ADAPTIVE ESCALATOR STRUCTURE
FOR LINEAR PREDICTION

by

DAE HEE YOUN

B.S., YONSEI UNIVERSITY, 1977

A MASTER'S REPORT

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Electrical Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1979

Approved by:

Nasir Ahmed
Major Professor

I would like to dedicate this work

to my parents and my wife.

TABLE OF CONTENTS

LIST OF FIGURES

Figure Page

CHAPTER 1

INTRODUCTION

The adaptive linear predictor configuration shown in Fig. 1-1 is a scheme which has found a variety of applications, such as line enhancement [1], spectral estimation [2], speech enhancement [3], and intrusion detecttion [4]. An integral part of the adaptive predictor is a finite impulse response (FIR) filter, whose coefficients (weights) are updated continuously via a variety of algorithms. They can be updated, for example, using information related to the prediction error e(k) (see Fig. 1-1), using Widrow's least-mean-square (LMS) algorithm [1]. Information pertaining to the input correlation matrix $\Sigma_{xx}(k)$ can also be used, as is the case with a class of sequential regression (SER) algorithms [5,6], one of which [5] is also referred to as Godard's algorithm [7]. The LMS and SER filters are usually implemented in the form of a tapped delay line model (see Fig. 1-2), where $g_i(k)$ denotes the i-th filter coefficient (weight) at time k, N is the number of weights, which also equals the number of past values x(k-1), x(k-2),...,x(k-N) used to predict x(k). This class of filters is optimum in that if the input is stationary,[†] then the predictor weights (or their expected value in the LMS case) converge to the Wiener solution.

Since the SER algorithms utilize information related to the input correlation matrix $\Sigma_{xx}(k)$ to update the filter weights, their implementations involve matrix operations. In contrast, the LMS algorithm involves

[†] i.e., the input correlation matrix is positive definite.

THIS BOOK CONTAINS NUMEROUS PAGES WITH DIAGRAMS THAT ARE CROOKED COMPARED TO THE REST OF THE INFORMATION ON THE PAGE.

THIS IS AS RECEIVED FROM CUSTOMER.

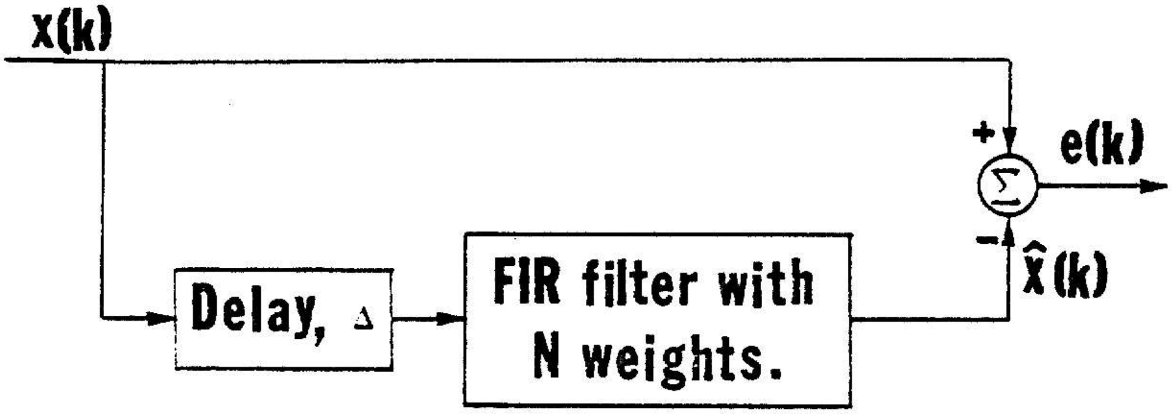


Fig. 1-1. Predictor configuration.

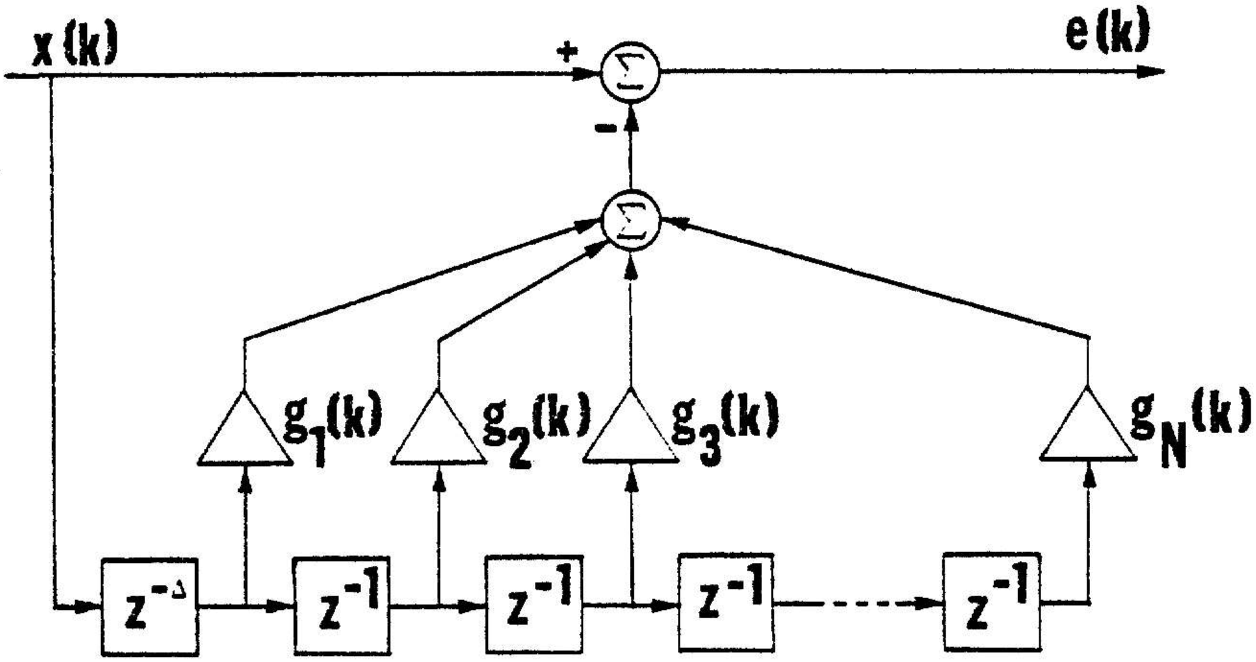


Fig. 1-2. Tapped delay line predictor.

only scalar operations, since no information related to $\Sigma_{xx}(k)$ is used. Thus the LMS algorithm is relatively easier to implement, while SER algorithms possess superior convergence properties.

Another realization for the predictor in Fig. 1-1 is a lattice structure [9-11]. For example, if $\Delta=1$ in Fig. 1-1, then the corresponding lattice structure is shown in Fig. 1-3. The lattice outputs $e_j(k)$ and $w_j(k)$ are called the "forward" and "backward" prediction errors, respectively, at the j-th stage, and $\ell_j(k)$ is the corresponding lattice weight. The error output $e_N(k)$ corresponds to $e(k)$ in Fig. 1-1. Several methods for updating the $\ell_j(k)$ using information related to the forward and reverse prediction errors are discussed in [8-10]. In each case, only scalar operations are necessary. It has been shown that these lattice structures have superior convergence properties relative to the LMS algorithm, since the backward prediction error $w_j(k)$, $j=1,2,\ldots,N$ are mutually uncorrelated. Again, such lattice structures are also optimum in the sense that convergence to the minimum mean-squared-error is achieved if the input is wide-sense stationary.[†]

The main objective of this study is to introduce the notion of escalator structures for implementing the predictor in Fig. 1-1. For the purpose of discussion, we consider the case $\Delta=1$ and $N=3$ in Fig. 1-1. Then the corresponding escalator structure is shown in Fig. 1-4, and will be shown to have the following properties:

(1) It consists of 3 stages corresponding to N=3. In general, there are N stages.

[†] i.e., the input correlation matrix is Toeplitz, in addition to being positive definite.

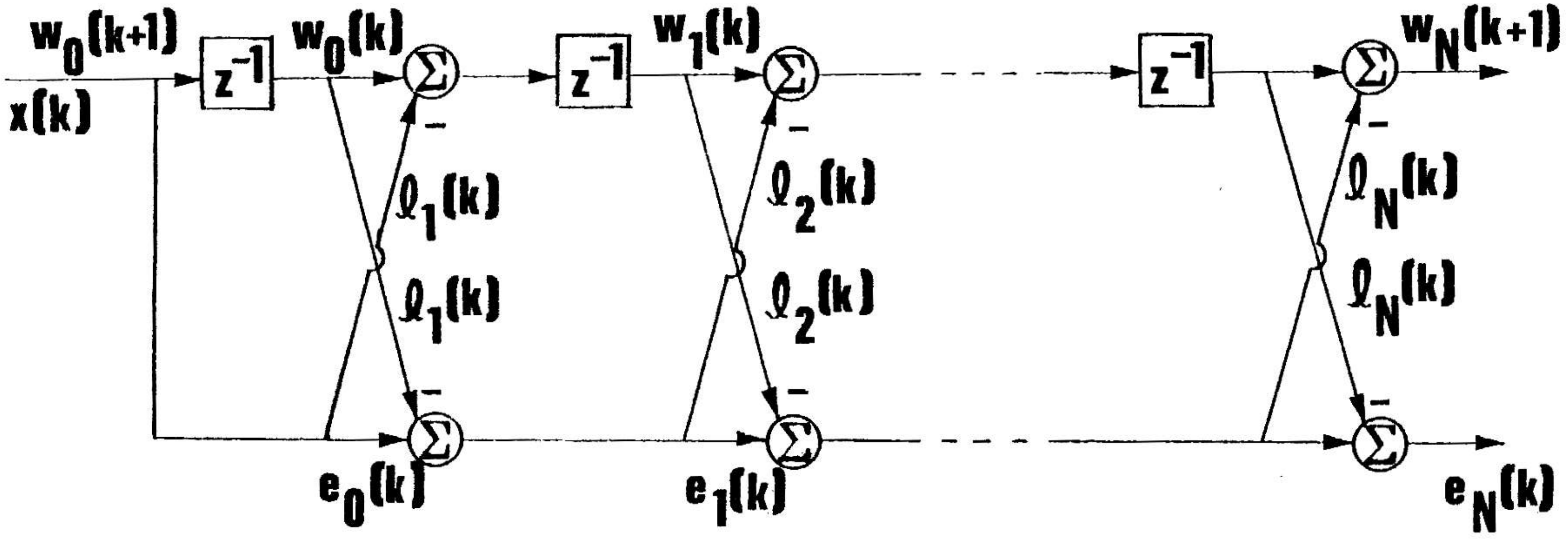


Fig. 1-3. Lattice predictor for the delay parameter Δ=1.

2) The output $y_i(k-m)$ are prediction errors at the i-th stage, as summarized in Fig. 1-4.

3) The prediction errors $y_1(k-2)$, $y_2(k-1)$, and $y_3(k)$ are mutually uncorrelated; in general, the errors $y_i(k-N+i)$, $i=1,2,\ldots,N$ are mutually uncorrelated.

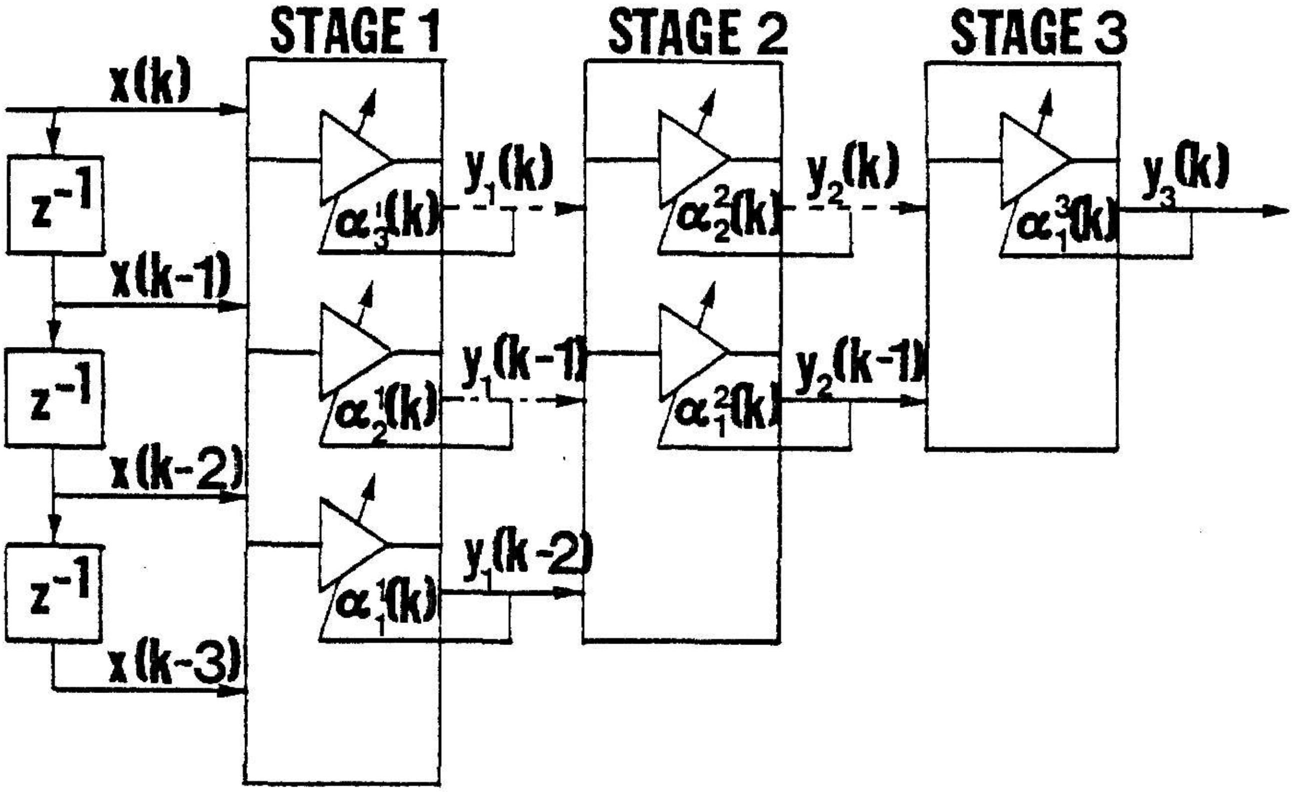
4) The i-th stage has (N+1-i) weights $\alpha_j^i(k)$, $j=1,2,\ldots,N+1-i$. Thus the total number of weights $\alpha_j^i(k)$ is given by

$$N + (N-1) + (N-2) + \ldots + 1 = N(N+1)/2 \tag{1-1}$$

These $\alpha_j^i(k)$ are updated using the steepest descent approach and minimizing the squared values of the errors $y_i(k-m)$, as indicated in Fig. 1-4.

5) The final prediction error is $y_3(k)$, which in the general case is $y_N(k)$, corresponds to e(k) in Fig. 1-1.

From the above discussion it is apparent that the desired prediction error $y_3(k)$ in Fig. 1-4 is attained via an escalator effect using sets of prediction errors associated with (N-1) prior stages. Hence we shall refer to the equations used to update the escalator weights $\alpha_j^i(k)$ as the adaptive escalator predictor (AEP) algorithm. An important property of the AEP algorithm is that if the input is stationary (i.e., $\Sigma_{xx}(k)$ is positive definite), then it is optimum in the sense that the final output mean squared error converges to the minimum mean squared error. In addition, only scalar operations are necessary to update the weights $\alpha_j^i(k)$. These operations are essentially the same as those used to update the lattice weights via an adaptive lattice predictor algorithm proposed in [9].



Stage 1: $y_1(k)$ = error to predict x(k) using x(k-3)

$y_1(k-1)$ = error to predict x(k-1) using x(k-3)

$y_1(k-2)$ = error to predict x(k-2) using x(k-3)

Stage 2: $y_2(k)$ = error to predict x(k) using x(k-2) and x(k-3)

$y_2(k-1)$ = error to predict x(k-1) using x(k-2) and x(k-3)

Stage 3: $y_3(k)$ = error to predict x(k) using x(k-1), x(k-2) and x(k-3)

Fig. 1-4. Prediction considerations via an escalator structure for Δ=1 and N=3.

While 2N update equations are involved in the lattice case, N(N+1)/2 equations are encountered in the case of escalators. However, escalator predictors possess two important advantages over their lattice counterparts. These are as follows:

1) For stationary inputs, they are optimal for a larger class of inputs, in that the input correlation matrix $\Sigma_{xx}(k)$ needs only be positive definite--i.e., optimality is not restricted to wide-sense stationary processes.

2) There are N mutually uncorrelated errors $y_i(k-N+j)$, $i=1,2,...,N$, involved, and hence the convergence rate is not dependent on the eigenvalue distribution of $\Sigma_{xx}(k)$, as is the case with the LMS algorithm. In addition, since the escalator predictor has N(N+1)/2 weights that are updated, compared to 2N weights in the lattice case, the AEP algorithm tends to converge faster. This aspect is illustrated via experimental results presented in Chapter 6.

CHAPTER 2

DIAGONALIZATION OF SYMMETRIC MATRICES

Given a symmetric matrix Σ_{xx}, it can be shown that there exists a nonsingular unit lower triangular (ULT) matrix W, such that $W \Sigma_{xx} W'$ is a diagonal matrix [10,11], which is unique. This W matrix can be computed in the form of a product of N ULT matrices [12-14], as discussed in what follows [13].

A symmetric matrix Σ_{xx} can be partitioned in the form

$$\Sigma_{xx} = \begin{bmatrix} a_{11} & V' \\ & \\ V & A_1 \end{bmatrix} \tag{2-1}$$

where A_1 is a matrix obtained by striking out the first column and the first row,

$$V' = [a_{12}\ a_{13} \cdots a_{1,N+1}] \quad ,$$

prime denoting matrix (vector) transpose, and a_{11} is assumed to be nonzero. Now, let

$$W_1 = \begin{bmatrix} 1 & 0 \\ & \\ -V/a_{11} & I_N \end{bmatrix} \tag{2-2}$$

where I_N is an identity matrix of order N. Then,

$$W_1 \Sigma_{xx} W_1' = \begin{bmatrix} a_{11} & 0 \\ & \\ 0 & B \end{bmatrix} \tag{2-3}$$

where B is a symmetric matrix of order N whose elements b_{ij} are given by

$$b_{ij} = a_{i+1,j+1} - \frac{a_{i+1,1}\, a_{j+1,1}}{a_{11}} \tag{2-3.a}$$

The matrix B is then treated in the same way and the process is continued until Σ_{xx} is reduced to a diagonal form D, such that

$$\begin{aligned} D &= W\, \Sigma_{xx}\, W' \\ &= \text{diag}\ (d_1, d_2, \ldots, d_{N+1}) \end{aligned} \tag{2-4}$$

where $W = W_N\, W_{N-1} \ldots W_2 W_1$ is the desired ULT matrix. This factorization process is now illustrated by an example for the case N=3. Then

$$\Sigma_{xx} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \tag{2-5}$$

where $a_{ij} = a_{ji}, a_{11} \neq 0$ and $i,j = 1,2,3,4$.

From (2-2), we have

$$W_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -\alpha_1^1 & 1 & 0 & 0 \\ -\alpha_2^1 & 0 & 1 & 0 \\ -\alpha_3^1 & 0 & 0 & 1 \end{bmatrix} \tag{2-6}$$

where $\alpha_j^1 = a_{j+1,1}/a_{11}$ and $j = 1,2,3$.

Substitution of (2-5) and (2-6) in (2-3) leads to

$$W_1\, \Sigma_{xx}\, W_1' = \begin{bmatrix} a_{11} & 0 & 0 & 0 \\ 0 & b_{11} & b_{12} & b_{13} \\ 0 & b_{21} & b_{22} & b_{23} \\ 0 & b_{31} & b_{32} & b_{33} \end{bmatrix} \tag{2-7}$$

where b_{ij} can be computed by using (2-3.a).

Similarly, for the submatrix B in (2-7), it follows that

$$W_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -\alpha_1^2 & 1 & 0 \\ 0 & -\alpha_2^2 & 0 & 1 \end{bmatrix}$$

where $\alpha_j^2 = b_{j+1,1}/b_{11}$ and $j = 1,2$, and

$$W_2\ W_1\ \Sigma_{xx}\ W_1'\ W_2' = \begin{bmatrix} a_{11} & 0 & 0 & 0 \\ 0 & b_{11} & 0 & 0 \\ 0 & 0 & c_{11} & c_{12} \\ 0 & 0 & c_{21} & c_{22} \end{bmatrix} \qquad (2\text{-}8)$$

Next, applying the above approach to the submatrix C in (2-8), we obtain

$$W_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -\alpha_1^3 & 1 \end{bmatrix}$$

where $\alpha_1^3 = c_{21}/c_{11}$.

Hence, the desired diagonal matrix D is obtained as the product

$W_3\ W_2\ W_1\ \Sigma_{xx}\ W_1'\ W_2'\ W_3'$; i.e.,

$$D = W\ \Sigma_{xx}\ W'$$

which yields

$$D = \begin{bmatrix} a_{11} & 0 & 0 & 0 \\ 0 & b_{11} & 0 & 0 \\ 0 & 0 & c_{11} & 0 \\ 0 & 0 & 0 & d_{11} \end{bmatrix} \qquad (2\text{-}9)$$

where $W = W_3\ W_2\ W_1$. The above computations are summarized in Fig. 2-1.

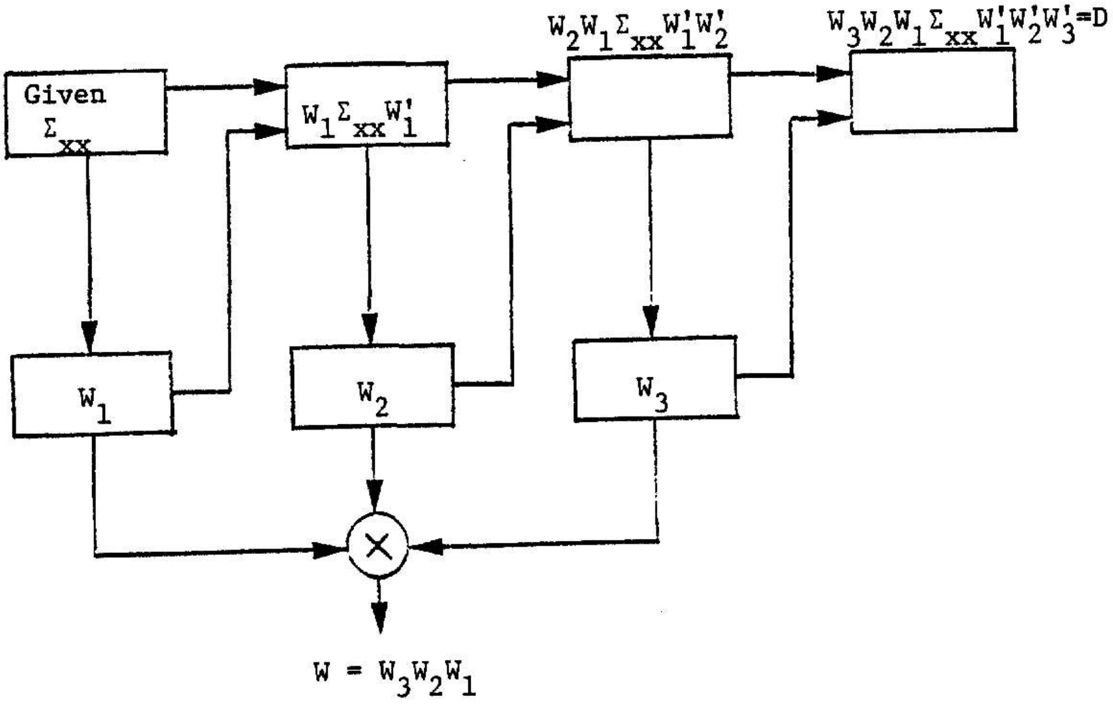


Fig. 2-1. Pertaining to computation of W, W_i and D.

CHAPTER 3

DIAGONALIZATION OF AUTOCOVARIANCE MATRICES

We consider the case when W is a unit lower triangular (ULT) transform matrix, as illustrated in Fig. 3-1.

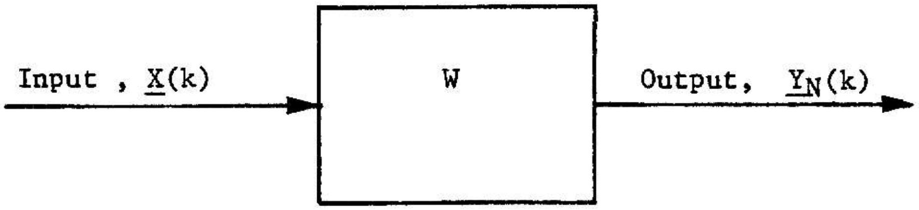


Fig. 3-1. A ULT transform interpretation.

The input vector $\underline{X}'(k) = [x(k-N) \ldots x(k-1)\ x(k)]$ consists of zero-mean random variables $x(k-i)$, $i=0,1,2,\ldots,N$. Thus, the output data vector $\underline{Y}_N(k)$ can be expressed in matrix form as

$$\underline{Y}_N(k) = W\,\underline{X}(k) \quad , \tag{3-1}$$

where

$$W = \begin{bmatrix} 1 & 0 & 0 & \ldots & 0 \\ -w_{1,1} & 1 & 0 & \ldots & 0 \\ -w_{2,2} & -w_{2,1} & 1 & \ldots & 0 \\ \vdots & \vdots & & & \vdots \\ -w_{N,N} & -w_{N,N-1} & \ldots & -w_{N,1} & 1 \end{bmatrix}$$

and $\underline{Y}_N(k) = [y_N(k-N)\ y_N(k-N+1) \ldots y_N(k-1)\ y_N(k)]$.

Let Σ_{xx} denote the input autocorrelation matrix which is assumed to be positive definite. Then, the corresponding output autocorrelation matrix Σ_{yy} is given by

$$\Sigma_{yy} = W \, \Sigma_{xx} \, W' \quad . \tag{3-2}$$

However, from the discussion in the previous chapter, it follows that Σ_{yy} is a diagonal matrix. Hence the components $\underline{Y}_N$ are <u>mutually</u> <u>uncorrelated</u>; i.e.,

$$\begin{aligned} E[y_N(k-i)\; y_N(k-j)] &= d_{ij} \quad ; \quad i=j \\ &= 0 \quad ; \quad i \neq j \end{aligned}$$

for $i,j=0,1,2,\ldots,N$.

Equation (3-1) can alternately be written as

$$\begin{aligned} y_N(k-N) &= x(k-N) \quad , \\ y_N(k-i) &= x(k-i) - \sum_{j=1}^{N-i} w_{N-i,j} \; x(k-i-j) \quad , \end{aligned} \tag{3-3}$$

for $i=0,1,2,\ldots,N-1$,

which represents a one-step delay predictor if $y_N(k-i)$ is treated as an error between $x(k-i)$ and its predicted value $\hat{x}(k-i)$,

$$\hat{x}(k-i) = \sum_{j=1}^{N-i} w_{N-i,j} \; x(k-i-j) \quad , \quad i=0,1,\ldots,N-1 \quad .$$

For the special case when the input is an N-th order Markov process, it has been shown that W in (3-1) is a banded ULT matrix [15]. Again, it can be shown that (see Appendix A) the elements of the (n+1)-th row of W correspond to the optimum (Wiener) coefficients of a one-step delay predictor which uses n past input values. The corresponding filter structure can be realized as a bank of N FIR filters, where (N-i) past input values are used to predict x(k-i). In the chapter that follows, we show that (3-3) can be realized in the form of an escalator structure.

CHAPTER 4

ESCALATOR STRUCTURE DERIVATION

From (2-4) it follows that we can factorize W in (3-1) into N+1 matrices W_i, i=1,2,3,...,N. Hence, we have the implementation shown in Fig. 4-1.

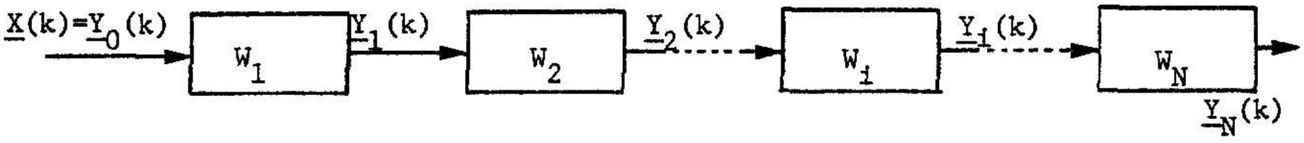


Fig. 4-1. ULT transform implementation in factored form.

We define the following vectors with respect to Fig. 4-1:

$$\underline{Y}_0'(k) = \underline{X}'(k) = [x(k-N) \ \dots \ x(k-1) \ x(k)] \ ,$$

$$\underline{Y}_i'(k) = [y_i(k-N) \ \dots \ y_i(k-1) \ y_i(k)] \ , \qquad (4\text{-}1)$$

for i = 1,2,...,N .

where x(k) is the current input sample, and $y_i(k-j)$, j=0,1,...,N are the output of the i-th stage. The output of the i-th stage at time k is given by

$$\underline{y}_i(k) = W_i \ \underline{Y}_{i-1}(k) \qquad \text{for } i=1,2,\dots,N \ . \qquad (4\text{-}2)$$

From the discussion given in Chapter 2, it follows that W_i is given by

$$W_i = \begin{array}{c} \begin{array}{ccccccc} & & & & \text{i-th column} & & \\ & & & & \downarrow & & \end{array} \\ \left[\begin{array}{ccccccc} 1 & 0 & 0 & \dots & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 & \dots & 0 \\ & & & & \vdots & & \\ & & & & 1 & & \\ \vdots & \vdots & \vdots & & -\alpha_1^i & 1 & \vdots \\ & & & & -\alpha_2^i & 0 & \\ 0 & 0 & 0 & & -\alpha_{N-i}^i & \dots & 1 \end{array}\right] \end{array} \qquad (4\text{-}3)$$

To illustrate, consider the case N=3. Then, the output of the first stage is given by

$$\underline{Y}_1(k) = W_1\ \underline{X}(k)$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ -\alpha_1^1 & 1 & 0 & 0 \\ -\alpha_2^1 & 0 & 1 & 0 \\ -\alpha_3^1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x(k-3) \\ x(k-2) \\ x(k-1) \\ x(k) \end{bmatrix}$$

$$= \begin{bmatrix} x(k-3) \\ x(k-2)\ -\alpha_1^1\ x(k-3) \\ x(k-1)\ -\alpha_2^1\ x(k-3) \\ x(k)\ -\alpha_3^1\ x(k-3) \end{bmatrix} = \begin{bmatrix} y_1(k-3) \\ y_1(k-2) \\ y_1(k-1) \\ y_1(k) \end{bmatrix} \qquad (4\text{-}4)$$

Similarly, the remaining stages can be implemented as illustrated in the signal flowgraph shown in Fig. 4-2.

For example, when N=3, the computations in Fig. 4-2 result in the escalator structure shown in Fig. 4-3. Again, from (3-3), $y_N(k-i)$, $i=0,1,\ldots,N-1$ is the error between x(k-i) and its predicted value $\hat{x}(k-i)$, where

$$\hat{x}(k-i) = \sum_{j=1}^{N-i} w_{N-i}(j)\ x(k-i-j) \quad , \qquad (4\text{-}5)$$

$$i=0,1,\ldots,N-1 \quad ,$$

where $w_{N-i}(j)$ are functions of the α_m^n which are elements of the W_i matrices in (4-2).

$$q_{ij} = w_{j-1,j-i}$$

$$q_{ij} = \alpha_{j-i}^i - \sum_{m=i+1}^{j-1} q_{mj} \qquad \text{for} \quad i < j$$

$$q_{ij} = \alpha_{j-i}^i \qquad \text{for} \quad j=i+1$$

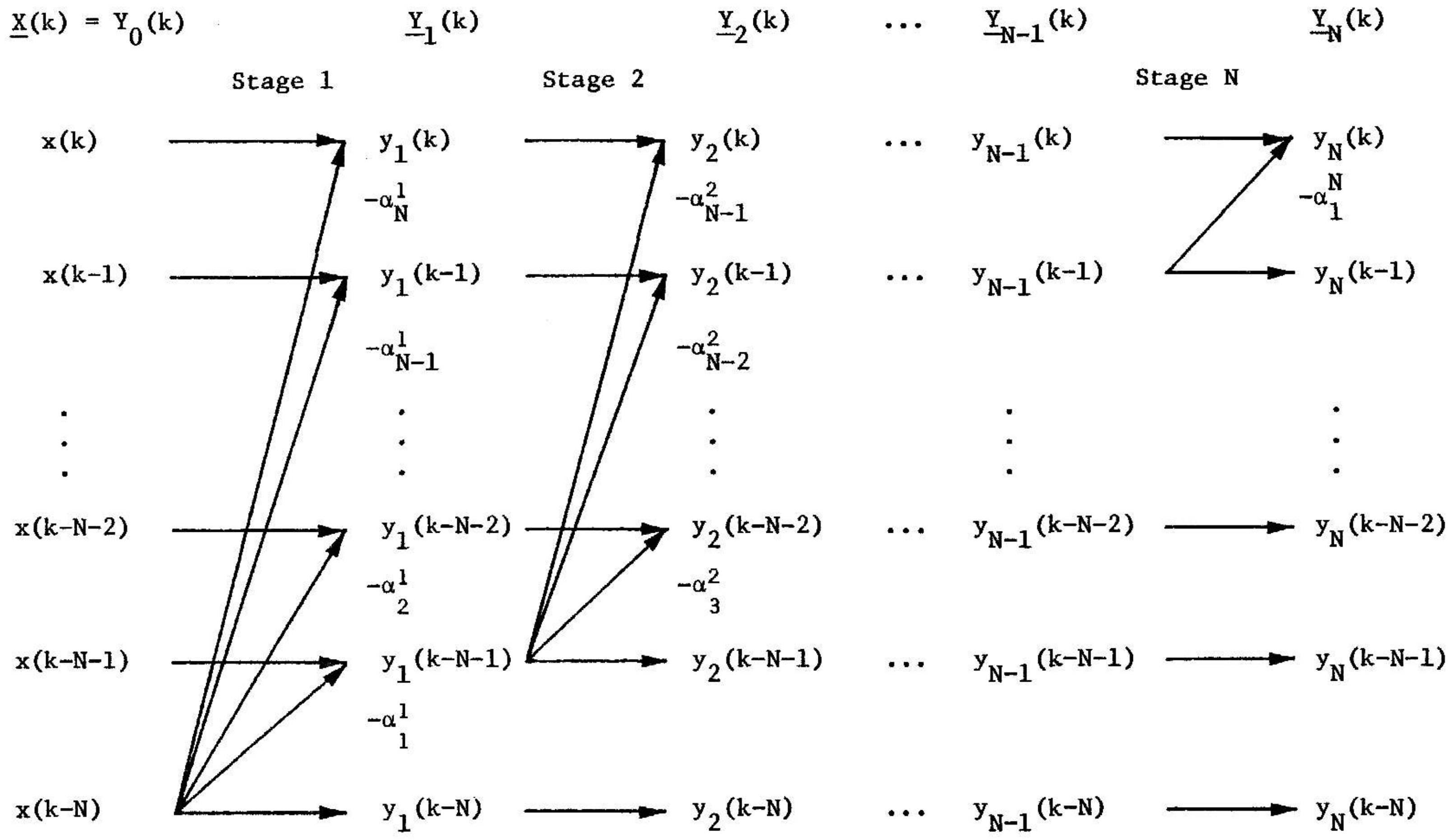


Fig. 4-2. Signal flowgraph to implement the ULT transform in factorized form.

$$q_{ij} = 1 \quad , \quad \text{for } i = j$$
$$q_{ij} = 0 \quad , \quad \text{for } i > j \quad : \quad i,j = 1,2,\ldots,N+1 \quad . \tag{4-6}$$

where q_{ij} are elements of W' .

A fundamental property of the escalator structure is that at the output of stage 1, $y_1(k-N)$, which is same as $x(k-N)$, is uncorrelated with respect to all the other outputs at that stage. Next, $y_2(k-N+1)$, which is the prediction error between $x(k-N+1)$ and its predicted value $\alpha_1^1\, x(k-N)$, is uncorrelated with respect to all other outputs of that stage, etc. Finally, at the output of the N-th stage, the prediction errors $y_N(k-i)$, $i=0,1,\ldots,N$ are mutually uncorrelated with respect to all other outputs at the N-th stage. In other words, the uncorrelated prediction error $y_N(k-i)$, $i=0,1,\ldots,N$, are obtained in N stages, one prediction error per stage.

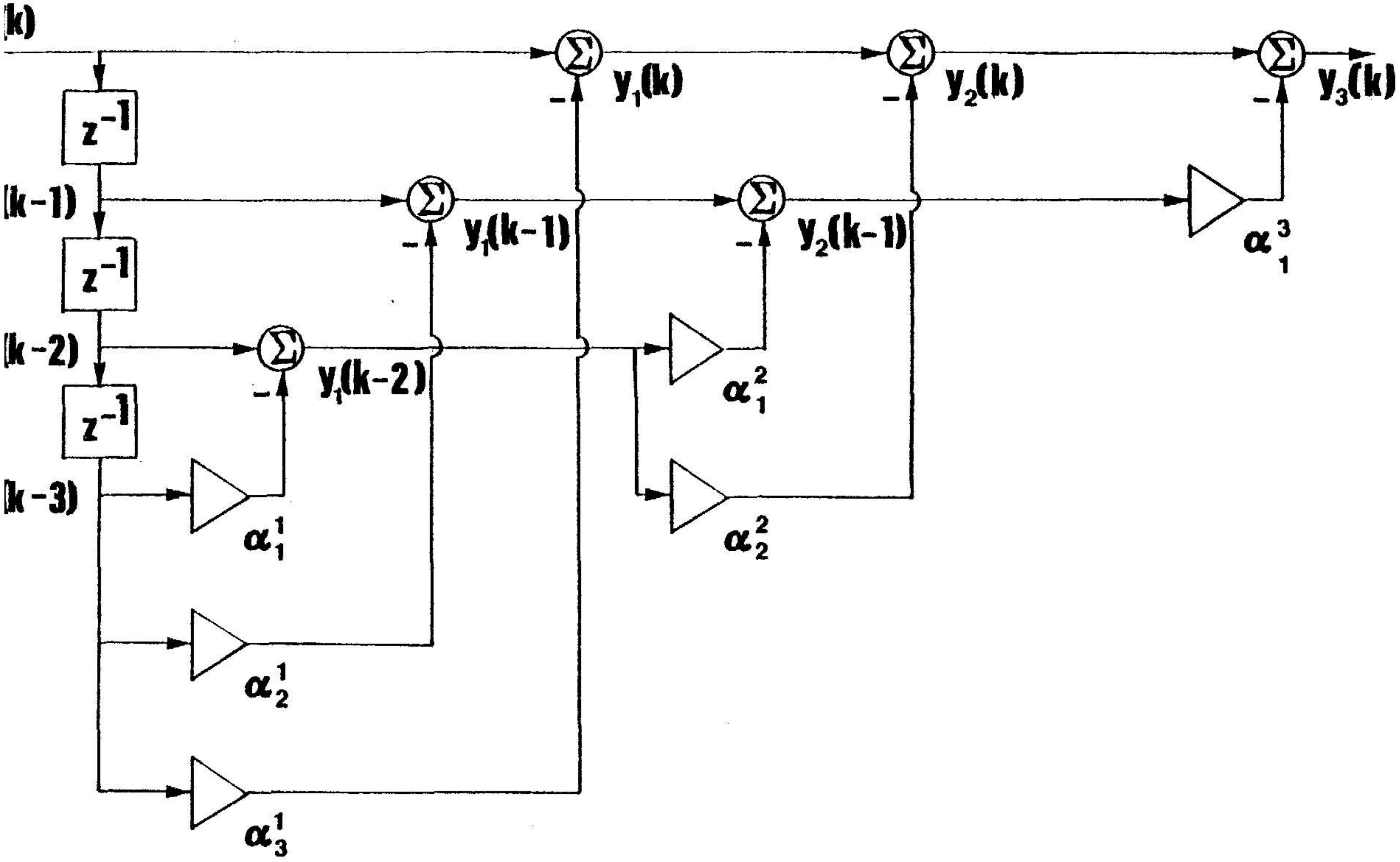


Fig. 4-3. Escalator predictor for N=3 and Δ=1.

CHAPTER 5

ADAPTIVE ESCALATOR STRUCTURES

When the input process is not stationary, the weights of filter must be updated. To this end, we introduce the following notation:

$$y_j(k-i) = y_{j-1}(k-i) - \alpha_m^j(k)\ y_{j-1}(k-n) \qquad (5\text{-}1)$$

for $j=1,2,\ldots,N$, $i=0,1,\ldots,(N-j)$, $m=N-i-j+1$, $n=N-j+1$, and $y_0(\cdot) \equiv x(\cdot)$. In (5-1), we note that there are $N(N+1)/2$ weights $\alpha_m^j(k)$ to be updated. Using the steepest-descent approach, $\alpha_m^j(k)$ is updated as follows:

$$\alpha_m^j(k+1) = \alpha_m^j(k) - \hat{\mu} \frac{\partial}{\partial\ \alpha_m^j(k)}\ y_j^2(k-i) \qquad (5\text{-}2)$$

Substitution of (5-1) in (5-2) results in

$$\alpha_m^j(k+1) = \alpha_m^j(k) + 2\hat{\mu}\ y_j(k-i)\ y_{j-1}(k-n) \quad . \qquad (5\text{-}3)$$

As in the case of adaptive lattice structures, (5-3) will have to be modified so as to account for the variance estimate of the prediction error $y_{j-1}(k-n)$ at the output of the (j-1)-th stage. Thus, we have

$$\alpha_m^j(k+1) = \alpha_m^j(k) + \frac{\mu}{\sigma_y^2(k)}\ [y_j(k-i)\ y_{j-1}(k-n)] \quad . \qquad (5\text{-}4)$$

where μ is a normalized adaptive step-size parameter, and the variance $\sigma_y^2(k)$ is estimated via the relation [8]

$$\sigma_y^2(k+1) = \beta\ \sigma_y^2(k) + (1-\beta)\ y_{j-1}^2(k-n) \quad . \qquad (5\text{-}5)$$

We refer to (5-4) as the adaptive escalator predictor (AEP) algorithm, where $\sigma_y^2(k)$ is updated via (5-5). The block diagram of AEP is drawn in Fig. 5-1.

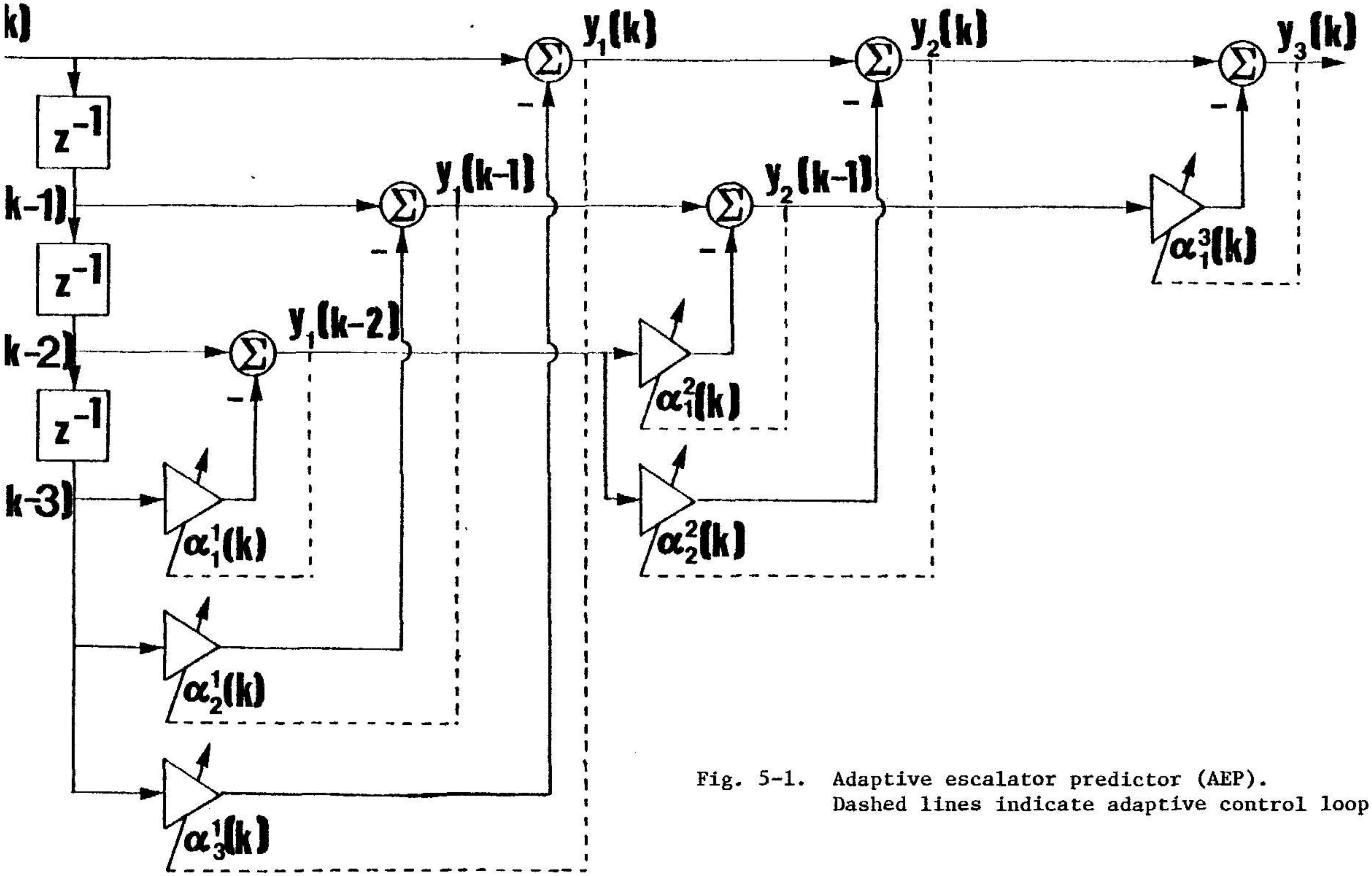


Fig. 5-1. Adaptive escalator predictor (AEP).
Dashed lines indicate adaptive control loop.

CHAPTER 6

EXPERIMENTAL RESULTS

A white noise sequence is sampled at the rate of 128 sps and passed through two 8-pole bandpass filters to generate two band-limited noise sequences. One filter has its cutoff frequencies at 15 Hz and 25 Hz, and the other at 40 Hz and 50 Hz, respectively. They are added to get an input signal whose power density spectrum is shown in Fig. 6-1. For the purpose of comparison, the adaptive lattice predictor (ALP) algorithm [9] is used in addition to the AEP algorithm. In each case, 8 sections (i.e., past input values) are used for prediction purposes.

The parameters μ and β were 0.05 and 0.95, respectively, for the AEP algorithm. Two different values were used for the ALP algorithm. The convergence parameter (α in [10]) and the smoothing parameter (β in [10]) were 0.02 and 0.98, respectively, to get the output shown in Fig. 6-2 (B) and 0.05 and 0.95 for Fig. 6-2 (C). The input and the corresponding outputs are shown in Fig. 6-2.

Next, the results in Fig. 6-2 were used to obtain the "learning curves" displayed in Fig. 6-3. These curves were obtained by computing the running mean-squared-error of the predictor outputs. A window width of 200 samples was used.

From Fig. 6-3, the superior performance of the escalator predictor is apparent relative to that of the lattice predictor, in both the transient and steady-state stages. We can also see that the output in Fig. 6-2 (B) converges faster than that in Fig. 6-2 (C). However, the steady state error in Fig. 6-2 (B) is slightly larger than that in Fig. 6-2 (C).

FIG. 6-1. POWER DENSITY SPECTRUM OF INPUT DATA.

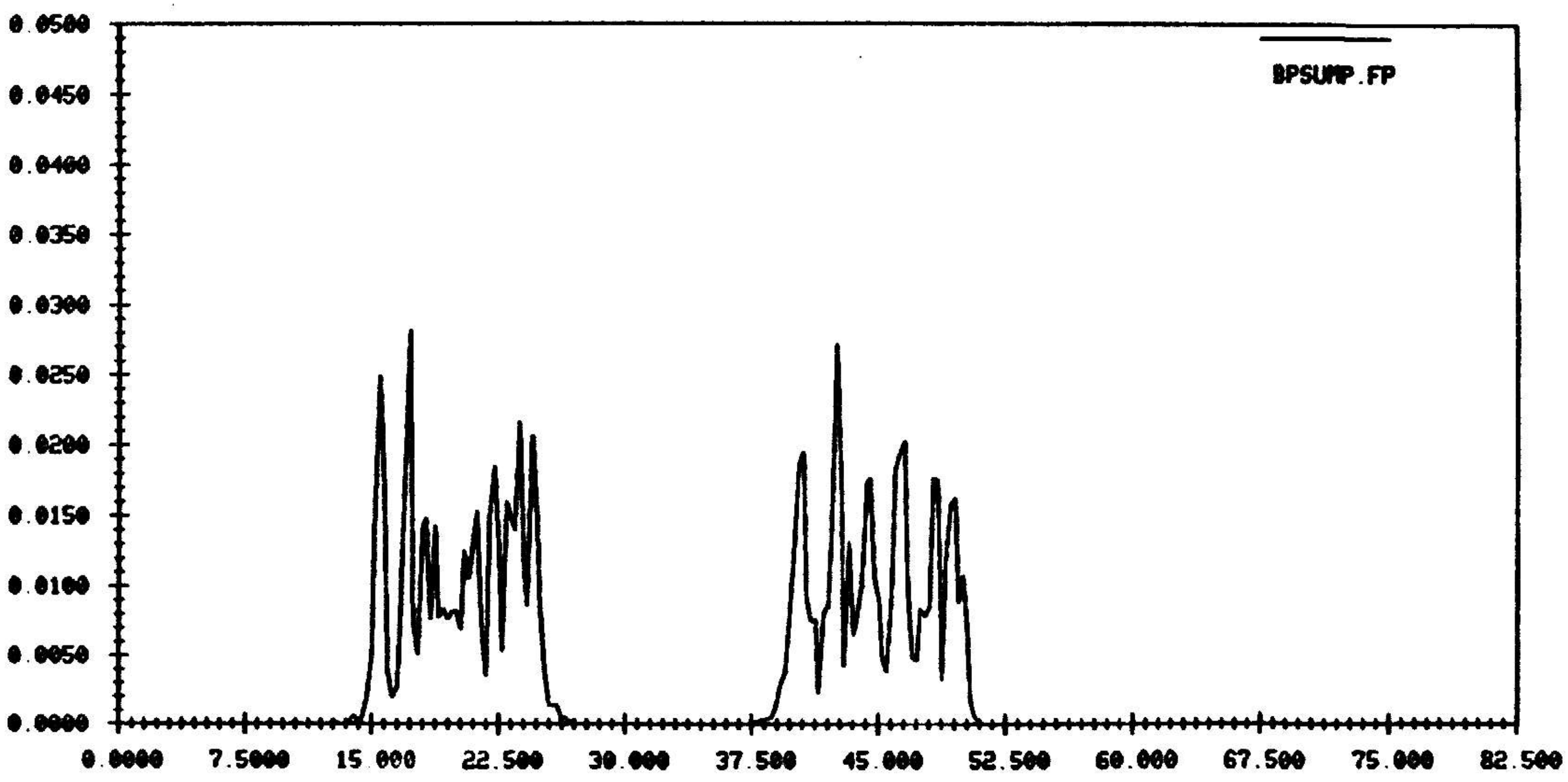
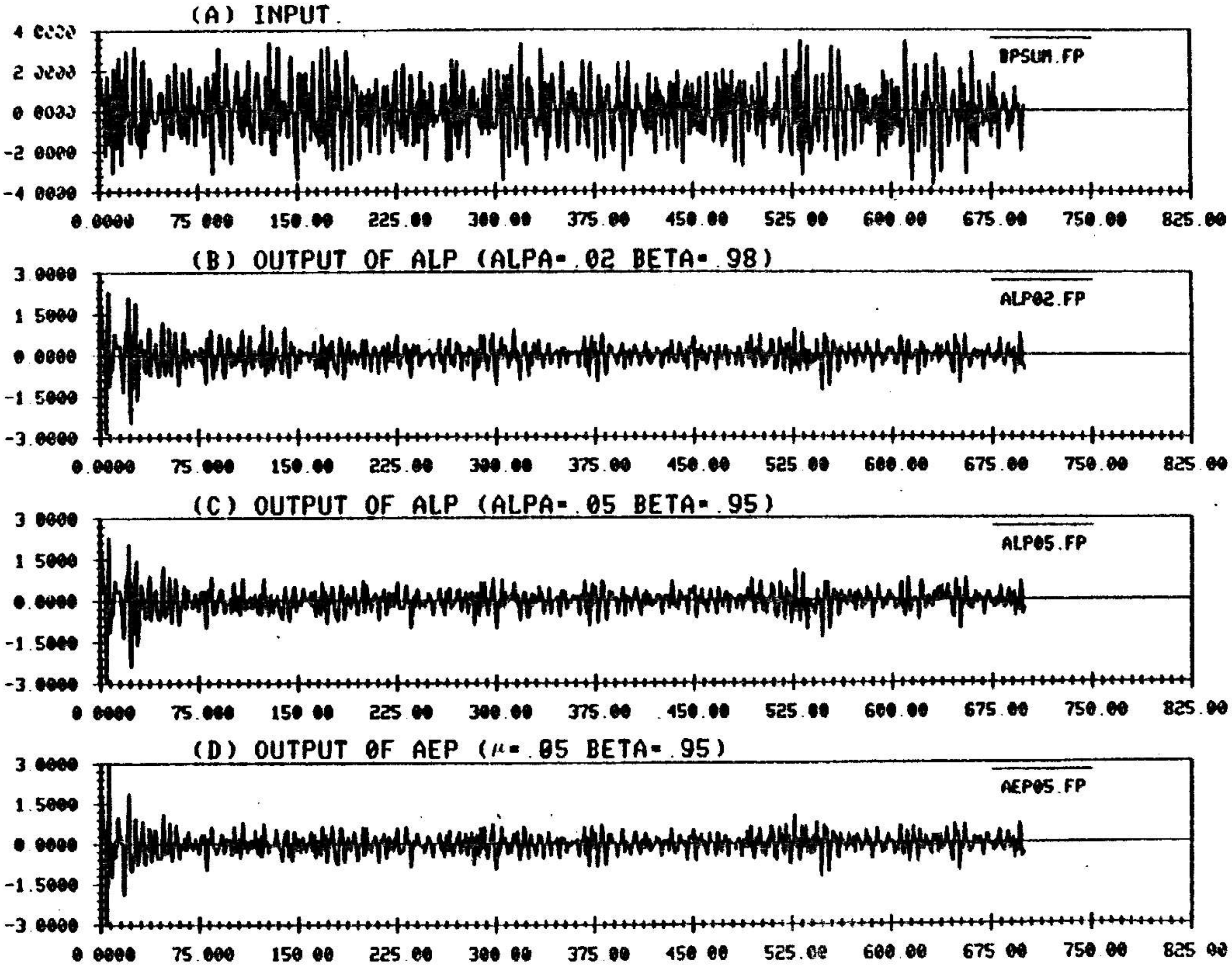


FIG. 6-2. INPUT AND OUTPUTS OF ALP AND AEP.



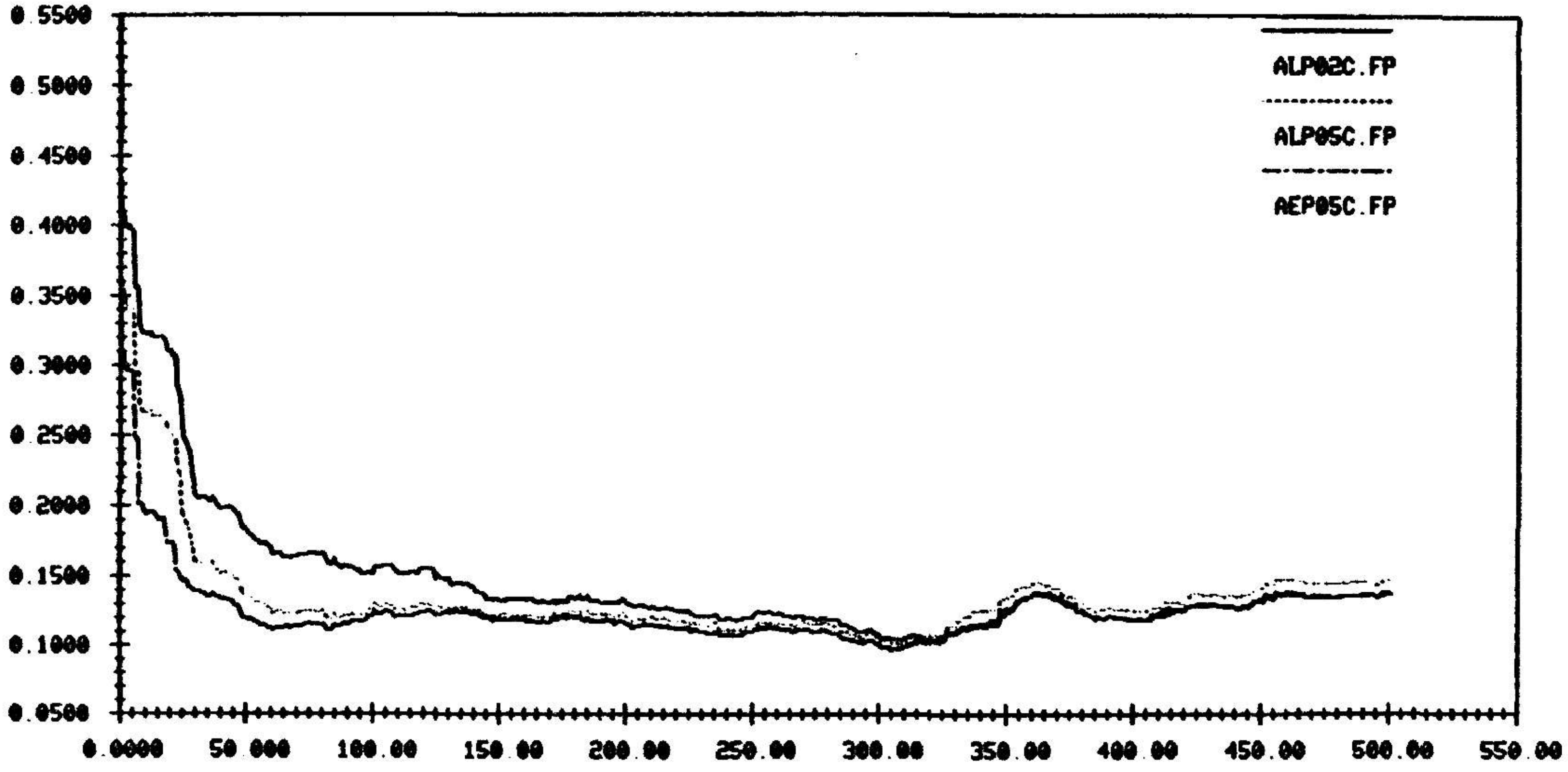
(A) INPUT.

(B) OUTPUT OF ALP (ALPA=.02 BETA=.98)

(C) OUTPUT OF ALP (ALPA=.05 BETA=.95)

(D) OUTPUT 0F AEP (μ=.05 BETA=.95)

FIG. 6-3. LEARNING CURVE.



CHAPTER 7

CONCLUSIONS

The AEP algorithm we have developed is pertinent to an escalator structure which enables one to realize Wiener predictors associated with positive definite correlation matrices; see Appendix A. It is important to note that no matrix operations are necessary to update the escalator weights, as is the case with the LMS and ALP algorithms. However, the experimental results that have been presented demonstrate that the AEP algorithm has superior convergence properties.

We add that another way of realizing Wiener predictors associated with positive definite correlation matrix is via the Gram-Schmidt structure [11]. However, the pertinent updating is done using the final output error, as opposed to local (or intermediate) errors that are used in the escalator structure. Thus, it is reasonable to expect that the AEP algorithm will converge faster than the algorithm proposed in [11].

(Future work will be devoted to deriving escalator structures for the filter and noise cancelling modes, and studying their properties.)

APPENDIX A

The escalator method [16] approach is used to show that the (n+1)-th row elements of W in (3-1) are the optimum (Wiener) weights to predict x(k-N+n) using n past input values.

Now, it is known that the optimum weights for predicting x(k) by using N past values, is given by [1]

$$G_N = -R_N^{-1} P_N$$

where $G'_N = [g_{1,N}\ g_{2,N} \cdots g_{N,N}]$ is the optimum weight vector,

$$R_N = E[\underline{X}(k-1)\ \underline{X}(k-1)'] \quad ,$$

$$P_N = E[x(k)\ \underline{X}(k-1)] \quad , \tag{A-1}$$

and $\underline{X}(k-1)' = [x(k-N) \ \ldots\ x(k-2)\ x(k-1)]$.

Thus, to predict x(k-N+n) by using n past values x(k-N+j), j=0,1,...,(n-1), we need n weights given by

$$G_n = -R_n^{-1} P_n \tag{A-2}$$

where $G'_n = [g_{1,n}\ g_{2,n} \cdots g_{n,n}]$,

$$R_n = E[\underline{X}(k-m)\ \underline{X}(k-m)'] \quad ,$$

$$P_n = E[x(k-N+m)\ \underline{X}(k-m)] \quad ,$$

for m = N-n+1, and $\underline{X}(k-m)' = [x(k-N) \ \ldots\ x(k-m)]$.

It is observed that the elements of G_n, n=1,2,...,N are the solutions of the following system of n equations:

$$\begin{aligned} a_{11}g_{1,n} + a_{12}g_{2,n} + \cdots + a_{1,n}g_{n,n} + a_{1,n+1} &= 0 \\ a_{21}g_{1,n} + a_{22}g_{2,n} + \cdots + a_{2,n}g_{n,n} + a_{2,n+1} &= 0 \\ &\vdots \\ a_{n,1}g_{1,n} + a_{n,2}g_{2,n} + \cdots + a_{n,n}g_{n,n} + a_{n,n+1} &= 0 \quad , \ n=1,2,\ldots,N \end{aligned} \tag{A-3}$$

where $\quad a_{ij} = E[x(k-(N-i+1))\ x(k-(N-j+1))]$

$\quad$ for $i = 1,2,\ldots,n$ and $j = 1,2,\ldots,(n+1)$.

With n=2, for example, (A-3) yields

$$\begin{aligned} &a_{11}g_{12} + a_{12}g_{21} + a_{13} = 0 \\ &a_{21}g_{12} + a_{22}g_{22} + a_{23} = 0 \end{aligned} \qquad \text{(A-3.a)}$$

where

$$a_{11} = E[x^2(k-N)]\ ,\ a_{12} = a_{21} = E[x(k-N)\ x(k-N+1)]\ ,$$

$$a_{22} = E[x^2(k-N+1)]\ ,\ a_{13} = E[x(k-N)\ x(k-N+2)]\ , \text{ and}$$

$$a_{23} = E[x(k-N+1)\ x(k-N+2)]\ .$$

We now define a unit upper triangular matrix Z whose elements are solutions of (A-3) for n=1,2,...,N , i.e.,

$$Z = \left[\begin{array}{ccccccc} & & & & (n+1)\text{-th column} & & \\ & & & & \downarrow & & \\ 1 & g_{11} & g_{12} & \cdots & g_{1,n} & \cdots & g_{1,N} \\ & 1 & g_{22} & \cdots & g_{2,n} & \cdots & g_{2,N} \\ & & 1 & \cdots & \vdots & & \vdots \\ & & & \ddots & g_{n,n} & & \\ & \bigcirc & & & 1 & & \\ & & & & & \ddots & g_{N,N} \\ & & & & & & 1 \end{array}\right] \qquad \text{(A-4)}$$

Then, from the escalator method [16] it follows that Z has the following property:

$$Z' \Sigma_{xx} Z = \tilde{D} \tag{A-5}$$

where $\Sigma_{xx} = E[\underline{X}(k)\ \underline{X}(k)']$ and $\tilde{D}$ is a diagonal matrix. Comparing (A-5) with the general form of (2-4) and using the fact that the ULT matrix W is unique for a given positive definite Σ_{xx}, we conclude that Z' = W. Thus we have the desired result

$$g_{i,n} = w_{n,n-i+1}$$

for $i=1,2,\ldots,n$ and $n=1,2,\ldots,N$.

APPENDIX B

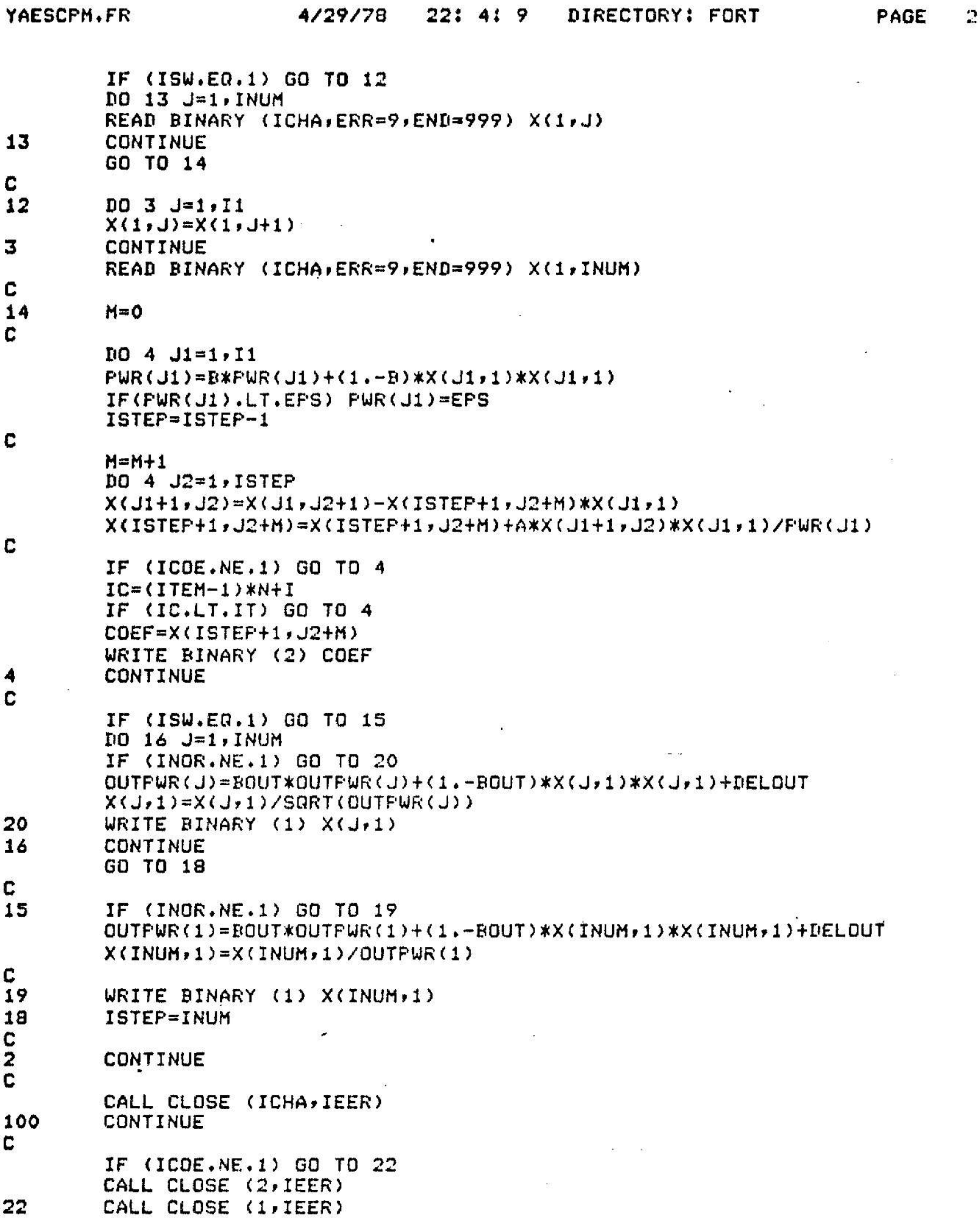
FORTRAN program to simulate the AEP algorithm as described in Chapter 5. Work was done with NOVA 1200 in the Department of Electrical Engineering, Kansas State University.

ILLEGIBLE DOCUMENT

THE FOLLOWING DOCUMENT(S) IS OF POOR LEGIBILITY IN THE ORIGINAL

THIS IS THE BEST COPY AVAILABLE

```
C       ****************************************
C
C       YAESCPM.FR
C
C       ISW=1 : INPUT DATA IS SHIFTED 1-BY-1
C          =0 : INPUT DATA IS VECTOR FORM
C             : BOTH OF THEM CAN BE NORMALIZED
C
C       ****************************************
C
C
        DIMENSION X(30,30),PWR(30),OUTPWR(30)
C
        CALL QUERY ('WANT COEF. ? (YES/NO)',ICOE)
        CALL QUERY ('NORMALIZE THE OUTPUT ? (YES/NO)',INOR)
C
        ACCEPT '# OF ITERATIONS OF SAME DATA SEQ. (ITER) ?',ITER
        ACCEPT 'SEQUENCE (1) OR BLOCK (0) DATA ? (ISW)',ISW
        ACCEPT 'NUMBER OF PAST INPUT DATA (I1) ?',I1
        ACCEPT 'HOW MANY DATA (N) ?',N
        ACCEPT 'VALUE OF EPSILON (EPS) ?',EPS
        ACCEPT 'VALUE OF ALPA (A) ?',A
        ACCEPT 'VALUE OF BETA (B) ?',B
C
        IF (INOR.NE.1) GO TO 21
        ACCEPT 'VALUE OF DELOUT (DELOUT) ?',DELOUT
        ACCEPT 'VALUE OF BOUT (BOUT) ?',BOUT
C
21      CONTINUE
        IF (ICOE.NE.1) GO TO 11
        ACCEPT 'EXECUTION TIME OF COEF. (IT) ?',IT
        CALL IOPEN (2,3,2,1,0,'OUTPUT FILE NAME FOR COEF. ?')
11      CONTINUE
        CALL IOPEN (1,3,2,1,0,'OUTPUT FILE NAME ?')
C
        ISTEP=I1+1
        INUM=ISTEP
C
        DO 1 J1=1,INUM
        PWR(J1)=0.
        OUTPWR(J1)=0.
        DO 1 J2=1,INUM
        X(J1,J2)=0.
1       CONTINUE
C
        IF (ISW.EQ.0) N=N/(I1+1)
C
C
        DO 100 ITEM=1,ITER
        ICHA=ITEM+2
        CALL IOPEN (ICHA,1,2,1,0,'INPUT FILE NAME ?')
C
        DO 2 I=1,N
C
```



```
YAESCPM.FR               4/29/78    22: 4: 9   DIRECTORY: FORT           PAGE   2

        IF (ISW.EQ.1) GO TO 12
        DO 13 J=1,INUM
        READ BINARY (ICHA,ERR=9,END=999) X(1,J)
13      CONTINUE
        GO TO 14
C
12      DO 3 J=1,I1
        X(1,J)=X(1,J+1)
3       CONTINUE
        READ BINARY (ICHA,ERR=9,END=999) X(1,INUM)
C
14      M=0
C
        DO 4 J1=1,I1
        PWR(J1)=B*PWR(J1)+(1.-B)*X(J1,1)*X(J1,1)
        IF(PWR(J1).LT.EPS) PWR(J1)=EPS
        ISTEP=ISTEP-1
C
        M=M+1
        DO 4 J2=1,ISTEP
        X(J1+1,J2)=X(J1,J2+1)-X(ISTEP+1,J2+M)*X(J1,1)
        X(ISTEP+1,J2+M)=X(ISTEP+1,J2+M)+A*X(J1+1,J2)*X(J1,1)/PWR(J1)
C
        IF (ICOE.NE.1) GO TO 4
        IC=(ITEM-1)*N+I
        IF (IC.LT.IT) GO TO 4
        COEF=X(ISTEP+1,J2+M)
        WRITE BINARY (2) COEF
4       CONTINUE
C
        IF (ISW.EQ.1) GO TO 15
        DO 16 J=1,INUM
        IF (INOR.NE.1) GO TO 20
        OUTPWR(J)=BOUT*OUTPWR(J)+(1.-BOUT)*X(J,1)*X(J,1)+DELOUT
        X(J,1)=X(J,1)/SQRT(OUTPWR(J))
20      WRITE BINARY (1) X(J,1)
16      CONTINUE
        GO TO 18
C
15      IF (INOR.NE.1) GO TO 19
        OUTPWR(1)=BOUT*OUTPWR(1)+(1.-BOUT)*X(INUM,1)*X(INUM,1)+DELOUT
        X(INUM,1)=X(INUM,1)/OUTPWR(1)
C
19      WRITE BINARY (1) X(INUM,1)
18      ISTEP=INUM
C
2       CONTINUE
C
        CALL CLOSE (ICHA,IEER)
100     CONTINUE
C
        IF (ICOE.NE.1) GO TO 22
        CALL CLOSE (2,IEER)
22      CALL CLOSE (1,IEER)
```

```
C
        STOP
9       TYPE'ERROR IN READ'
999     CONTINUE
        STOP
        END
```

REFERENCES

[1] B. Widrow, *et al.*, "Adaptive noise cancelling: Principles and Applications," *Proc. IEEE*, Vol. 63, No. 12, pp. 1692-1975, Dec. 1975.

[2] L. J. Griffiths, "Rapid measurement of digital instantaneous frequency," *IEEE Trans. Acoust., Speech and Sig. Proc.*, Vol. ASSP-23, pp. 207-222, Apr. 1975.

[3] M. R. Sambur, "Adaptive noise cancelling for speech signals," *IEEE Trans. Acoust. Speech and Sig. Proc.*, Vol. ASSP-26, pp. 419-423, Oct. 1978.

[4] N. Ahmed, *et al.*, "On an intrusion-detection approach via adaptive prediction," to appear in the *IEEE Trans. Aerospace and Electronic Systems*.

[5] D. Soldan and N. Ahmed, "On a sequential regression predictor," *Electronics Letters*. Vol. 13, pp. 446-447, July 1977.

[6] N. Ahmed, D. R. Hummels, and M. Uhl, "A short-term sequential regression algorithm," to appear in the *IEEE Trans. Acoust., Speech and Sig. Proc*.

[7] R. D. Gitlin and F. R. Magee, Jr., "Self-orthogonalizing adaptive equalization algorithms," *IEEE Trans. Commun.*, Vol. COM-25, No. 7, pp. 666-672, July 1977.

[8] L. J. Griffiths, "A continuously-adaptive filter implemented as a lattice structure," Proc. ICASSP, Hartford, CT, May 1977, pp. 683-686.

[9] L. J. Griffiths, "An adaptive lattice structure for noise-cancelling applications," Proc. ICASSP, Tulsa, OK, Apr. 1978, pp. 87-90.

[10] M. Makhoul, "A class of all-zero lattice digital filters: Properties and applications," *IEEE Trans. Acoust., Speech and Sig. Proc.*, Vol. ASSP-26, Aug. 1978, pp. 304-314.

[11] L. J. Griffiths, "Adaptive structures for multiple-input noise cancelling applications," Proc. ICASSP. Washington, D.C., Apr. 1979, pp. 925-928.

[12] H. Margenau and G. M. Murphy, The Mathematics of Physics and Chemistry. New York: D. Van Nostrand, 1943, pp. 307-309.

[13] G. Hadley, Linear Algebra, Reading, MA: Addison-Wesley, 1964, pp. 253-259.

[14] R. A. Beaumont, Linear Algebra, New York: Harcourt, Brace & World, Inc., 1965, pp. 195-201.

[15] A. Habibi and R. S. Hershel, "A unified representation fo differential pulse-code modulation (DPCM) and transform coding systems," IEEE Trans. Commun., May 1974, pp. 642-646.

[16] D. K. Faddeev and V. N. Faddeeva, Computational Methods of Linear Algebra, W. H. Freeman and Co., 1963, pp. 168-171.

ACKNOWLEDGEMENT

I would like to take this opportunity to thank all those who assisted me during my study. I would especially like to thank Dr. Nasir Ahmed for his encouragement and guidance, along with Dr. Donald R. Hummels and Dr. Leonard E. Fuller for serving on my committee. I am also grateful to the Electrical Engineering Department of Kansas State University for its financial support.

The continuing encouragement received from my wife, Young, and my parents, has made this report a reality. Finally, I would like to thank Dakshesh Parikh for his assistance and encouragement.

ADAPTIVE ESCALATOR STRUCTURE FOR LINEAR PREDICTION

by

DAE HEE YOUN

B.S., YONSEI UNIVERSITY, 1977

AN ABSTRACT OF A MASTER'S REPORT

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Electrical Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1979

ABSTRACT

In this report, an escalator structure for linear prediction of a process whose correlation matrix is positive definite, is introudced. A corresponding adaptive escalator predictor (AEP) algorithm is also developed. If N past input values are used for prediction purposes, the AEP algorithm requires N(N+1)/2 escalator weights to be updated. However, the updating process requires no matrix equations, as is the case with adaptive lattice structures. However, while lattice structures are realized assuming that the input correlation matrix is Toeplitz, escalator structures require that it need only be positive definite. Thus, escalator structures correspond to larger class of Wiener solution.