0 & I_{2,2}^{1/2}(n-2) & 0 & 0 \\ \times & \times & \times & B_1^{1/2}(n-2,n-2) & \times \\ \times & 0 & \times & 0 & B_3^{1/2}(n-1,n-2) \end{bmatrix} \quad (3)$$

3. QRD-LSL INTERPOLATION ALGORITHM

In this section we formulate a computationally efficient as well as numerically stable algorithm for solving the linear least-squares lattice interpolation problem within the framework of the modified QR-decomposition for interpolation. In this algorithm, order-update as well as time-update recursions for obtaining the interpolation errors as one additional future data and one additional past data are taken into account respectively to estimate the present data are derived. This may be accomplished by applying a sequence of Givens rotations directly to the result of the modified QR-decomposition for interpolation discussed in the previous section. The derivation will be proceeded in the following six stages. In each stage, either a single Givens rotation or a sequence of Givens rotations is applied.

1. For convenience, we may let n = n-2 in (3) of [7] and multiply both sides of (3) of [7] by $\Lambda^{1/2}(\mathbf{n}-2)$, where $\Lambda(\mathbf{n}-2) = \mathrm{diag}[\lambda^{n-3}, \lambda^{n-4}, \ldots, 1]$. This yields

$$\Lambda^{1/2}(\mathbf{n}-2)\mathbf{e}_{p,f}^{I}(\mathbf{n}-\mathbf{f}-2) = \Lambda^{1/2}(\mathbf{n}-2)\mathbf{Y}_{N+1}(\mathbf{n}-2)\mathbf{b}_{p,f}(\mathbf{n}-\mathbf{f}-2). \quad (4)$$

We then apply a sequence of N Givens rotations that define the (n-2)-by-(n-2) orthogonal matrix $\overline{\mathbf{Q}(\mathbf{n}-2)}$ to transform $\mathbf{Y}_{N+1}(\mathbf{n-2})$ into matrix

$$\mathbf{R}_{p,f}(\mathbf{n}-2) = \begin{bmatrix} \mathbf{L}_{f\times f}(\mathbf{n}-2) & \mathbf{p}_f(\mathbf{n}-2) & \mathbf{R}_{f\times p}(\mathbf{n}-2) \\ \mathbf{o}_f^T & I_{p,f}^{1/2}(n-f-2) & \mathbf{o}_p^T \\ \mathbf{R}_{p\times f}(\mathbf{n}-2) & \mathbf{p}_p(\mathbf{n}-2) & \mathbf{U}_{p\times p}(\mathbf{n}-2) \end{bmatrix}$$

that is in the LU triangular form for a $(p,f)^{th}$ order interpolator as shown in (2) with y(n-2) being the most recent data sample used. According to (1) and (4), we may thus write

$$\overline{\mathbf{Q}(\mathbf{n}-2)}\Lambda^{1/2}(\mathbf{n}-2)\mathbf{e}_{p,f}^{I}(\mathbf{n}-\mathbf{f}-2) = \Lambda^{1/2}(\mathbf{n}-2)\begin{bmatrix} \mathbf{R}_{p,f}(\mathbf{n}-2) \\ \mathbf{O}_{(n-N-3)\times(N+1)} \end{bmatrix}\mathbf{b}_{p,f}(\mathbf{n}-\mathbf{f}-2) \quad (5)$$

Note the matrix $\mathbf{R}_{p,f}(\mathbf{n-2})$ can automatically generate the optimum interpolation coefficients in $\mathbf{b}_{p,f}(\mathbf{n-f-2})$ by using the back substitution.

2. If we also apply the transformation produced by $\overline{\mathbf{Q}(\mathbf{n}-2)}$ to the 1-by-(n-1) data vector $\mathbf{y}^T(\mathbf{n-1}) = [y(1)\ y(2)\ \ldots\ y(n-1)]$ and 1-by-(n-2) data vector $\mathbf{y}^T(\mathbf{n-N-3}) = [0\ \ldots\ 0\ y(1)\ \ldots\ y(n-N-3)]$, which are the data vectors containing next future and next past data samples respectively to be included in order to obtain the order-updated recursion for the interpolation error, we thus find

$$\begin{bmatrix} 1 & \\ & \overline{\mathbf{Q}(\mathbf{n}-2)} \end{bmatrix}\Lambda^{1/2}(\mathbf{n}-1)\ \mathbf{y}(\mathbf{n}-1) = \left[\lambda^{(n-2)/2}y(1), \mathbf{p}_f^{F'}(\mathbf{n}-1), \Delta_{N+1}^{F'}(n-1), \mathbf{p}_p^{F'}(\mathbf{n}-1), \mathbf{v}_{n-N-3}^{F'}(\mathbf{n}-1)\right]^T \quad (6)$$

and

$$\overline{\mathbf{Q}(\mathbf{n}-2)}\ \Lambda^{1/2}(\mathbf{n}-2)\ \mathbf{y}(\mathbf{n}-\mathbf{N}-3) = \left[\mathbf{p}_f^{B'}(\mathbf{n}-2), \Delta_{N+1}^{B'}(n-2), \mathbf{p}_p^{B'}(\mathbf{n}-2), \mathbf{v}_{n-N-3}^{B'}(\mathbf{n}-2)\right]^T, \quad (7)$$

where both $\Delta_{N+1}^{F'}(n-1)$ and $\Delta_{N+1}^{B'}(n-2)$ are auxiliary parameters which will be used later to obtain the intermediate prediction errors. By appending both transformed data vectors obtained in (6) and (7), respectively, as the leftmost column and rightmost column of the matrix $\Lambda^{1/2}(\mathbf{n}-2)\begin{bmatrix} \mathbf{R}_{p,f}(\mathbf{n}-2) \\ \mathbf{O}_{(n-N-3)\times(N+1)} \end{bmatrix}$ in (5), together with the new data sample vector for time n at the bottom row, we obtain an expanded matrix $\mathbf{D}(\mathbf{n})$:

$$\mathbf{D}(\mathbf{n}) = \begin{bmatrix} \lambda^{(n-1)/2}y(1) & \mathbf{o}_f^T & 0 & \mathbf{o}_p^T & 0 \\ \lambda^{1/2}\mathbf{p}_f^{F'}(\mathbf{n}-1) & \lambda^{1/2}\mathbf{L}_{f\times f}(\mathbf{n}-2) & \lambda^{1/2}\mathbf{p}_f(\mathbf{n}-2) & \lambda^{1/2}\mathbf{R}_{f\times p}(\mathbf{n}-2) & \lambda^{1/2}\mathbf{p}_f^{B'}(\mathbf{n}-2) \\ \lambda^{1/2}\Delta_{N+1}^{F'}(n-1) & \mathbf{o}_f^T & \lambda^{1/2}I_{p,f}^{1/2}(n-f-2) & \mathbf{o}_p^T & \lambda^{1/2}\Delta_{N+1}^{B}(n-2) \\ \lambda^{1/2}\mathbf{p}_p^{F'}(\mathbf{n}-1) & \lambda^{1/2}\mathbf{R}_{p\times f}(\mathbf{n}-2) & \lambda^{1/2}\mathbf{p}_p(\mathbf{n}-2) & \lambda^{1/2}\mathbf{U}_{p\times p}(\mathbf{n}-2) & \lambda^{1/2}\mathbf{p}_p^{B'}(\mathbf{n}-2) \\ \lambda^{1/2}\mathbf{v}_{n-N-3}^{F'}(\mathbf{n}-1) & \mathbf{O}_{(n-N-3)\times f} & \mathbf{0}_{n-N-3} & \mathbf{O}_{(n-N-3)\times p} & \lambda^{1/2}\mathbf{v}_{n-N-3}^{B'}(\mathbf{n}-2) \\ y(n) & y(n-1)\ldots & y(n-f-1) & \ldots & y(n-N-2) \end{bmatrix} \quad (8)$$

3. We then apply a sequence of Givens rotations to annihilate all the elements in the bottom row of the matrix $\mathbf{D}(\mathbf{n})$ except for the (n,1), (n,f+2), and (n,N+3) elements. These rotations include a combination of a sequence of f Givens rotations (for "F"s) proceeding leftwards from the element (n,f+1) to the element (n,2) and a sequence of p

Givens rotations (for "B"s) proceeding rightwards from the element (n,f+3) to the element (n,N+2) with the order of annihilating the elements being performed in accordance with the sequencing chosen (e.g., BFBFBF...) to preserve the LU triangular form for an interpolator. For example, if the sequence BFBFBF... is chosen, then the elements in the following order: (n,f+3), (n,f+1), (n,f+4), (n,f), (n,f+5), (n,f-1),... will be annihilated successively. Such a sequence of N Givens rotations defines the n-by-n orthogonal matrix $\mathbf{L(n)}$ that transforms the matrix $\mathbf{D(n)}$ to matrix $\mathbf{E(n)}$ (i.e., $\mathbf{L(n)D(n)} = \mathbf{E(n)}$). As a result of this transformation, only the following elements in matrix $\mathbf{D(n)}$ change: Those elements from the 2nd row to the (f+1)th row would be time-updated to be $\left[\mathbf{p}_f^{F'}(\mathbf{n})\ \mathbf{L}_{f\times f}(\mathbf{n}\text{-}1)\ \mathbf{p}_f(\mathbf{n}\text{-}1)\ \mathbf{R}_{f\times p}(\mathbf{n}\text{-}1)\ \mathbf{p}_f^{B'}(\mathbf{n}\text{-}1)\right]$; those elements from the (f+3)th row to (N+2)th row would be time-updated to be $\left[\mathbf{p}_p^{F'}(\mathbf{n})\ \mathbf{R}_{p\times f}(\mathbf{n}\text{-}1)\ \mathbf{p}_p(\mathbf{n}\text{-}1)\ \mathbf{U}_{p\times p}(\mathbf{n}\text{-}1)\ \mathbf{p}_p^{B'}(\mathbf{n}\text{-}1)\right]$, and the elements at the bottom row of the matrix $\mathbf{E(n)}$ can be shown to be $\left[e^{F}_{N+1}(n,n\text{-}f\text{-}1)\ \mathbf{o}_f^T\ e^{I}_{p,f}(n\text{-}f\text{-}1)\ \mathbf{o}_p^T\ e^{B}_{N+1}(n\text{-}1,n\text{-}f\text{-}1)\right]$.

4. A single Givens rotation that annihilates $e^{I}_{p,f}(n\text{-}f\text{-}1)$ at the bottom row of the matrix $\mathbf{E(n)}$ is applied such that the intermediate forward and backward prediction errors and the forward and backward prediction errors of order (N+1), respectively, can be related. We may thus write

$$\begin{bmatrix} I_{f+1} & & & \\ & c_{I,N}(n\text{-}1) & & s_{I,N}(n\text{-}1) \\ & & I_{n\text{-}f\text{-}2} & \\ & -s_{I,N}(n\text{-}1) & & c_{I,N}(n\text{-}1) \end{bmatrix} \mathbf{E(n)} = \mathbf{F(n)} \tag{9}$$

As a result of this transformation, the (f+2)th row of matrix $\mathbf{E(n)}$ becomes $\left[\Delta^{F'}_{N+1}(n)\ \mathbf{o}_f^T\ I^{1/2}_{p,f}(n\text{-}f\text{-}1)\ \mathbf{o}_p^T\ \Delta^{B'}_{N+1}(n\text{-}1)\right]$ and the bottom row of $\mathbf{E(n)}$ becomes $\left[e^{F}_{N+1}(n)\ \mathbf{o}_f^T\ 0\ \mathbf{o}_p^T\ e^{B}_{N+1}(n\text{-}1)\right]$. The rest of elements of $\mathbf{E(n)}$ remain the same. By using (9), the interpolation cosine and sine parameters can then be defined as

$$c_{I,N}(n\text{-}1) = \frac{\lambda^{1/2} I^{1/2}_{p,f}(n\text{-}f\text{-}2)}{I^{1/2}_{p,f}(n\text{-}f\text{-}1)} \tag{10}$$

$$s_{I,N}(n\text{-}1) = \frac{e^{I}_{p,f}(n\text{-}f\text{-}1)}{I^{1/2}_{p,f}(n\text{-}f\text{-}1)} \tag{11}$$

where $I_{p,f}(n\text{-}f\text{-}1) = \lambda I_{p,f}(n\text{-}f\text{-}2) + \left(e^{I}_{p,f}(n\text{-}f\text{-}1)\right)^2$ (12)

is the time-updated recursion for the minimum weighted sum of the interpolation error square. Again by using (9), we obtain the following relations:

$$e^{F'}_{N+1}(n,n\text{-}f\text{-}1) = \frac{e^{F}_{N+1}(n) + \lambda^{1/2} s_{I,N}(n\text{-}1)\Delta^{F'}_{N+1}(n\text{-}1)}{c_{I,N}(n\text{-}1)} \tag{13}$$

and

$$e^{B}_{N+1}(n,n\text{-}f) = \frac{e^{B}_{N+1}(n) + \lambda^{1/2} s_{I,N}(n)\Delta^{B'}_{N+1}(n\text{-}1)}{c_{I,N}(n)}, \tag{14}$$

where

$$\Delta^{F'}_{N+1}(n) = \frac{\lambda^{1/2}\Delta^{F'}_{N+1}(n\text{-}1) + s_{I,N}(n\text{-}1)e^{F}_{N+1}(n)}{c_{I,N}(n\text{-}1)} \tag{15}$$

and

$$\Delta^{B'}_{N+1}(n) = \frac{\lambda^{1/2}\Delta^{B'}_{N+1}(n\text{-}1) + s_{I,N}(n)e^{B}_{N+1}(n)}{c_{I,N}(n)} \tag{16}$$

Equations (13) and (14) that correspond to (8) and (7) of [7] respectively will be used to compute the order-updated recursions for the interpolation error in stage 6.

5. The order-update recursion for the interpolation error as one additional *future* data sample is used can be obtained by applying an n-by-n orthogonal matrix $\mathbf{P(n)}$ to matrix $\mathbf{E(n)}$, that is, $\mathbf{P(n)E(n)} = \mathbf{I(n)}$. The matrix $\mathbf{P(n)}$ represents the combined transformation produced by a sequence of (n-N-2) Givens rotations which has the following two effects: (a) annihilating all the (n-N-3) elements of vector $\lambda^{1/2}\mathbf{v}^{F'}_{n\text{-}N\text{-}3}(\mathbf{n}\text{-}1)$ in the first column of matrix $\mathbf{E(n)}$ and (b) rotating the element $\lambda^{1/2}\Delta^{F'}_{N+1}(n\text{-}1)$ in the first column of matrix $\mathbf{E(n)}$ into the (1,f+2) element of the matrix. As a result, the first row of matrix $\mathbf{E(n)}$ becomes $[\lambda^{1/2}F^{1/2}_{N+1}(n\text{-}1)\ \mathbf{o}_f^T\ \lambda^{1/2}\rho^{F}_{p,f+1}(n\text{-}1)\ \mathbf{o}_p^T\ \times]$, where $F^{1/2}_{N+1}(n\text{-}1)$ is the minimum sum of the forward prediction error square and $\rho^{F}_{p,f+1}(n\text{-}1)$ is an auxiliary parameter. Furthermore, the upper-left (N+2)×(N+2) submatrix of $\mathbf{I(n)}$ turns out to be in the LU triangular form for a (p,f+1)th order interpolator.

6. We then apply one single Givens rotation to $\mathbf{I(n)}$ so as to annihilate $e^{F}_{N+1}(n,n\text{-}f\text{-}1)$ at the bottom row of $\mathbf{I(n)}$ and obtain

$$\begin{bmatrix} c'_{F,N+1}(n) & & s'_{F,N+1}(n) \\ & I_{n\text{-}2} & \\ -s'_{F,N+1}(n) & & c'_{F,N+1}(n) \end{bmatrix} \mathbf{I(n)} = \mathbf{J(n)} \tag{17}$$

As a result of this transformation, only the first row and the bottom row of matrix $\mathbf{I(n)}$ would change and they become $\left[F^{1/2}_{N+1}(n,n\text{-}f\text{-}1)\ \mathbf{o}_f^T\ \rho^{F}_{p,f+1}(n)\ \mathbf{o}_p^T\ \times\right]$ and $\left[0\ \mathbf{o}_f^T\ e^{I}_{p,f+1}(n\text{-}f\text{-}1)\ \mathbf{o}_p^T\ \times\right]$ respectively, where × denotes an element whose value is not of direct interest to us. This transformation defines the following intermediate forward cosine and sine parameters:

$$c'_{F,N+1}(n) = \frac{\lambda^{1/2}F^{1/2}_{N+1}(n\text{-}1,n\text{-}f\text{-}2)}{F^{1/2}_{N+1}(n,n\text{-}f\text{-}1)} \tag{18}$$

$$s'_{F,N+1}(n) = \frac{e^{F}_{N+1}(n,n\text{-}f\text{-}1)}{F^{1/2}_{N+1}(n,n\text{-}f\text{-}1)} \tag{19}$$

where

$$F_{N+1}(n,n\text{-}f\text{-}1) = \lambda F_{N+1}(n\text{-}1,n\text{-}f\text{-}2) + \left(e^{F}_{N+1}(n,n\text{-}f\text{-}1)\right)^2 \tag{20}$$

is the time updated recursion for the minimum weighted sum of the intermediate forward prediction error square. Again by using (17), the order-update recursion for the interpolation error as one additional future data sample is used can then be obtained by

$$e^{I}_{p,f+1}(n-f-1) =$$
$$c'_{F,N+1}(n)e^{I}_{p,f}(n-f-1) - \lambda^{1/2}s'_{F,N+1}(n)\rho^{F}_{p,f+1}(n-1) \quad (21)$$

and the time update for computing its interpolation auxiliary parameter is

$$\rho^{F}_{p,f+1}(n) = \lambda^{1/2}c'_{F,N+1}(n)\rho^{F}_{p,f+1}(n-1) + s'_{F,N+1}(n)e^{I}_{p,f}(n-f-1) \quad (22)$$

The order-update recursion for the interpolation error as one additional *past* data sample is used can be similarly obtained. We summarize the final results without deriving them:

$$e^{I}_{p+1,f}(n-f) = c'_{B,N+1}(n)e^{I}_{p,f}(n-f) - \lambda^{1/2}s'_{B,N+1}(n)\rho^{B}_{p+1,f}(n-1) \quad (23)$$
$$\rho^{B}_{p+1,f}(n) = \lambda^{1/2}c'_{B,N+1}(n)\rho^{B}_{p+1,f}(n-1) + s'_{B,N+1}(n)e^{I}_{p,f}(n-f) \quad (24)$$

where

$$c'_{B,N+1}(n) = \frac{\lambda^{1/2}B^{1/2}_{N+1}(n-1,n-f-1)}{B^{1/2}_{N+1}(n,n-f)} \quad (25)$$
$$s'_{B,N+1}(n) = \frac{e^{B}_{N+1}(n,n-f)}{B^{1/2}_{N+1}(n,n-f)} \quad (26)$$

which defines the intermediate backward cosine and sine parameters by using

$$B_{N+1}(n,n-f) = \lambda B_{N+1}(n-1,n-f-1) + \left(e^{B}_{N+1}(n,n-f)\right)^2 \quad (27)$$

Equations (10) – (16), and (18) – (27) constitute the *QRD-LSL interpolation algorithm*. To initialize the algorithm, at time n=0 we set $\Delta^{F'}_{m}(0) = \Delta^{B'}_{m}(0) = 0$ for all m and $\rho^{F}_{p,f}(0) = \rho^{B}_{p,f}(0) = 0$ for all p and f. Note that the total number of computations needed for the computation of a $(p,f)^{th}$ order interpolation error is proportional to O(N) including the computation of the prediction errors, where p+f = N. This is in contrast to the $O(N^2)$ computations needed by the algorithms developed in [4][5].

4. COMPUTER SIMULATIONS

In this section we describe a computer simulation experiment which compares the performances of the QRD-LSL prediction and QRD-LSL interpolation using the QRD-LSL interpolation algorithm. For purposes of comparison, we use an AR(2) data process defined as $y(n) + a_1\, y(n-1) + a_2\, y(n-2) = \varepsilon(n)$, where the driving process, $\varepsilon(n)$, is a computer generated sequence which simulates a zero-mean Gaussian white noise process with variance σ^2_{ε}. The symbol < > is used to denote a 200 sample ensemble average. For convenience, the AR parameter values are from [3, p.351]. The results of the simulations are presented in Figure 1 corresponding to an eigenvalue spread of 100 using a $\log_{10}$ scale. In this figure, the ensemble-averaged squared errors for $e^{F}_{2}(n)$ and $e^{F}_{4}(n)$ for $0 \le n \le 151$ and $e^{I}_{2,2}(n-2)$ for $2 \le n \le 153$ were computed over 200 trials each. Each trial used an independent realization of the white-noise process $\varepsilon(n)$. Figure 1 clearly shows that interpolation yields smaller steady-state values of the average squared error (about 7 dB) than prediction does.

5. CONCLUSIONS

This paper shows that the QRD-LSL algorithm that combines the good numerical properties of QR-decomposition and the desirable features of a lattice structure can be extended from linear prediction to linear interpolation. The simulation results reveal that the QRD-LSL interpolation, which makes better use of the correlation between the nearest neighboring data samples than the QRD-LSL prediction, may achieve much better performance than that of the QRD-LSL prediction.

REFERENCES

[1] 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.

[2] 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.

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

[4] C. K. Coursey, and J. A. Stuller, "Interpolation Lattice Filter," IEEE Trans. on Acoustics, Speech, and Signal Processing, Vol. ASSP-39, No. 4, pp. 965-967, April 1991.

[5] J. T. Yuan, "Asymmetric Interpolation Lattice," IEEE Trans. on Signal Processing, Vol 44, No.5, May 1996, pp.1256-1261.

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

[7] J. T. Yuan, "QRD-Based LSL Interpolators – Part I: An Exact Decoupling Property of the LSL Interpolator," Proceedings of IEEE 1998 International Conference on Acoustics, Speech and Signal Processing (ICASSP-98).

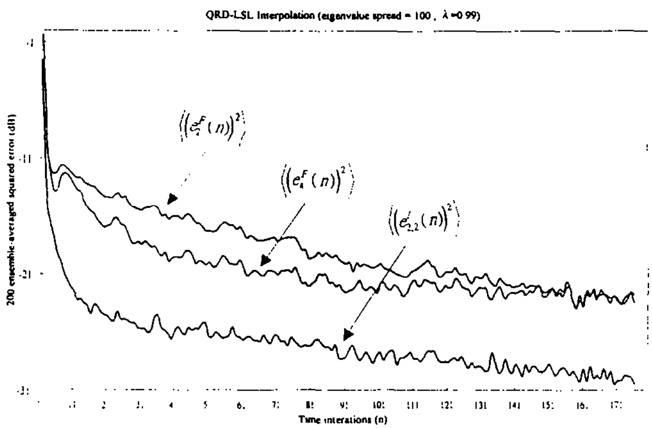

Figure 1 Learning curves of the QRD-LSL algorithm for linear predictiors and interpolator