Least Squares Order-Recursive Lattice Smoothers

Jenq-Tay Yuan, *Member, IEEE* and John A. Stuller, *Senior Member, IEEE*

***Abstract*— Conventional least squares order-recursive lattice (LSORL) filters use present and past data values to estimate the present value of a signal. This paper introduces LSORL smoothers which use past, present and future data for that purpose. Except for an overall delay needed for physical realization, LSORL smoothers can substantially outperform LSORL filters while retaining all the advantages of an order-recursive structure.**

I. Introduction

Least squares order-recursive lattice (LSORL) filters have several advantages over fast Kalman and fast transversal filters [1], [2], [4], [10], [11]. An N-stage LSORL filter automatically generates all N of the outputs that would be provided by N separate transversal filters of length 1, 2, $\cdots, N$. Higher order lattice filters are obtained from lower order ones by simply adding more stages, leaving the original stages unchanged. This *modular structure* permits dynamic assignment, and rapid automatic determination of the most effective filter length. The order-recursive property also lends itself to the use of efficient VLSI hardware implementations. A final advantage of LSORL filters is superior numerical stability.

To our knowledge, previous references to LSORL filters have been concerned primarily with *causal* filters. In causal LSORL filtering, the present value of a desired sequence (the primary sequence), $x(n)$, is estimated through a linear combination of the present and past values of the data sequence, $y(n)$, (the observations or reference sequence). For any filter with N stages, a suitable delay can be introduced to produce the smallest mean square error (MSE) [12]. The introduction of delay makes the filter "noncausal" in the sense that a linear combination of the present, past and future observations, $y(n)$, can be used to estimate the present value of a desired signal sequence, $x(n)$. It is well known that a noncausal filter, or *smoother*, can outperform a causal filter in terms of minimum mean square error (MMSE) (see p. 157 of [5] and p. 279 of [6]). However, once delay is introduced into a LSORL filter, the order-recursive property no longer holds. Higher order "noncausal" filters cannot be built from lower-order ones simply by adding more lattice stages as more *"future"* observations are used to estimate the present value of the desired sequence.

Manuscript received October 6, 1993; revised July 25, 1994. The associate editor coordinating the review of this paper and approving it for publication was Prof. Tamal Bose.

J.-T. Yuan is with the Department of Electronic Engineering, Fu Jen Catholic University, Taipei, Taiwan, R.O.C.

J. A. Stuller is with the Department of Electrical Engineering, University of Missouri-Rolla, Rolla, MO 65401 USA.

IEEE Log Number 9410285.

In a previous correspondence [3], we described an order-recursive lattice smoother design based on a minimum mean square error performance criterion. The analysis required prior knowledge of the second order statistics of the observations and the desired sequences. Under the least squares criterion used in the present paper, knowledge of these statistics is not needed. We demonstrate by simulation experiment that the resulting LSORL smoothers can substantially outperform conventional LSORL filters while retaining the order-recursive structure with all its advantages.

II. Least Squares Smoothers

Consider the direct-form realization of an Nth-order FIR least-squares smoother shown in Fig. 1. The desired sequence $x(i)$ is estimated from its current, p past, and f future observations $y(i)$, for $i = 1, 2, \cdots, n$. The length of the observations, n, is variable. The order, $N = p + f$. We will refer to any Nth-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

$$e_{p,f}(i) = x(i) - \hat{x}_{p,f}(i) = x(i) - \boldsymbol{h}_{p,f}^{T}(\boldsymbol{n}-\boldsymbol{f}) \cdot \boldsymbol{y}_{N+1}(\boldsymbol{i}+\boldsymbol{f}), \qquad 1-f \leq i \leq n-f \quad (1)$$

where

$$\boldsymbol{y}_{N+1}^{T}(\boldsymbol{i}+\boldsymbol{f}) = [y(i+f), y(i+f-1), \cdots, y(i-p)] \quad (2)$$

and

$$\boldsymbol{h}_{p,f}^{T}(n-f) = [h_{(p,f),f}(n-f), \cdots, h_{(p,f),1}(n-f), h_{(p,f),0}(n-f), h_{(p,f),-1}(n-f), \cdots, h_{(p,f),-p}(n-f)]. \quad (3)$$

The vector $\boldsymbol{h}_{p,f}(\boldsymbol{n}-\boldsymbol{f})$ contains the fixed coefficients of the (p, f)th-order FIR smoother and will be chosen for least-squares estimation error over the time interval $1-f \leq i \leq n-f$ with prewindowing of data, that is

$$x(i) = y(i) = 0 \quad \text{for } i \leq 0. \quad (4)$$

Equation (1) can be written in matrix form as

$$\boldsymbol{e}_{(\boldsymbol{p},\boldsymbol{f})}(\boldsymbol{n}-\boldsymbol{f}) = \boldsymbol{x}(\boldsymbol{n}-\boldsymbol{f}) - \boldsymbol{A}_{N+1}(\boldsymbol{n})\boldsymbol{h}_{\boldsymbol{p},\boldsymbol{f}}(\boldsymbol{n}-\boldsymbol{f}) \quad (5)$$

where

$$\boldsymbol{x}^{T}(\boldsymbol{n}-\boldsymbol{f}) = [x(1-f), x(2-f), \cdots, x(n-f)], \quad (6)$$

$$\boldsymbol{e}_{\boldsymbol{p},\boldsymbol{f}}^{T}(\boldsymbol{n}-\boldsymbol{f}) = [e_{p,f}(1-f), e_{p,f}(2-f), \cdots, e_{p,f}(n-f)], \quad (7)$$

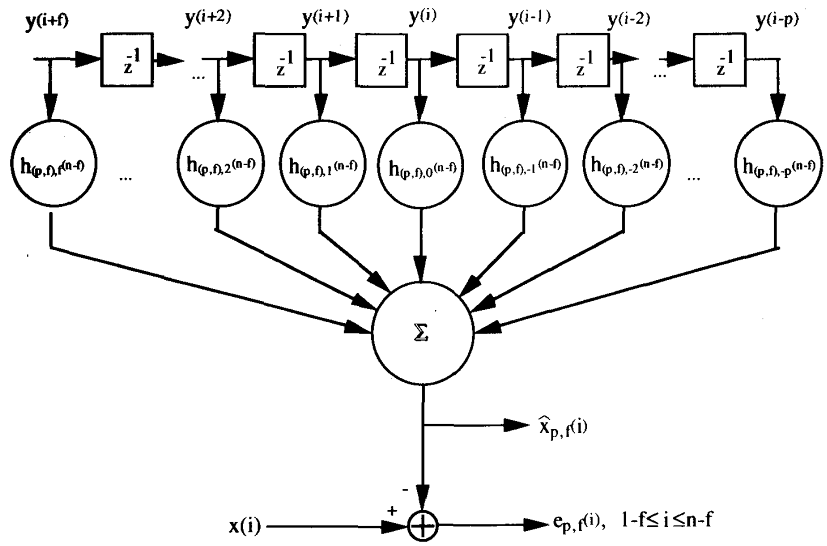


Fig. 1. Direct-form realization of the Nth or (p,f)th-order FIR least squares smoother.

$$A^T_{N+1}(n) = \begin{bmatrix} y(1) & y(2) & & y(n) \\ 0 & y(1) & \cdots & y(n-1) \\ 0 & 0 & & y(n-2) \\ \vdots & \vdots & & \vdots \\ 0 & 0 & & y(n-N) \end{bmatrix}. \quad (8)$$

The subscript $N+1$ in the symbol for the data matrix $A_{N+1}(n)$ signifies the number of columns. The optimum coefficients in (3) can be chosen by using the Hilbert space projection theorem (see p. 20 of [13]). The Hilbert space H in this case consists of all finite- energy-norm limits of sequences of linear combinations of the data and desired sequences. The inner product (u, v) of any two vectors, u, v, in H is given by $(u, v) = \Sigma_{i=1}^n u_i v_i$. If $(u, v) = 0$, we say that u and v are *least squares orthogonal*. The squared norm of a vector u is $(u, u) = |u|^2$, where u and v are both real-valued vector time series defined by $u^T = [u_1, u_2, \cdots, u_n]$ and $v^T = [v_1, v_2, \cdots, v_n]$, respectively. We define $Y(n)$ as the Hilbert subspace of H spanned by the rows of matrix $A^T_{N+1}(n)$ (i.e., of all available observations: $y(1)$, $y(2)$, $\cdots, y(n)$), and use bold brackets [ ] to represent each column of elements of matrix $A^T_{N+1}(n)$ as the Hilbert subspace of $Y(n)$. For example, the right-most column of elements of matrix $A^T_{N+1}(n)$ constitutes the subspace $Y_{p,f}(n, f) = [y(n-N), y(n-N-1), \cdots, y(n-1), y(n)]$. According to the Hilbert space projection theorem (see p. 26 of [13]), there is a unique vector $\hat{x}_{p,f}(n-f)$ in the subspace $Y(n)$ that minimizes the norm of the error vector

$$|e_{p,f}(n-f)| = |x(n-f) - \hat{x}_{p,f}(n-f)|. \quad (9)$$

The vector $\hat{x}^T_{p,f}(n-f) = [\hat{x}_{p,f}(1-f), \hat{x}_{p,f}(2-f), \cdots, \hat{x}_{p,f}(n-f)]$ is the *orthogonal projection* of vector $x(n-f)$ onto the subspace $Y(n)$. It is characterized by the following least squares *orthogonality principle*:

$$e^T_{p,f}(n-f)A_{N+1}(n) = 0^T_{N+1} \quad (10)$$

where 0_{N+1} is a zero column vector of length $N+1$. The following augmented normal equation of order (p, f) can be obtained by substituting (5) and (10) into the matrix operations of $|e_{p,f}(n-f)|^2$:

$$\begin{bmatrix} A^T_{N+1}(n)A_{N+1}(n) & A^T_{N+1}(n)x(n-f) \\ x^T(n-f)A_{N+1}(n) & x^T(n-f)x(n-f) \end{bmatrix} \cdot \begin{bmatrix} -h_{p,f}(n-f) \\ 1 \end{bmatrix} = \begin{bmatrix} 0_{N+1} \\ E_{p,f_{\min}}(n-f) \end{bmatrix} \quad (11)$$

where $E_{p,f_{\min}}(n-f)$ is the minimum value of $|e_{p,f}(n-f)|^2$.

III. Order-Recursive Least Squares Lattice Smoothers

To develop an order-update recursion for the coefficients of vector $h_{p,f}(n-f)$, we write the augmented normal equation for the least-squares smoother of one higher order $N \to N+1$. This may be done by increasing either f or p. We consider first the problem of increasing N by using one additional *future* observation: $f \to f+1$. The augmented normal equation of order $(p, f+1)$ can be deduced from (11),

$$\begin{bmatrix} A^T_{N+2}(n)A_{N+2}(n) & A^T_{N+2}(n)x(n-f-1) \\ x^T(n-f-1)A_{N+2}(n) & x^T(n-f-1)x(n-f-1) \end{bmatrix} \begin{bmatrix} -h_{p,f+1}(n-f-1) \\ 1 \end{bmatrix} = \begin{bmatrix} 0_{N+2} \\ E_{p,f+1_{\min}}(n-f-1) \end{bmatrix} \quad (12)$$

where

$$x^T(n-f-1) = [x(-f), x(1-f), \cdots, x(n-f-1)] \quad (13)$$

TABLE I
LSORL SMOOTHING ALGORITHM

(I) LSL ALGORITHM (Predictions): [4, pp.619]:

For n=1,2,3,... compute the various order updates: m=1,2,...,N, where N is the final order of the least squares lattice predictor:

$$\Delta_{m-1}(n) = \lambda\Delta_{m-1}(n-1) + \frac{e^B_{m-1}(n-1)\,e^F_{m-1}(n)}{\alpha_{m-1}(n-1)} \qquad \text{(T-1)}$$

$$e^F_m(n) = e^F_{m-1}(n) - \frac{\Delta_{m-1}(n)\,e^B_{m-1}(n-1)}{P^B_{m-1}(n-1)} \qquad \text{(T-2)}$$

$$e^B_m(n) = e^B_{m-1}(n-1) - \frac{\Delta_{m-1}(n)\,e^F_{m-1}(n)}{P^F_{m-1}(n)} \qquad \text{(T-3)}$$

$$P^F_m(n) = P^F_{m-1}(n) - \frac{\Delta^2_{m-1}(n)}{P^B_{m-1}(n-1)} \qquad \text{(T-4)}$$

$$P^B_m(n) = P^B_{m-1}(n-1) - \frac{\Delta^2_{m-1}(n)}{P^F_{m-1}(n)} \qquad \text{(T-5)}$$

$$\alpha_m(n-1) = \alpha_{m-1}(n-1) - \frac{(e^B_{m-1}(n-1))^2}{P^B_{m-1}(n-1)} \qquad \text{(T-6)}$$

(II) Smoothing:

For n=1,2,3,... start from f=0 and p=-1. Additional p "past" and f "future" stages can be increased by computing any of C^f_N combinations of (T-7) and (T-8).

$$\rho^B_{p+1,f}(n-f) = \rho^B_{p+1,f}(n-f-1) + \frac{e^B_{N+1}(n)}{\alpha_{N+1}(n)}e_{p,f}(n-f) \qquad \text{(T-7a)}$$

$$e_{p+1,f}(n-f) = e_{p,f}(n-f) - \frac{\rho^B_{p+1,f}(n-f)}{P^B_{N+1}(n)}e^B_{N+1}(n) \qquad \text{(T-7b)}$$

$$\rho^F_{p,f+1}(n-f-1) = \rho^F_{p,f+1}(n-f-2) + \frac{e^F_{N+1}(n)}{\alpha_{N+1}(n-1)}e_{p,f}(n-f-1) \qquad \text{(T-8a)}$$

$$e_{p,f+1}(n-f-1) = e_{p,f}(n-f-1) - \frac{\rho^F_{p,f+1}(n-f-1)}{P^F_{N+1}(n)}e^F_{N+1}(n) \qquad \text{(T-8b)}$$

Initialization: delta is a small positive constant

$$\Delta_{m-1}(0) = 0 \qquad \text{(T-9)}$$

$$P^F_{m-1}(0) = \text{delta} \qquad \text{(T-10)}$$

$$P^B_{m-1}(0) = \text{delta} \qquad \text{(T-11)}$$

$$e^F_0(n-1) = e^B_0(n) = y(n),\ n\geq 1 \qquad \text{(T-12)}$$

$$P^F_0(n) = P^B_0(n) = P^F_0(n-1) + y^2(n),\ n\geq 1 \qquad \text{(T-13)}$$

$$\alpha_0(n-1) = 1,\ n\geq 1$$

$$e_{-1,0}(n) = x(n),\ n\geq 1$$

$$\rho^F_{p,f}(i) = 0,\ p\geq 0,\ f\geq 0,\ 0\geq i\geq -f$$

$$\rho^B_{p,0}(0) = 0,\ p\geq 0$$

The forgetting factor [4, pp.478], λ, is set to unity throughout this paper. Our simulations, like those reported in [1] and [4], did not encounter stability problems for $\lambda = 1$.

$$\boldsymbol{A}^T_{\boldsymbol{N}+2}(\boldsymbol{n}) = \begin{bmatrix} y(1) & y(2) & & y(n) \\ 0 & y(1) & \cdots & y(n-1) \\ 0 & 0 & & y(n-2) \\ \vdots & \vdots & & \vdots \\ 0 & 0 & & y(n-N-1) \end{bmatrix} \qquad (14)$$

$$\boldsymbol{h}^T_{\boldsymbol{p},\boldsymbol{f}+1}(\boldsymbol{n}-\boldsymbol{f}-1) = [h_{(p,f+1),f+1}(n-f-1), \cdots, h_{(p,f+1),0}(n-f-1), \cdots, h_{(p,f+1),-p}(n-f-1)] \qquad (15)$$

and $E_{p,f+1_{\min}}(n-f-1)$ is the minimum value of $|e_{\boldsymbol{p},\boldsymbol{f}+1}(\boldsymbol{n}-\boldsymbol{f}-1)|^2$. Appendix A shows that the $(p, f+1)$st order optimum coefficient vector $\boldsymbol{h}_{\boldsymbol{p},\boldsymbol{f}+1}(\boldsymbol{n}-\boldsymbol{f}-1)$ is given recursively from the (p, f)th-order optimum vector $\boldsymbol{h}_{\boldsymbol{p},\boldsymbol{f}}(\boldsymbol{n}-\boldsymbol{f}-1)$ by

$$\boldsymbol{h}_{\boldsymbol{p},\boldsymbol{f}+1}(\boldsymbol{n}-\boldsymbol{f}-1) = \begin{bmatrix} 0 \\ \boldsymbol{h}_{\boldsymbol{p},\boldsymbol{f}}(\boldsymbol{n}-\boldsymbol{f}-1) \end{bmatrix} + \frac{\rho^F_{p,f+1}(n-f-1)}{P^F_{N+1}(n)} \cdot \boldsymbol{a}_{\boldsymbol{N}+1}(\boldsymbol{n}) \qquad (16)$$

where

$$\boldsymbol{a}^T_{\boldsymbol{N}+1}(\boldsymbol{n}) = [1, a_{N+1,1}(n), \cdots, a_{N+1,N+1}(n)], \qquad (17)$$

$$\rho^F_{p,f+1}(n-f-1) = \sum_{i=-f}^{n-f-1} x(i)e^F_{N+1}(i+f+1) = \sum_{i=-f}^{n-f-1} y(i+f+1)e_{p,f}(i) \qquad (18)$$

and $P^F_{N+1}(n)$ is the minimum value of the sum of the $(N+1)$st order forward prediction-error squares (see p. 575 of [4]). Note that $a_{N+1,i}(n), i = 1, 2, \cdots, N+1$ and $e^F_{N+1}(n)$ are $(N+1)$st-order forward prediction coefficients and $(N+1)$st-order forward prediction error respectively. It follows from (A10)–(A9) in Appendix A that the minimum value of the sum of the estimation-error squares update is

$$E_{p,f+1_{\min}}(n-f-1) = E_{p,f_{\min}}(n-f-1) - \frac{(\rho^F_{p,f+1}(n-f-1))^2}{P^F_{N+1}(n)}. \qquad (19)$$

We obtain the final result by premultiplying both sides of (16) by row vector $[y(n), y(n-1), \cdots, y(n-N-1)]$. This step yields the following order-recursion for the estimate

$$\boxed{\hat{x}_{p,f+1}(n-f-1) = \hat{x}_{p,f}(n-f-1) + \frac{\rho^F_{p,f+1}(n-f-1)}{P^F_{N+1}(n)}e^F_{N+1}(n).} \qquad (20)$$

The order-recursion for the estimation error can be found by subtracting the above equation from $x(n-f-1)$. This yields

$$\boxed{e_{p,f+1}(n-f-1) = e_{p,f}(n-f-1) - \frac{\rho^F_{p,f+1}(n-f-1)}{P^F_{N+1}(n)}e^F_{N+1}(n).} \qquad (21)$$

The following order-update recursion between $\boldsymbol{h}_{\boldsymbol{p}+1,\boldsymbol{f}}(\boldsymbol{n}-\boldsymbol{f})$ and $\boldsymbol{h}_{\boldsymbol{p},\boldsymbol{f}}(\boldsymbol{n}-\boldsymbol{f})$ can be obtained similarly by using one additional *past* observation $p \to p+1$

$$\boldsymbol{h}_{\boldsymbol{p}+1,\boldsymbol{f}}(\boldsymbol{n}-\boldsymbol{f}) = \begin{bmatrix} \boldsymbol{h}_{\boldsymbol{p},\boldsymbol{f}}(\boldsymbol{n}-\boldsymbol{f}) \\ 0 \end{bmatrix} + \frac{\rho^B_{p+1,f}(n-f)}{P^B_{N+1}(n)} \cdot \boldsymbol{c}_{\boldsymbol{N}+1}(\boldsymbol{n}) \qquad (22)$$

where

$$\boldsymbol{c}_{\boldsymbol{N}+1}^{\boldsymbol{T}}(\boldsymbol{n}) = [c_{N+1,N+1}(n), \cdots, c_{N+1,1}(n), 1] \quad (23)$$

$$\rho_{p+1,f}^{B}(n-f) = \sum_{i=1-f}^{n-f} y(i-p-1)e_{p,f}(i) = \sum_{i=1-f}^{n-f} x(i)e_{N+1}^{B}(i+f) \quad (24)$$

and $P_{N+1}^{B}(n)$ is the minimum value of the sum of the $(N+1)$st order backward prediction-error squares. Note that $c_{N+1,i}(n), i = 1, 2, \cdots, N+1$ and $e_{N+1}^{B}(n)$ are $(N+1)$st-order backward prediction coefficients and $(N+1)$st-order backward prediction error, respectively. By premultiplying both sides of (22) by vector $[y(n), y(n-1), \cdots, y(n-N-1)]$, we obtain

$$\boxed{\hat{x}_{p+1,f}(n-f) = \hat{x}_{p,f}(n-f) + \frac{\rho_{p+1,1}^{B}(n-f)}{P_{N+1}^{B}(n)} e_{N+1}^{B}(n).} \quad (25)$$

Finally, we find by derivations similar to that of (21) and (19),

$$\boxed{e_{p+1,f}(n-f) = e_{p,f}(n-f) - \frac{\rho_{p+1,f}^{B}(n-f)}{P_{N+1}^{B}(n)} e_{N+1}^{B}(n)} \quad (26)$$

and

$$E_{p+1,f_{\min}}(n-f) = E_{p,f_{\min}}(n-f) - \frac{(\rho_{p+1,f}^{B}(n-f))^2}{P_{N+1}^{B}(n)}. \quad (27)$$

It follows from (18) and (21) that $\rho_{p,f+1}^{F}(n-f-1)$ needs to be computed before computing the estimation error. Haykin [4] developed an efficient time-update recursion for computing $\rho_N(n)$ for the causal case only. A generalized result for the two-sided noncausal case can be obtained by using a derivation similar to that in pp. 625–627 of [4] by starting with (18) and (24). This yields

$$\boxed{\begin{aligned}&\rho_{p,f+1}^{F}(n-f-1)\\ &\quad = \rho_{p,f+1}^{F}(n-f-2) + \frac{e_{N+1}^{F}(n)}{\alpha_{N+1}(n-1)} e_{p,f}(n-f-1)\end{aligned}} \quad (28)$$

and

$$\boxed{\begin{aligned}&\rho_{p+1,f}^{B}(n-f)\\ &\quad = \rho_{p+1,f}^{B}(n-f-1) + \frac{e_{N+1}^{B}(n)}{\alpha_{N+1}(n)} e_{p,f}(n-f)\end{aligned}} \quad (29)$$

where the likelihood variable $\alpha_{N+1}(n)$ is a measure of the likelihood of deviation of successive data samples from a Gaussian distribution [7]. By combining (21), (26), (28), and (29) with the well-known recursive least squares lattice algorithm for linear prediction (see p. 619 of [4]), we can obtain a time as well as order update recursion, referred to as the LSORL *smoothing algorithm* summarized in Table I, for the estimation error for the LSORL smoother. The order-update recursion of the likelihood variable given in Table I is based on p. 615 [4]. Due to the efficient order-update recursion for $\rho_{p,f+1}^{F}(n-f-1)$ and $\rho_{p+1,f}^{B}(n-f)$ in (28) and (29) respectively, the number of computations required for (p, f)th-order LSORL smoothing is $O(N)$ per time iteration, the same as that needed for Nth-order LSORL filtering. The LSORL smoothing algorithm is described in more detail in Section V.

To construct a LSORL smoother of order (p, f), equation pairs (29), (26), and (28), (21) must be applied p and f times, respectively. However, any sequencing between these two equations is permissible. Consequently, there are $C_N^p = C_N^f = N!/p!f!$ permissible lattice realizations for a LSORL smoother of order (p, f). Two of six possible lattice realizations of an order (2 ,2) smoother are shown in Figs. 2 and 3. These realizations are identified by the sequences FFBB and BFBF of forward (F) and backward (B) prediction errors used. Figs. 2 and 3 may be related to Fig. 1 by setting $p = f = 2$ and $i = n - f$ in Fig. 1.

Lower order signal estimates corresponding to the sequencing of (20) and (25) are directly accessible from LSORL smoothers. For example, the estimates $\hat{x}_{0,1}(n-1), \hat{x}_{0,2}(n-2)$ and $\hat{x}_{1,2}(n-2)$ are accessible from the FFBB realization of Fig. 2 and the estimates $\hat{x}_{1,0}(n), \hat{x}_{1,1}(n-1)$ and $\hat{x}_{2,1}(n-1)$ are accessible from the BFBF realization of Fig. 3. We have not determined the theoretically optimum sequencing of (20) and (25) for channel equalization or other applications. We conjecture that an alternating sequence BFBF $\cdots$ is typically most appropriate because signal autocorrelation functions are typically monotonically decreasing. This conjecture is supported by the simulation experiments described in Section V. The computer simulation results comparing the BFBFBFBFBF to the BBBBBFFFFF realization indeed showed that the former is faster in convergence than the latter. As with LSORL filters, the order of a LSORL smoother can be determined by adding stages until a sufficiently small estimation error is obtained.

IV. LS Orthogonal Basis Set

Conventional LSORL filters produce a sequence of least squares (LS) uncorrelated backward prediction errors $e_0^B(n), e_1^B(n), \cdots, e_N^B(n)$ at all instants of time (see p. 469 of [4]). Haykin refers to this property as the exact *decoupling property* of the LSORL algorithm. The decoupling property, however, is confined to causal data. To process the future observations adaptively, we will need a broader decoupling property. In this section, we show that when a sequence of p past and f future observations is considered, appropriate combinations of f delayed forward prediction errors and p backward prediction errors form C_N^f sets of LS orthogonal bases. We will refer to the LS orthogonality among all the elements within each of these orthogonal bases as the *LS orthogonal basis theorem*. The LS orthogonal basis theorem is

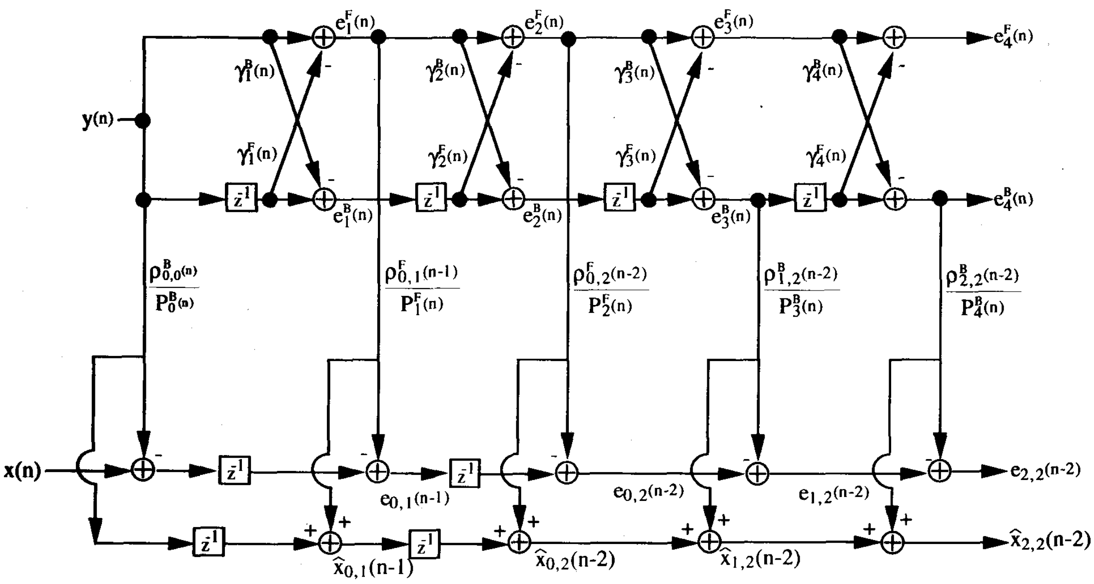


Fig. 2. (2, 2)th-order LSORL smoother using sequence FFBB.

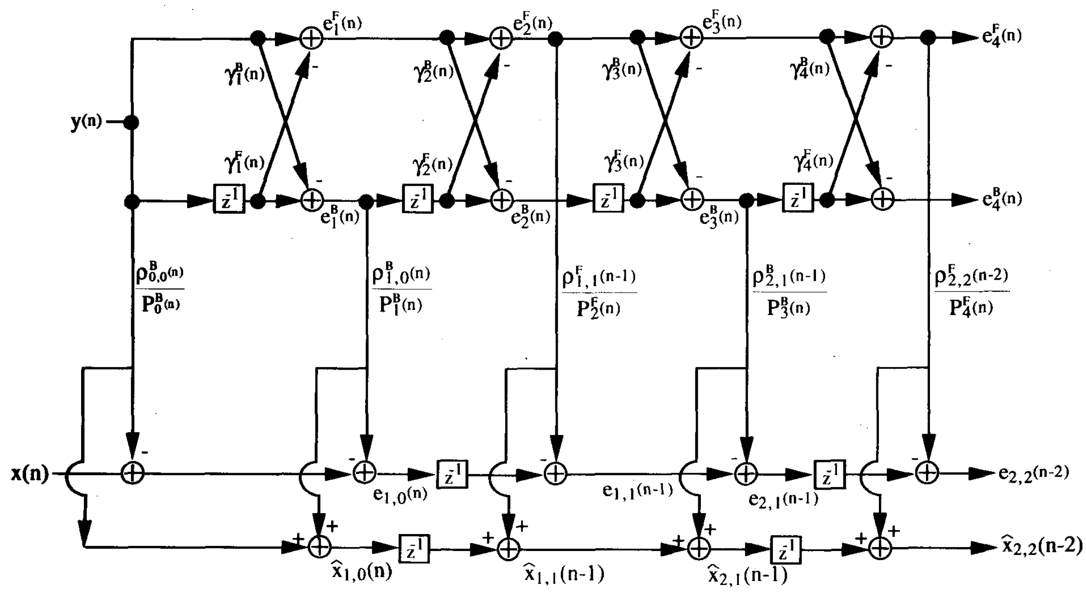


Fig. 3. (2, 2)th-order LSORL smoother using sequence BFBF.

more general than the exact decoupling property of [4]. It can be viewed as the *deterministic counterpart of the orthogonal basis theorem* introduced in [3].

The LS orthogonal basis theorem can be stated as follows:

There are C_N^f possible sets of $(p + f+1)$ LS orthogonal bases directly accessible from an Nth-order prediction error lattice that can be embedded into a LSORL smoother of order (p, f). The following conditions must be satisfied for a set of $f + p+1$ prediction errors to form a LS 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 prediction errors are delayed by one time unit. A proof of the LS orthogonal basis theorem is given in Appendix II.

Understanding of the LS orthogonal basis theorem is facilitated by referring to Table II. This table depicts the development of the Hilbert subspaces for the (2, 2)th-order LS smoother of Fig. 3. The top row in the table denotes the order of the estimate as the data progresses deeper into the lattice.

TABLE II
DEVELOPMENT OF HILBERT SUBSPACES

Order N	0	1	2	3	4
Sequence		B	F	B	F
(p,f)	(0,0)	(1,0)	(1,1)	(2,1)	(2,2)
$\widehat{x_{p,f}}(n)$	$\widehat{x_{0,0}}(n) = \frac{\rho^B_{0,0}(n)}{P^B_0(n)} y(n) = \frac{\rho^B_{0,0}(n)}{P^B_0(n)} e^B_0(n)$	$\widehat{x_{1,0}}(n) = \widehat{x_{0,0}}(n) + \frac{\rho^B_{1,0}(n)}{P^B_1(n)} e^B_1(n)$	$\widehat{x_{1,1}}(n-1) = \widehat{x_{1,0}}(n-1) + \frac{\rho^F_{1,1}(n-1)}{P^F_2(n)} e^F_2(n)$	$\widehat{x_{2,1}}(n-1) = \widehat{x_{1,1}}(n-1) + \frac{\rho^B_{2,1}(n-1)}{P^B_3(n)} e^B_3(n)$	$\widehat{x_{2,2}}(n-2) = \widehat{x_{2,1}}(n-2) + \frac{\rho^F_{2,2}(n-2)}{P^F_4(n)} e^F_4(n)$
$Y_{p,f}(n)$	$Y_{0,0}(n) = [y(n)] = [e_0(n)]$	$Y_{1,0}(n) = [y(n-1), y(n)] = [e_0(n), e^B_1(n)]$	$Y_{1,1}(n-1) = [y(n-2), y(n-1), y(n)] = [e_0(n-1), e^B_1(n-1), e^F_2(n)]$	$Y_{2,1}(n-1) = [y(n-3), y(n-2), y(n-1), y(n)] = [e_0(n-1), e^B_1(n-1), e^F_2(n), e^B_3(n)]$	$Y_{2,2}(n-2) = [y(n-4), y(n-3), y(n-2), y(n-1), y(n)] = [e_0(n-2), e^B_1(n-2), e^F_2(n-1), e^B_3(n-1), e^F_4(n)]$

The second row gives the sequence of Bs and Fs associated with the developing smoother. The third row gives the values (p, f) associated with the sequence of Bs and Fs. In the fourth row, the smoothed estimates having order 0, 1, 2, 3, 4 are evaluated by means of (20) and (25). In the bottom row, the Hilbert subspaces which contain $y(n)$ as the most recent observation (i.e., the right-most column in matrix $\boldsymbol{A}^T_{\boldsymbol{N}+1}(n)$) are described both in terms of the data basis and in terms of a forward and backward prediction error basis. The latter basis can be easily verified to be orthogonal.

There are $C_4^2 = 6$ possible sets of LS orthogonal bases that can be similarly used in a LSORL smoother of order (2, 2). Each of the six orthogonal basis set provides an orthogonal basis for the Hilbert subspace $\boldsymbol{Y}_{2,2}(\boldsymbol{n}-2) = [y(n-4), y(n-3), y(n-2), y(n-1), y(n)]$. The basis sets are

$$\begin{aligned}
&[y(n-2), e_1^B(n-2), e_2^B(n-2), e_3^F(n-1), e_4^F(n)]\\
&[y(n-2), e_1^F(n-1), e_2^F(n), e_3^B(n), e_4^B(n)]\\
&[y(n-2), e_1^F(n-1), e_2^B(n-1), e_3^F(n), e_4^B(n)]\\
&[y(n-2), e_1^B(n-2), e_2^F(n-1), e_3^B(n-1), e_4^F(n)]\\
&[y(n-2), e_1^F(n-1), e_2^B(n-1), e_3^B(n-1), e_4^F(n)]\\
&[y(n-2), e_1^B(n-2), e_2^F(n-1), e_3^F(n), e_4^B(n)].
\end{aligned}$$

Conventional LSORL filters have $(p, f) = (N, 0)$ and employ the set of backward prediction errors $\{y(n-N), e_1^B(n), e_2^B(n), \cdots, e_N^B(n)\}$ as the orthogonal basis for $\boldsymbol{Y}_{\boldsymbol{N},0}(\boldsymbol{n}) = [y(n-N), y(n-N+1), \cdots, y(n)]$. Because $C_N^0 = 1$, this set is unique for use in an order recursive filter.

V. Computer Simulations on Adaptive Equalization

We present results of computer simulations of adaptive equalization of a linear channel having unknown distortion. The simulation closely follow that of [1] and p. 634 of [4]. A polar form pseudo-random signal $x(n)$ is applied to a channel having unit pulse response:

$$h_n = \begin{cases} \dfrac{1}{2}\left[1 + \cos\left(\dfrac{2\pi}{W}(n-2)\right)\right], & n = 1, 2, 3 \\ 0, & \text{otherwise.} \end{cases} \tag{30}$$

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 tenth-order LSORL filter of the type described in [7]. Equalizer #2 was a tenth-order LSL filter with five units time delay (i.e., five "future" samples were used) of the type described in [1]. 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 LSORL filter, the order-recursive property is lost. Equalizer #3 was a (5, 5)th-order LSORL smoother of the type described in this paper. Of the $C_{10}^5 = 252$ possible realizations of a (5, 5)th-order LSORL smoother, we used the form BFBFBFBFBF. The LSORL smoothing algorithm of Table I involves division by updated parameters at some steps. To obviate computational errors, we applied Friedlander's suggestion [10] to set terms involving divisors less than a preassigned threshold, t, equal to zero (see p. 618 of [4]). The parameter W in (30) was set equal to 2.9 and 3.5 to provide for eigenvalue spreads $S =$ 6.078 and 46.82, respectively.

The learning curves for the three equalizers are shown in Figs. 4 and 5. The initial values $P_{m-1}^F(0)$ and $P_{m-1}^B(0)$ in (T-10) and (T-11) in Table I were set equal to 0.001 in both figures. 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. The transient performance of the (5, 5)th-order smoother is seen to be much less sensitive to the varying value of the preassigned threshold t than that of the tenth-order filter with delay. It can also be seen that the rate of convergence of the (5, 5)th-order smoother is as fast or faster than that of the tenth-order filter with delay, depending on the value of threshold, t. Additional realizations including the sequencing BBBBBFFFFF and the sequencing FFFFFBBBBB were tried. The simulation results revealed that the sequencing BFBFBFBFBF displayed the fastest initial transient performance compared to other realizations of the (5, 5)th-order smoother although their differences were small. In addition, the initial values of $P_{m-1}^F(0)$ and $P_{m-1}^B(0)$ do not have a significant effect on the initial transient performance of the adaptive equalizer.

VI. Conclusions

The stage-to-stage modularity of adaptive LS lattice filters provides a capability for rapid expansion or contraction of filter length to adapt to unknown and nonstationary data signals [11]. This capability also leads to efficient hardware implementation. This paper has shown that modularity can be extended from LS filters to LS smoothers which have superior performance and identical computational cost. Our simulations have involved equalizer learning curves. The LSORL smoother developed in this paper may also find application in blind equalization where a linear adaptive smoothing algorithm is used to obtain equalized output with a finite delay [8].

Appendix A

To verify (16), we rewrite it as

$$\begin{bmatrix} -\boldsymbol{h}_{\boldsymbol{p},\boldsymbol{f}+1}(\boldsymbol{n}-\boldsymbol{f}-1) \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ -\boldsymbol{h}_{\boldsymbol{p},\boldsymbol{f}}(\boldsymbol{n}-\boldsymbol{f}-1) \\ 1 \end{bmatrix} - \frac{\rho^F_{p,f+1}(n-f-1)}{P^F_{N+1}(n)} \cdot \begin{bmatrix} \boldsymbol{a}_{\boldsymbol{N}+1}(\boldsymbol{n}) \\ 0 \end{bmatrix} \tag{A1}$$

we also rewrite (12) as

$$\begin{bmatrix} -\boldsymbol{h}_{\boldsymbol{p},\boldsymbol{f}+1}(\boldsymbol{n}-\boldsymbol{f}-1) \\ 1 \end{bmatrix} = \begin{bmatrix} 0_{\boldsymbol{N}+2} \\ E_{p,f+1_{\min}}(n-f-1) \end{bmatrix} \tag{A2}$$

Note that the matrix $\boldsymbol{R}'_{\boldsymbol{p},\boldsymbol{f}+1}(\boldsymbol{n})$ can be written as in (A3), which appears at the bottom of the page, and as

$$\boldsymbol{R}'_{\boldsymbol{p},\boldsymbol{f}+1}(\boldsymbol{n}) = \begin{bmatrix} \boldsymbol{R}_{\boldsymbol{N}+2}(\boldsymbol{n}) & \boldsymbol{A}^{\boldsymbol{T}}_{\boldsymbol{N}+2}(\boldsymbol{n})\boldsymbol{x}(\boldsymbol{n}-\boldsymbol{f}-1) \\ \boldsymbol{x}^{\boldsymbol{T}}(\boldsymbol{n}-\boldsymbol{f}-1)\boldsymbol{A}_{\boldsymbol{N}+2}(\boldsymbol{n}) & \boldsymbol{x}^{\boldsymbol{T}}(\boldsymbol{n}-\boldsymbol{f}-1)\boldsymbol{x}(\boldsymbol{n}-\boldsymbol{f}-1) \end{bmatrix} \tag{A4}$$

where

$$\theta_{\boldsymbol{N}+1}(\boldsymbol{n})^{\boldsymbol{T}} = \left[\sum_{i=-f}^{n-f-1} y(i+f+1)y(i+f), \sum_{i=-f}^{n-f-1} y(i+f+1)y(i+f-1), \cdots, \sum_{i=-f}^{n-f-1} y(i+f+1)y(i-p) \right] \tag{A5}$$

and vector $\boldsymbol{a}_{\boldsymbol{N}+1}(\boldsymbol{n})$ and scalar $P^F_{N+1}(n)$ were both defined in Section III. Matrix $R_{n+2}(n)$ in (A4) is the $(q+2)$-by-$(q+2)$ deterministic correlation matrix defined as

$$\boldsymbol{R}_{\boldsymbol{N}+2}(\boldsymbol{n}) = \boldsymbol{A}^{\boldsymbol{T}}_{\boldsymbol{N}+2}(\boldsymbol{n})\boldsymbol{A}_{\boldsymbol{N}+2}(\boldsymbol{n}). \tag{A6}$$

Equation (A1) can then be verified by substituting it into (12) and using (A3) and (A4). This yields

$$\begin{bmatrix} 0_{\boldsymbol{N}+2} \\ E_{p,f+1_{\min}}(n-f-1) \end{bmatrix} = \begin{bmatrix} \sum_{i=-f}^{n} y^2(i) & \theta_{\boldsymbol{N}+1}(\boldsymbol{n})^{\boldsymbol{T}} \sum_{i=-f}^{n-f-1} x(i)y(i+f+1) \\ \theta_{\boldsymbol{N}+1}(\boldsymbol{n}) & \\ \sum_{i=-f}^{n-f-1} x(i)y(i+f+1) & \boldsymbol{R}'_{\boldsymbol{p},\boldsymbol{f}}(\boldsymbol{n}-1) \end{bmatrix} \begin{bmatrix} 0 \\ -\boldsymbol{h}_{\boldsymbol{p},\boldsymbol{f}}(\boldsymbol{n}-\boldsymbol{f}-1) \\ 1 \end{bmatrix} - \frac{\rho^F_{p,f+1}(n-f-1)}{P^F_{N+1}(n)} \begin{bmatrix} \boldsymbol{R}_{\boldsymbol{N}+2}(\boldsymbol{n}) & \boldsymbol{A}^{\boldsymbol{T}}_{\boldsymbol{N}+2}(\boldsymbol{n})\boldsymbol{x}(\boldsymbol{n}-\boldsymbol{f}-1) \\ \boldsymbol{x}^T(\boldsymbol{n}-\boldsymbol{f}-1)\boldsymbol{A}_{\boldsymbol{N}+2}(\boldsymbol{n}) & \boldsymbol{x}^{\boldsymbol{T}}(\boldsymbol{n}-\boldsymbol{f}-1)\boldsymbol{x}(\boldsymbol{n}-\boldsymbol{f}-1) \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{a}_{\boldsymbol{N}+1}(\boldsymbol{n}) \\ 0 \end{bmatrix} \tag{A7}$$

where

$$\begin{aligned} \rho^F_{p,f+1}(n-f-1) &= \sum_{i=-f}^{n-f-1} x(i)y(i+f+1) - \boldsymbol{h}_{\boldsymbol{p},\boldsymbol{f}}(\boldsymbol{n}-\boldsymbol{f}-1)\theta_{\boldsymbol{N}+1}(\boldsymbol{n}) \\ &= \sum_{i=-f}^{n-f-1} y(i+f+1) \cdot [x(i) - h_{(p,f),f}(n-f-1)y(i+f) \\ &\quad - h_{(p,f),f-1}(n-f-1)y(i+f-1) \\ &\quad - \cdots - h_{(p,f),-p}(n-f-1)y(i-p)] \\ &= \sum_{i=-f}^{n-f-1} y(i+f+1)e_{p,f}(i) \end{aligned} \tag{A8}$$

$$R'_{p,f+1}(n) = \begin{bmatrix} \sum_{i=-f}^{n} y^2(i) & \theta_{N+1}(n)^T & \sum_{i=-f}^{n-f-1} x(i)y(i+f+1) \\ \theta_{N+1}(n) & & \\ \sum_{i=-f}^{n-f-1} x(i)y(i+f+1) & R'_{p,f}(n-1) & \end{bmatrix} \tag{A3}$$

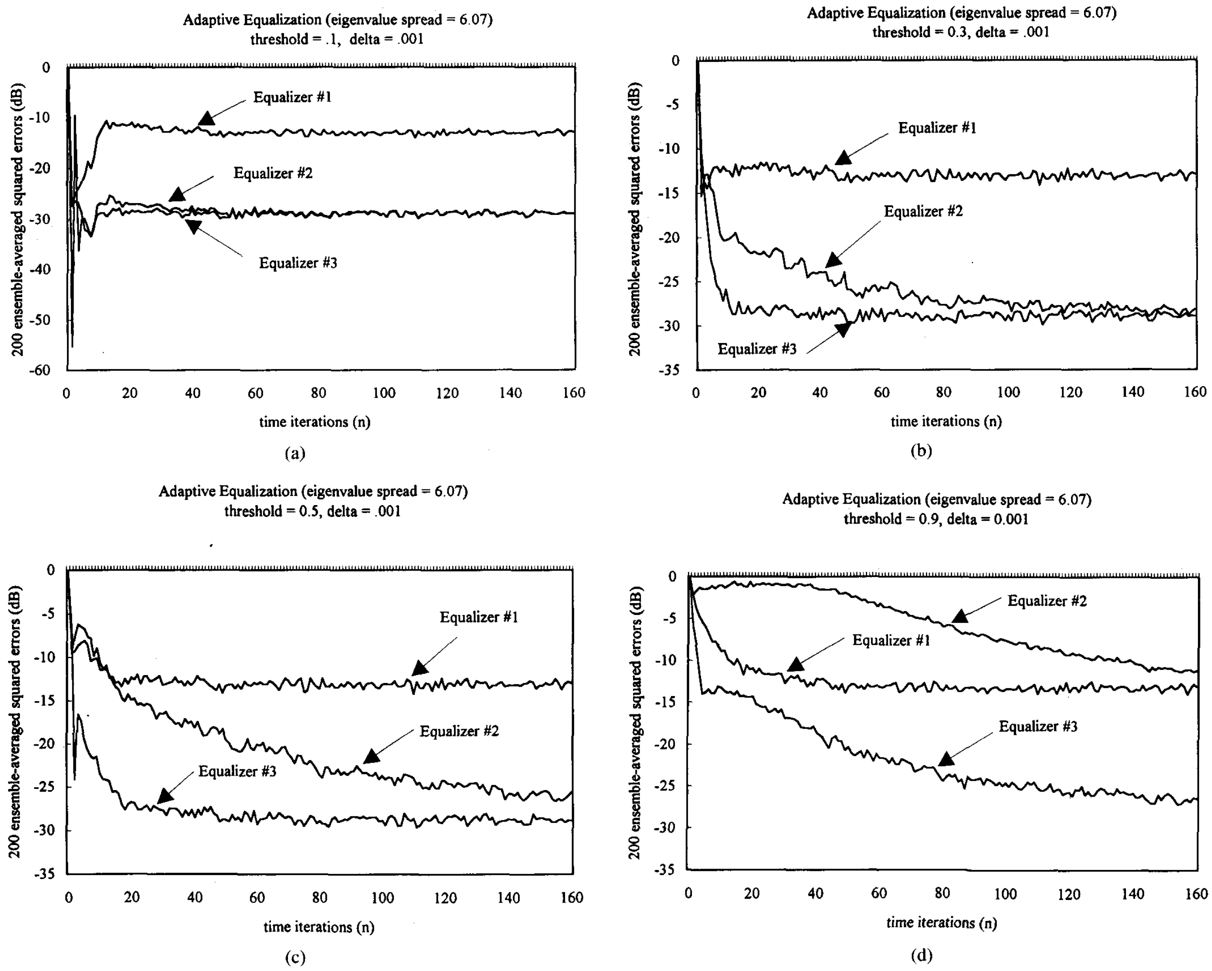


Fig. 4. Learning curves for the three equalizers (eigenvalue spread 6.07).

and

$$\begin{aligned}\rho^F_{p,f+1}(n-f-1) &= \boldsymbol{x}(\boldsymbol{n}-\boldsymbol{f}-1)^{\boldsymbol{T}}\boldsymbol{A}_{\boldsymbol{N}+2}(\boldsymbol{n})\boldsymbol{a}_{\boldsymbol{N}+1}(\boldsymbol{n})\\ &= \sum_{i=-f}^{n-f-1} x(i)[y(i+f+1)+a_{N+1,1}(n)\\ &\quad \cdot y(i+f)+\cdots+a_{N+1,N+1}(n)\\ &\quad \cdot y(i-p)]\\ &= \sum_{i=-f}^{n-f-1} x(i)e^F_{N+1}(i+f+1). \end{aligned} \tag{A9}$$

Equation (A7) can be simplified to be

$$\begin{aligned}&\begin{bmatrix} 0_{\boldsymbol{N}+2} \\ E_{p,f+1_{\min}}(n-f-1)\end{bmatrix}\\ &\quad = \begin{bmatrix} \rho^F_{p,f+1}(n-f-1) \\ 0_{\boldsymbol{N}+1} \\ E_{p,f_{\min}}(n-f-1)\end{bmatrix} - \frac{\rho^F_{p,f+1}(n-f-1)}{P^F_{N+1}(n)}\\ &\quad\quad \cdot \begin{bmatrix} P^F_{N+1}(n) \\ 0_{\boldsymbol{N}+1} \\ \rho^F_{p,f+1}(n-f-1)\end{bmatrix}. \end{aligned} \tag{A10}$$

Note that in (16), a unit *time delay* is needed when one more *future* observation is taken into account to estimate the desired sequence. The order-update recursion between $\boldsymbol{h}_{\boldsymbol{p}+1,\boldsymbol{f}}(\boldsymbol{n}-\boldsymbol{f})$ and $\boldsymbol{h}_{p,f}(n-f)$ shown in (22) can be obtained similarly.

APPENDIX B

The LS orthogonal basis theorem can be proven once we can show that $\hat{x}_{p,f}(n-f-1)$ is orthogonal to $e^F_{N+1}(n)$ and $\hat{x}_{p,f}(n-f)$ is orthogonal to $e^B_{N+1}(n)$ in (20) and (25) respectively in a time-averaged sense. We prove the latter case here. The former case can be proven similarly.

The least squares coefficient vector of (11) is given by:

$$\begin{aligned}\boldsymbol{h}_{\boldsymbol{p},\boldsymbol{f}}(\boldsymbol{n}-\boldsymbol{f}) &= (\boldsymbol{A}^{\boldsymbol{T}}_{\boldsymbol{N}+1}(\boldsymbol{n})\boldsymbol{A}_{\boldsymbol{N}+1}(\boldsymbol{n}))^{-1}\boldsymbol{A}^{\boldsymbol{T}}_{\boldsymbol{N}+1}(\boldsymbol{n})\\ &\quad \cdot \boldsymbol{x}(\boldsymbol{n}-\boldsymbol{f}). \end{aligned} \tag{B1}$$

By substituting (B1) into (5), we obtain

$$\boldsymbol{e}_{\boldsymbol{p},\boldsymbol{f}}(\boldsymbol{n}-\boldsymbol{f}) = (\boldsymbol{I}-\boldsymbol{P}_{\boldsymbol{N}+1})\boldsymbol{x}(\boldsymbol{n}-\boldsymbol{f}) \tag{B2}$$

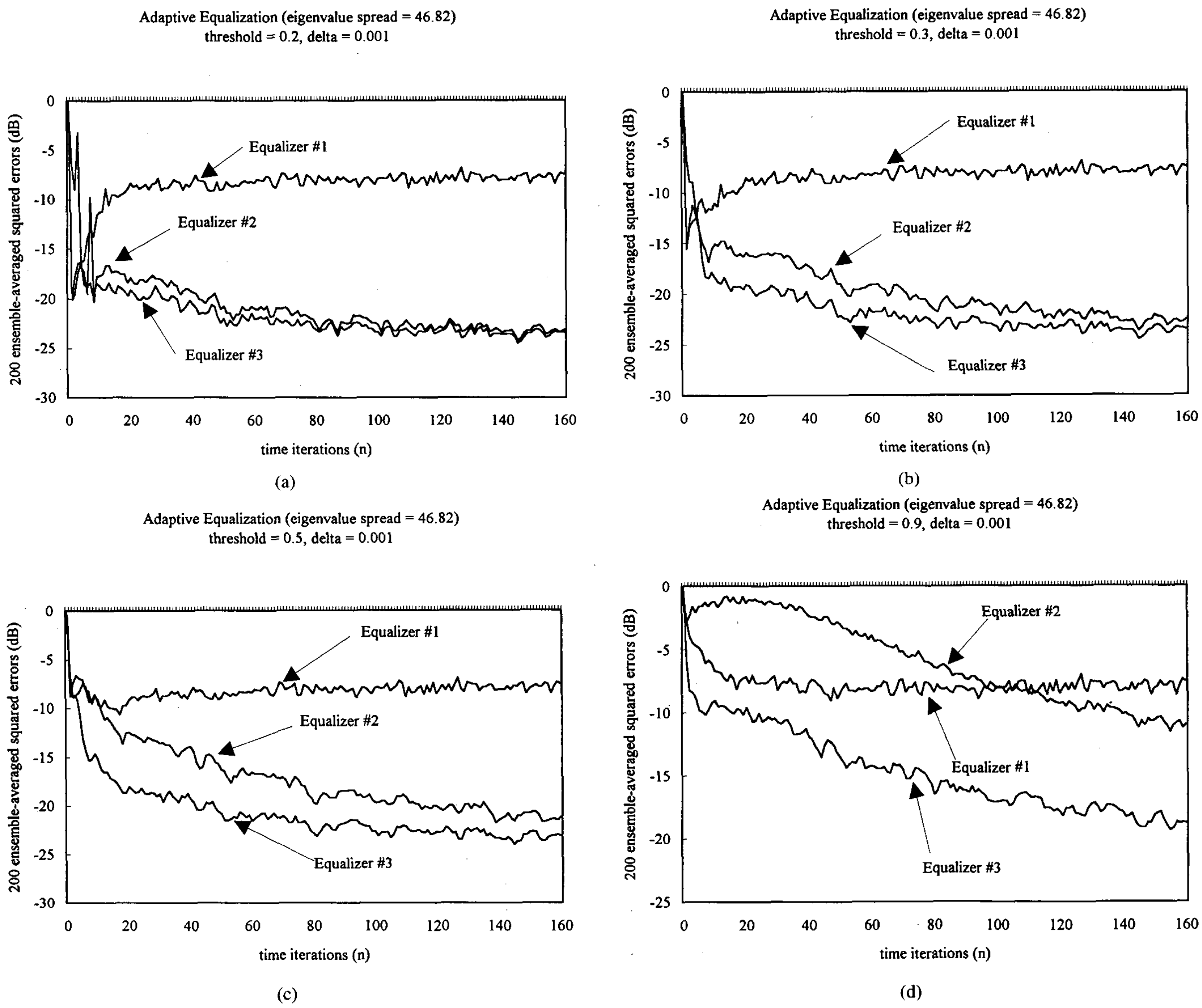


Fig. 5. Learning curves for the three equalizers (eigenvalue spread 46.82).

where

$$P_{N+1} = A_{N+1}(n)(A^T_{N+1}(n)A_{N+1}(n))^{-1}A^T_{N+1}(n). \tag{B3}$$

P_{N+1} is a projection operator [9]. The LS estimate $\hat{x}_{p,f}(n-f)$ is the orthogonal projection of $x(n-f)$ onto the subspace $Y(n)$

$$\hat{x}_{p,f}(n-f) = P_{N+1}x(n-f). \tag{B4}$$

It follows from the Hilbert space orthogonal projection theorem that

$$e^{B^T}_{N+1}(n)A_{N+1}(n) = 0^T_{N+1} \tag{B5}$$

where

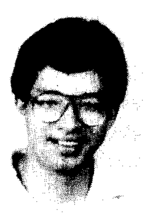
$$e^{B^T}_{N+1}(n) = [e^B_{N+1}(1), e^B_{N+1}(2), \cdots, e^B_{N+1}(n)]. \tag{B6}$$

Equation (B5) states that vector $e^B_{N+1}(n)$ is orthogonal to the subspace $Y(n)$, which, by (B4), contains the vector $\hat{x}_{p,f}(n-f)$. Consequently, vectors $e^B_{N+1}(n)$ and $\hat{x}_{p,f}(n-f)$ are orthogonal to each other. This in turn implies that $\hat{x}_{p,f}(n-f)$ is indeed LS orthogonal to $e^B_{N+1}(n)$. The orthogonality between $\hat{x}_{p,f}(n-f-1)$ and $e^F_{N+1}(n)$ can be similarly obtained. With these features in mind, as we proceed to lower orders, we must be able to express $\hat{x}_{p,f}(n-f)$ by a combination of N mutually orthogonal forward and backward prediction errors with appropriate delay. Since any sequencing between the use of (20) and (25) is permissible, there are C^f_N possible sets of $(p+f+1)$ LS orthogonal basis vectors at all instants of time. This proves the LS orthogonal basis theorem.

REFERENCES

[1] E. H. Satorius and J. D. Pack, "Application of least squares lattice algorithms to adaptive equalization," *IEEE Trans. Commun.*, vol. COM-29, pp. 136–142, Feb. 1981.

[2] S. Qureshi, "Adaptive equalization," in *Proc. IEEE*, vol. 73, pp. 1349–1387, Sept. 1985.
[3] J. T. Yuan and J. A. Stuller, "Order-recursive FIR smoothers," *IEEE Trans. Signal Processing*, vol. 42, pp. 1242–1246, May 1994.
[4] S. Haykin, *Adaptive Filter Theory*. Englewood Cliffs, NJ: Prentice Hall, 1991, 2nd ed.
[5] A. Gelb, *Applied Optimal Estimation*. Cambridge. MA: MIT Press, 1986.
[6] R. G. Brown, *Introduction to Random Signal Analysis and Kalman Filtering*. New York: Wiley, 1983.
[7] M. Morf and D. T. Lee, "Recursive least squares ladder forms for fast parameter tracking," in *Proc. 1978 IEEE Conf. Decision Contr.*, San Diego, CA, pp. 1362–1367.
[8] A. M. Benveniste, M. Goursat, and G. Ruget, "Robust identification of a nonminimum phase system: Blind adjustment of a linear equalizer in data communications," *IEEE Trans. Automat. Contr.*, vol. AC-25, pp. 385–399, June 1980.
[9] D. T. Lee, M. Morf, and B. Friedlander, "Recursive least squares ladder estimation algorithms," *IEEE Trans. Acoust. Speech Signal Processing*, vol. ASSP-29, June 1981.
[10] B. Friedlander, "Lattice filters for adaptive processing," in *Proc. IEEE*, vol. 70, pp. 829–867, Aug. 1982.
[11] R. C. North, J. R. Zeidler, W. H. Ku, and T. R. Albert, "A floating-point arithmetic error analysis of direct and indirect coefficient updating techniques for adaptive lattice filters," *IEEE Trans. Signal Processing*, vol. 41, pp. 1809–1823, May 1993.
[12] E. H. Satorius and S. T. Alexander, "Channel equalization using adaptive lattice algorithms," *IEEE Trans. Commun.*, vol. COM-27, pp. 899–905, June 1979.
[13] A. A. Giordano and F. M. Hsu, *Least Squares Estimation with Applications to Digital Signal Processing*. New York: Wiley, 1985.



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 the Fu Jen Catholic University, Taipei, Taiwan, R.O.C., in 1981 and the M.S. and the Ph.D. degrees, both in electrical engineering, from the University of Missouri–Rolla, Rolla, MO, 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 interest include fast algorithms for digital signal processing, adaptive filtering, spectral estimation, and image processing.

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

John A. Stuller (SM'91) was born in Norwalk, CT, in 1941. He received the B.S. degree in electrical engineering from the Massachusetts Institute of Technology, Cambridge, in 1963, the M.Sc. degree in electrical engineering from the University of Southern California, Los Angeles, in 1967, and the Ph.D. degree in electrical engineering from the University of Connecticut, Storrs, in 1971.

He has held technical staff positions with Northrop Corporation, Palos Verdes, CA, the TRW Systems Group, Redondo Beach, CA, the Perkin Elmer Corporation, Danbury, CT, and Bell Laboratories, Holmdel, NJ. From 1971 to 1977, he was a member of the Faculty of the University of New Brunswick, Fredericton, Canada. From August 1985 to August 1986, he was on leave from the University of Missouri-Rolla, at the Naval Underwater Systems Center, New London, CT. He has been a Professor of Electrical Engineering at the University of Missouri-Rolla since 1979.