THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 61, No. 1, January 1982
Printed in U.S.A.

Fast Recursive Estimation Using the Lattice Structure

By E. SHICHOR

(Manuscript received April 10, 1981)

This paper presents the theory for a rapidly converging adaptive linear digital filter. The filter weights are updated for every new input sample. This way the filter is optimal (in the minimum mean square error sense) for all past data up to the present, at all instants of time. This adaptive filter has thus the fastest possible rate of convergence. Such an adaptive filter, which is highly desirable for use in dynamical systems, e.g., digital equalizers, used to require on the order of N^2 multiplications for an N-tap filter at each instant of time. Recent "fast" algorithms have reduced this number to like 10 N. One of these algorithms has the lattice form, and is shown here to have some interesting properties: It decorrelates the input data to a new set of orthogonal components using an adaptive, Gram-Schmidt like, transformation. Unlike other fast algorithms of the Kalman form, the filter length can be changed at any time with no need to restart or modify previous results. It is conjectured that these properties will make it less sensitive to digital quantization errors in finite word-length implementation.

I. INTRODUCTION

Gradient algorithms are widely used in adaptive tapped delay-line filters, such as equalizers, to derive a set of tap coefficients that gives the desired output with a minimum mean square error (mmse). It is widely recognized[1] that when the input samples presented to the adaptive system are highly correlated, convergence to the optimum filter coefficients is slow. An important contribution to solving this problem of slow convergence was made by Godard[2] who obtained an adaptive algorithm that minimizes the total mse at all instants of time. Consequently, the Godard algorithm has the fastest possible rate of convergence in an mmse sense, and is usually referred to as the optimal mean-square adaptive estimator. This algorithm has the structure of a Kalman filter, and its complexity is on the order of N^2 multiplications and additions per iteration, where N is the number of filter coefficients being adjusted. Fast convergence results when successive corrections to the coefficients' vector are adaptively decorrelated. Based on this observation, other practical, less complex, schemes of orthogonalizing the corrections were proposed, see for example Ref. 1. Recently, an efficient (or "fast") computing procedure, called the fast Kalman algorithm, was obtained which provides a fast-converging estimator identical to that of Godard, but which requires only on the order of 10 N multiplications.[3,4,5]

Another approach to accelerated convergence is to transform the input data to obtain uncorrelated inputs to the estimator.[6] When the characteristics of the channel are fixed and known, the transformation can be found from the data autocorrelation matrix. When this matrix is unknown, the transformation has to be adaptive. Since the lattice structure, whose computational complexity grows only like N, is known to generate "white" uncorrelated outputs by a process called inverse filtering that keeps removing correlated components from the input signal,[7–11] it has been proposed for this application. However, the outputs of the lattice structure are uncorrelated only after it has converged to its steady state; therefore, it may not converge as fast as the Godard algorithm. Recently, Morf was able to formulate the lattice algorithm in a special form such that its outputs are uncorrelated in the mean square sense for all instants of time.[12,13] Our purpose is to extend Morf's works to compute an adaptive estimator which is equivalent in performance to Godard's. Moreover, we will demonstrate that the computational complexity of the adaptive lattice algorithm compares well with the fast Kalman algorithm of Falconer and Ljung.[5] The advantage of the lattice structure is the ease of changing the number of coefficients. It is also conjectured that the lattice algorithm will be less sensitive than the Kalman algorithm to finite-precision digital implementation. Recently, this was observed in Ref. 14. It is also discussed in Ref. 15, where the development of an equalizer based on the adaptive lattice algorithm is presented in a form similar to the one given here. One case to illustrate this property of the lattice will be given at the end of this paper.

In the next section, the optimal least mean square estimator and predictor are precisely defined, and the minimal error that results is given. In Section III, several properties of the optimal predictor are explored and are related to the estimation problem. In Section IV, an efficient (in the sense of small number of computations) lattice form is derived, using the relations developed in Section III, that maintains the optimal convergence. In Section V, the properties of this lattice form are compared to the steady-state lattice structure. Suggestions for further work are also included.

II. OPTIMAL MEAN SQUARE ESTIMATION

2.1 Notation and definitions

Given a discrete time input data sequence $\{y_i\}$ $i = 0, 1, \cdots$, it is desired to find the set of weights for a transversal tapped delay-line filter such that the output of this filter be a good estimate of another sequence $\{d_i\}$. An adaptive equalizer, for example, has the received signal as its input, while its output should provide an estimate of the transmitted data. In a transversal filter, a vector of filter coefficients, of length $p + 1$, operates on vectors of data that are shifted versions of the input data for time $0 \leq t \leq T$ defined by

$$y_{p,T}^t = (y_T, y_{T-1}, \cdots, y_{T-p}), \tag{1}$$

with y^t being the transpose of y and it is assumed that $y_i = 0$ for $i < 0$. As we are concerned with an adaptive, i.e., time varying filter, its weight vector of order $p + 1$ will be denoted

$$w_{p,T}^t = [w_{p,T}(0), w_{p,T}(1), \cdots, w_{p,T}(p)]. \tag{2}$$

Using these definitions, the output of the $p + 1$ long linear estimator at time T is $\hat{d}_{p,T}$ given by

$$\hat{d}_{p,T} = w_{p,T}^t y_{p,T}. \tag{3}$$

Now suppose that $w_{p,T}$ is the best predictor for time T, then the total, or accumulated, mean square estimation error up to time T, when using this predictor, is given by

$$E(w_{p,T}) = \sum_{i=0}^{T} (d_i - w_{p,T}^t y_{p,i})^2. \tag{4}$$

The sequence of weight vectors that minimizes eq. (4) at every instant of time T, is the most rapidly converging sequence, and is called the optimal adaptive filter.

Making use of the following time-domain definitions of the cross correlation vector and the autocorrelation matrix,

$$\sum_{i=0}^{T} d_i y_{p,i} \equiv g_{p,T} \tag{5}$$

$$\sum_{i=0}^{T} y_{p,i} y_{p,i}^t \equiv R_{p,T}, \tag{6}$$

one obtains

$$E(w_{p,T}) = \sum_{i=0}^{T} d_i^2 - 2w_{p,T}^t g_{p,T} + w_{p,T}^t R_{p,T} w_{p,T}. \tag{7}$$

2.2 The optimal estimator

Equating the gradient of eq. (7) with respect to $w_{p,T}$ to zero gives

$$R_{p,T} w_{p,T} = g_{p,T}. \tag{8}$$

It is seen that solving for the optimal estimator is equivalent to inverting a matrix at every new sample point:

$$w_{p,T} = R_{p,T}^{-1} g_{p,T}. \tag{9}$$

Godard showed that $w_{p,T}$ can be updated with on the order of p^2 calculations,[1,2] which is an improvement over the simple matrix inversion, requiring on the order of p^3 calculations. Algorithms that require only on the order of $10p$ operations for obtaining the optimal estimator appeared[3,4] subsequently, and are called "fast" algorithms.

The essence of this paper is to derive the fast algorithm in a special form, called the lattice form. This form was proposed to speed the convergence of the weight vector of an adaptive predictor to its optimal value (see Refs. 7 to 11). As will be seen in the next paragraph, the estimator problem is closely related to the prediction problem.

From eqs. (7) and (8) it is seen that the minimal total mse that results using the optimal estimator is simply

$$E_{\text{opt}}(w_{p,T}) = \sum_{i=0}^{T} d_i^2 - g_{p,T}^t R_{p,T}^{-1} g_{p,T}. \tag{10}$$

This is in constrast to the adaptive gradient algorithm whose performance is more difficult to analyze.

2.3 Optimal prediction

The problem of prediction is more basic, but similar to the problem of estimation. Solving this problem will be shown to simplify the solution of the estimation problem. For linear prediction, a set of weights is used to linearly estimate the present input point from past values of the input data. Let the set of p weights at time T be $\{-a_{p,T}(1), -a_{p,T}(2), \cdots, -a_{p,T}(p)\}$, so that the error when predicting the input point y_i is given by

$$e_{p,i} = A_{p,T}^t y_{p,i}, \tag{11}$$

with

$$A_{p,T}^t = [1, a_{p,T}(1), \cdots, a_{p,T}(p)]. \tag{12}$$

The error generated in predicting the input is that part of the input which is uncorrelated to past values of the input. This is a desired feature for fast convergence of adaptive filters.

The total square error up to time T is, thus, given by

$$\sum_{i=0}^{T} e_{p,i}^2 = A_{p,T}^t R_{p,T} A_{p,T}. \tag{13}$$

Taking derivatives with respect to $a_{p,T}(1)$ to $a_{p,T}(p)$, it is found that the predictor weight vector that will minimize the total mse up to time

T is the solution of the last p equations of the expression

$$R_{p,T}A_{p,T} = \begin{pmatrix} R^e_{p,T} \\ 0^p \end{pmatrix}, \tag{14}$$

with $R^e_{p,T}$ yet unknown and $0^p = (0, \cdots, 0)^t$ vector of order p. Using this optimal predictor in eq. (13) gives

$$\sum_{i=0}^{T} e^2_{p,i} = A^t_{p,T}R_{p,T}A_{p,T} = A^t_{p,T}\begin{pmatrix} R^e_{p,T} \\ 0^p \end{pmatrix} = R^e_{p,T}. \tag{15}$$

Therefore, $R^e_{p,T}$ is the minimal total mse that will result.

As before, obtaining the optimal predictor $A_{p,T}$ for all T is equivalent to inverting the matrix $R_{p,T}$ for all T. An efficient algorithm for doing this will be described. It should be noted, from comparing eqs. (8) and (14), that the latter is a simpler "homogenous" set of equations, except for the end term $R^e_{p,T}$; therefore, its solutions can serve as a basis for the solution of eq. (8).

III. DERIVATION OF THE ORDER AND TIME UPDATE RELATIONS

3.1 Time shift properties of $R_{p,T}$

The vectors $y_{p,T}$ for successive values of T are shifts of each other. As these vectors build up the matrix $R_{p,T}$ in eq. (6), it is expected that shifted versions of the solutions to the predictor and estimator equations will serve in updating these solutions. For doing this, the shift properties of $R_{p,T}$ are explored. For the (i, j) term in eq. (6), we have

$$R_{p,T}(i, j) = \sum_{k=0}^{T} y_{k+1-i}y_{k+1-j} = R_{p-1,T}(i, j)$$

$$\text{for all } p - 1 \geq i, j \geq 1 \tag{16}$$

$$R_{p,T}(i+1, j+1) = \sum_{k=0}^{T} y_{k-i}y_{k-j} = \sum_{k=-1}^{T-1} y_{k+1-i}y_{k+1-j}$$

$$= \sum_{k=0}^{T-1} y_{k+1-i}y_{k+1-j} = R_{p-1,T-1}(i, j)$$

$$\text{for all } p - 1 \geq i, j \geq 1, \tag{17}$$

where the fact that $y_i \equiv 0$ for $i < 0$ was used. Using Morf's notation, these relations can conveniently be written as

$$R_{p,T} = \begin{pmatrix} R_{p-1,T} & X \\ X & X \end{pmatrix} = \begin{pmatrix} X & X \\ X & R_{p-1,T-1} \end{pmatrix}, \tag{18}$$

with X being any other term in the matrix. It is clear from eq. (18) that $R_{p,T}$ is symmetric, but not Toeplitz, if steady state is not reached. Therefore, the properties of Toeplitz matrices cannot be used, as is done for example in claiming fast convergence in Ref. 7.

3.2 Order update relations

It will be useful to define, similar to eq. (12), a backward prediction vector

$$B^t_{p,T} = (b_{p,T}(p),\, b_{p,T}(p-1),\, \cdots,\, b_{p,T}(1),\, 1), \tag{19}$$

with the backward error given by

$$r_{p,T} = B^t_{p,T} y_{p,T}, \tag{20}$$

i.e., it is the error in predicting y_{T-p} from y_T to y_{T-P+1}. To minimize the total mean square backward prediction error up to time T, $B_{p,T}$ should be the solution of

$$R_{p,T} B_{p,T} = \begin{pmatrix} 0^p \\ R^r_{p,T} \end{pmatrix}. \tag{21}$$

It is seen that this is another set of homogenous equations, except for the lower one. Again, the optimal error $r_{p,T}$ will be orthogonal to $y_{T-p+1}, \cdots, y_T$.

A recursive procedure will be derived in the Appendix for generating solutions to eqs. (14) and (21) for increasing order p.

It is shown to be

$$A_{p+1,T} = \begin{pmatrix} A_{p,T} \\ 0 \end{pmatrix} - k_{p,T} R^{-r}_{p,T-1} \begin{pmatrix} 0 \\ B_{p,T-1} \end{pmatrix} \tag{22}$$

for $k_{p,T}$ as defined in the Appendix, and the higher order total error is

$$R^e_{p+1,T} = R^e_{p,T} - k^2_{p,T} R^{-r}_{p,T-1} = R^e_{p,T}(1 - k^2_{p,T} R^{-e}_{p,T} R^{-r}_{p,T-1}). \tag{23}$$

As increasing the predictor order would not increase, and typically will decrease the error, it should be that

$$1 \geq k^2_{p,T} R^{-e}_{p,T} R^{-r}_{p,T-1} \geq 0. \tag{24}$$

Similarly,

$$B_{p+1,T} = \begin{pmatrix} 0 \\ B_{p,T-1} \end{pmatrix} - k_{p,T} R^{-e}_{p,T} \begin{pmatrix} A_{p,T} \\ 0 \end{pmatrix} \tag{25}$$

and

$$R^r_{p+1,T} = R^r_{p,T-1} - k^2_{p,T} R^{-e}_{p,T}. \tag{26}$$

Similar relations hold for the prediction error, when multiplying eqs. (22) and (25) by $y_{p+1,T}$

$$e_{p+1,T} = e_{p,T} - k_{p,T} R^{-r}_{p,T-1} r_{p,T-1} \tag{27}$$

$$r_{p+1,T} = r_{p,T-1} - k_{p,T} R^{-e}_{p,T} e_{p,T}. \tag{28}$$

The following auxiliary quantities are needed

$$C_{p,T} = R^{-1}_{p,T} y_{p,T} \tag{29}$$

$$\gamma_{p,T} = C^t_{p,T} y_{p,T} = y^t_{p,T} R^{-1}_{p,T} y_{p,T}. \tag{30}$$

The order update of these quantities will also be derived in the Appendix. It is shown to be

$$C_{p+1,T} = \begin{pmatrix} C_{p,T} \\ 0 \end{pmatrix} + r_{p+1,T} R^{-r}_{p+1,T} B_{p+1,T}$$

$$= \begin{pmatrix} C_{p,T} \\ 0 \end{pmatrix} + \mu_{p+1,T} B_{p+1,T} = \begin{pmatrix} X \\ \mu_{p+1,T} \end{pmatrix}, \tag{31}$$

with $\mu_{p+1,T}$ defined by

$$\mu_{p+1,T} = r_{p+1,T} R^{-r}_{p+1,T} \tag{32}$$

and, as seen, is the last term of $C_{p+1,T}$.

3.3 Time update relations

To obtain the time update of $A_{p,T}$, use is made of the following:

$$R_{p,T+1} = R_{p,T} + y_{p,T+1} y^t_{p,T+1}. \tag{33}$$

This relation is shown to give

$$A_{p,T+1} = A_{p,T} - e^0_{p,T+1} \begin{pmatrix} 0 \\ C_{p-1,T} \end{pmatrix}. \tag{34}$$

Here, the definition

$$e^0_{p,T+1} = A^t_{p,T} y_{p,T+1} \tag{35}$$

is used for the tentative prediction error before updating the prediction coefficients. As for the minimal total mse, it is updated according to

$$R^e_{p,T+1} = A^t_{p,T} \begin{pmatrix} R^e_{p,T+1} \\ 0^p \end{pmatrix} = A^t_{p,T} R_{p,T+1} A_{p,T+1}$$

$$= A^t_{p,T}(R_{p,T} + y_{p,T+1} y^t_{p,T+1}) A_{p,T+1} = R^e_{p,T} + e^0_{p,T+1} e_{p,T+1}. \tag{36}$$

It should be mentioned here that only in the stationary case $e_{p,T+1} = e^0_{p,T+1}$ and, thus, $R^e_{p,T+1} = R^e_{p,T} + e^2_{p,T+1}$. As for updating $B_{p,T}$, two different possibilities are derived in the Appendix:

$$B_{p,T+1} = B_{p,T} - r^0_{p,T+1} \begin{pmatrix} C_{p-1,T+1} \\ 0 \end{pmatrix} \tag{37}$$

or

$$B_{p,T+1} = (B_{p,T} - r^0_{p,T+1} C_{p,T+1}) \times \frac{1}{1 - r^0_{p,T+1} \mu_{p,T+1}}. \tag{38}$$

From eq. (37), a relation like eq. (36) can be obtained

$$R^r_{p,T+1} = R^r_{p,T} + r_{p,T+1} r^0_{p,T+1}. \tag{39}$$

Again, $r^0_{p,T+1}$ is the tentative backward prediction error. The time update of $C_{p,T}$ is also obtained in the Appendix.

From these results, a simple update for $k_{p,T}$ is found to be

$$k_{p,T+1} = k_{p,T} + e^0_{p,T+1} r_{p,T}. \tag{40}$$

or alternatively, from eqs. (79) and (83)

$$k_{p,T+1} = k_{p,T} + e^0_{p,T+1} r^0_{p,T} (1 - \gamma_{p-1,T}) = k_{p,T} + e_{p,T+1} r^0_{p,T}. \tag{41}$$

All these relations are derived in the Appendix. They form the basis for the lattice network which update these quantities both in order and time.

IV. EFFICIENT CALCULATION OF THE OPTIMAL ESTIMATOR

4.1 Tapped delay line estimator

The optimal estimator for any order p and for each time T is given in eq. (9). Using eqs. (5) and (6), we get

$$\begin{aligned} R_{p,T+1} w_{p,T+1} &= g_{p,T+1} = g_{p,T} + d_{T+1} y_{T+1} \\ &= R_{p,T} w_{p,T} + d_{T+1} y_{T+1} \\ &= R_{p,T+1} w_{p,T} + (d_{T+1} - w^t_{p,T} y_{p,T+1}) y_{p,T+1}. \end{aligned} \tag{42}$$

Therefore,

$$\begin{aligned} w_{p,T+1} &= w_{p,T} + (d_{T+1} - w^t_{p,T} y_{p,T+1}) R^{-1}_{p,T+1} y_{p,T+1} \\ &= w_{p,T} + (d_{T+1} - \hat{d}^0_{p,T+1}) C_{p,T+1}. \end{aligned} \tag{43}$$

Note that updating $w_{p,T}$ involves the tentative estimate $\hat{d}^0_{p,T+1} = w_{p,T} y_{p,T+1}$ using the new data and present estimator weights. This makes it possible to implement this scheme in decision-directed equalizers, where the decision on which d_{T+1} was transmitted is based on $\hat{d}^0_{p,T+1}$. Also note that the correction to $w_{p,T}$ is in the direction of $C_{p,T+1} = R^{-1}_{p,T+1} y_{p,T+1}$ rather than $y_{p,T+1}$, as in the gradient algorithm. These vectors are parallel only if $R_{p,T+1}$ is a unit matrix times a scalar; thus, all its eigenvalues are equal. When this is not the case, and $y_{p,T+1}$ contains eigenvectors corresponding to different eigenvalues, $R^{-1}_{p,T+1}$ equalizes the gains for these vectors. Also note the similarity in the updating equations (34), (37), and (43) which is to be expected, since prediction is a special case of estimation.

The fast Kalman algorithm is an efficient recursive procedure to obtain $C_{p,T+1}$. This is given in Ref. 4 as follows:

1. Assume that all vectors are available up to and including time T.
2. Use eq. (35) to obtain $e^0_{p,T+1} = A^t_{p,T} y_{p,T+1}$.
3. Use eq. (34) to calculate $A_{p,T+1} = A_{p,T} - e^0_{p,T+1} \begin{pmatrix} 0 \\ C_{p-1,T} \end{pmatrix}$.

4. Use eq. (11) to calculate $e_{p,T+1} = A^t_{p,T+1} y_{p,T+1}$.
5. Use eq. (36) to calculate $R^e_{p,T+1} = R^e_{p,T} + e^0_{p,T+1} e_{p,T+1}$.
6. Calculate $e_{p,T+1} R^{-e}_{p,T+1}$.
7. Use eq. (89) to calculate

$$C_{p,T+1} = \begin{pmatrix} 0 \\ C_{p-1,T} \end{pmatrix} + e_{p,T+1} R^{-e}_{p,T+1} A_{p,T+1}.$$

8. From eq. (31), $\mu_{p,T+1}$ is found.
9. Find $r^0_{p,T+1} = B^t_{p,T} y_{p,T+1}$.
10. Use eq. (38) to calculate

$$B_{p,T+1} = (B_{p,T} - r^0_{p,T+1} C_{p,T+1}) \frac{1}{1 - r^0_{p,T+1} \mu_{p,T+1}}.$$

11. Use eq. (73) to calculate

$$\begin{pmatrix} C_{p-1,T+1} \\ 0 \end{pmatrix} = C_{p,T+1} - \mu_{p,T+1} B_{p,T+1}.$$

12. Calculate the tentative estimate $\hat{d}^0_{p,T+1} = w^t_{p,T} y_{p,T+1}$.
13. Use eq. (43) to update the estimator weights

$$w_{p,T+1} = w_{p,T} + (d_{T+1} - \hat{d}^0_{T+1}) C_{p,T+1}.$$

The initial conditions, when there is not enough input data so that $R_{p,T}$ in eq. (6) does not have an inverse, are discussed in Ref. 4. There are $10p + 5$ multiplications, $9p + 4$ additions, and 2 divisions for one complete updating cycle. Note that there are no matrix operations, only additions and products of scalars and vectors is involved. By comparison, the simple fixed-step gradient algorithm requires $2p + 1$ multiplications and $2p + 1$ additions per cycle.

4.2 Lattice structure

Here an equivalent algorithm that also gives the optimal estimator with about the same number of computations is derived. It is assumed that the input data are transformed by the lower triangular transformation matrix

$$L^t_{p,T} = \begin{pmatrix} B_{0,T} & B_{1,T} & \cdots & B_{p,T} \\ 0^p & 0^{p-1} & & \end{pmatrix}. \tag{44}$$

From eq. (20), the transformed data are

$$L_{p,T} y_{p,T} = (r_{0,T} r_{1,T} \cdots r_{p,T})^t \equiv \bar{r}_{p,T}. \tag{45}$$

Note that as the dimension p increases, new terms are included in $\bar{r}$, but the previous ones are unchanged. This is an important property of the lattice algorithm that enables us to change the estimation order without the need to recalculate all previous values, as for example in the fast Kalman algorithm. As before, we let the tentative transformed vector be

$$\bar{r}^{0}_{p,T+1} = L_{p,T}y_{p,T+1}. \tag{46}$$

Define a new vector of weights

$$H^t_{p,T} \equiv (h_{0,T}h_{1,T} \cdots h_{p,T}), \tag{47}$$

that now operates on $\bar{r}_{p,T}$ to give the estimate

$$\hat{d}_{p,T} = H^t_{p,T}\bar{r}_{p,T} = H^t_{p,T}L_{p,T}y_{p,T}. \tag{48}$$

It is seen that for this estimate to be equivalent to the optimal estimator, $w_{p,T}$ is the transform of $H_{p,T}$, and from eq. (8)

$$L_{p,T}g_{p,T} = L_{p,T}R_{p,T}w_{p,T} = L_{p,T}R_{p,T}L^t_{p,T}H_{p,T}. \tag{49}$$

Using eq. (21), it can be shown that

$$R_{p,T}L^t_{p,T} = \begin{pmatrix} R^r_{0,T} & 0^1 & & 0^P \\ & R^r_{1,T} & \cdots & \\ X & X & & R^r_{p,T} \end{pmatrix}, \tag{50}$$

which is a lower triangular matrix. The product $L_{p,T}R_{p,T}L^t_{p,T}$ is, thus, a symmetric product of two lower triangular matrices; therefore, it should be symmetric, lower triangular, and diagonal, i.e.,

$$L_{p,T}R_{p,T}L^t_{p,T} \equiv D_{p,T}. \tag{51}$$

The diagonal terms are easily found using eq. (50)

$$D_{p,T}(i, i) = [B^t_{i,T}(0^{P-i})^t] \begin{pmatrix} 0^i \\ R^r_{i,T} \\ X \end{pmatrix} = R^r_{i,T} \tag{52}$$

and, again, they are independent of p.

At this point, a closer inspection of $L_{p,T}$ is of interest. It is a lower triangular matrix with 1 on the main diagonal. Therefore, $L^{-1}_{p,T}$ has the same structure. Therefore, eq. (45) can be rewritten as

$$\bar{r}_{p,T} = y_{p,T} - (L^{-1}_{p,T} - I_p)\bar{r}_{p,T}. \tag{53}$$

This can be looked upon as a Gram-Schmidt procedure to calculate new orthogonal components of $\bar{r}_{p,T}$ from the components of $y_{p,T}$, minus their projections on the previous $\bar{r}_{p,T}$ components. Thus, eq. (51) represents the fact that the autocorrelation matrix of the transformed data is indeed diagonal.

Using eqs. (49) and (51), $H_{p,T}$ can be found from the transformed $g_{p,T}$ by

$$H_{p,T} = D^{-1}_{p,T}L_{p,T}g_{p,T}. \tag{54}$$

It should be noted that only scalar divisions rather than matrix inversion is needed here and increasing the order of the lattice *estimator does not change previously calculated values of H.* This is why double indices are used in eq. (47) as compared to triple indices in eq. (2). Equation (54) can be broken to p scalar equations

$$h_{p,T} = R_{p,T}^{-r} B_{p,T}^{t} g_{p,T}. \tag{55}$$

To see how the right-hand side develops in time, define

$$\rho_{p,T} = B_{p,T}^{t} g_{p,T}. \tag{56}$$

Then from eqs. (37), (5), (29), (9), and (3) in that order

$$\begin{aligned} \rho_{p,T+1} &= B_{p,T+1}^{t} g_{p,T+1} \\ &= [B_{p,T}^{t} - r_{p,T+1}^{0}(C_{p-1,T+1}0)](g_{p,T} + d_{T+1}y_{p,T+1}) \\ &= \rho_{p,T} + (d_{T+1} - \hat{d}_{p-1,T+1})r_{p,T+1}^{0} = \rho_{p,T} + V_{p-1,T+1}r_{p,T+1}^{0}, \end{aligned} \tag{57}$$

where $V_{p,T}$ is the estimation error after the pth order estimator. For $p = 0$

$$\rho_{0,T+1} = g_{0,T+1} = \rho_{0,T} + d_{T+1}y_{T+1}, \tag{58}$$

i.e., $\hat{d}_{-1,T+1} \equiv 0$. Obviously,

$$V_{p,T} = d_T - \hat{d}_{p,T} = d_T - H_{p,T}^{t}\bar{r}_{p,T} = V_{p-1,T} - h_{p,T}r_{p,T}. \tag{59}$$

The recursive solution to eq. (55) that corresponds to eq. (43) is:

$$\begin{aligned} h_{p,T+1} &= R_{p,T+1}^{-r}(\rho_{p,T} + V_{p-1,T+1}r_{p,T+1}^{0}) \\ &= R_{p,T+1}^{-r}[(R_{p,T+1}^{r} - r_{p,T+1}r_{p,T+1}^{0})h_{p,T} + V_{p-1,T+1}r_{p,T+1}^{0}] \\ &= h_{p,T} + R_{p,T+1}^{-r}(V_{p-1,T+1} - h_{p,T}r_{p,T+1})r_{p,T+1}^{0}. \end{aligned} \tag{60}$$

This is equivalent to the first tap of the conventional tapped delay line equalizer for $p = 0$ only. The tentative estimate as in eq. (43) is now

$$\hat{d}_{p,T+1}^{0} = w_{p,T}^{t}y_{p,T+1} = H_{p,T}^{t}L_{p,T}y_{p,T+1} = H_{p,T}^{t}\bar{r}_{p,T+1}^{0}. \tag{61}$$

The minimal total squared error is from eq. (51)

$$E_{p,T} = \sum_{i=0}^{T} d_i^2 - g_{p,T}^{t}R_{p,T}^{-1}g_{p,T} = \sum_{i=0}^{T} d_i^2 - (L_{p,T}g_{p,T})^{t}D_{p,T}^{-1}(L_{p,T}g_{p,T}). \tag{62}$$

From the structure of D and L it follows that

$$E_{p+1,T} = E_{p,T} - (B_{p+1,T}^{t}g_{p+1,T})^2R_{p+1,T}^{-r} = E_{p,T} - \rho_{p+1,T}^{2}R_{p+1,T}^{-r}. \tag{63}$$

Using eq. (57), the residual error can be found for all instants of time. It is then simple to decide whether p should be increased, decreased, or unchanged to meet the desired performance. As mentioned before, when adding or deleting sections no recalculation of the coefficients is needed.

The procedure for recursively obtaining the estimator $H_{p,T}$ and the estimate $\hat{d}_{p,T+1}$ is as follows:

1. Assume that all quantities are known up to and including time T.
2. Start with $e_{p,T+1} = e^0_{p,T+1} = r_{p,T+1} = r^0_{p,T+1} = y_{T+1}$ for $p = 0$.
3. Use eqs. (22) and (25) to compute

$$e^0_{1,T+1} = e^0_{0,T+1} - (k_{0,T} R^{-r}_{0,T-1}) r^0_{0,T}$$

$$r^0_{1,T+1} = r^0_{0,T} - (k_{0,T} R^{-e}_{0,T}) e^0_{0,T+1}.$$

4. Use eq. (26) to compute $k_{0,T+1} = k_{0,T} + e^0_{0,T+1} r_{0,T}$.
5. Use eqs. (36) and (39) to compute

$$R^e_{0,T+1} = R^e_{0,T} + e_{0,T+1} e^0_{0,T+1}$$

$$R^r_{0,T+1} = R^r_{0,T} + r_{0,T+1} r^0_{0,T+1}.$$

6. Compute the gain terms $k_{0,T+1} R^{-r}_{0,T}$ $k_{0,T+1} R^{-e}_{0,T+1}$ to obtain from eqs. (27) and (28)

$$e_{1,T+1} = e_{0,T+1} - (k_{0,T+1} R^{-r}_{0,T}) r_{0,T}$$

$$r_{1,T+1} = r_{0,T} - (k_{0,T+1} R^{-e}_{0,T+1}) e_{0,T+1}.$$

These gain terms can be saved for the next recursion.

7. Repeat steps 3 to 6 for $p = 1, 2, \cdots$
8. Use eq. (61) to compute the tentative estimate

$$\hat{d}^0_{p,T+1} = H^t_{p,T} \bar{r}^0_{p,T+1}.$$

9. To update $H_{p,T}$ start with $V_{-1,T+1} = d_{T+1}$ from eqs. (58) and (59) and use eq. (60) to compute

$$h_{0,T+1} = h_{0,T} + R^{-r}_{0,T+1} (V_{-1,T+1} - h_{0,T} r_{0,T+1}) r^0_{0,T+1}.$$

10. Use eq. (59) to compute

$$V_{0,T+1} = V_{-1,T+1} - h_{0,T+1} r_{0,T+1}.$$

11. Repeat steps 9 and 10 for $p = 1, 2, \cdots$.
12. Use eq. (48) to compute $d_{p,T+1} = H^t_{p,T+1} \overline{r_{p,T+1}}$.

Steps 3, 6, 8, and 10 can be drawn in a block diagram like in Fig. 1. The variable gain terms are $k_{p,T} R^{-r}_{p,T-1}$, $k_{p,T} R^e_{p,T}$, and $h_{p,T}$, and they are updated in steps 4, 6, and 9. When the system reaches a steady state, it can be illustrated in a simpler form as shown in Fig. 2.

4.2.1 Starting the algorithm

The problem of finding the optimal predictor/estimator of order p is not well defined if there are less than p input points. Therefore, when starting the algorithm, p should be 0 in the first recursion, 1 in the second, and p should grow linearly in time until it reaches the desired number of sections of the lattice. This is in contrast with the

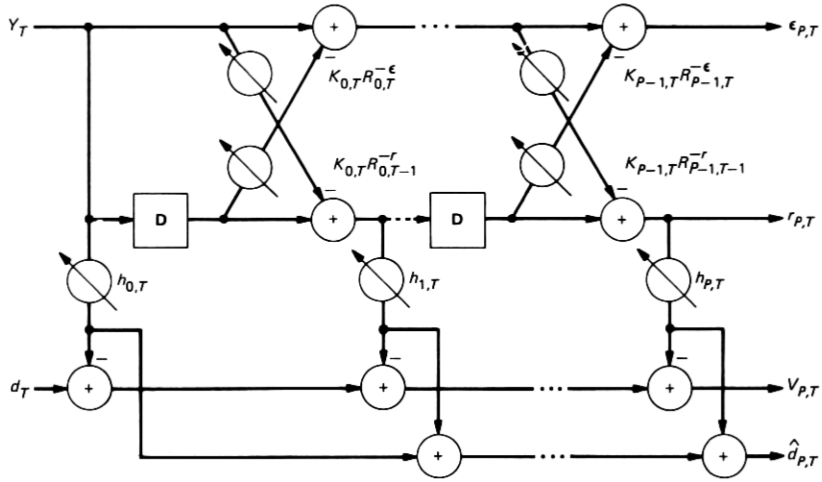

Fig. 1—The basic form of the lattice estimator.

fast Kalman algorithm, where a small diagonal matrix is assumed for $R_{p,0}$.

4.2.2 Number of operations

The number of operations required for the three algorithms, the simple gradient, the fast Kalman, and the lattice, are given below where p is the number of adaptive parameters:

Algorithm	Multiplications	Additions	Divisions
Gradient	$2p$	$2p$	—
Fast Kalman	$10p$	$9p$	2
Lattice	$12p$	$11p$	$3p$

V. DISCUSSION

It was shown in eq. (9) that the optimal linear estimator that yields the least total mse is obtained by matrix inversion. A recursive algorithm to update the optimal estimator also involves an inversion of the correlation matrix of the data as in eq. (43). If the input data are uncorrelated (i.e., low signal embedded in flat noise, or data signal with Nyquist spectral shape), then multiplying by $R_{p,t}^{-1}$ is equivalent to scalar division, which is the simple gradient algorithm. However, if the data are highly correlated and $R_{p,T}$ has its eigenvalues spread out ($\lambda_{max}/\lambda_{min} \gg 1$), then the optimal recursive algorithm for the fastest convergence of the estimator is more complex: The estimator can still have the form of a tapped delay line, but now the shift properties of

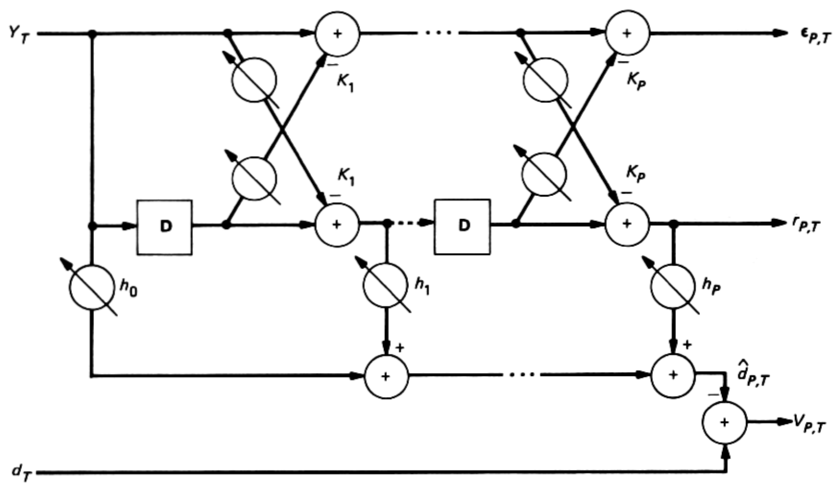

Fig. 2—The basic form of the steady state lattice estimator.

$R_{p,T}$ are used to update $w_{p,T}$ in $0(p)$ multiplications. A different approach demonstrated in this paper is to transform the input data using the lattice network to get uncorrelated inputs to the estimator. The estimator weights are now different but related to the original set through the same transformation eq. (48). This is similar to Ref. 6, except that the transformation matrix is time variant. The performance of the nonstationary lattice and the fast Kalman are the same—both give the minimal error—and in Ref. 2 it is demonstrated that convergence time can be reduced by a factor of 15, compared to the simple gradient algorithm.

The differences between the lattice and the fast Kalman algorithms in practical, finite precision digital implementation should be fully discussed elsewhere. However, an example can be given here. Examining step 5 for the Kalman algorithm, eq. (36) may occasionally render $R^e_{p,T+1}$ which is nonpositive because of the accumulation of arithmetic errors. The algorithm is useless from that time on, and has to be restarted. On the other hand, for the lattice algorithm, step 5, if either $R^e_{p,T+1}$ or $R^r_{p,T+1}$ become nonpositive, force them to be some small but positive number, and make all $k_{i,T+1}$ equal zero for $i \geqslant p$. The updating algorithm then falls back to the gradient algorithm from tap p and on, or the filter length can be shortened to length p, as desired.

Future work should try and make use of the above recurrence update relations for the exponentially weighted errors, the "fading memory" case under time-varying situations. Also simpler, suboptimal algorithms can be derived and should be compared to the exact algorithm in terms of performance and complexity.

VI. CONCLUSIONS

(*i*) The lattice algorithm gives identical results to the fast Kalman algorithm for adapting filter coefficients when both have the same number of coefficients.

(*ii*) The number of multiplications for the two algorithms is about the same, but the lattice requires more divisions for normalization by the residual error energy at each stage.

(*iii*) Changing the number of taps is easier under the lattice algorithm.

(*iv*) In limited-precision implementation under severe amplitude distorting channels, the last property of the lattice algorithm may be valuable in providing better performance.

APPENDIX

A.1 Derivation of the order update of $A_{p,T}$

From eqs. (14), (18), and (20), it is found that

$$R_{p+1,T}\begin{pmatrix} A_{p,T} \\ 0 \end{pmatrix} = \begin{pmatrix} R_{p,T} & X \\ X & X \end{pmatrix}\begin{pmatrix} A_{p,T} \\ 0 \end{pmatrix} = \begin{pmatrix} R^e_{p,T} \\ 0^p \\ k_{p,T} \end{pmatrix}, \tag{64}$$

and similarly

$$R_{p+1,T}\begin{pmatrix} 0 \\ B_{p,T-1} \end{pmatrix} = \begin{pmatrix} k'_{p,T} \\ 0^p \\ R^r_{p,T-1} \end{pmatrix} \tag{65}$$

for some $k_{p,T}, k'_{p,T}$. From the fact that $A_{p,T}(0) = B_{p,T-1}(p) \equiv 1$ it can be shown that

$$\begin{aligned} k_{p,T} &= (0\ B^t_{p,T-1})\begin{pmatrix} R^e_{p,T} \\ 0^p \\ k_{p,T} \end{pmatrix} = (0\ B^t_{p,T-1})R_{p+1,}T\begin{pmatrix} A_{p,T} \\ 0 \end{pmatrix} \\ &= (k'_{p,T}(0^p)^t R^r_{p,T-1})\begin{pmatrix} A_{p,T} \\ 0 \end{pmatrix} = k'_{p,T}. \end{aligned} \tag{66}$$

Combining eqs. (64) and (65) in a proper way, it is found that

$$R_{p+1,T}\left[\begin{pmatrix} A_{p,T} \\ 0 \end{pmatrix} - k_{p,T}R^{-r}_{p,T-1}\begin{pmatrix} 0 \\ B_{p,T-1} \end{pmatrix}\right] = \begin{pmatrix} R^e_{p+1,T} \\ 0^{p+1} \end{pmatrix}, \tag{67}$$

with

$$R^{-r}_{p,T-1} \equiv (R^r_{p,T-1})^{-1}. \tag{68}$$

Therefore,

$$A_{p+1,T} = \begin{pmatrix} A_{p,T} \\ 0 \end{pmatrix} - k_{p,T}R^{-r}_{p,T-1}\begin{pmatrix} 0 \\ B_{p,T-1} \end{pmatrix}. \tag{69}$$

A.2 The order update of $C_{p,T}$

For the order update of $C_{p,T}$, note that

$$R_{p+1,T}\begin{pmatrix} C_{p,T} \\ 0 \end{pmatrix} = \begin{pmatrix} y_{p,T} \\ X \end{pmatrix} \tag{70}$$

and

$$R_{p+1,T}B_{p+1,T} = \begin{pmatrix} 0^{p+1} \\ R^{r}_{p+1,T} \end{pmatrix}. \tag{71}$$

From this it can be seen that $C_{p+1,T}$ is a linear combination of $\begin{pmatrix} C_{p,T} \\ 0 \end{pmatrix}$ and $B_{p+1,T}$.
From the relation

$$\begin{aligned} ((0^{p+1})^{t}R^{r}_{p+1,T})C_{p+1,T} &= B^{t}_{p+1,T}R_{p+1,T}C_{p+1,T} \\ &= B^{t}_{p+1,T}y_{p+1,T} = r_{p+1,T}, \end{aligned} \tag{72}$$

it then must be that

$$\begin{aligned} C_{p+1,T} &= \begin{pmatrix} C_{p,T} \\ 0 \end{pmatrix} + r_{p+1,T}R^{-r}_{p+1,T}B_{p+1,T} \\ &= \begin{pmatrix} C_{p,T} \\ 0 \end{pmatrix} + \mu_{p+1,T}B_{p+1,T} = \begin{pmatrix} X \\ \mu_{p+1,T} \end{pmatrix}, \end{aligned} \tag{73}$$

with the definition

$$\mu_{p+1,T} = r_{p+1,T}R^{-r}_{p+1,T}, \tag{74}$$

i.e., $\mu_{p+1,T}$ is the last term of $C_{p+1,T}$.
Therefore,

$$\gamma_{p+1,T} = C^{t}_{p+1,T}y_{p+1,T} = \gamma_{P,T} + r^{2}_{p+1,T}R^{-r}_{p+1,T}, \tag{75}$$

and, thus,

$$\gamma_{p,T} = \sum_{i=0}^{p} r^{2}_{i,T}R^{-r}_{i,T}. \tag{76}$$

A.3 The time update relations

From eq. (33) the time update of $A_{p,T}$ is obtained as follows:

$$\begin{aligned} R_{p,T+1}A_{p,T} &= (R_{p,T} + y_{p,T+1}y^{t}_{p,T+1})A_{p,T} \\ &= \begin{pmatrix} R^{e}_{p,T} \\ 0^{p} \end{pmatrix} + y_{p,T+1}e^{0}_{p,T+1} \\ &= \begin{pmatrix} R^{e}_{p,T+1} \\ 0^{p} \end{pmatrix} + R_{p,T+1}\begin{pmatrix} 0 \\ C_{p-1,T} \end{pmatrix} e^{0}_{p,T+1} \end{aligned} \tag{77}$$

for some $R^e_{p,T+1}$, with the definition (35) for $e^0_{p,T+1}$. From eq. (77), it can be seen that

$$A_{p,T+1} = A_{p,T} - e^0_{p,T+1}\begin{pmatrix} 0 \\ C_{p-1,T} \end{pmatrix}, \tag{78}$$

and multiplying both sides by $y_{p,T+1}$ gives the relation

$$e_{p,T+1} = e^0_{p,T+1}(1 - \gamma_{p-1,T}) \quad \text{for} \quad p = 1, 2, \cdots \tag{79}$$

As for $p = 0$, we get from the definition that

$$e_{0,T+1} = e^0_{0,T+1} \quad \text{or} \quad \gamma_{-1,T} \equiv 0.$$

The time update of $B_{p,T}$ is obtained similarly:

$$R_{p,T+1}B_{p,T} = (R_{p,T} + y_{p,T+1}y^t_{p,T+1})B_{p,T} = \begin{pmatrix} 0^p \\ R^r_{p,T} \end{pmatrix} + y_{p,T+1}r^0_{p,T+1}$$

$$= \begin{pmatrix} 0^p \\ R^r_{p,T+1} \end{pmatrix} + R_{p,T+1}\begin{pmatrix} C_{p-1,T+1} \\ 0 \end{pmatrix} r^0_{p,T+1}. \tag{80}$$

Therefore,

$$B_{p,T+1} = B_{p,T} - r^0_{p,T+1}\begin{pmatrix} C_{p-1,T+1} \\ 0 \end{pmatrix}. \tag{81}$$

As in eqs. (36), (78), and (79),

$$R^r_{p,T+1} = R^r_{p,T} + r_{p,T+1}r^0_{p,T+1} \tag{82}$$

and

$$r_{p,T+1} = r^0_{p,T+1}(1 - \gamma_{p-1,T+1}) \quad \text{for} \quad p = 1, 2, \cdots \tag{83}$$

and

$$r_{0,T+1} = r^0_{0,T+1} = y_{t+1}.$$

The time update of $B_{p,T}$ can also be obtained as follows:

$$R_{p,T+1}B_{p,T} = (R_{p,T} + y_{p,T+1}y^t_{p,T+1})B_{p,T}$$

$$= \begin{pmatrix} 0^p \\ R^r_{p,T} \end{pmatrix} + y_{p,T+1}r^0_{p,T+1} = \begin{pmatrix} 0^p \\ R^r_{p,T} \end{pmatrix} + R_{p,T+1}C_{p,T+1}r^0_{p,T+1}. \tag{84}$$

Thus,

$$B_{p,T+1} = (B_{p,T} - r^0_{p,T+1}C_{p,T+1}) \times \frac{1}{1 - r^0_{p,T+1}\mu_{p,T+1}}, \tag{85}$$

where the denominator is chosen to make $b_{p,T+1}(0) \equiv 1$ using the definition (32). The time update of $C_{p,T}$ is obtained as in eqs. (70) to (73).

$$R_{p,T+1}\begin{pmatrix} 0 \\ C_{p-1,T} \end{pmatrix} = \begin{pmatrix} X \\ y_{p-1,T} \end{pmatrix} \tag{86}$$

and

$$R_{p,T+1}A_{p,T+1} = \begin{pmatrix} R^e_{p,T+1} \\ 0^p \end{pmatrix}. \tag{87}$$

Therefore, $C_{p,T+1}$ is a linear combination of

$$\begin{pmatrix} 0 \\ C_{p-1,T} \end{pmatrix}$$

and $A_{p,T+1}$. From

$$[R^e_{p,T+1}(0^p)^t]C_{p,T+1} = A^t_{p,T+1}R_{p,T+1}C_{p,T+1} = A^t_{p,T+1}y_{p,T+1} = e_{p,T+1}, \tag{88}$$

it is found that

$$C_{p,T+1} = \begin{pmatrix} 0 \\ C_{p-1,T} \end{pmatrix} + e_{p,T+1}R^{-e}_{p,T+1}A_{p,T+1}. \tag{89}$$

Multiplying by $y_{p,T+1}$ gives

$$\gamma_{p,T+1} = \gamma_{p-1,T} + e^2_{p,T+1}R^{-e}_{p,T+1} \tag{90}$$

and

$$\gamma_{p,T+1} = \sum_{i=0}^{p} e_{i,T+i+1-p}R^{-e}_{i,T+i+1-p}. \tag{91}$$

For the time update of $K_{p,T}$, use the definitions (64) and (65):

$$\begin{aligned} k_{p,T+1} &= [k_{p,T+1}(0^p)^tR^r_{p,T}] \begin{pmatrix} A_{p,T} \\ 0 \end{pmatrix} \\ &= (0\ B^t_{p,T})(R_{p+1,T} + y_{p+1,T+1}y^t_{p+1,T+1}) \begin{pmatrix} A_{p,T} \\ 0 \end{pmatrix} \\ &= (0\ B^t_{p,T}) \begin{pmatrix} R^e_{p,T} \\ 0^p \\ k_{p,T} \end{pmatrix} + r_{p,T}e^0_{p,T+1} = k_{p,T} + e^0_{p,T+1}r_{p,T} \end{aligned} \tag{92}$$

Using eqs. (79) and (83) an alternative form is

$$k_{p,T+1} = k_{p,T} + e^0_{p,T+1}r^0_{p,T}(1 - \gamma_{p-1,T}) = k_{p,T} + e_{p,T+1}r^0_{p,T}. \tag{93}$$

REFERENCES

1. R. D. Gitlin and F. R. Magee, Jr., "Self Orthogonalizing Adaptive Equalization Algorithms," IEEE Trans. Commun. *COM-25,* No. 7 (July 1977), pp. 666–72.
2. D. Godard, "Channel Equalization Using Kalman Filter for Fast Data Transmission," IBM J. Res. and Dev., *18,* No. 3, (May 1974), pp. 267–73.
3. D. D. Falconer and L. Ljung, "Application of Fast Kalman Estimation to Adaptive Equalization," Report LiTH-ISY-I-0158, Dept. of Electrical Engineering, Linkoping University, S-58183 Linkoping, Sweden, 1977.
4. L. Ljung, M. Morf, and D. Falconer, "Fast Calculation of Gain Matrices for Recursive Estimation Schemes," Int. J. Control, *27,* No. 1 (January 1978), pp. 1–19.
5. D. D. Falconer and L. Ljung, "Application of Fast Kalman Estimation to Adaptive Equalization," IEEE Trans. Commun., *COM-26,* No. 10 (October 1978), pp. 1439–46.

6. R. W. Chang, "A New Equalizer Structure for Fast Start-Up Digital Communication," B.S.T.J., *50,* No. 6 (July–August 1971), pp. 1969–2014.
7. J. Makhoul, "A Class of All-Zero Lattice Digital Filters: Properties and Applications," IEEE Trans. Acoust., Speech and Sig. Proc., *ASSP-26,* No. 4 (August 1978), pp. 304–14.
8. E. Shichor and J. Reiss, "Sequential Algorithm for Estimation of Parameters of an Autoregressive Process," Proc. IEEE Int. Conf. Acoust., Speech, and Signal Processing, Hartford, Connecticut (May 1977), pp. 744–6.
9. L. J. Griffith, "A Continuously Adaptive Filter Implemented as a Lattice Structure," ibid., pp. 683–6.
10. L. J. Griffiths, "An Adaptive Lattice Structure for Noise Canceling Applications," Proc. IEEE Int. Conf. Acoust., Speech, and Signal Processing, Tulsa, Okla., pp. 87–90, 1978.
11. E. H. Satorious and S. T. Alexander, "Rapid Equalization of Highly Dispersive Channels Using Lattice Algorithms," Naval Ocean Systems Center Tech. Report 249, San Diego, California, April, 1978.
12. M. Morf, "Ladder Forms in Estimation and System Identification," IEEE 11th Annual Asilomar Conf. Circuits, Systems, and Computers, Pacific Grove, California, November 1977, pp. 424–9.
13. M. Morf, A. Vieira, and D. T. Lee, "Ladder Forms for Identification and Speech Processing," Proc. 1977 IEEE Conf. Decision and Control, New Orleans, Louisiana (December 1977), pp. 1074–8.
14. T. L. Lim and M. S. Mueller, "Rapid Equalizer Start-up Using Least Square Algorithms," Int. Conf. Comm., Seattle, Washington, June 1980.
15. E. H. Satorius and J. D. Pack, "Application of Least Squares Lattice Algorithms to Adaptive Equalization," IEEE Trans. Comm., *COM-29,* No. 2 (February 1981), pp. 136–42.