(4)

Lattice Algorithms for Computing QR and Cholesky Factors in Least Squares Theory of Linear Predict

Cédric J. DEMEURE and Louis L. SCHARF

Department of Electrical and Computer Engineering

UNIVERSITY OF COLORADO

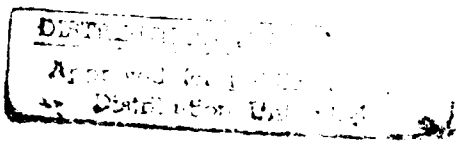
BOULDER, COLORADO

(4)

Lattice Algorithms for Computing QR and Cholesky Factors in the Least Squares Theory of Linear Prediction

Cédric J. DEMEURE and Louis L. SCHARF

TECHNICAL REPORT
SEPTEMBER 1987



Lattice Algorithms for Computing QR and Cholesky Factors in the Least Squares Theory of Linear Prediction†

Cédric J. DEMEURE, Student Member IEEE

Louis L. SCHARF, Fellow IEEE

Electrical and Computer Engineering Department

University of Colorado

Campus Box 425

Boulder CO 80309.

Tel. (303) 492 8283

Abstract‡

In this paper we pose a sequence of linear prediction problems that are a little different from those previously posed. By solving the sequence of problems we are able to QR factor data matrices of the type usually associated with correlation, pre and post-windowed, and covariance methods of linear prediction. Our solutions cover the forward, backward and forward-backward problems. The QR factor orthogonalizes the data matrix and solves the problem of Cholesky factoring the experimental correlation matrix and its inverse. This means we may use generalized Levinson algorithms to derive generalized QR algorithms, which are then used to derive generalized Schur algorithms. All three algorithms are true lattice algorithms that may be implemented either on a vector machine or on a multi-tier lattice, and all three algorithms generate generalized reflection coefficients that may be used for filtering or classification.

† This work was supported by the Office of Naval Research, Arlington, VA under contract N00014-85-K-0256.

‡ Permission to publish this abstract separately is granted.

I. Introduction

In this paper we pose a sequence of least squares problems from the theory of linear prediction. The problems are a little different from those originally posed in the paper by Morf, et al. [6]. The solution to this sequence of problems produces QR factorizations of the kinds of data matrices that are usually associated with the covariance, pre-windowed, post-windowed and correlation methods of linear prediction. By QR factorization we mean the computation of an orthogonal matrix Q and an upper triangular matrix R such that $Y = QR$. That factorization is very often used to solve an overdermined system of equations $Ya = b$ in the least squares sense by $Ra = Q^T b$. The results apply to forward, backward or forward-backward linear prediction problems. We interpret the QR factorization in analysis, synthesis and orthogonalizing terms, and use it to generate Cholesky factors of the experimental covariance matrix (the Grammian of the data matrix) and its inverse. Then we use the generalized Levinson recursions derived by Friedlander et al. [1] to derive generalized recursions for computing the orthogonal matrix in the QR factorization of any of the Toeplitz (or concatenation of Toeplitz) matrices that can arise in linear prediction. These recursions generalize those first discovered by Cybenko [13] for the correlation method of linear prediction. We then use these recursions to derive generalized Schur recursions for Cholesky factoring any of the close-to-Toeplitz covariance matrices that can arise in linear prediction.

All of our generalized recursions are true lattice recursions that generate generalized reflection coefficients. All three algorithms may be implemented on a vector machine or on a multi-tier lattice. In a procedure similar to that of Robinson and Treitel [7] in the scalar Toeplitz case, and Friedlander [8] in the multi-variable Toeplitz case, we use the autoregressive lattice filter associated with the generalized reflection coefficients to complete the QR factorization with the computation of the upper triangular matrix.

Finally, we show how our generalized Levinson, QR, and Schur algorithms may be extended to multivariable (or vector) linear prediction problems. The multivariable results may be applied to a variety of multidimensional problems, as well.

II. Least Squares Problems in Linear Prediction

Let $\underline{y} = [y_0, y_1, \ldots, y_{N-1}]^T$ denote an N sample snapshot of the stationary time series

$\{y_t\}$. From this snapshot, we would like to identify an autoregressive or whitening model for the time series $\{y_t\}$. This model takes the form

$$\sum_{i=0}^{n} a_i^n y_{t-i} = u_t^n \quad ; \qquad a_0^n = 1. \tag{II.1}$$

where $\{u_t^n\}$ is a white sequence with zero mean and variance σ_n^2. The interpretation is that the digital filter $A_n(z) = \sum_{i=0}^{n} a_i^n z^{-i}$ whitens the time series $\{y_t\}$. The whitening model may be written as the predictor model

$$y_t = \widehat{y}_t + u_t^n \qquad\qquad \widehat{y}_t = -\sum_{i=1}^{n} a_i^n y_{t-i} \tag{II.2}$$

The variance of the white sequence $\{u_t\}$, or equivalently the mean squared error between y_t and the one-step ahead predictor $\widehat{y}_t$, is σ_n^2.

Prediction : Our procedure for identifying a model $A_n(z)$ will be to form a sequence of predictions of the form

$$\widehat{y}_t^p = -\sum_{i=1}^{p} a_i^p y_{t-i} \quad ; \quad a_0^p = 1 \tag{II.3}$$

and to let the predictor order range from $p = 0$ to $p = n$. The error between y_t and the p^{th} order predictor $\widehat{y}_t^p$ is, of course,

$$u_t^p = y_t - \widehat{y}_t^p = \sum_{i=0}^{p} a_i^p y_{t-i} \tag{II.4}$$

We shall be interested in a window of these errors for which the time index t satisfies the condition

$$k \leq t + (n - p) \leq l. \tag{II.5}$$

As the predictor order increases from $p = 0$ to $p = n$, the window of length $(l - k + 1)$ moves from left to right across the data set, as illustrated in Figure 1. The indexes k and l may be chosen to select among the various techniques of linear prediction. The data values outside the range $[0, \ldots, N-1]$ are set to zero. In the covariance method of linear prediction, $k = n$ and $l = N - 1$, in the correlation method, $k = 0$ and $l = N - 1 + n$, as illustrated in Figure 2.

Let us write out the error equations, over the window just defined, for the p^{th} order predictor as follows :

$$\begin{bmatrix} y_{l-n} & \cdots & y_{l-1} & y_l \\ \vdots & \ddots & & y_{l+1} \\ & \ddots & & \vdots \\ \vdots & & & y_{l+n} \\ y_k & & & \vdots \\ \vdots & \ddots & & \\ & & \ddots & y_{k+1} \\ y_{k-n} & \cdots & y_{k-1} & y_k \end{bmatrix} \begin{bmatrix} a_p^p \\ \vdots \\ a_1^p \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} u_{l+n-p}^p \\ u_{l+n-p-1}^p \\ \vdots \\ \\ \\ \\ \vdots \\ u_{k-n-p+1}^p \\ u_{k-n-p}^p \end{bmatrix} \tag{II.6}$$

The compact notation is

$$\Upsilon \begin{bmatrix} \underline{a}^p \\ \underline{0} \end{bmatrix} = \Upsilon A^p = U^p \tag{II.7}$$

This scheme may be reproduced for $p = 0$ to $p = n$ to obtain the set of equations

$$\Upsilon A = U \tag{II.8}$$

where Υ is the Toeplitz data matrix just defined, and the matrices A and U are given by

$$A = \begin{bmatrix} 1 & a_1^1 & a_2^2 & \cdots & a_n^n \\ & 1 & a_1^2 & & a_{n-1}^n \\ & & 1 & & \\ & & & \ddots & \vdots \\ & & & 1 & a_1^n \\ & & & & 1 \end{bmatrix} = [A^0, \ldots, A^p, \ldots, A^n \tag{II.9}$$

$$U = \begin{bmatrix} u_{l-n}^0 & \cdots & u_{l-n-p}^p & \cdots & u_l^n \\ u_{l-n+1}^0 & & & & u_{l-1}^n \\ \vdots & & & & \vdots \\ & & & & \\ \vdots & & & & \vdots \\ u_{k-n+1}^0 & & & & u_{k+1}^n \\ u_{k-n}^0 & \cdots & u_{k-n+p}^p & \cdots & u_k^n \end{bmatrix} = [U^0, \ldots, U^p, \ldots, U^n \tag{II.10}$$

Least Squares : The Grammian of the error matrix U is

$$U^T U = A^T \Upsilon^T \Upsilon A \tag{II.11}$$

The pp element of the Grammian is

$$\sigma_p^2 = (U^p)^T U^p = [(a^p)^T 0^T] Y^T Y \begin{bmatrix} a^p \\ 0 \end{bmatrix} = (A^p)^T Y^T Y A^p, \qquad (II.12)$$

which is just the accumulated squared prediction error over the window defined for the predictor of order p. This squared error may be minimized with respect to the coefficients $\underline{a}^p$ under the constraint that $(\underline{a}^p)^T \underline{\delta} = a_0^p = 1$. The appropriate regression equation is

$$\nabla_{\underline{a}^p} \left([(\underline{a}^p)^T 0^T] Y^T Y \begin{bmatrix} a^p \\ 0 \end{bmatrix} - \lambda (\underline{a}^p)^T \underline{\delta} \right) = 0 \qquad (II.13)$$

The solution for $\underline{a}^p$ is then

$$[I \;\; 0] Y^T Y \begin{bmatrix} \underline{a}^p \\ 0 \end{bmatrix} = \sigma_p^2 \underline{\delta} \;\; ; \qquad \underline{\delta} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \qquad (II.14)$$

the matrix $[I \; 0]$ is just there to reduce the length of the vector to $p+1$. When this solution is written out for $p = 0, 1, \ldots, n$, the result is

$$Y^T Y A = L D^2 \qquad (II.15)$$

where L is a lower triangular matrix with ones on its main diagonal and D^2 is the diagonal matrix containing the prediction errors :

$$D^2 = \text{Diag}\,[\sigma_0^2, \sigma_1^2, \ldots, \sigma_n^2] \qquad (II.16)$$

QR Factors : The equation $Y^T Y A = L D^2$ characterizes the least squares predictors for $p = 0$ to $p = n$. The right hand side is lower triangular. If both sides of the equation are pre-multiplied by A^T, then the left hand side, namely $A^T Y^T Y A$ is symmetric and the right hand side is lower triangular. So the right hand side must be diagonal. This means the least squares solution for the columns of A produce the following equations :

$$Y A = U \qquad A^T Y^T Y A = U^T U = D^2 \qquad (II.17)$$

It follows that $Y A = U$ is a QR factorization of the data matrix Y when A solves the sequence of least squares problems we have posed. This QR factorization (really a Gram-Schmidt orthogonalization of Y) may be rewritten as

$$Y = U H^T \quad ; \quad H^T = A^{-1} \qquad (II.18)$$

where H^T is upper triangular. Then the QR factorization may be given the following analysis, synthesis and orthogonalization interpretations :

i) $\Upsilon A = U$: Υ is analyzed or whitened by the analysis matrix A to produce the white matrix U $(U^T U = D^2)$.

ii) $\Upsilon = UH^T$: Υ is synthesized from the white matrix U by the upper triangular synthesis matrix $H^T = A^{-1}$.

iii) $\Upsilon^T U = HD^2$: the data matrix Υ and the white matrix U are causally orthogonal, with H describing the cross-correlation between the input Υ and the output U.

Second-Order Equations : Let us define the experimental correlation matrix R to be the Grammian of the data matrix Υ :

$$R = \Upsilon^T \Upsilon \tag{II.19}$$

Then, from the analysis and synthesis equations we can interpret the analysis matrix A as a Cholesky factor of R^{-1} and the synthesis matrix H as a Cholesky factor of R :

i) $A^T RA = D^2$: A is a Cholesky factor of R^{-1}.

ii) $R = HD^2 H^T$: H is a Cholesky factor of R.

iii) $RA = HD^2$: A analyzes R to produce H.

Backward equations : The prediction operation may also be written backwards in time (with respect to future values of the time-series). The one-step backward model becomes

$$\sum_{i=0}^{n} b_i^n y_{t+i} = v_t^n \quad ; \qquad b_0^n = 1. \tag{II.20}$$

where $\{v_t^n\}$ is a white sequence with zero mean and variance τ_n^2. As before, lower order predictors are introduced to form a family of order increasing predictors

$$\sum_{i=0}^{p} b_i^p y_{t+i} = v_t^p \quad ; \qquad b_0^p = 1. \tag{II.21}$$

for $p = 0$ to $p = n$. The prediction errors are written for the time index satisfying the following condition :

$$k \leq t + p \leq l. \tag{II.22}$$

The error equations for the order p predictor may be written as

$$\begin{bmatrix} y_k & y_{k-1} & \cdots & y_{k-n} \\ y_{k+1} & \ddots & \ddots & \vdots \\ & \ddots & & y_{k-1} \\ \vdots & & & y_k \\ y_{l-n} & & & \vdots \\ \vdots & \ddots & & \\ y_{l-1} & & \ddots & \vdots \\ y_l & y_{l-1} & \cdots & y_{l-n} \end{bmatrix} \begin{bmatrix} b_p^p \\ \vdots \\ b_1^p \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} v_{k-p}^p \\ v_{k-p-1}^p \\ \vdots \\ \\ \\ \\ \vdots \\ v_{l-p-1}^p \\ v_{l-p}^p \end{bmatrix} \qquad (II.23)$$

The data matrix X in this equation is related to Y by the exchange matrix J : $X = JYJ$. The compact notation is then

$$X \begin{bmatrix} b^p \\ \underline{0} \end{bmatrix} = XB^p = V^p \qquad (II.24)$$

This scheme may be reproduced for $p = 0$ to $p = n$ to obtain the set of equations

$$XB = V \qquad (II.25)$$

$$B = [B^0, \ldots, B^p, \ldots, B^n] \qquad V = [V^0, \ldots, V^p, \ldots, V^n] \qquad (II.26)$$

The least squares solution under the constraint $b_0^p = 1$, when written out for $p = 0$ to $p = n$, leads to the matrix equation :

$$B^T X^T XB = V^T V = E^2 = \text{Diag}\,[\tau_0^2, \ldots, \tau_p^2, \ldots, \tau_n^2] \qquad (II.27)$$

It then follows that $X = VB^{-1} = VG^T$ is a QR factorization of X, and the second order Cholesky equations for the Grammian $X^T X = JRJ$ are

$$B^T X^T XB = E^2 \qquad X^T X = GE^2 G^T \qquad (II.28)$$

The connections with the forward case come from the relationship between X and Y. The QR factorization of X is a "QL" factorization on Y. The upper-lower Cholesky factorization of $(X^T X)^{-1}$ is a lower-upper Cholesky factorization of $(Y^T Y)^{-1}$ and the lower-upper Cholesky factorization of $X^T X$ is an upper-lower Cholesky factorization of $Y^T Y$.

Forward-Backward equations : When the time series is known to be stationary, its statistical properties are not modified when time is reversed. This property may be very important for some applications, and the family of models may be forced to satisfy this property. In other words, the forward and backward prediction errors are written with respect to the same predictor coefficients :

$$\sum_{i=0}^{p} a_i^p y_{t-i} = u_t^p \qquad \sum_{i=0}^{p} a_i^p y_{t+i} = v_t^p \quad : \quad a_0^p = 1. \tag{II.29}$$

The compact notation is

$$\begin{bmatrix} Y \\ X \end{bmatrix} \begin{bmatrix} a^p \\ 0 \end{bmatrix} = \begin{bmatrix} U^p \\ V^p \end{bmatrix} \quad : \quad X = JYJ \tag{II.30}$$

When these equations are written out for prediction orders from $p = 0$ to $p = n$, we have the coupled prediction equations

$$\begin{bmatrix} Y \\ X \end{bmatrix} A = \begin{bmatrix} U \\ V \end{bmatrix} \tag{II.31}$$

The Grammian of the error matrix is

$$\begin{bmatrix} U^T & V^T \end{bmatrix} \begin{bmatrix} U \\ V \end{bmatrix} = A^T \begin{bmatrix} Y^T & X^T \end{bmatrix} \begin{bmatrix} Y \\ X \end{bmatrix} A = A^T \left[Y^T Y + X^T X \right] A \tag{II.32}$$

The pp^{th} element of this Grammian is

$$\sigma_p^2 = (A^p)^T \left[Y^T Y + X^T X \right] A^p \tag{II.33}$$

which is just the accumulated squared prediction errors in the forward and backward predictions.

As before, the solution of the family of least squares problems produces a matrix A that orthogonalizes the data matrix $\begin{bmatrix} Y \\ X \end{bmatrix}$, and Cholesky factors the Grammian $Y^T Y + X^T X$.

Linear Statistical models : There is one more useful comment to be made about the equations of linear prediction. Suppose $\underline{y}^i$ is the i^{th} snapshot of a data set. It might be obtained from a multi-sensor array or from overlapped windows of a time series. This snapshot may be analyzed into a white vector $\underline{u}^i$ by fitting a linear prediction or analysis model :

$$A^T \underline{y}^i = \underline{u}^i \tag{II.34}$$

where A^T is a lower triangular matrix with unit diagonal elements. Call R the experimental correlation matrix and D^2 the experimental analysis matrix :

$$\sum_{i}^{l} y^i (y^i)^T \quad R \qquad \sum_{i}^{l} u^i (u^i)^T \quad D^2 \tag{II.35}$$

Then the connection between R and D^2 is

$$A^T R A = D^2 \tag{II.36}$$

If the columns of A are chosen to minimize the diagonal elements of D^2, term by term, then D^2 is diagonal, and the factorization obtained is a Cholesky factorization. Similarly the synthesis model may be written

$$y^i = H u^i \qquad H^T = A^{-1} \tag{II.37}$$

The connection with the prediction equations comes with the choice of y^i as the i^{th} column of Y^T, the Toeplitz data matrix. With such a choice for y^i, the linear statistical model is really another way to write down the whole set of increasing order prediction problems. When the matrix R is replaced by the expectation of $y^i(y^i)^T$, then this formalism reduces to that of 3.

Summary : By solving the right sequence of least squares problems, we QR factor a data matrix and produce Cholesky factors of the experimental covariance matrix and its inverse. This is fundamental. We can either think of a QR factoring of the data matrix as a square root method of factoring the experimental correlation matrix and its inverse, or we can think of a Cholesky factoring of the experimental covariance matrix and its inverse as a square method of obtaining the "R" part of a QR factorization. We can also think of the QR factor $YA = U$, the Cholesky factor $A^T R A = D^2$, and the Cholesky factor $R \quad H D^2 H^T$ as three different ways of characterizing the matrix A (or its inverse) that contains order-increasing prediction filters. In these characterizations, the Toeplitz structure of Y, and the close-to-Toeplitz structure of R, may be used to derive fast algorithms for computing A (or its inverse). In these algorithms, generalized reflection coefficients are used to update recursions.

In section III of this paper, we review how the generalized Levinson recursions derived by Friedlander et al. [1] produce a fast algorithm for computing A from the factorization $A^T R A = D^2$. These recursions are used in section IV to produce a fast algorithm for computing the orthogonal matrix U from the factorization $YA = U$. Then the recursions for U are used in section V to produce a fast algorithm for H from the factorization $R = HD^2H^T$. All these algorithms are true lattice algorithms that produce generalized reflection coefficients. In this way we will have derived fast QR algorithms for any of the data matrices that arise in linear prediction problems and generalized Schur algorithms for any of associated experimental covariance matrices.

III. Factoring R^{-1} into its Cholesky factors

The problem of factoring R^{-1} is the problem of finding A in the diagonalization :

$$A^T R A = D^2 = \text{Diag}\{\sigma_0^2, \sigma_1^2, \ldots, \sigma_n^2\} \qquad (III.1)$$

This equation may be written as

$$RA = HD^2 \quad ; \quad H = A^{-T} \qquad (III.2)$$

and read out as follows :

$$R_i a^i = \sigma_i^2 \delta_i = \sigma_i^2 [0, \ldots, 0, 1]^T \qquad (III.3)$$

Where R_i is the $(i+1)$ by $(i+1)$ top left submatrix of R. When i is incremented to $i+1$ then, of course, a new column and a new row are added to R_i. If the resulting matrix R_{i+1} has a simple recursive dependence on R_i, then there is reason to hope for a recursive dependence of a^{i+1} on a^i. This was the insight of Friedlander et al. [1].

The matrix R is $Y^T Y$ (or $X^T X$, or $Y^T Y + X^T X$) with Y and X Toeplitz. This means R is close-to-Toeplitz. Let us denote the $(n+1)$ by $(n+1)$ symmetric, non-negative definite correlation matrix R as follows :

$$R \begin{bmatrix} r_{0,0} & r_{0,1} & \cdots & r_{0,n-1} & r_{0,n} \\ r_{1,0} & r_{1,1} & & & r_{1,n} \\ \vdots & & & & \vdots \\ r_{n-1,0} & & & & r_{n-1,n} \\ r_{n,0} & r_{n,1} & \cdots & r_{n,n-1} & r_{n,n} \end{bmatrix} \qquad (III.4)$$

The shifted difference matrix [1] is the n by n matrix $\delta[R]$:

$$\delta[R] = \begin{bmatrix} r_{1,1} & \cdots & r_{1,n} \\ \vdots & & \vdots \\ r_{n,1} & \cdots & r_{n,n} \end{bmatrix} - \begin{bmatrix} r_{0,0} & \cdots & r_{0,n-1} \\ \vdots & & \vdots \\ r_{n-1,0} & \cdots & r_{n-1,n-1} \end{bmatrix} \tag{III.5}$$

The rank of $\delta[R]$ is the displacement rank α. We stress the cases where $0 \leq \alpha < 4$, as these correspond to the least squares problems of linear prediction. The decomposition of $\delta[R]$ may be written [1]

$$\delta[R] = C\Sigma C^T \tag{III.6}$$

where C is an n by α matrix and Σ is an α by α diagonal signature matrix, containing $+1$ or -1 on its diagonal.

The fundamental equation used in the derivation of fast algorithms is the update for the matrix R_i, using C_i, which consists of the first $(i+1)$ rows of C :

$$\begin{aligned} R_{i+1} &= \left[\begin{array}{ccc|c} & & & r_{0,i+1} \\ & R_i & & \vdots \\ & & & r_{i,i+1} \\ \hline r_{i+1,0} & \cdots & r_{i+1,i} & r_{i+1,i+1} \end{array}\right] \\ &= \left[\begin{array}{c|ccc} r_{0,0} & r_{0,1} & \cdots & r_{0,i+1} \\ \hline r_{1,0} & & & \\ \vdots & & R_i & \\ r_{i+1,0} & & & \end{array}\right] + \left[\begin{array}{c|ccc} 0 & 0 & \cdots & 0 \\ \hline 0 & & & \\ \vdots & & C_i\Sigma C_i^T & \\ 0 & & & \end{array}\right] \end{aligned} \tag{III.7}$$

The idea here is to correct a Toeplitz approximation for R_{i+1} with a low rank matrix $C_i\Sigma C_i^T$. Note that $R = R_n$ and $R_0 = r_{0,0}$.

When R is Toeplitz then $r_{i,j} = r_{i-j}$, which means that δR is zero. In the more general case, Y^TY has a displacement rank equal to zero in the correlation method of linear prediction, one in the pre- and post-windowed methods of linear prediction, two in the covariance case of linear prediction and four in the forward-backward covariance method of linear prediction. Intermediate forward-backward methods with a displacement rank of two may be introduced, by using forward-backward methods in either the pre-windowed or the post-windowed methods of linear prediction.

Let $\underline{a}^i = [a_i^i, \ldots, a_i^1, 1]^T$, denote the first $(i+1)$ elements of A^i, the i^{th} column of A. From the Cholesky decomposition $R^{-1} = AD^{-2}A^T$, it follows that $RA = HD^2$, where

$H = A^{-T}$. Read the i^{th} column of this equation :

$$R_i a^i = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \sigma_i^2 \qquad (III.8)$$

If $\underline{a}^{i-1}$ is approximated by $\begin{bmatrix} 0 \\ \underline{a}^i \end{bmatrix}$, then the $(i+1)$ version of this equation is :

$$R_{i+1} \begin{bmatrix} 0 \\ \underline{a}^i \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \sigma_i^2 + \left[\begin{array}{c|ccc} 0 & r_{0,1} & \cdots & r_{0,i+1} \\ \hline 0 & & & \\ \vdots & & C_i \Sigma C_i^T & \\ 0 & & & \end{array}\right] \begin{bmatrix} 0 \\ \underline{a}^i \end{bmatrix} \qquad (III.9)$$

which may be written as

$$R_{i+1} \begin{bmatrix} 0 \\ \underline{a}^i \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \sigma_i^2 - \begin{bmatrix} 1 & \underline{0}^T \\ 0 & \\ \vdots & C_i \\ 0 & \end{bmatrix} F_i \sigma_i^2 \qquad (III.10)$$

with

$$\sigma_i^2 F_i = \begin{bmatrix} r_{0,0} & r_{1,0} & \cdots & r_{i+1,0} \\ \underline{0} & & \Sigma C_i^T & \end{bmatrix} \begin{bmatrix} 0 \\ \underline{a}^i \end{bmatrix} \qquad (III.11)$$

Define the $(i+1)$ by $(\alpha+1)$ matrix $\widehat{\underline{a}}^i$ as

$$R_i \widehat{\underline{a}}^i = \begin{bmatrix} 1 & \underline{0}^T \\ 0 & \\ \vdots & C_{i-1} \\ 0 & \end{bmatrix} \Delta_i \sigma_i^2 \qquad (III.12)$$

where Δ_i is an $(\alpha+1)$ by $(\alpha+1)$ error matrix, so that

$$R_{i+1} \begin{bmatrix} \widehat{\underline{a}}^i \\ 0 \end{bmatrix} = \left[\begin{array}{cc} 1 & \underline{0}^T \\ 0 & \\ \vdots & C_{i-1} \\ 0 & \\ \hline & E_i \Delta_i^{-1} \end{array}\right] \Delta_i \sigma_i^2 \qquad (III.13)$$

$$\sigma_i^2 E_i = [r_{i+1,0}, \ldots, r_{i+1,i}] \widehat{\underline{a}}^i. \qquad (III.14)$$

Then we have the recursions

$$\begin{aligned} \underline{a}^{i+1} &= \begin{bmatrix} 0 \\ \underline{a}^i \end{bmatrix} + \begin{bmatrix} \widehat{a}^i \\ 0 \end{bmatrix} \Delta_i^{-1} F_i \\ \widehat{\underline{a}}^{i+1} &= \begin{bmatrix} \widehat{a}^i \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \underline{a}^i \end{bmatrix} D_i \end{aligned} \qquad (III.15)$$

with $D_i = [0, c_i]\,\Delta_i = E_i$, and c_i is the last row of C_i. The proof that $D_i^T = F_i = K_{i+1}$ is contained in [1]. K_{i+1} is the generalized reflection coefficient.

$\underline{a}^i$ contains the forward prediction coefficients, the first column of $\widehat{\underline{a}}^i$ contains the backward prediction coefficients (on the same time window), and the other columns of $\widehat{\underline{a}}^i$ are used to correct the shift difference. In fact, these columns perform the transformation from the sample covariance matrix to the final and/or initial conditions, to cancel out "end-effects" due to the non-Toeplitz nature of R.

These equations may be written in a format where all vectors are of fixed dimension, $(n+1)$ by 1. Append $\widehat{a}^i$ with zeros to get the $(n+1)$ by $(\alpha+1)$ matrix $\widehat{A}^i$. Then the generalized Levinson recursions of Friedlander et al. [1] may be written as follows :

$$\begin{aligned} \widehat{A}^{i+1} &= \widehat{A}^i + Z\,A^i\,K_{i+1}^T \\ A^{i+1} &= Z\,A^i + \widehat{A}^i\,\Delta_i^{-1} K_{i+1} \qquad \text{for } i = 0 \text{ to } n-1. \end{aligned} \qquad (III.16)$$

The matrix Z in these equations is the delay matrix :

$$Z = \left[\begin{array}{ccc|c} 0 & \cdots & 0 & 0 \\ \hline & & & 0 \\ & I & & \vdots \\ & & & 0 \end{array}\right] \qquad (III.17)$$

These recursions are initialized by

$$\begin{aligned} A^0 &= [1\,0\cdots 0]^T & \widehat{A}^0 &= \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \end{bmatrix}^T \\ \sigma_0^2 &= r_{0,0} & \Delta_0 &= \begin{bmatrix} 1 & 0^T \\ 0 & -\sigma_0^{-2}\Sigma \end{bmatrix} \end{aligned} \qquad (III.18)$$

The vector K_{i+1} is an $(\alpha+1)$ vector which generalizes the usual scalar reflection coefficient k_{i+1}, and is computed by

$$\sigma_i^2\,K_{i+1}^T = -(Z\,A^i)^T \begin{bmatrix} r_{0,0} & 0^T \\ r_{1,0} & \\ \vdots & C\Sigma \\ r_{n,0} & \end{bmatrix} \qquad (III.19)$$

The prediction error σ_i^2, and the error matrix Δ_i are updated as follows :

$$\begin{aligned} \sigma_{i+1}^2 &= \sigma_i^2(1 - K_{i+1}^T \Delta_i^{-1} K_{i+1}) \\ \sigma_{i+1}^2 \Delta_{i+1} &= \sigma_i^2(\Delta_i - K_{i+1} K_{i+1}^T) \end{aligned} \qquad (III.20)$$

Note that these two equations are the same if $\Delta_i = 1$ and K_{i+1} is a scalar, and that Δ_i never needs to be inverted, as only $\Delta_i^{-1} K_{i+1}$ is used in the recursions.

IV. Factoring Y into its QR factors

Using the recursions for the columns of A, we propose to find the corresponding recursions for the columns of the orthogonal matrix U, using the QR equation $YA = U$. For the correlation method of linear prediction we use the technique of Rialan and Scharf [4]. For all the other cases of linear prediction we generalize their technique.

Correlation method of LP : In the correlation method of linear prediction, $k = 0$ and $l = N - 1 + n$, which means that the data matrix Y looks like this :

$$Y = \begin{bmatrix} y_{N-1} & \cdots & y_{N-n} & y_{N-n-1} & \cdots & y_0 & 0 & \cdots & 0 \\ 0 & \ddots & & & & & \ddots & \ddots & \vdots \\ \vdots & \ddots & y_{N-1} & & & & & y_0 & 0 \\ 0 & \cdots & 0 & y_{N-1} & \cdots & y_n & \cdots & y_1 & y_0 \end{bmatrix}^T \qquad (IV.1)$$

Then the correlation matrix $Y^T Y = R$ is Toeplitz, symmetric, and positive semi-definite.

The Cybenko recursions [10] on the columns of U may be derived by using the Levinson recursions on the columns of A to induce recursions on the columns of U, using $YA = U$. Define Y_i to be the first $(i+1)$ columns of Y and U^i to be the i^{th} column of U. Then introduce an auxiliary vector $\hat{U}^i$, so that

$$Y A^i = Y_i \underline{a}^i = U^i \qquad Y \hat{A}^i = Y_i \underline{\hat{a}}^i = \hat{U}^i \qquad (IV.2)$$

We saw in section II that the i^{th} column of U contains the forward prediction errors of order i. Similarly $\hat{U}^i$ contains the backward prediction errors of the same order. Reproduce these equations for $(i+1)$ and use the Levinson recursions to get

$$Y A^{i+1} = U^{i+1} = Y \left(Z A^i - k_{i+1} \hat{A}^i \right) \qquad (IV.3)$$

As we have $Y Z A^i = Z Y A^i$ (as long as the last element in A^i is equal to zero), then

$$U^{i+1} = Z U^i + k_{i+1} \hat{U}^i \qquad (IV.4)$$

Doing for the same for $\hat{U}^{i+1}$ leads to the following recursions :

$$\begin{aligned} \hat{U}^{i+1} &= \hat{U}^i + k_{i+1}\, Z\, U^i \\ U^{i+1} &= Z\, U^i + k_{i+1}\, \hat{U}^i \qquad \text{for } i = 0, \ldots, n-1. \end{aligned} \tag{IV.5}$$

These recursions are initialized using

$$U^0 = \hat{U}^0 = [\, y_{N-1} \quad \ldots \quad y_1 \quad y_0 \quad 0 \quad \ldots \quad 0\,]^T \qquad \sigma_0^2 = (U^0)^T U^0 \tag{IV.6}$$

k_{i+1} is the reflection coefficient, and is computed from the internal variables by the following inner product

$$k_{i+1}\, \sigma_i^2 = -(U^0)^T\, Z\, U^i \tag{IV.7}$$

This comes from the familiar computation of the reflection coefficient :

$$\begin{aligned} k_{i+1}\, \sigma_i^2 &= -[\, r_0 \quad r_1 \quad \ldots \quad r_n\,]\, Z\, A^i \\ &= -[\, y_{N-1} \quad \ldots \quad y_0 \quad 0 \quad \ldots \quad 0\,]\, Y\, Z\, A^i \\ &= -(U^0)^T\, Z\, U^i \end{aligned} \tag{IV.8}$$

Alternates formulas for the reflection coefficient are given by the fact that U^{i+1} is orthogonal to the previous U^j $(j = 0, \ldots, i)$ as well as the previous $\hat{U}^j$:

$$k_{i+1} = -\frac{(U^i)^T Z U^i}{(U^i)^T \hat{U}^i} = -\frac{(\hat{U}^i)^T Z U^i}{(\hat{U}^i)^T \hat{U}^i}$$

Note that $\sigma_i^2 = (U^i)^T U^i = (\hat{U}^i)^T \hat{U}^i$. This algorithm for computing the orthogonal matrix in the QR factorization, is composed of parallel updates for the vectors U^i and $\hat{U}^i$, and a side computation containing an inner product for $k_{i+1}\sigma_i^2$, and the update for σ_{i+1}^2 :

$$\sigma_{i+1}^2 = \sigma_i^2 (1 - k_{i+1}^2) \tag{IV.9}$$

The Cybenko algorithm may be extended to more general matrices Y, corresponding to the other cases of linear prediction, using the generalized Levinson recursions. The only difference lies in the dimension $(\alpha + 1)$ of the auxiliar matrix $\hat{A}^i$, which induces a different dimension in the matrix $\hat{U}^i$. We saw that the very special shape of Y in the correlation method of linear prediction allowed us to permute factors in $Y Z A^i$. The loss

of the upper triangle of zeros in Y in the covariance method forbids us from making the same manipulation. We therefore develop a "generic" algorithm based on the extension of Y with the upper triangle of zeros (when it is missing). The matrix U in the QR factorization of Y for the covariance method of linear prediction will then be contained within the orthogonal matrix of the "generic" case. The pre- and post-windowed methods follow as special cases.

In the correlation method of Linear Prediction, the matrix H may be computed, if needed, by using the "impulse" response and internal variables of the AR (or recursive) lattice as described in [3]. In section VI, we will show how this procedure generalizes in the other cases.

Generic case : We suppose now that we have the following Toeplitz matrix W :

$$W = \left[\begin{array}{cccc} y_l & 0 & \cdots & 0 \\ y_{l-1} & y_l & \ddots & \vdots \\ \vdots & & \ddots & 0 \\ \hline y_{l\ n} & & & y_l \\ \vdots & \ddots & & \vdots \\ y_{k-n} & & & y_k \end{array}\right] = \begin{bmatrix} T \\ Y \end{bmatrix} \qquad (IV.10)$$

The data matrix Y has QR factor $YA = U$. Therefore

$$WA = \begin{bmatrix} \mathcal{U} \\ U \end{bmatrix} \qquad (IV.11)$$

which means the orthogonal matrix U is embedded in a larger matrix. Define the i^{th} column of WA as follows :

$$WA^i = \begin{bmatrix} \mathcal{U}^i \\ U^i \end{bmatrix} \qquad (IV.12)$$

Similarly

$$W\widehat{A}^i = \begin{bmatrix} \widehat{\mathcal{U}}^i \\ \widehat{U}^i \end{bmatrix} \qquad (IV.13)$$

As in the correlation case, we reproduce these equations for $(i+1)$ and use the generalized Levinson recursions for A^i and $\widehat{A}^i$ to get

$$\begin{aligned} WA^{i+1} &= \begin{bmatrix} \mathcal{U}^{i+1} \\ U^{i+1} \end{bmatrix} = Z \begin{bmatrix} \mathcal{U}^i \\ U^i \end{bmatrix} + \begin{bmatrix} \widehat{\mathcal{U}}^i \\ \widehat{U}^i \end{bmatrix} \Delta_i^{-1} K_{i+1} \\ W\widehat{A}^{i+1} &= \begin{bmatrix} \widehat{\mathcal{U}}^{i+1} \\ \widehat{U}^{i+1} \end{bmatrix} = \begin{bmatrix} \widehat{\mathcal{U}}^i \\ \widehat{U}^i \end{bmatrix} + Z \begin{bmatrix} \mathcal{U}^i \\ U^i \end{bmatrix} K_{i+1}^T \end{aligned} \qquad (IV.14)$$

The computation of K_{i+1} depends then on the method used.

Covariance method of LP : For the covariance method of linear prediction, $k = n$ and $l = N - 1$. Then the shifted difference matrix δR has rank 2 :

$$Y = \begin{bmatrix} y_{N-n-1} & \cdots & y_1 & y_0 \\ \vdots & \ddots & & \vdots \\ y_{N-2} & & y_n & y_{n-1} \\ y_{N-1} & y_{N-2} & \cdots & y_{n+1} & y_n \end{bmatrix}^T \qquad (IV.15)$$

$$C = \begin{bmatrix} y_0 & \cdots & y_{n-1} \\ y_{N-n} & \cdots & y_{N-1} \end{bmatrix}^T \quad \text{and} \quad \Sigma = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \qquad (IV.16)$$

Write

$$YA^i = Y_i a^i = U^i \qquad Y\hat{A}^i = Y_i \hat{a}^i = \hat{U}^i \qquad (IV.17)$$

with $\hat{U}^i$ an $(N - n)$ by 3 matrix. The "generic" recursions may be used for U^i and $\hat{U}^i$:

$$\begin{bmatrix} U^{i+1} \\ U^{i+1} \end{bmatrix} = Z \begin{bmatrix} U^i \\ U^i \end{bmatrix} + \begin{bmatrix} \hat{U}^i \\ \hat{U}^i \end{bmatrix} \Delta_i^{-1} K_{i+1}$$
$$\begin{bmatrix} \hat{U}^{i+1} \\ \hat{U}^{i+1} \end{bmatrix} = \begin{bmatrix} \hat{U}^i \\ \hat{U}^i \end{bmatrix} + Z \begin{bmatrix} U^i \\ U^i \end{bmatrix} K_{i+1}^T \qquad (IV.18)$$

The initialization is

$$\begin{bmatrix} U^0 \\ \hline U^0 \end{bmatrix} = \begin{bmatrix} y_{N-1} \\ \vdots \\ y_{N-n} \\ \hline y_{N-n-1} \\ \vdots \\ y_1 \\ y_0 \end{bmatrix} \qquad \begin{bmatrix} \hat{U}^0 \\ \hline \hat{U}^0 \end{bmatrix} = \begin{bmatrix} y_{N-1} & 0 & 0 \\ \vdots & \vdots & \vdots \\ y_{N-n} & 0 & 0 \\ \hline y_{N-n-1} & 0 & 0 \\ \vdots & \vdots & \vdots \\ y_1 & 0 & 0 \\ y_0 & 0 & 0 \end{bmatrix} \qquad (IV.19)$$

$$\sigma_0^2 = r_{0,0} = (U^0)^T U^0 \qquad \Delta_0 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \sigma_0^{-2} & 0 \\ 0 & 0 & -\sigma_0^{-2} \end{bmatrix} \qquad (IV.20)$$

The three components of the reflection coefficient K_{i+1} are computed using

$$\sigma_i^2 K_{i+1}(1) = -\begin{bmatrix} (U^0)^T & 0 \end{bmatrix} \begin{bmatrix} U^i(n) \\ U^i \end{bmatrix}$$
$$\sigma_i^2 K_{i+1}(2) = U^i(N - n) \qquad (IV.21)$$
$$\sigma_i^2 K_{i+1}(3) = -U^i(n)$$

which comes from the use of

$$\sigma_i^2 K_{i+1}^T = -(ZA^i)^T \begin{bmatrix} r_{0,0} & 0 & 0 \\ r_{1,0} & -y_0 & y_{N-n} \\ \vdots & \vdots & \vdots \\ r_{n,0} & y_{n-1} & y_{N-1} \end{bmatrix} \qquad (IV.22)$$

Only one inner product is necessary per update. Note that $U^i(N-n)$ is the last value contained in the vector U^i, and that $\mathcal{U}^i(n)$ is the last value of the vector $\mathcal{U}^i$. Reflection coefficients are computed without first computing correlations.

An alternate formula may also be obtained as in the correlation method, using the orthogonality property of the columns of U (but is much more expensive) :

$$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 & (\hat{U}^i)^T \end{bmatrix} Z \begin{bmatrix} \mathcal{U}^i \\ U^i \end{bmatrix} + \begin{bmatrix} 0 & (\hat{U}^i)^T \end{bmatrix} \begin{bmatrix} \hat{\mathcal{U}}^i \\ \hat{U}^i \end{bmatrix} \Delta_i^{-1} K_{i+1}$$

At this point, if one needs the whitening matrix A (or its last column, the coefficients of $A_n(z)$), then the generalized Levinson recursions may be used together with the reflection coefficients computed by the QR recursions. In this approach, what is really done is to compute the impulse response of the lattice filter associated with these reflection coefficients (see section VI).

Post-windowed method of LP : In the post-windowed method of linear prediction, $k = n$ and $l = N + n - 1$. Then Y and the shifted difference matrix δR are defined by

$$Y = \begin{bmatrix} y_{N-1} & y_{N-2} & \cdots & y_{N-n-1} & \cdots & y_0 \\ 0 & y_{N-1} & & & & y_1 \\ \vdots & \ddots & \ddots & & & \vdots \\ 0 & \cdots & 0 & y_{N-1} & \cdots & y_n \end{bmatrix}^T \qquad (IV.23)$$

$$C = \begin{bmatrix} y_0 & \cdots & y_{n-1} \end{bmatrix}^T \quad \text{and} \quad \Sigma = -1 \qquad (IV.24)$$

The displacement rank α is unity which means that the generalized Levinson algorithm will use an n by 2 vector. The "generic" recursions may be simplified in this case to :

$$\begin{aligned} U^{i+1} &= Z\, U^i + \hat{U}^i\, \Delta_i^{-1}\, K_{i+1} \\ \hat{U}^{i+1} &= \hat{U}^i + Z\, U^i\, K_{i+1}^T \qquad \text{for } i = 0, \ldots, n-1. \end{aligned} \qquad (IV.25)$$

The initialization is

$$U^0 = \begin{bmatrix} y_{N-1} & \cdots & y_1 & y_0 \end{bmatrix}^T \qquad \sigma_0^2 = (U^0)^T U^0$$

$$\widehat{U}^0 = \begin{bmatrix} y_{N-1} & \cdots & y_1 & y_0 \\ 0 & \cdots & 0 & 0 \end{bmatrix}^T \qquad \Delta_0 = \begin{bmatrix} 1 & 0 \\ 0 & \sigma_0^2 \end{bmatrix} \tag{IV.26}$$

Now the reflection coefficient K_{i+1} is given by

$$\sigma_i^2 K_{i+1}^T = \begin{bmatrix} -(U^0)^T Z U^i, & U^i(N) \end{bmatrix} \tag{IV.27}$$

where $U^i(N)$ is the last element of U^i. Note that the side computations involve an inner product, and the updates for Δ_i and σ_i^2, as in the corresponding generalized Levinson algorithm.

Pre-windowed method of LP : For the pre-windowed method of linear prediction, $k = 0$ and $l = N - 1$. Then δR is rank 1, and Y and δR are defined by

$$Y = \begin{bmatrix} y_{N-n-1} & \cdots & y_0 & 0 & \cdots & 0 \\ \vdots & & & \ddots & \ddots & \vdots \\ y_{N-2} & & & & y_0 & 0 \\ y_{N-1} & \cdots & y_n & \cdots & y_1 & y_0 \end{bmatrix}^T \tag{IV.28}$$

$$C = \begin{bmatrix} y_{N-n} & \cdots & y_{N-1} \end{bmatrix}^T \quad \text{and} \quad \Sigma = 1 \tag{IV.29}$$

Note that the pre-window matrix Y is related to the post-window matrix Y by the exchange matrix J. JYJ has the same triangle of zeros on the top as in the post-windowed method. The only difference lies in the fact that time is reversed, or in other words y_i is replaced by y_{N-1-i}. The fast algorithm for U in the post-windowed case may then be used on JYJ.

Backward Linear Prediction : In the backward method of linear prediction, the only difference with the forward recursions is that the entries in the data matrix Y are altered. The shape of Y remains unchanged. In other words, time is reversed. This means that the fast algorithms for U may be used for Y with just a modification of the initialization of the vectors.

Forward-Backward Linear Prediction : In the forward-backward method of linear prediction, the data matrix is the concatenation of two Toeplitz matrices. For the correlation method, $Y^TY = JY^TYJ$ meaning that the forward-backward extention leads to the same

Cholesky factors A and H, and D^2 is just multiplied by a factor of two. In the covariance method, the forward-backward extension leads to the so-called modified covariance method of linear prediction, and one way to compute the highest order predictor $A_n(z)$ is to use the algorithm of Marple [13]. But the lower order predictors introduced in [13] do not solve the same set of least squares problems that we have defined. We introduce in the Appendix an alternate to this algorithm to find a characterization of the highest order predictor, based on the generalized reflection coefficients computed with the recursions for the columns of the orthogonal matrix. The impulse response of the associated lattice filter is then equal to the highest order predictor.

V. Factoring R into its Cholesky Factors

The LU factorization of R may be written $R = HD^2H^T$. In the Toeplitz case, the LeRoux-Gueguen algorithm [2] may be used to compute H directly. In all the linear prediction cases, or in other words when $R = Y^TY$, the recursions for the columns of H are easily induced from the QR factor recursions by simply premultiplying the recursions by Y^T. This was first done in [4] for the Toeplitz case.

Denote by H^i the vector given by $Y^TU^i = H^i$, and by $\hat{H}^i$ the matrix given by $Y^T\hat{U}^i = \hat{H}^i$. Use the recursions for U^i and $\hat{U}^i$ to obtain the recusrions :

$$\begin{aligned} \hat{H}^{i+1} &= \hat{H}^i + Y^T Z U^i K_{i+1}^T \\ H^{i+1} &= Y^T Z U^i + \hat{H}^i \Delta_i^{-1} K_{i+1} \end{aligned} \tag{V.1}$$

Set the first row of $\hat{H}^i$ to zero to be able to replace $Y^T Z U^i$ by $Z H^i$. The algorithm consists then of the recursions

$$\begin{aligned} \hat{H}^{i+1} &= \hat{H}^i + Z H^i K_{i+1}^T \\ H^{i+1} &= Z H^i + \hat{H}^i \Delta_i^{-1} K_{i+1} \end{aligned} \tag{V.2}$$

A more general approach to this derivation consists of using the generalized Levinson recursions directly to induce recursions for the columns of H. Then there is no need for R to equal Y^TY. This algorithm is a generalization of the vector version of the Leroux-Gueguen

algorithm derived in [3]. Define h^i and g^i by

$$RA^i = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ \hline \underline{h}^i \end{bmatrix} \sigma_i^2 \qquad R\widehat{A}^i = \begin{bmatrix} 1 & 0^T \\ 0 & \\ \vdots & C_{i-1} \\ 0 & \\ \hline & g^i \end{bmatrix} \Delta_i \sigma_i^2 \tag{V.3}$$

Note that $\underline{h}^i$ is a vector of length $(n - i - 1)$ and g^i is an $(n - i - 1)$ by $(\alpha + 1)$ matrix. Our goal is to derive recursions for $\underline{h}^i$ and g^i, without carrying the update equations for A^i and $\widehat{A}^i$. Some simplifications arise with the definitions of H^i and $\widehat{H}^i$:

$$RA^i = H^i \qquad \widehat{H}^i \Delta_i^{-1} \sigma_i^{-2} = \begin{bmatrix} 0 & \underline{0}^T \\ \vdots & C_{i-1} \\ 0 & \\ \hline & g^i \end{bmatrix} - \begin{bmatrix} 0 & \underline{0}^T \\ & C \\ \vdots & \\ 0 & \end{bmatrix} \tag{V.4}$$

The coupled recursions are :

$$\begin{aligned} \widehat{H}^{i+1} &= \widehat{H}^i + Z\ H^i K_{i+1}^T \\ H^{i+1} &= Z\ H^i + \widehat{H}^i\ \Delta_i^{-1}\ K_{i+1} \qquad \text{for } i = 0, \ldots, n-1. \end{aligned} \tag{V.5}$$

with the initialization

$$H^0 = \begin{bmatrix} r_{0,0} \\ r_{1,0} \\ \vdots \\ r_{n,0} \end{bmatrix} \qquad \widehat{H}^0 = \begin{bmatrix} 0 & \underline{0}^T \\ r_{1,0} & \\ \vdots & C\Sigma \\ r_{n,0} & \end{bmatrix}$$

$$\sigma_0^2 = r_{0,0} \qquad \Delta_0 = \begin{bmatrix} 1 & \underline{0}^T \\ \underline{0} & -\sigma_0^{-2}\Sigma \end{bmatrix} \tag{V.6}$$

Note that H^i is the i^{th} column of HD^2, or similarly $H^i \sigma_i^{-2}$ is the i^{th} column of H. Note also that the first i elements of H^i and the first $(i-1)$ rows of $\widehat{H}^i$ are equal to zero. The coefficient K_{i+1} may be read out of the recursions as

$$\widehat{H}^{i+1}(i) = [0, \ldots, 0] = \widehat{H}^i(i) + K_{i+1}^T \sigma_i^2 \tag{V.7}$$

so that $-K_{i-1}^T \sigma_i^2$ equals the first non-zero row in $\hat{H}'$. These coupled recursions include the update for $\sigma_i^2 = H^i(i)$, the first non-zero element in H^i. The only necessary side computation is the update of Δ_i :

$$\sigma_{i-1}^2 \Delta_{i-1} = \sigma_i^2(\Delta_i \quad K_{i-1} K_{i-1}^T) \tag{V.8}$$

This algorithm generalizes the LeRoux-Gueguen algorithm to close-to-Toeplitz matrices and places the recursions in a vector format. The algorithm is fixed point and no inner product is necessary to compute the reflection coefficients. The recursions are equivalent to the lattice algorithm of Friedlander [5]. The only differences lie in the normalization of the lattice and in the organization of the cells.

If the reflection coefficients are known, it is a challenging problem to see if and how the correlation coefficients and the matrix H may be computed from them. In the correlation case, Robinson and Treitel [7] solved this problem by observing that the all-pole lattice filter has an output equal to the causal part of the correlation sequence when the input is zero and the state is initialized at $[r_0, 0, \ldots, 0]$. The multi-channel case was studied by Friedlander [8]. The fact that internal variables of the lattice are entries in the Cholesky factor H is explained in [3]. We present the generalizations of these results to the close-to-Toeplitz cases in the next section.

VI. Lattice Presentation

The lattice representation of the coupled recursions in the correlation method of linear prediction yields additional insight. The transformation from reflection coefficients to correlation values is particularly easy using the all-pole (AR) lattice filter [7]. With a zero input sequence and an initial state set to $[r_0, 0, \ldots, 0]$, the output of the filter is the causal part of the correlation sequence. It was also shown in [3] that internal variables are scaled entries of the lower-triangular matrix H, the Cholesky factor of the Toeplitz matrix R. In [5], a lattice structure of more then two lines was used to implement the normalized version of the generalized Leroux-Gueguen algorithm, for all the cases of linear prediction.

Figure 3 showns an all-zero (MA) lattice filter that may be used to implement the recursions of the previous sections for the covariance case of linear prediction. The other cases correspond to simplified versions of the filter, except for the forward-backward co-

variance case which requires two extra lines. The conventions used in Figure 3 are the following :

$$K^i = \begin{bmatrix} K^i(1) \\ K^i(2) \\ K^i(3) \end{bmatrix} \qquad \hat{K}^i = \Delta_i^{-1} K^i = \begin{bmatrix} \hat{K}^i(1) \\ \hat{K}^i(2) \\ \hat{K}^i(3) \end{bmatrix} \qquad (VI.1)$$

The lattice of Figure 3 is the same as that of [5] except for the internal organization of the cells and the normalization of the variables. The impulse response of the lattice is obtained by placing an impulse on the first two lines and zero on the last lines. This produces the Generalized Levinson algorithm of section III (without the computation of the reflection coefficients, of course). In Figure 4, the corresponding all-pole (AR) lattice filter is shown. This lattice structure has one very interesting application. It may be used to duplicate the procedures of [3], [7], [8] for computing correlations in the close-to-Toeplitz cases. When the inputs is zero and the internal state is initialized at $r_{0,0}, 0, \ldots, 0$, the first row (or column) of the covariance matrix is generated on the first output line, the last n data values on the second line, and the first n data values are generated on the third line. The internal variables at the inputs (or outputs) on the first line of the cells reproduce the entries of the Cholesky factor HD^2. The entries of cell i reproduce, in time sequence, the entries of the i^{th} column of H, exactly as in the Toeplitz case.

Using the following notation for the entries of HD^2 :

$$HD^2 = \begin{bmatrix} H^0(0) & & & & \\ H^0(1) & H^1(1) & & \phi & \\ \vdots & & \ddots & & \\ & & & H^{n-1}(n-1) & \\ H^0(n) & H^1(n) & \ldots & H^{n-1}(n) & H^n(n) \end{bmatrix} \quad H^0 H^1 \cdots H^n \qquad (VI.2)$$

then the input of cell $(i+1)$ (output of cell i) at time $j+i$ (on the top line) is $H^i(j)$. Because of the delay, one can see that this variable will be zero for $j < i$. Equivalently, entries on the k^{th} line are equal to the corresponding entry of the $(k-1)^{\text{th}}$ column of $\hat{H}^i$. This is illustrated in Figure 5. The algorithm for actually computing all of these variables from the reflection coefficients is a generalization of that given in [7]. We may take advantage of the fact that almost half of the computed variables in the algorithm of [8] are equal to zero, as $H^i(j)$ and $\hat{H}^i(j-1)$ are zero for $j < i$, to reduce the number of computations. The algorithm is then

Initialization : $H^0(0) = r_{0,0}$

For $j = 1, \ldots, n$:

$\hat{H}^j(j) = 0, \ldots, 0$

For $i = j \quad 1, \ldots, 0$:

$\hat{H}^i(j) = \hat{H}^{i-1}(j) - H^i(j-1)K_{i-1}^T$

$H^0(j) = \delta^T \hat{H}^0(j)$

For $i = 0, \ldots, j \quad 1$:

$H^{i-1}(j) = H^i(j-1) + \hat{H}^i(j)\, \Delta_i^{-1} K_{i-1}$

If the algorithm is initialized with $H^0(0) = 1$ instead of $r_{0,0}$, then all the variables are scaled down by $r_{0,0}$. When only the reflection coefficients are known then both the prediction errors σ_i^2 and Δ_i may be recovered using the update formula given in section III. Note that in this algorithm $\hat{H}^i(j)$ is the j^{th} row of $\hat{H}^i$, which is a row vector made up of $(\alpha - 1)$ elements. Note also that the algorithm is not restricted to the methods of linear prediction but is applicable for all the possible values of the displacement rank α. When this algorithm is run in conjonction with the recursions for $U^i, \hat{U}^i$ and K_i, then we have a complete QR algorithm for computing U and H in the QR factorization $Y = UH^T$.

VII. Multi-variable case

In several applications, multi-variable linear prediction is necessary, by which we mean that the datum y_i is a vector of length d. The data may come from several sensors in a multi-channel system, or simply be a collection of scalar variables in one row (or column) of a two dimensional image. The prediction coefficients a_j become d by d matrices. The forward linear prediction problem is to predict the value of y_i as

$$\hat{y}_i = \sum_{j=1}^{n} a_j y_{i-j} \quad \text{for } i = k, \ldots, l. \tag{VII.1}$$

The prediction error is

$$u_i = y_i - \hat{y}_i = y_i - \sum_{j=1}^{n} a_j y_{i-j} \tag{VII.2}$$

which can still be organized in a matrix fashion as $Y\theta = e$

$$\begin{bmatrix} y_{l-n}^T & \cdots & y_{l-1}^T & y_l^T \\ & & & y_{l-1}^T \\ \vdots & & & \vdots \\ & & \ddots & y_{k-1}^T \\ y_{k-n}^T & \cdots & y_{k-1}^T & y_k^T \end{bmatrix} \begin{bmatrix} a_n^T \\ \vdots \\ a_1^T \\ I \end{bmatrix} = \begin{bmatrix} u_l^T \\ u_{l-1}^T \\ \vdots \\ u_{k-1}^T \\ u_k^T \end{bmatrix} \qquad (VII.3)$$

Y is a $(l - k + 1)$ by $(n + 1)d$ block Toeplitz matrix. Then, depending on the values of k and l, one ends up with different methods of linear prediction. The solution of the problem is the same as in the scalar case except for element dimensions and occasional transpose. Our purpose in this section is just to give the generalization of our vector algorithms to the multi-variable case. The Grammian matrix R is now block close-to-Toeplitz with $r_{i,j} = r_{j,i}^T$. The generalized Levinson algorithm performs the block Cholesky factorization of R^{-1}, the generalized QR algorithm the block QR factorization of Y, and the generalized LeRoux-Gueguen algorithm the block Cholesky factorization of R.

The generalized Levinson recursions are

$$\begin{aligned} B^{i+1} &= B^i + Z\, A^i\, N_i^{-1}\, K_{i+1}^T \\ A^{i+1} &= Z\, A^i + B^i\, M_i^{-1}\, K_{i-1} \qquad \text{for } i = 0, \ldots, n-1. \end{aligned} \qquad (VII.4)$$

with the initializations

$$A^0 = [I\, 0 \ldots 0]^T \qquad B^0 = \begin{bmatrix} I & 0 & \ldots & 0 \\ 0 & 0 & \ldots & 0 \end{bmatrix}^T$$

$$N_0 = r_{0,0} \qquad M_0 = \begin{bmatrix} r_{0,0} & 0 \\ 0 & \Sigma \end{bmatrix} \qquad (VII.5)$$

The dimensions of the variables involved are the following : A^i is $(n - 1)d$ by d, B^i is $(n + 1)d$ by $(\alpha + 1)d$, N_i is d by d, M_i is $(\alpha - 1)d$ by $(\alpha - 1)d$, Σ is αd by αd, C is nd by αd, and finally Z is $(n - 1)d$ by $(n + 1)d$.

The matrix Z in these equations is the delay matrix :

$$Z = \left[\begin{array}{ccc|c} 0 & \cdots & 0 & 0 \\ \hline I & & & 0 \\ & \ddots & & \vdots \\ & & I & 0 \end{array}\right] \qquad (VII.6)$$

K_{i-1} is an $(\alpha + 1)d$ by d matrix and is computed by

$$K_{i-1}^T = -(ZA^i)^T \begin{bmatrix} r_{0,0} & 0 \\ r_{1,0} & \\ \vdots & C\Sigma \\ r_{n,0} & \end{bmatrix} \qquad (VII.7)$$

The prediction error matrix N_i, and the error matrix M_i are updated as follows :

$$\begin{aligned} N_{i+1} &= N_i - K_{i+1}^T M_i^{-1} K_{i+1} \\ M_{i-1} &= M_i - K_{i+1} N_i^{-1} K_{i+1}^T \end{aligned} \qquad (VII.8)$$

The lattice recursions for the matrix columns of U in the covariance case are

$$\begin{aligned} \hat{S}^{i-1} &= \hat{S}^i + Z S^i N_i^{-1} K_{i-1}^T \\ S^{i-1} &= Z S^i + \hat{S}^i M_i^{-1} K_{i-1} \qquad \text{for } i = 0, \ldots, n-1. \end{aligned} \qquad (VII.9)$$

with the initializations

$$S^0 = \begin{bmatrix} y_{N-1}^T \\ \vdots \\ y_{N-n}^T \\ \hline y_{N-n-1}^T \\ \vdots \\ y_0^T \end{bmatrix} = \begin{bmatrix} T^0 \\ \hline U^0 \end{bmatrix} \qquad \hat{S}^0 = \begin{bmatrix} y_{N-1}^T & 0 & 0 \\ \vdots & \vdots & \vdots \\ y_{N-n}^T & 0 & 0 \\ \hline y_{N-n-1}^T & 0 & 0 \\ \vdots & \vdots & \vdots \\ y_0^T & 0 & 0 \end{bmatrix}$$

$$N_0 = r_{0,0} = (U^0)^T U^0 \qquad M_0 = \begin{bmatrix} N_0 & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & -I \end{bmatrix} \qquad (VII.10)$$

The three matrix components of the reflection coefficient K_{i-1} are computed using

$$\begin{aligned} K_{i+1}^T(1) &= -\left[(U^0)^T, 0\right] \begin{bmatrix} S^i(n) \\ U^i \end{bmatrix} \\ K_{i+1}^T(2) &= U^i(N-n) = S^i(N) \\ K_{i-1}^T(3) &= -S^i(n) = T^i(n) \end{aligned} \qquad (VII.12)$$

The other forward or backward cases of linear prediction are simplified version of this algorithm, and the forward-backward case consists of two of these recursions.

Finally, the Schur recursions are

$$\begin{aligned} \hat{H}^{i+1} &= \hat{H}^{i} - Z\, H^{i}\, N_i^{-1}\, K_{i+1}^{T} \\ H^{i+1} &= Z\, H^{i} + \hat{H}^{i}\, M_i^{-1}\, K_{i+1} \qquad \text{for } i = 0, \ldots, n-1. \end{aligned} \tag{VII.13}$$

with the initializations

$$H^0 = \begin{bmatrix} r_{0,0} \\ r_{1,0} \\ \vdots \\ r_{n,0} \end{bmatrix} \qquad \hat{H}^0 = \begin{bmatrix} 0 & 0 \\ r_{1,0} & \\ \vdots & C\Sigma \\ r_{n,0} & \end{bmatrix}$$

$$N_0 = r_{0,0} \qquad M_0 = \begin{bmatrix} N_0 & 0 \\ 0 & -\Sigma \end{bmatrix} \tag{VII.14}$$

Note that H^i is the i^{th} block column of HD^2, or similarly $H^i\, N_i^{-1}$ is the i^{th} block column of H. D is the block diagonal matrix which i^{th} block diagonal element is N_i. Note also that the first i blocks of H^i and the first $(i+1)$ block rows of $\hat{H}^i$ are equal to zero. The coefficient K_{i+1} may be read out of the recursions as

$$\hat{H}^{i+1}(i) = [0, \ldots, 0] = \hat{H}^{i}(i) - K_{i+1}^{T} \tag{VII.15}$$

so that $-K_{i+1}^{T}$ equals the first non-zero block row in $\hat{H}^i$. These coupled recursions include the update for $N_i = H^i(i)$, the first non-zero element in H^i. The only necessary side computation is the update of M_i :

$$M_{i+1} = M_i - K_{i+1}\, N_i^{-1}\, K_{i+1}^{T} \tag{VII.16}$$

Note that the use of a lattice to compute the Cholesky factor H from reflection coefficients is still valid in the multi-variable case.

VIII. Conclusion

We have derived vector algorithms for Cholesky and QR factoring Toeplitz and close-to-Toeplitz matrices for all of the cases of linear prediction. The same coupled recursions are used in all the algorithms, namely

$$\begin{aligned} N^{i+1} &= N^{i} + Z\, M^{i} K_{i+1}^{T} \\ M^{i+1} &= Z\, M^{i} + N^{i}\, \Delta_i^{-1}\, K_{i+1} \end{aligned}$$

The vector M^i contains the i^{th} column of the matrix A, HD^2, or U, depending upon which factorization is being computed. The inner products required to compute the reflection coefficients and to initialize the variables are

- $M^i = A^i$: inner product required for computing $r_{i,0}$ and K_i.
- $M^i = U^i$: inner product required for computing K_i only.
- $M^i = H^i\sigma_i^2$: inner product required for computing $r_{i,0}$ only.

The Cholesky algorithms have complexity $n^2\alpha$. The fast algorithms for the orthogonal matrix U have complexity $Nn\alpha$, where N is the number of data values available.

The QR factorization of the data matrix Y in the covariance case of linear prediction may be used for a general Toeplitz matrix T. This gives a fast algorithm to obtain $T = U\mathcal{R}$ in a QR method to obtain eigenvalues. If Y is symmetric (meaning that the time series is symmetric $y_{n+i} = y_{n-i}$) then the forward and backward predictors are reversals of each other, or in other words U^i is the reversal of the first column of $\hat{U}^i$, which simplifies the computation.

IX. References

[1] B. Friedlander, M. Morf, T. Kailath and L. Ljung, "New Inversion Formulas for Matrices Classified in terms of Their Distance from Toeplitz Matrices," Linear Algebra and its Application, Vol. 27, pp. 31-60, 1979.

[2] J. Le Roux and C.J. Gueguen, "A Fixed Point Computation of Partial Correlation Coefficients," IEEE Transactions on ASSP, Vol. 25, pp. 257-259, June 1977.

[3] C.J. Demeure and L.L. Scharf, "Linear Statistical Models for Stationary Sequences and Related Algorithms for Cholesky Factorization of Toeplitz Matrices," IEEE transactions on ASSP, Vol. 35, pp. 29-42, Jan. 1987.

[4] C.P. Rialan and L.L. Scharf, "Fast Algorithms for QR and Cholesky Factors of Toeplitz Operators,", IEEE Proceedings of the Inter. Conf. on ASSP, Dallas, pp. 41-44, April 1987, also submitted to IEEE Trans. on ASSP, Oct. 1986.

[5] B. Friedlander, "Lattice Filters for Adaptive Processing," IEEE Proceedings, Vol. 70, No.8, pp. 829-867, August 1982.

[6] M. Morf, B. Dickinson, T. Kailath and A. Vieira, "Efficient Solution of Covariance Equations for Linear Prediction," IEEE Trans. on ASSP, Vol. 25, No. 5, pp. 429-433, Oct.

1977.

[7] E.A. Robinson and S. Treitel, "Maximum Entropy and the Relationship of the Partial Autocorrelation to the Reflection Coefficients of a Layered System," IEEE Transactions on ASSP, Vol. 28, No. 2, pp. 224-235, April 1980.

[8] B. Friedlander, "Efficient Computation of the Covariance Sequence of an Autoregressive Process," IEEE Proceedings of the Int. Conf. on ASSP, Boston, pp. 182-185, April 1983.

[9] C.J. Demeure and L.L. Scharf, "Vector Algorithms for Computing QR and Cholesky Factors of Close-to-Toeplitz matrices," IEEE Proceedings of the Int. Conf. on ASSP, Dallas, pp. 1851-1854, April 1987.

[10] G. Cybenko, "A General Orthogonalization Technique with Applications to Time Series Analysis and Signal Processing," Math. of Comp., Vol. 40, No. 161, pp. 323-336, Jan. 1983.

[11] N. Levinson, "The Wiener RMS Criterion in Filter Design and Prediction," J. Math. Phys., Vol. 25, pp. 261-278, 1947.

[12] J. Durbin, "The Fitting of Times Series Models," Revue de l'Institut International de Statistique, Vol. 28, pp. 233-243, 1960.

[13] L. Marple, " A New Autoregressive Spectrum Analysis Algorithm," IEEE Trans. on ASSP, Vol. 28, No. 4, pp. 441-454, Aug. 1980.

Appendix

Fast QR algorithm for the forward-backward covariance method of linear prediction.

The data matrix in the forward-backward covariance method of linear prediction, is the following :

$$Y_{fb} = \begin{bmatrix} Y \\ X \end{bmatrix} \tag{A.1}$$

where Y and X are the forward and backward data matrices, respectively, in the covariance method of linear prediction. The Grammian is

$$R = Y_{fb}^T Y_{fb} = Y^T Y + X^T X = Y^T Y + J Y^T Y J \tag{A.2}$$

which means that R is a centro-symmetric matrix. Its displacement rank is four, and the

displacement matrix $\delta[R]$ is given by

$$C = \begin{bmatrix} y_0 & y_{N-n} & y_{n-1} & y_{N-1} \\ y_1 & y_{N-n+1} & y_{n-2} & y_{N-2} \\ \vdots & \vdots & \vdots & \vdots \\ y_{n-2} & y_{N-2} & y_1 & y_{N-n+1} \\ y_{n-1} & y_{N-1} & y_0 & y_{N-n} \end{bmatrix} \qquad \Sigma = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{A.3}$$

Note that $\delta[R] = -J\delta[R]J$. As the generalized Levinson algorithm does not use the centro-symmetry property of R (only R_0 and R_n are centro-symmetric), the matrix $\hat{A}^i$ is n by 5. As before, the recursions for A^i and $\hat{A}^i$ induce recursions for the orthogonal matrix in the QR factorization :

$$Y_{fb}A = \begin{bmatrix} Y \\ X \end{bmatrix} A = \begin{bmatrix} U \\ V \end{bmatrix} \tag{A.4}$$

and its i^{th} column

$$Y_{fb}A^i = \begin{bmatrix} Y \\ X \end{bmatrix} A^i = \begin{bmatrix} U^i \\ V^i \end{bmatrix} \tag{A.5}$$

Once more the "generic" recursions have to be used to get recursions for the columns of the orthogonal matrix.

Define the matrices W and $\mathcal{W}$ to be the Toeplitz extension of Y and X, respectively :

$$W = \begin{bmatrix} T \\ Y \end{bmatrix} \qquad \mathcal{W} = \begin{bmatrix} \mathcal{T} \\ X \end{bmatrix} \tag{A.6}$$

$$W = \left[\begin{array}{cccc} y_{N-1} & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ y_{N-n} & \cdots & y_{N-1} & 0 \\ \hline & & Y & \end{array}\right] \qquad \mathcal{W} = \left[\begin{array}{cccc} y_0 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ y_{n-1} & \cdots & y_0 & 0 \\ \hline & & X & \end{array}\right] \tag{A.7}$$

The recursions are then derived exactly as before with

$$WA^i = \begin{bmatrix} U^i \\ V^i \end{bmatrix} \qquad W\hat{A}^i = \begin{bmatrix} \hat{U}^i \\ \hat{V}^i \end{bmatrix} \tag{A.8}$$

and

$$\mathcal{W}A^i = \begin{bmatrix} \mathcal{V}^i \\ V^i \end{bmatrix} \qquad \mathcal{W}\hat{A}^i = \begin{bmatrix} \hat{\mathcal{V}}^i \\ \hat{V}^i \end{bmatrix} \tag{A.9}$$

The recursions are

$$\begin{bmatrix} \hat{\mathcal{U}}^{i+1} \\ \hat{U}^{i+1} \end{bmatrix} = \begin{bmatrix} \hat{\mathcal{U}}^{i} \\ \hat{U}^{i} \end{bmatrix} + Z \begin{bmatrix} \mathcal{U}^{i} \\ U^{i} \end{bmatrix} K_{i+1}^{T}$$

$$\begin{bmatrix} \mathcal{U}^{i+1} \\ U^{i+1} \end{bmatrix} = Z \begin{bmatrix} \mathcal{U}^{i} \\ U^{i} \end{bmatrix} + \begin{bmatrix} \hat{\mathcal{U}}^{i} \\ \hat{U}^{i} \end{bmatrix} \Delta_i^{-1} K_{i+1}$$

$$\begin{bmatrix} \hat{\mathcal{V}}^{i+1} \\ \hat{V}^{i+1} \end{bmatrix} = \begin{bmatrix} \hat{\mathcal{V}}^{i} \\ \hat{V}^{i} \end{bmatrix} + Z \begin{bmatrix} \mathcal{V}^{i} \\ V^{i} \end{bmatrix} K_{i+1}^{T} \qquad (A.10)$$

$$\begin{bmatrix} \mathcal{V}^{i+1} \\ V^{i+1} \end{bmatrix} = Z \begin{bmatrix} \mathcal{V}^{i} \\ V^{i} \end{bmatrix} + \begin{bmatrix} \hat{\mathcal{V}}^{i} \\ \hat{V}^{i} \end{bmatrix} \Delta_i^{-1} K_{i+1} \qquad \text{for } i = 0, \ldots, n-1.$$

with the initializations

$$\begin{bmatrix} \mathcal{U}^{0} \\ U^{0} \end{bmatrix} = \begin{bmatrix} y_{N-1} & \cdots & y_{N-n} & | & y_{N-n-1} & \cdots & y_1 & y_0 \end{bmatrix}^T$$

$$\begin{bmatrix} \mathcal{V}^{0} \\ V^{0} \end{bmatrix} = \begin{bmatrix} y_0 & \cdots & y_{n-1} & | & y_n & \cdots & y_{N-1} \end{bmatrix}^T$$

$$\begin{bmatrix} \hat{\mathcal{U}}^{0} \\ \hat{U}^{0} \end{bmatrix} = \begin{bmatrix} y_{N-1} & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ y_{N-n} & 0 & 0 & 0 & 0 \\ \hline y_{N-n-1} & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ y_0 & 0 & 0 & 0 & 0 \end{bmatrix} \qquad \begin{bmatrix} \hat{\mathcal{V}}^{0} \\ \hat{V}^{0} \end{bmatrix} = \begin{bmatrix} y_0 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ y_{n-1} & 0 & 0 & 0 & 0 \\ \hline y_n & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ y_{N-1} & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\sigma_0^2 = r_{0,0} = (U^0)^T U^0 \qquad \Delta_0 = \sigma_0^{-2} \begin{bmatrix} \sigma_0^2 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \qquad (A.11)$$

The five components of the reflection coefficient K_{i+1} are computed using

$$-\sigma_i^2 K_{i+1}(1) = \begin{bmatrix} (U^0)^T, 0 \end{bmatrix} \begin{bmatrix} \mathcal{U}^i(n) \\ U^i \end{bmatrix} + \begin{bmatrix} (V^0)^T, 0 \end{bmatrix} \begin{bmatrix} \mathcal{V}^i(n) \\ V^i \end{bmatrix}$$

$$\sigma_i^2 K_{i+1}(2) = U^i(N-n) \qquad \sigma_i^2 K_{i+1}(3) = -\mathcal{U}^i(n)$$

$$\sigma_i^2 K_{i+1}(4) = -\mathcal{V}^i(n) \qquad \sigma_i^2 K_{i+1}(5) = V^i(N-n) \qquad (A.12)$$

which comes from the use of

$$\sigma_i^2 K_{i+1}^T = -(ZA^i)^T \begin{bmatrix} r_{0,0} & 0 & 0 & 0 & 0 \\ r_{1,0} & -y_0 & y_{N-n} & y_{n-1} & -y_{N-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ r_{n,0} & -y_{n-1} & y_{N-1} & y_0 & -y_{N-n} \end{bmatrix} \qquad (A.13)$$

There are just two inner products per update.

List of Figures

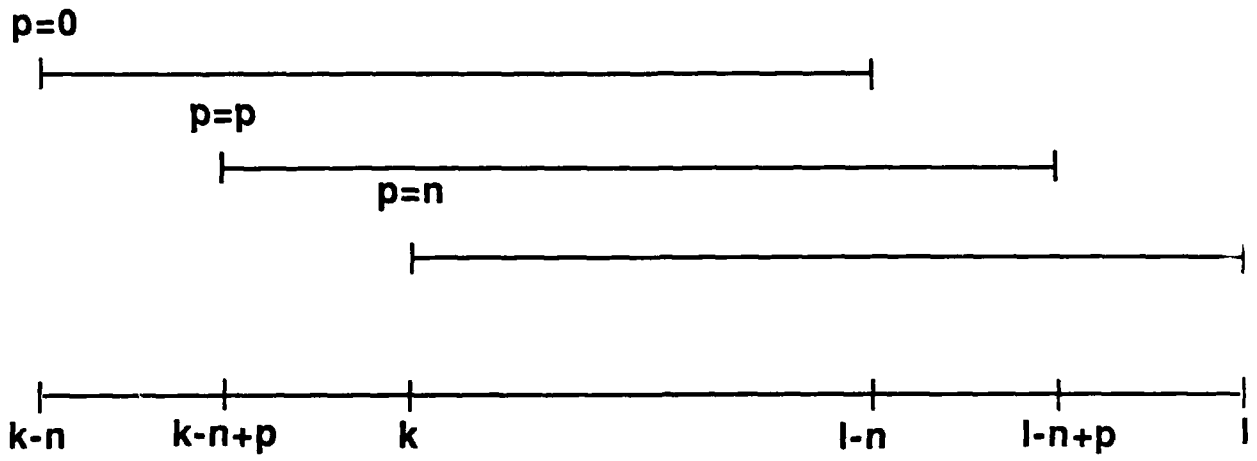


Figure 1 : Illustration of the error windows for prediction.

orders increasing from p=0 to p=n.

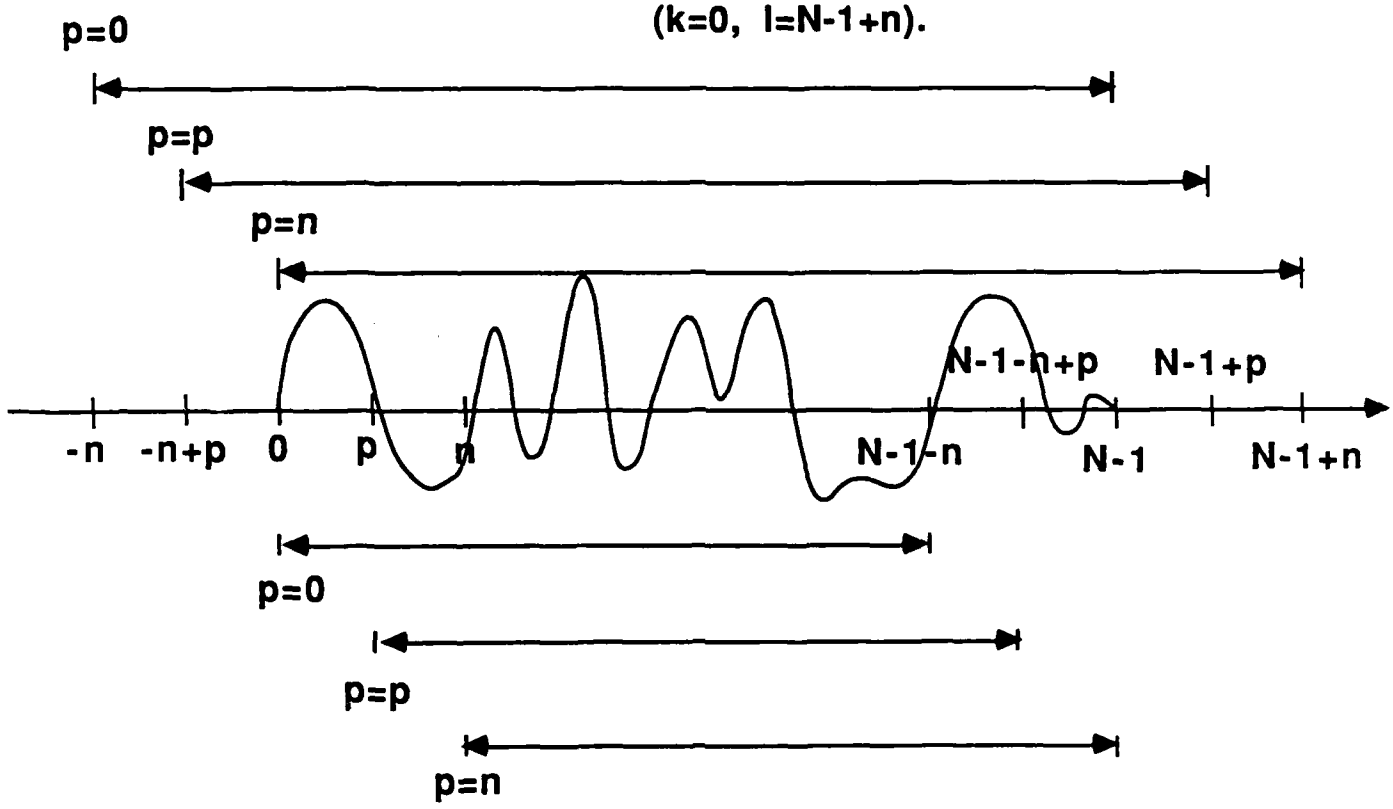


Figure 2 : Error Window for the Correlation and Covariance Methods of Linear Prediction.

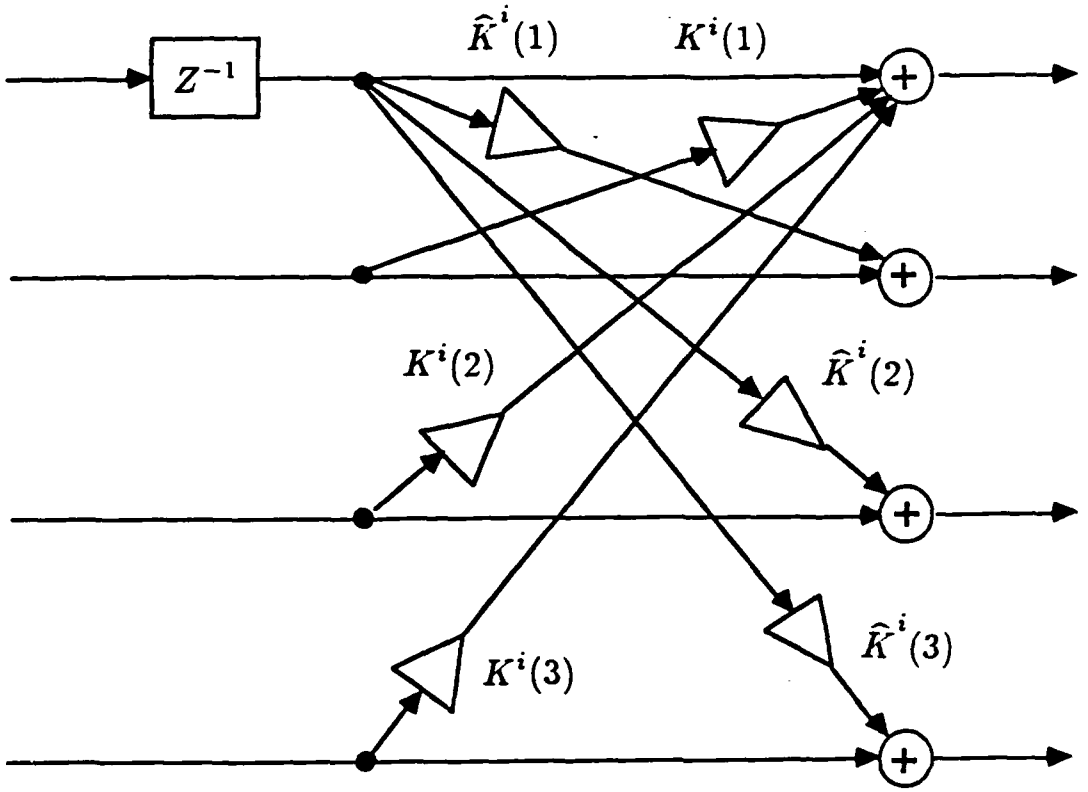


Figure 3 : MA Lattice Filter Cell

(Covariance Case)

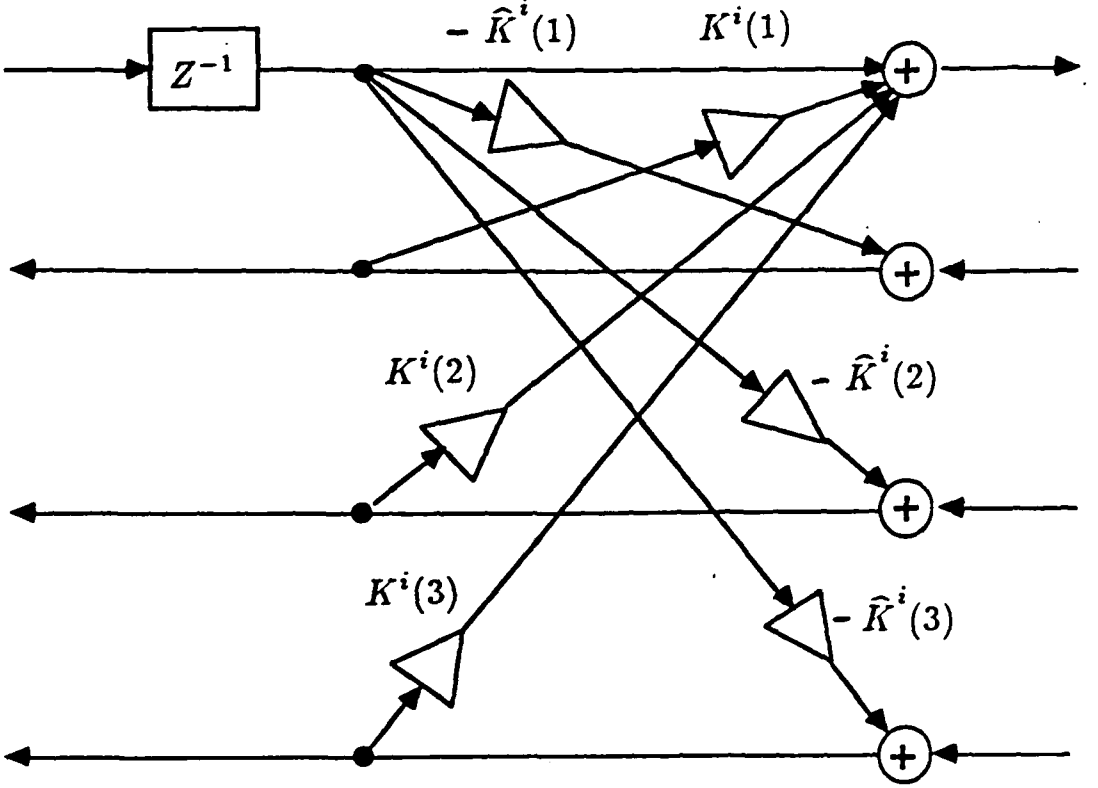


Figure 4 : AR Lattice Filter Cell

(Covariance Case)

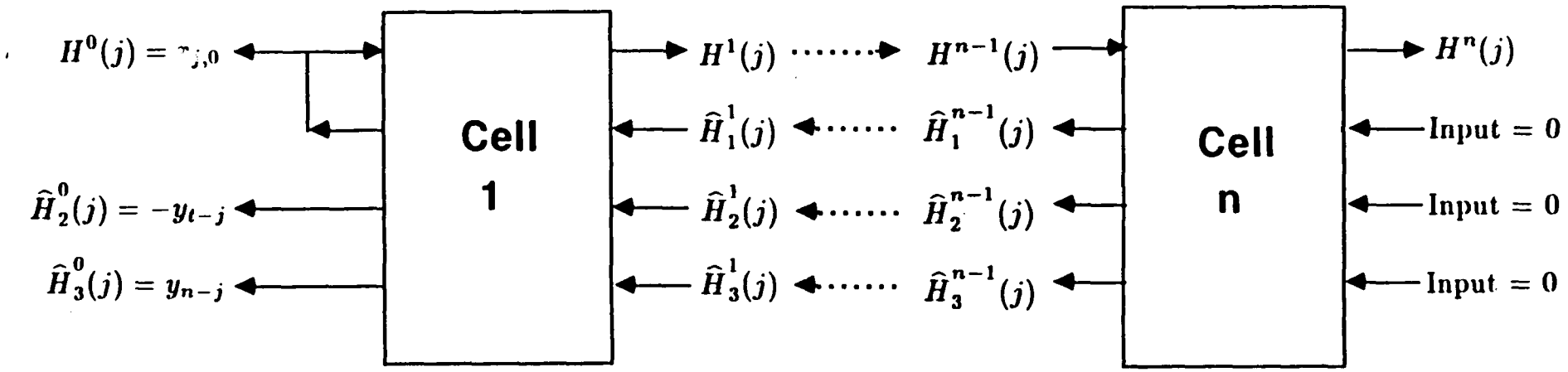


Figure 5 : AR Lattice Filter (Covariance Case).

Recover Covariance and Cholesky Factor from Reflection Coefficients

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS None
2a. SECURITY CLASSIFICATION AUTHORITY DISCASS	3. DISTRIBUTION/AVAILABILITY OF REPORT Unrestricted
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) CU/ONR/1	5. MONITORING ORGANIZATION REPORT NUMBER(S) N/A
6a. NAME OF PERFORMING ORGANIZATION Dept. Elect. & Computer Eng. University of Colorado 6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION Office of Naval Research Mathematics Division
6c. ADDRESS (City, State and ZIP Code) Campus Box 425 Boulder, CO 80309-0425	7b. ADDRESS (City, State and ZIP Code) 800 North Quincy Street Arlington, VA 22217-5000
8a. NAME OF FUNDING/SPONSORING ORGANIZATION 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (City, State and ZIP Code)	10. SOURCE OF FUNDING NOS. PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT NO.
11. TITLE (Include Security Classification) Lattice Algorithms for Computing QR and Cholesky Factors in the Least Squares Theory of Linear Prediction (U)	
12. PERSONAL AUTHOR(S) Cedric J. Demeure and Louis L. Scharf	
13a. TYPE OF REPORT Technical 13b. TIME COVERED FROM _____ TO _____	14. DATE OF REPORT (Yr., Mo., Day) September 1987 15. PAGE COUNT 39
16. SUPPLEMENTARY NOTATION	
17. COSATI CODES FIELD / GROUP / SUB. GR.	18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

In this paper we pose a sequence of linear prediction problems that are a little different from those previously posed. By solving the sequence of problems we are able to QR factor data matrices of the type usually associated with correlation, pre and post-windowed, and covariance methods of linear prediction. Our solutions cover the forward, backward, and forward-backward problems. The QR factor orthogonalizes the data matrix and solves the problem of Cholesky factoring the experimental correlation matrix and its inverse. This means we may use generalized Levinson algorithms to derive generalized QR algorithms, which are then used to derive generalized Schur algorithms. All three algorithms are true lattice algorithms that may be implemented either on a vector machine or on a multi-tier lattice, and all three algorithms generate generalized reflection coefficients that may be used for filtering or classification.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS ☒	21. ABSTRACT SECURITY CLASSIFICATION U
22a. NAME OF RESPONSIBLE INDIVIDUAL Louis L. Scharf	22b. TELEPHONE NUMBER (Include Area Code) 303/492-8283 22c. OFFICE SYMBOL

DD FORM 1473, 83 APR EDITION OF 1 JAN 73 IS OBSOLETE.

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DISTRIBUTION:

Dr. Neil L. Gerr
Office of Naval Research
800 North Quincy Street
Arlington, VA 22217-5000

Office of Naval Research
Resident Representative
University of New Mexico
Room 223, Bandolier Hall West
Albuquerque, NM 87131

Director, Naval Research Laboratory
Attn: Code 2627
Washington, DC 20375

Defense Technical Information Center
Building 5
Cameron Station
Alexandria, VA 22314

Martha Campbell
Government Publications
University of Colorado
Campus Box 184
Boulder, CO 80309-0184

Office of Contracts & Grants
University of Colorado
Campus Box 19
Boulder, CO 80309-0019