QR-Decomposition-Based Least-Squares Lattice Interpolators

Jenq-Tay Yuan, *Member, IEEE*

***Abstract*—QR-decomposition-based least-squares lattice (QRD-LSL) predictors have been extensively applied in the design of order-recursive adaptive filters. This work presents QRD-LSL interpolators that use both past and future data samples to estimate the present data sample based on a novel decoupling property. We show that the order-recursive QRD-LSL interpolators display better numerical properties and achieve a higher level of computational efficiency than the conventional LSL interpolators. Except for an overall delay needed for physical realization, QRD-LSL interpolators can substantially outperform QRD-LSL predictors while retaining all the desirable features of the latter.**

***Index Terms*—Adaptive filtering, interpolation, least-squares lattice, QR-decomposition.**

I. Introduction

LINEAR interpolation estimates an unknown data sample based on a weighted sum of the surrounding data samples [1]. In the special case of one-dimensional (1-D) signal processing, linear interpolation of order $(p,\ f)$ estimates the present data sample of an input data sequence by forming a weighted sum of the p past and f future data samples [2]. Clearly, interpolation differs from prediction in that such an estimation considers the subspace of past data samples and the subspace of future data samples. Therefore, as expected, the interpolation process is more accurate than the prediction process. Previous investigations [1]–[4] have confirmed the ability of linear interpolation to perform much better than linear prediction in terms of minimum mean squared error. Moreover, linear interpolators perform better than their prediction counterparts in applications such as data compression, speech (or image) coding and restoration, and spread spectrum communications [5]–[12].

The *lattice structure* [13]–[20] has the merits of modularity, robustness and better numerical conditioning than the transversal structure. A lattice predictor (i.e., the lattice structure of a prediction-error filter) can be constructed in two steps: a) successively construct an *orthogonal basis* of the subspace of past data samples, and b) project the present data vector successively onto the orthogonal basis vectors. Successive stages of the lattice predictor are *decoupled* since projections are performed onto orthogonal basis vectors. Haykin [14, p. 651] referred to the uncorrelated least-squares backward prediction errors produced at various stages of the lattice predictor as the exact decoupling property of the least-squares lattice (LSL) predictor. The decoupling property, however, is confined to causal data only. Adaptively processing both past and *future* data samples requires a broader decoupling property. In this work, we present a novel *exact decoupling property for LSL interpolation*. The exact decoupling property proposed herein includes the exact decoupling property of the LSL predictor as a special case and provides the general framework for the design of fast *order-recursive LSL interpolators*.

Previous works on LSL interpolators [1], [2] have been developed by directly inverting the correlation matrix of the input data and can only be implemented by using the coefficients (or tap weights) of a transversal predictor. Consequently, $O(N^2)$ operations are required, where N denotes the dimension of the interpolation problem. Moreover, when finite precision arithmetic is used, the effect of robustness of these lattice interpolators toward round-off noise is reduced due to the use of the transversal predictor coefficients [13]. This work develops an order-recursive QR-decomposition-based least-squares lattice (QRD-LSL) interpolator by combining the proposed exact decoupling property with the QR-decomposition technique. Implementing the QRD-LSL interpolator merely involves using both forward and backward prediction errors produced at the various stages of a QRD-LSL predictor. Consequently, its total computational complexity scales linearly with the interpolation order N. Furthermore, when finite precision arithmetic is used, simulation results in Section IV verify that the QRD-LSL interpolators are more numerically robust than the conventional LSL interpolators.

In this paper, matrices are set in uppercase boldface type and column vectors in lowercase boldface type, whereas scalars appear in plain text type. Dimensions of matrices and vectors appear as subscripts. For example, $\boldsymbol{O}_{f\times p}$ and $\boldsymbol{o}_p$ represent a $f \times p$ null matrix and a column vector of p zeros, respectively.

The paper is organized as follows. An exact decoupling property of LSL interpolators is presented in Section II. A modified QR-decomposition for interpolation and a QRD-LSL interpolation algorithm are presented in Section III. In Section IV, simulation results are presented, and conclusions are drawn in Section V.

Manuscript received April 17, 1998; revised June 30, 1999. This work was supported by the National Science Council, Taiwan, R.O.C., under Contract NSC 88-2218-E-030-001. The associate editor coordinating the review of this paper and approving it for publication was Dr. Phillip A. Regalia.

The author is with the Department of Electronic Engineering, Fu Jen Catholic University, 24205 Taipei, Taiwan, R.O.C. (e-mail: yuan@mars.ee.fju.edu.tw).

Publisher Item Identifier S 1053-587X(00)00091-X.

II. LSL Interpolator and its Exact Decoupling Property

A. Least-Squares Interpolators of Order (p, f)

In (p, f)th-order linear interpolation, we linearly estimate the present input data sample $x(i)$ from its p past and f future neighboring data samples, viz.

$$\hat{x}_{p,f}(i) = -\sum_{\substack{k=-p \\ k\neq 0}}^{f} b_{(p,f),k}(n-f)x(i+k) \qquad 1-f \le i \le n-f \tag{1}$$

where $b_{(p,f),k}(n-f)$ is the interpolation coefficient at time n, which remains fixed during the observation interval $1-f \le i \le n-f$ with the prewindowing condition on the data (i.e., $x(i)=0$ for $i \le 0$). The length of the signal, n, is variable. The order, $N = p+f$. By using (1), the (p, f)th order interpolation error at each time unit can be defined as

$$\begin{aligned} e^I_{p,f}(i) &= x(i) - \hat{x}_{p,f}(i) \\ &= x(i) + \sum_{\substack{k=-p \\ k\neq 0}}^{f} b_{(p,f),k}(n-f)x(i+k), \\ & \qquad 1-f \le i \le n-f. \end{aligned} \tag{2}$$

Herein, we refer to any Nth-order interpolation filter that operates on the present data sample as well as p past and f future data samples to produce the (p, f)th-order interpolation error at its output as a *(p, f)th–order interpolator*, where $N = p+f$ is assumed implicitly. If the most recent data sample used is $x(n)$, then (2) can be written in a matrix form for $1-f \le i \le n-f$ as in (2a), shown at the bottom of the page, or, more compactly

$$\boldsymbol{e^I_{p,f}(n-f)} = \boldsymbol{x}(n-f) - \boldsymbol{X}_{p,f}(n)\boldsymbol{b}_{p,f}(n-f). \tag{3}$$

The least-squares solution to (3) can be determined as follows by using the projection approach [13], [15], [21], [22]. The projection operator that projects vectors onto the subspace spanned by the columns of $\boldsymbol{X}_{p,f}(n)$ (i.e., of p past data samples and f future data samples) can be expressed by

$$\boldsymbol{P}_X = \boldsymbol{X}_{p,f}(n)\left[\boldsymbol{X}^T_{p,f}(n)\boldsymbol{X}_{p,f}(n)\right]^{-1}\boldsymbol{X}^T_{p,f}(n). \tag{4}$$

Its orthogonal complement is denoted by $\boldsymbol{P}^{\perp}_X \triangleq (\boldsymbol{I} - \boldsymbol{P}_X)$, where $\boldsymbol{I}$ represents the identity matrix. The interpolation error vector of (3) can therefore be determined by $\boldsymbol{e^I_{p,f}(n-f)} = \boldsymbol{P}^{\perp}_X\boldsymbol{x}(n-f)$.

B. Order-Recursive LSL Interpolators

To develop order-recursive LSL interpolators, we first introduce a modified version of linear forward and backward predictions, which is referred to herein as *intermediate forward and backward predictions*, respectively, since they form a bridge between linear predictions and linear interpolations as will become apparent later (see (8) and (9)). Consider a sequence of n data samples $\{x(i)\}$, $1 \le i \le n$. Herein, $x(i)$ is referred to as the most recent data sample used for linear interpolation, and $x(i-f)$ is referred to as the present data sample that will be estimated by its p past and f future neighboring data samples, where $0 \le f \le N$ and $p = N - f$. The Nth-order intermediate forward predictor $\hat{x}_N(i, i-f)$is defined as the best possible predictor of $x(i)$ based on a weighted linear combination of the N previous data samples without, however, taking the present data sample $x(i-f)$ into consideration, viz.

$$\hat{x}_N(i, i-f) = -\sum_{\substack{k=1 \\ k\neq f}}^{N} a_{Nk}(n)x(i-k)$$

where $a_{Nk}(n)$, $k = 1, 2, \cdots, f-1, f+1, \cdots, N$ are Nth-order intermediate forward prediction coefficients that are

$$\begin{bmatrix} e^I_{p,f}(1-f) \\ \vdots \\ e^I_{p,f}(-1) \\ e^I_{p,f}(0) \\ e^I_{p,f}(1) \\ \vdots \\ e^I_{p,f}(n-f) \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ x(1) \\ \vdots \\ x(n-f-1) \\ x(n-f) \end{bmatrix} - \begin{bmatrix} x(1) & 0 & 0 & & 0 \\ x(2) & \vdots & \vdots & & 0 \\ x(3) & x(1) & x(1) & & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ x(n) & x(n-f+1) & x(n-f-1) & \cdots & x(n-N) \end{bmatrix} \cdot \begin{bmatrix} -b_{(p,f),f}(n-f) \\ -b_{(p,f),f-1}(n-f) \\ \vdots \\ -b_{(p,f),1}(n-f) \\ -b_{(p,f),-1}(n-f) \\ \vdots \\ -b(p,f),\ -p(n-f) \end{bmatrix} \tag{2a}$$

TABLE I
CHOICES FOR $\boldsymbol{y}$, $\boldsymbol{v}$, AND $\boldsymbol{w}$ FOR THE DERIVATION OF (6)–(9)

y	v	w	$\mathbf{w}^T\mathbf{P}^{\perp}_{[X\ Y]}\mathbf{v}$	$\mathbf{w}^T\mathbf{P}^{\perp}_{X}\mathbf{v}$	$\left[\mathbf{y}^T\mathbf{P}^{\perp}_{X}\mathbf{y}\right]^{-1}\mathbf{y}^T\mathbf{P}^{\perp}_{X}\mathbf{v}$	$\mathbf{w}^T\mathbf{P}^{\perp}_{X}\mathbf{y}$
$\mathbf{x}^B_n$	$\mathbf{x}(n-f)$	π_n	$e^I_{p+1,f}(n-f)$	$e^I_{p,f}(n-f)$	$k^B_{p+1,f}(n)$	$e^B_{N+1}(n,n-f)$
$\mathbf{x}^F_n$	$\mathbf{x}(n-f-1)$	π_n	$e^I_{p,f+1}(n-f-1)$	$e^I_{p,f}(n-f-1)$	$k^F_{p,f+1}(n)$	$e^F_{N+1}(n,n-f-1)$
$\mathbf{x}(n-f)$	$\mathbf{x}^B_n$	π_n	$e^B_{N+1}(n)$	$e^B_{N+1}(n,n-f)$	$l^B_{N+1}(n)$	$e^I_{p,f}(n-f)$
$\mathbf{x}(n-f-1)$	$\mathbf{x}^F_n$	π_n	$e^F_{N+1}(n)$	$e^F_{N+1}(n,n-f-1)$	$l^F_{N+1}(n)$	$e^I_{p,f}(n-f-1)$

optimized in the least-squares sense over the entire observation interval $1 \le i \le n$. The corresponding Nth-order intermediate forward prediction error can therefore be written as

$$e^F_N(i, i-f) = x(i) - \hat{x}(i, i-f) = x(i) + \sum_{\substack{k=1 \\ k\neq f}}^{N} a_{Nk}(n)x(i-k), \qquad 1 \le i \le n.$$

The Nth-order intermediate backward prediction error can similarly be found to be

$$e^B_N(i, i-f) = x(i-N) + \sum_{\substack{k=1 \\ k\neq p}}^{N} c_{Nk}(n)x(i+k-N)$$

where $c_{Nk}(n)$, $k = 1, 2, \cdots, p-1, p+1, \cdots, N$ are Nth–order intermediate backward prediction coefficients.

Next, the order-recursive LSL interpolator is developed by using a projection-operator update formula [15], [21], [23]

$$\boldsymbol{w}^T\boldsymbol{P}^{\perp}_{[XY]}\boldsymbol{v} = \boldsymbol{w}^T\boldsymbol{P}^{\perp}_{X}\boldsymbol{v} - \boldsymbol{w}^T\boldsymbol{P}^{\perp}_{X}\boldsymbol{y}\left[\left(\boldsymbol{y}^T\boldsymbol{P}^{\perp}_{X}\boldsymbol{y}\right)^{-1}\left(\boldsymbol{y}^T\boldsymbol{P}^{\perp}_{X}\boldsymbol{v}\right)\right] \tag{5}$$

where $\boldsymbol{y}$, $\boldsymbol{v}$, $\boldsymbol{w}$ are some vectors, and $[XY]$ denotes the subspace spanned by the columns of $\boldsymbol{X}_{p,f}(n)$ and $\boldsymbol{y}$ taken together. Using (5) with choices for $\boldsymbol{y}$, $\boldsymbol{v}$, $\boldsymbol{w}$ shown in the top two rows of Table I generates the following two equations that characterize the order-recursive LSL interpolator as an additional past data sample and an additional future data sample are taken into account, respectively, to estimate the present data sample:

$$e^I_{p+1,f}(n-f) = e^I_{p,f}(n-f) - k^B_{p+1,f}(n) \cdot e^B_{N+1}(n, n-f) \tag{6}$$

$$e^I_{p,f+1}(n-f-1) = e^I_{p,f}(n-f-1) - k^F_{p,f+1}(n) \cdot e^F_{N+1}(n, n-f-1) \tag{7}$$

where $\boldsymbol{x}^F_n = [x(1), x(2), \cdots, x(n)]^T$, $\boldsymbol{x}^B_n = [0, \cdots, 0, x(1), \cdots, x(n-N-1)]^T$, $\boldsymbol{\pi}_n = [0, \cdots, 0, 1]^T$, and $k^B_{p+1,f}(n)$ and $k^F_{p,f+1}(n)$ are the interpolator's scalar coefficients.

To construct an LSL interpolator of order (p, f), (6) and (7) must be applied p and f times, respectively. However, any sequencing between these two equations is permissible. Consequently, there are $C^p_N = C^f_N = N!/p!f!$ permissible realizations for an LSL interpolator of order (p, f). For example, to construct a (2,2)th-order interpolator, a total of six permissible realizations may be identified by the sequences BFBF, FBFB, BBFF, FFBB, FBBF, and BFFB of intermediate backward (B) and intermediate forward (F) prediction errors used in (6) and (7), respectively.

Equations (6) and (7) reveal that the intermediate forward and backward prediction errors are necessary to compute the order-updated interpolation error. These errors, however, can be computed by using (5) with choices for $\boldsymbol{y}$, $\boldsymbol{v}$, $\boldsymbol{w}$ shown in the last two rows of Table I

$$e^B_{N+1}(n, n-f) = e^B_{N+1}(n) + 1^B_{N+1}(n)e^I_{p,f} \cdot (n-f) \tag{8}$$

$$e^F_{N+1}(n, n-f-1) = e^F_{N+1}(n) + 1^F_{N+1}(n)e^I_{p,f} \cdot (n-f-1) \tag{9}$$

where $1^B_{N+1}(n)$ and $1^F_{N+1}(n)$ are scalar coefficients. Notably, $e^B_{N+1}(n)$and $e^F_{N+1}(n)$ are directly accessible from an $(N+1)$st-order LSL predictor that can be embedded into an LSL interpolator, and $e^I_{p,f}(n-f)$[or $e^I_{p,f}(n-f-1)$] is already computed from the previous interpolation lattice stage of the LSL interpolator. The relations shown in (8) and (9) connect together the linear predictions, intermediate predictions, and linear interpolations. In fact, as shown in Section III (see Table II), (6)–(9), together with the well-known LSL predictor [14], constitute an order-recursive *LSL interpolator*. The four equations can also be gathered as two occurrences of a biorthogonalization process [24].

C. Exact Decoupling Property of LSL Interpolators

When considering a sequence of p past and f future data samples to estimate the present data sample $x(n-f)$, appropriate combinations of f delayed intermediate forward prediction errors and p delayed intermediate backward prediction errors, followed by the interpolation error $e^I_{p,f}(n-f)$, form C^f_N sets of orthogonal bases. Each C^f_Northogonal basis set provides an orthogonal basis for the subspace $\boldsymbol{x}_{p,f}(n-f) = [x(n-N), \cdots, x(n-f-1), x(n-f), x(n-f+1), \cdots, x(n)]$ spanned by the present data sample $x(n-f)$, p past data samples, and f future data samples. The orthogonality among all the elements within each of these orthogonal bases is referred

TABLE II
SUMMARY OF THE QRD-LSL INTERPOLATION ALGORITHM

1. Computations

(I) Predictions: For time n = 1, 2,..., and prediction order m = 1, 2,..., N, where N is the final prediction order [14].

$$B_{m-1}(n-1) = \lambda B_{m-1}(n-2) + \left(e^{B}_{m-1}(n-1)\right)^2 \tag{14}$$

$$c_{B,m-1}(n-1) = \frac{\lambda^{1/2}\left(B_{m-1}(n-2)\right)^{1/2}}{\left(B_{m-1}(n-1)\right)^{1/2}} \tag{15}$$

$$s_{B,m-1}(n-1) = \frac{e^{B}_{m-1}(n-1)}{\left(B_{m-1}(n-1)\right)^{1/2}} \tag{16}$$

$$e^{F}_{m}(n) = c_{B,m-1}(n-1)e^{F}_{m-1}(n) - \lambda^{1/2}s_{B,m-1}(n-1)\pi^{F}_{m-1}(n-1) \tag{17}$$

$$\pi^{F}_{m-1}(n) = \lambda^{1/2}c_{B,m-1}(n-1)\pi^{F}_{m-1}(n-1) + s_{B,m-1}(n-1)e^{F}_{m-1}(n) \tag{18}$$

$$F_{m-1}(n) = \lambda F_{m-1}(n-1) + \left(e^{F}_{m-1}(n)\right)^2 \tag{19}$$

$$c_{F,m-1}(n) = \frac{\lambda^{1/2}\left(F_{m-1}(n-1)\right)^{1/2}}{\left(F_{m-1}(n)\right)^{1/2}} \tag{20}$$

$$s_{F,m-1}(n) = \frac{e^{F}_{m-1}(n)}{\left(F_{m-1}(n)\right)^{1/2}} \tag{21}$$

$$e^{B}_{m}(n) = c_{F,m-1}(n)e^{B}_{m-1}(n-1) - \lambda^{1/2}s_{F,m-1}(n)\pi^{B}_{m-1}(n-1) \tag{22}$$

$$\pi^{B}_{m-1}(n) = \lambda^{1/2}c_{F,m-1}(n)\pi^{B}_{m-1}(n-1) + s_{F,m-1}(n)e^{B}_{m-1}(n-1) \tag{23}$$

(II) Interpolations: For time n = 1, 2,..., start from f = 0 and p = 0. N = p + f. Additional p "past" and f "future" stages can be increased by computing one of C^{f}_{N} combinations of part (a) and part (b).

(a) An additional future data sample is taken into account:

$$I_{p,f}(n-f-1) = \lambda I_{p,f}(n-f-2) + \left(e^{I}_{p,f}(n-f-1)\right)^2 \tag{24}$$

to herein as the *exact decoupling property of the LSL interpolator*. The exact decoupling property of the LSL interpolator can be directly verified by using a simplified form of (5), which can be written as

$$\boldsymbol{P}^{\perp}_{[XY]}\boldsymbol{v} = \boldsymbol{P}^{\perp}_{X}\boldsymbol{v} - k(n)\left(\boldsymbol{P}^{\perp}_{X}\boldsymbol{y}\right) \tag{10}$$

where $k(n)$ is a scalar coefficient computed by using $[(\boldsymbol{y}^T\boldsymbol{P}^{\perp}_{X}\boldsymbol{y})^{-1}(\boldsymbol{y}^T\boldsymbol{P}^{\perp}_{X}\boldsymbol{v})]$ in (5). It is well known that $\boldsymbol{P}^{\perp}_{[XY]}\boldsymbol{v}$ and $\boldsymbol{P}^{\perp}_{X}\boldsymbol{y}$ are orthogonal due to the Gram–Schmidt orthogonalization procedure. For illustration, we consider the simple case of estimating the present data sample $x(n-2)$ from its two previous data samples and its two future data samples. Using the orthogonal property revealed in (10) recursively allows us to obtain the following orthogonal basis identified by BFBF: $[x(n-3), \boldsymbol{P}^{\perp}_{x(n-3)}x(n-1), \boldsymbol{P}^{\perp}_{[x(n-3)x(n-1)]}x(n-4), \boldsymbol{P}^{\perp}_{[x(n-3)x(n-1)x(n-4)]}x(n), \boldsymbol{P}^{\perp}_{[x(n-3)x(n-1)x(n-4)x(n)]}x(n-2)]$, which is equivalent to $[e^{B}_{1}(n-2, n-2), e^{F}_{2}(n-1, n-2), [e^{B}_{1}(n-2, n-2), e^{B}_{3}(n-1, n-2), e^{I}_{2,2}(n-2)]$. Owing to the arbitrary order in which the data samples are chosen, there are $C^{2}_{4} = 6$ different orthogonal bases. The other orthogonal bases are

$$[e^{F}_{1}(n-1, n-2), e^{B}_{2}(n-1, n-2), e^{F}_{3}(n, n-2)$$
$$e^{B}_{4}(n, n-2), e^{I}_{2,2}(n-2)]$$

$$[e_1^B(n-2,\,n-2),\,e_2^B(n-2,\,n-2),\,e_3^F(n-1,\,n-2)\;\;e_4^F(n,\,n-2),\,e_{2,2}^I(n-2)]$$
$$[e_1^F(n-1,\,n-2),\,e_2^F(n,\,n-2),\,e_3^B(n,\,n-2)\;\;e_4^B(n,\,n-2),\,e_{2,2}^I(n-2)]$$
$$[e_1^F(n-1,\,n-2),\,e_2^B(n-1,\,n-2),\,e_3^B(n-1,\,n-2)\;\;e_4^F(n,\,n-2),\,e_{2,2}^I(n-2)]$$
$$[e_1^B(n-2,\,n-2),\,e_2^F(n-1,\,n-2),\,e_3^F(n,\,n-2)\;\;e_4^B(n,\,n-2),\,e_{2,2}^I(n-2)].$$

Each orthogonal basis can be used to construct an *order-recursive* realization for a (2,2)th-order LSL interpolator (see Fig. 1). Notably, the well-known exact decoupling property of the LSL predictor [14] can be readily shown to be a special case of the exact decoupling property of the LSL interpolator by setting $(p,\,f)=(N,\,0)$ (i.e., an Nth-order LSL predictor).

III. Order-Recursive QRD-LSL Interpolators

A. A Modified QR-Decomposition for Interpolation

It can be shown that by using the exact decoupling property developed in Section II, an $n\times n$ orthogonal matrix $\overline{\boldsymbol{Q}(n)}$ can always be constructed from one of the C_N^f orthonormal basis sets, each providing an orthonormal basis for the subspace spanned by the columns of data matrix as shown in (11), shown at the bottom of the page, such that it applies a generalized orthogonal triangularization to $\boldsymbol{X}_{N+1}(n)$ as shown by

$$\overline{\boldsymbol{Q}(n)}\boldsymbol{X}_{N+1}(n)=\begin{bmatrix}\boldsymbol{Q}_{p,\,f}^T(n)\\ \boldsymbol{S}^T(n)\end{bmatrix}\boldsymbol{X}_{N+1}(n)=\begin{bmatrix}\boldsymbol{R}_{p,\,f}(n)\\ \boldsymbol{O}_{(n-N-1)\times(N+1)}\end{bmatrix}\tag{12}$$

where $\boldsymbol{Q}_{p,\,f}^T(n)$ contains the first $(N+1)$ rows of $\overline{\boldsymbol{Q}(\boldsymbol{n})}$, whereas $\boldsymbol{S}^T(n)$ contains the remaining rows. Since C_N^f possible sequences can be used, the matrix $\boldsymbol{R}_{p,\,f}(n)$ in (12) can display C_N^f different forms, all of which contain one $f\times f$ lower triangular matrix and one $p\times p$ upper triangular matrix with zero elements filling the $(f+1)$st row, except for the $(f+1,\,f+1)$th element, as shown in

$$\mathbf{R}_{p,f}(n)=\begin{bmatrix}\boxed{\begin{matrix}F_N^{1/2}(n,n-f) & 0 & \cdots & 0\\ \times & \ddots & \ddots & \vdots\\ \vdots & \ddots & F_4^{1/2}(n-f+2,n-f) & 0\\ \times & \cdots & \times & F_2^{1/2}(n-f+1,n-f)\end{matrix}} & \begin{matrix}\times & \times & \cdots & \times\\ \vdots & \vdots & \ddots & \vdots\\ \times & \times & \cdots & \times\end{matrix}\\ \begin{matrix}0 & \cdots & 0 & I_{p,f}^{1/2}(n-f)\\ \times & \cdots & \times & \times\\ \vdots & \ddots & \vdots & \vdots\\ \times & \cdots & \times & \times\end{matrix} & \boxed{\begin{matrix}B_1^{1/2}(n-f,n-f) & \times & \cdots & \times\\ 0 & B_3^{1/2}(n-f+1,n-f) & \ddots & \vdots\\ \vdots & \ddots & \ddots & \times\\ 0 & \cdots & 0 & B_{N-1}^{1/2}(n-1,n-f)\end{matrix}}\end{bmatrix}\tag{13}$$

More specifically, our results indicate that $\boldsymbol{R}_{p,\,f}(n)$ using the BFBFBF$\cdots$ sequence with $p=f$ can be shown in (13), where $F_m^{1/2}(n-j,\,n-f)$ and $B_m^{1/2}(n-j,\,n-f)$ are the square roots of the minimum sum of mth-order intermediate forward and backward prediction error squares, respectively, whereas $I_{p,\,f}^{1/2}(n-f)$ is the square root of the minimum sum of $(p,\,f)$th-order interpolation error square; the symbol $\times$ denotes either a zero or a nonzero element whose value is not of direct interest. Notably, the first f diagonal elements of the matrix $\boldsymbol{R}_{p,\,f}(n)$ give the square roots of the minimum sum of the intermediate forward prediction error squares corresponding to the f future data samples $x(n),\cdots,x(n-f+1)$, whereas the last p diagonal elements give the square roots of the minimum sum of the intermediate backward prediction error squares corresponding to the p past data samples $x(n-f-1),\cdots,x(n-N)$. The $(f+1,\,f+1)$th element that gives $I_{p,\,f}^{1/2}(n-f)$ corresponds to the present data sample $x(n-f)$ to be estimated. We refer to the result in (12) as the *modified QR-decomposition for interpolation* and refer to the form in $\boldsymbol{R}_{p,\,f}(n)$ of (13) as the *standard lower/upper (LU) triangular form for a $(p,\,f)$th order interpolator* based on the QR-decomposition.

B. QRD-LSL Interpolation Algorithm

We now formulate a computationally efficient as well as numerically stable order-recursive algorithm for solving the least-squares lattice interpolation problem within the framework of the modified QR-decomposition for interpolation. This formulation is accomplished by recursively updating the standard LU triangular form for an interpolator from order $(p,\,f)$ to order $(p,\,f+1)$ and order $(p+1,\,f)$, respectively, by a sequence of Givens rotations. Table II summarizes the QRD-LSL interpolation algorithm. The derivation of the algorithm is long and, therefore, is omitted herein. Notably, the total number of operations required by the proposed algorithm is approximately $40N$ per iteration, whereas the total

$$\boldsymbol{X}_{N+1}(n)=\begin{bmatrix}x(1) & 0 & & 0 & & 0\\ x(2) & x(1) & & \vdots & & 0\\ x(3) & x(2) & & x(1) & & 0\\ \vdots & \vdots & \cdots & \vdots & \cdots & \vdots\\ x(n) & x(n-1) & & \cdots x(n-f)\cdots & & x(n-N)\end{bmatrix}\tag{11}$$

TABLE II *(Continued)*
SUMMARY OF THE QRD-LSL INTERPOLATION ALGORITHM

$$c_{I,N}(n-1) = \frac{\lambda^{1/2} I_{p,f}^{1/2}(n-f-2)}{I_{p,f}^{1/2}(n-f-1)} \tag{25}$$

$$s_{I,N}(n-1) = \frac{e_{p,f}^{I}(n-f-1)}{I_{p,f}^{1/2}(n-f-1)} \tag{26}$$

$$e_{N+1}^{F}(n,n-f-1) = \frac{e_{N+1}^{F}(n) + \lambda^{1/2} s_{I,N}(n-1)\Delta_{N+1}^{F}(n-1)}{c_{I,N}(n-1)} \tag{27}$$

$$\Delta_{N+1}^{F}(n) = \frac{\lambda^{1/2}\Delta_{N+1}^{F}(n-1) + s_{I,N}(n-1)e_{N+1}^{F}(n)}{c_{I,N}(n-1)} \tag{28}$$

$$F_{N+1}(n,n-f-1) = \lambda F_{N+1}(n-1,n-f-2) + \left(e_{N+1}^{F}(n,n-f-1)\right)^2 \tag{29}$$

$$c_{F,N+1}'(n) = \frac{\lambda^{1/2} F_{N+1}^{1/2}(n-1,n-f-2)}{F_{N+1}^{1/2}(n,n-f-1)} \tag{30}$$

$$s_{F,N+1}'(n) = \frac{e_{N+1}^{F}(n,n-f-1)}{F_{N+1}^{1/2}(n,n-f-1)} \tag{31}$$

$$e_{p,f+1}^{I}(n-f-1) = c_{F,N+1}'(n)e_{p,f}^{I}(n-f-1) - \lambda^{1/2} s_{F,N+1}'(n)\rho_{p,f+1}^{F}(n-1) \tag{32}$$

$$\rho_{p,f+1}^{F}(n) = \lambda^{1/2} c_{F,N+1}'(n)\rho_{p,f+1}^{F}(n-1) + s_{F,N+1}'(n)e_{p,f}^{I}(n-f-1) \tag{33}$$

Let f+1 $\rightarrow$ f and continue to compute part (a) or part (b) in accordance with a particular sequence chosen until a selected interpolation order (p,f) is reached.

(b) An additional past data sample is taken into account:

$$I_{p,f}(n-f) = \lambda I_{p,f}(n-f-1) + \left(e_{p,f}^{I}(n-f)\right)^2 \tag{34}$$

$$c_{I,N}(n) = \frac{\lambda^{1/2} I_{p,f}^{1/2}(n-f-1)}{I_{p,f}^{1/2}(n-f)} \tag{35}$$

$$s_{I,N}(n) = \frac{e_{p,f}^{I}(n-f)}{I_{p,f}^{1/2}(n-f)} \tag{36}$$

$$e_{N+1}^{B}(n,n-f) = \frac{e_{N+1}^{B}(n) + \lambda^{1/2} s_{I,N}(n)\Delta_{N+1}^{B}(n-1)}{c_{I,N}(n)} \tag{37}$$

$$\Delta_{N+1}^{B}(n) = \frac{\lambda^{1/2}\Delta_{N+1}^{B}(n-1) + s_{I,N}(n)e_{N+1}^{B}(n)}{c_{I,N}(n)} \tag{38}$$

number of operations required by the LSL interpolation algorithms developed in [1] and [2] is approximately $2N^2+53N$ per iteration. The computational efficiency of the proposed algorithm is obtained by exploiting the intermediate forward and backward predictions introduced in Section II. Fig. 1 presents a signal-flow graph of the QRD-LSL interpolation algorithm showing the (2,2)th -order QRD-LSL interpolator using the sequence BFBF. The auxiliary parameters symbolized by π, Δ, and ρ in this figure are computed by using (18) and (23), (28) and (38), and (33) and (43), respectively. Symbols $\boldsymbol{F}$ and $\boldsymbol{B}$ in Fig. 1 denote the computations of the forward and backward cosine-sine pairs of rotation parameters [see (20) and (21) and (15) and (16)], respectively, whereas symbols $\boldsymbol{I^F}$ and $\boldsymbol{I^B}$ denote the computations of the interpolation cosine-sine pairs of rotation parameters as an additional future data sample and an additional past data sample are considered [see (25) and (26) and (35) and (36)], respectively. These computational results are then used

TABLE II *(Continued)*
SUMMARY OF THE QRD-LSL INTERPOLATION ALGORITHM

$$B_{N+1}(n,n-f) = \lambda B_{N+1}(n-1,n-f-1) + \left(e^{B}_{N+1}(n,n-f)\right)^2 \quad (39)$$

$$c'_{B,N+1}(n) = \frac{\lambda^{1/2} B^{1/2}_{N+1}(n-1,n-f-1)}{B^{1/2}_{N+1}(n,n-f)} \quad (40)$$

$$s'_{B,N+1}(n) = \frac{e^{B}_{N+1}(n,n-f)}{B^{1/2}_{N+1}(n,n-f)} \quad (41)$$

$$e^{I}_{p+1,f}(n-f) = c'_{B,N+1}(n) e^{I}_{p,f}(n-f) - \lambda^{1/2} s'_{B,N+1}(n) \rho^{B}_{p+1,f}(n-1) \quad (42)$$

$$\rho^{B}_{p+1,f}(n) = \lambda^{1/2} c'_{B,N+1}(n) \rho^{B}_{p+1,f}(n-1) + s'_{B,N+1}(n) e^{I}_{p,f}(n-f) \quad (43)$$

Let p+1 $\rightarrow$ p and continue to compute part (a) or part (b) in accordance with a particular sequence chosen until a selected interpolation order (p,f) is reached.

2. Initializations

(c) Auxiliary parameter initialization: For order m = 1, 2,...,N, set

$\pi^{F}_{m-1}(0) = \pi^{B}_{m-1}(0) = 0$

$\Delta^{F}_{m}(0) = \Delta^{B}_{m}(0) = 0$

$\rho^{B}_{p,f}(0) = \rho^{F}_{p,f}(0) = 0$, for all p and f

(d) Soft constraint initialization: For order m = 0, 1,...,N, set

$B_m(-1) = \delta$

$F_m(0) = \delta$

$B_m(0,0) = F_m(0,0) = \delta$

$I_{p,f}(n) = \delta$, for $n \leq 0$ and all p and f

where δ is a small positive constant.

(e) Data initialization: For n = 1, 2,..., set

$e^{F}_{0}(n) = e^{B}_{0}(n) = x(n)$

$e^{I}_{0,0}(n) = x(n)$

$e^{I}_{p,f}(n) = 0$, for $n \leq 0$ and all p and f

where x(n) is the input to the QRD-LSL interpolator.

to construct the orthogonal basis of the subspace of two past data samples and two future data samples (i.e., $[e_1^B(n-2, n-2), e_2^F(n-1, n-2), e_3^B(n-1, n-2), e_4^F(n, n-2)]$) successively by employing the BFBF sequence [see (24)–(28) and (34)–(38)]. Symbols $\boldsymbol{F}'$ and $\boldsymbol{B}'$ in Fig. 1 denote the computations of the intermediate forward and intermediate backward cosine-sine pairs of rotation parameters, respectively [see (30) and (31) and (40) and (41)]. These computational results are then used to project the present data vector successively onto the orthogonal basis just constructed [see (29)–(33) and (39)–(43)]. Since the projections are performed onto orthogonal basis sets, successive stages of the QRD-LSL interpolator are decoupled. Consequently, higher order QRD-LSL interpolators are obtained from lower order ones by simply adding more "past" stages or more "future" stages, leaving the original stages unchanged. This *modular structure* permits a dynamic assignment as well as rapid automatic determination of the most effective filter length.

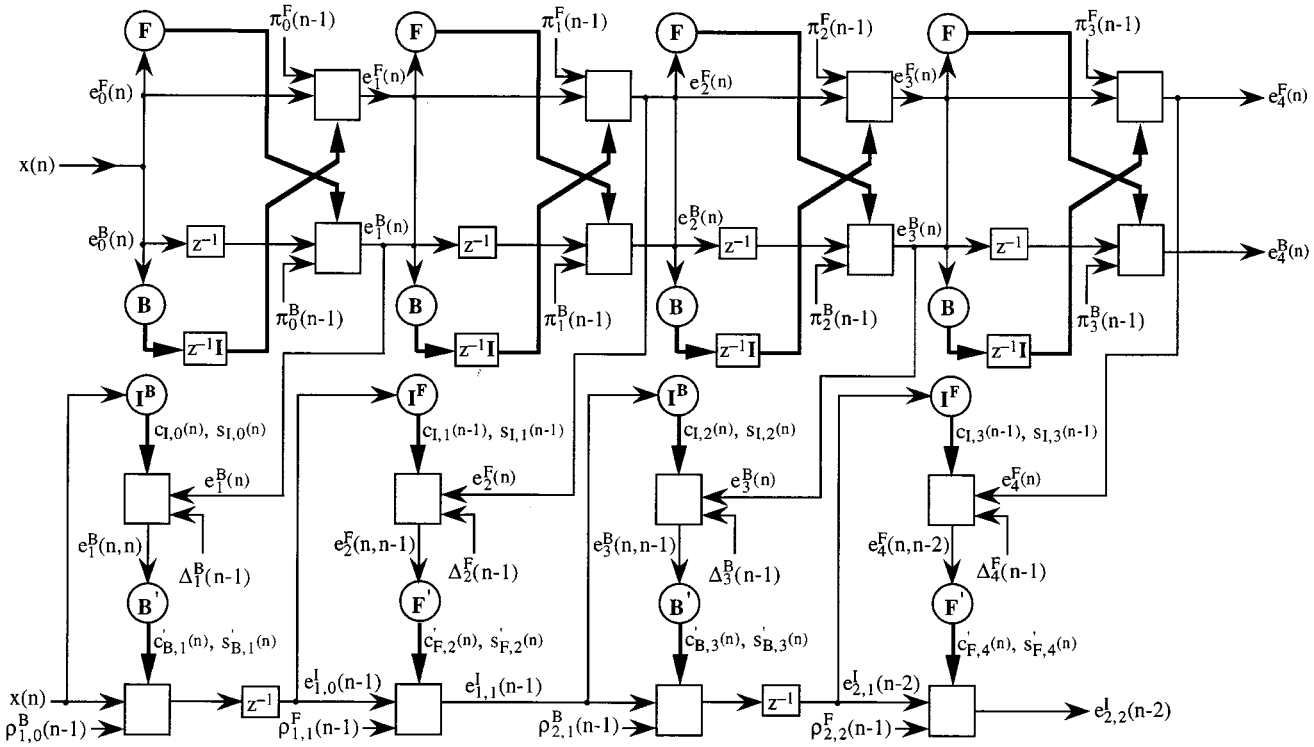


Fig. 1. Signal-flow graph of the (2,2)th-order QRD-LSL interpolator using sequence BFBF.

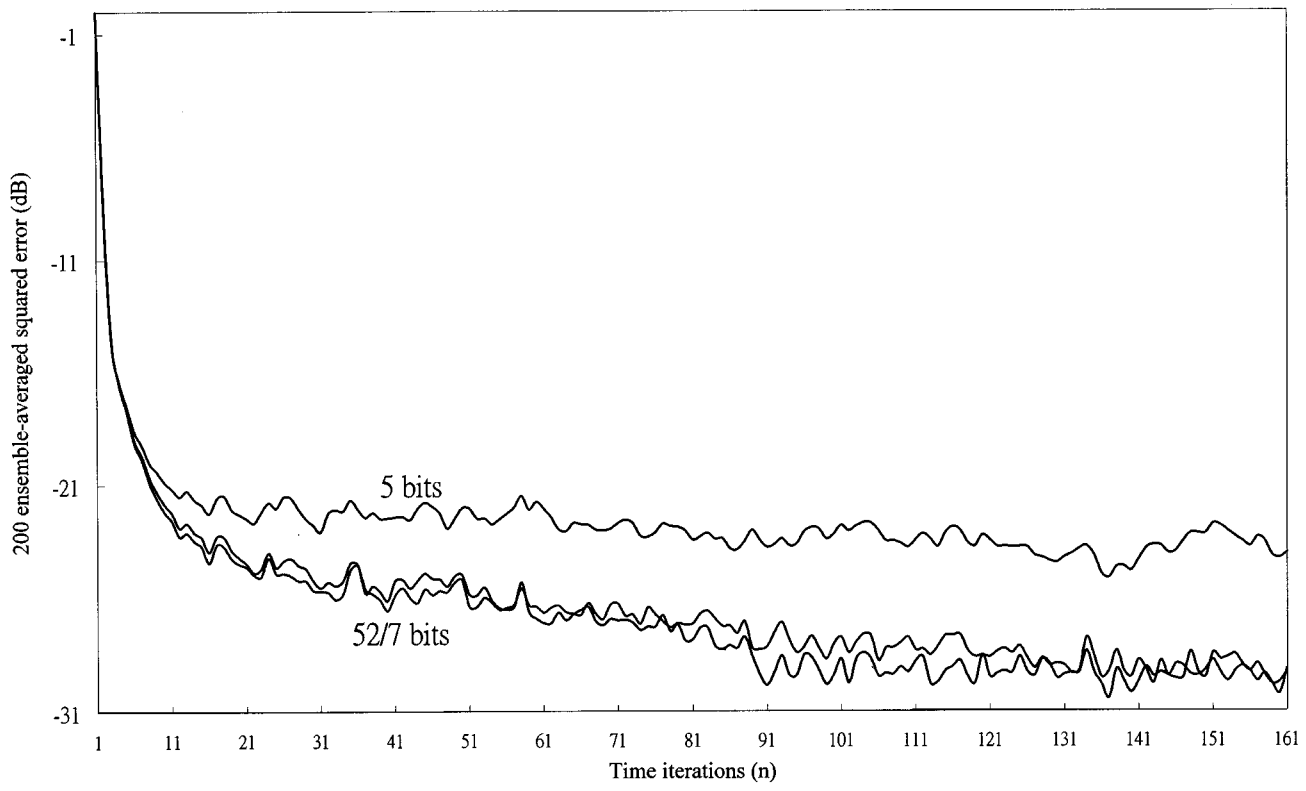


Fig. 2. Robustness of the QRD-LSL interpolation algorithm (eigenvalue spread = 100, $\lambda = 0.99, \delta = 1$).

IV. Computer Simulations

According to a previous investigation [2], the LSL interpolation that more effectively utilizes the correlation between neighboring samples than the LSL prediction can yield a much better performance than that of the latter. In this section, simulations are performed to compare the numerical behavior of the QRD-LSL interpolation algorithm with that of the conventional LSL interpolation algorithm. All simulations are performed on a PC with a floating-point processor. The finite-precision

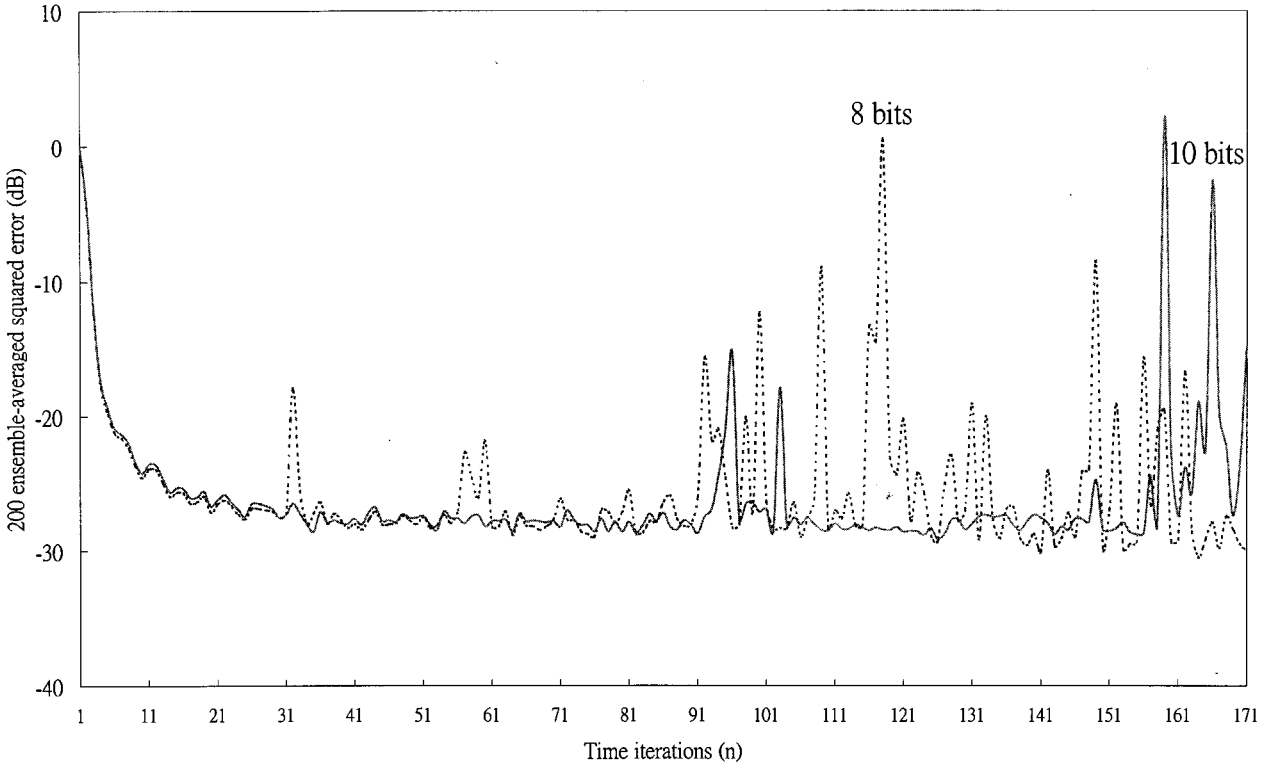


Fig. 3. Numerical behavior of the LSL algorithm developed in [2] (eigenvalue spread = 100, $\lambda = 0.99$, $\delta = 1$).

effects are observed by truncating the mantissa at a predefined position without affecting the exponent. Herein, a second-order AR process defined as $x(n)+a_1x(n-1)+a_2x(n-2)=\varepsilon(n)$ is used, where the driving process $\varepsilon(n)$ is a computer-generated sequence that simulates a zero-mean Gaussian white noise process with variance σ_ε^2. The AR parameters $a_1 = -1.9114$ and $a_2 = 0.95$ are chosen so that the AR process $x(n)$ has unity variance. For convenience, the AR parameter values are from [14, p. 351]. Figs. 2 and 3 summarize the simulation results that correspond to an eigenvalue spread 100. In both figures, the ensemble-averaged squared errors for $e_{2,2}^I(n-2)$ were computed over 200 trials each. Each trial used an independent realization of the white-noise process $\varepsilon(n)$. The forgetting factor and the initialization constant in both figures remain fixed at $\lambda = 0.99$ and $\delta = 1$, respectively.

Fig. 2 depicts the numerical robustness of the QRD-LSL interpolation algorithm. The mantissa length takes values 52, 7, and 5. This figure clearly indicates that 52 and 7 bits do not markedly differ. Even for five mantissa bits, the algorithm still functions properly but with a slower convergence rate. Our simulations were performed for 1000 iterations, and no noticeable error accumulation effects as a result of round-off noise were observed for five mantissa bits. Fig. 3 displays the numerical behavior of the recursive LSL interpolation algorithm developed in [2] with the same eigenvalue spread under the same simulation conditions. While the recursive LSL interpolation algorithm functions properly for 11 mantissa bits, the performance becomes unacceptable for less precision and it breaks down for less than eight mantissa bits.

V. Conclusions

This work extends the well known QRD-LSL predictors to QRD-LSL interpolators. Applying the intermediate forward and backward predictions introduced in Section II allows us to implement a QRD-LSL interpolator without computing the transversal predictor coefficients. Consequently, computational cost is significantly reduced. Furthermore, owing to the exact decoupling property developed in Section II, the proposed QRD-LSL interpolators are order-recursive. When finite precision arithmetic is used, computer simulation results confirm that the QRD-LSL interpolators are more numerically robust than the conventional LSL interpolators due to the use of the well-conditioned and numerically stable QR-decomposition technique. The QRD-LSL interpolators has potential implications for signal processing and communication problems, including data compression, coding and restoration of speech and image, and narrowband interference suppression in spread spectrum communications. Except for an overall delay needed for physical realization, QRD-LSL interpolators can substantially outperform QRD-LSL predictors while retaining all the desirable features of the latter.

Acknowledgment

The author would like to thank the anonymous reviewers for their careful reading of the manuscript and for helpful suggestions.

REFERENCES

[1] C. K. Coursey and J. A. Stuller, "Interpolation lattice filter," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 39, pp. 965–967, Apr. 1991.

[2] J. T. Yuan, "Asymmetric interpolation lattice," *IEEE Trans. Signal Processing*, vol. 44, pp. 1256–1261, May 1996.

[3] B. Picinobono and J. M. Kerilis, "Some properties of prediction and interpolation errors," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 36, Apr. 1988.

[4] S. Kay, "Some results in linear interpolation theory," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-31, pp. 746–749, June 1983.

[5] A. K. Jain, "Image coding via a nearest neighbors image model," *IEEE Trans. Commun.*, vol. COMM-23, pp. 318–331, Mar. 1975.

[6] M. L. Sethia and J. B. Anderson, "Interpolative DPCM," *IEEE Trans. Commun.*, vol. COMM-32, pp. 729–736, June 1984.

[7] N. S. Jayant, R. Steele, N. W. Chan, and C. E. Schmidt, "On soft-decision demodulation for PCM- and DPCM-encoded speech," *IEEE Trans. Commun.*, vol. COMM-28, pp. 308–311, Feb. 1980.

[8] N. S. Jayant and S. W. Christensen, "Effects of packet losses in waveform coded speech and improvements due to an odd-even sample-interpolation procedure," *IEEE Trans. Commun.*, vol. COM-29, pp. 101–110, Feb. 1981.

[9] Y. Yashima and K. Sawada, "An extrapolative-interpolative prediction coding method for HDTV signals," *IEEE Trans. Commun.*, vol. 38, pp. 1779–1785, Oct. 1990.

[10] B. Zeng and Y. Neuvo, "Interpolative BTC image coding with vector quantization," *IEEE Trans. Commun.*, vol. 41, pp. 1436–1438, Oct. 1993.

[11] L. Li and L. B. Milstein, "Rejection of narrow-band interference in PN spread-spectrum systems using transversal filters," *IEEE Trans. Commun.*, vol. COMM-30, pp. 925–928, May 1982.

[12] E. Masry, "Closed-form analytical results for the rejection of narrow-band interference in PN spread spectrum systems—Part II: Linear interpolation filters," *IEEE Trans. Commun.*, vol. COMM-33, pp. 10–19, Jan. 1985.

[13] P. Strobach, *Linear Prediction Theory—A Mathematical Basis for Adaptive Systems*. Berlin, Germany: Springer–Verlag, 1990, pp. 525–531.

[14] S. Haykin, *Adaptive Filter Theory*, 3rd ed. Englewood Cliffs, NJ: Prentice–Hall, 1996.

[15] B. Friedlander, "Lattice filters for adaptive processing," *Proc. IEEE*, vol. 70, no. 8, pp. 829–867, Aug. 1982.

[16] S. J. Orfanidis, *Optimum Signal Processing*. New York: Macmillan, 1985.

[17] I. K. Proudler, J. G. McWhirter, and T. J. Shepherd, "Computationally efficient QR decomposition approach to least squares adaptive filtering," *Proc. Inst. Elect. Eng. F*, vol. 138, no. 4, pp. 341–353, Aug. 1991.

[18] F. Ling, "Givens rotation based least squares lattice and related algorithms," *IEEE Trans. Signal Processing*, vol. 39, pp. 1541–1551, July 1991.

[19] 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 Processing*, vol. 39, pp. 879–891, Apr. 1991.

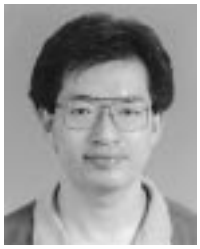
[20] A. H. Sayed and T. Kailath, "A state-space approach to adaptive RLS filtering," *IEEE Signal Processing Mag.*, pp. 18–60, July 1994.

[21] B. Porat, B. Friedlander, and M. Morf, "Square root covariance ladder algorithms," *IEEE Trans. Automat. Contr.*, vol. AC-27, pp. 813–829, Aug. 1982.

[22] D. T. L. Lee, M. Morf, and B. Friedlander, "Recursive least squares ladder estimation algorithms," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-29, pp. 627–641, June 1981.

[23] H. Lev-Ari, T. Kailath, and J. Cioffi, "Least-squares adaptive lattice and transversal filters: A unified geometric theory," *IEEE Trans. Inform. Theory*, vol. IT-30, pp. 222–236, Mar. 1984.

[24] D. T. M. Slock, "Reconciling fast RLS lattice and QR algorithms," in *Proc. ICASSP*, Albuquerque, NM, 1990, pp. 1591–1594.



Jenq-Tay Yuan (M'92) was born in Taipei, Taiwan, R.O.C., on September 9, 1957. He received the B.S. degree in electronic engineering from Fu Jen Catholic University, Taipei, in 1981, the M.S. and the Ph.D. degrees, both in electrical engineering, from the University of Missouri, Rolla, in 1986 and 1991, respectively.

From 1992 to 1993, he was a system engineer in the Formosa Plastics Corporation, Point Comfort, TX. Since 1993, he has been with the Department of Electronic Engineering, Fu Jen Catholic University, where he is currently an associate professor. His current research interests are in statistical and adaptive signal processing and their application to communication systems.

Dr. Yuan is a member of the IEEE Communication Society and the IEEE Signal Processing Society.