A *D*-Step Predictor in Lattice and Ladder Form

MICHEL R. GEVERS, MEMBER, IEEE, AND VINCENT J. WERTZ

Abstract—We use the orthogonalizing property of the two-multiplier linear prediction lattice filter to construct a d-step ahead predictor in lattice form. The predictor generates d-step forward and backward residuals in a recursive way and possesses most of the interesting properties of the basic one-step prediction lattice filter. An exact solution is presented first assuming a stationary observation process, using orthogonal projections in Hilbert space. Two adaptive implementations are also proposed for the case where the statistics of the signal process are unknown or time varying: a gradient method and a recursive least-squares scheme. Finally, we show how to construct an adaptive d-step ahead predictor by adding a ladder part to the d-step lattice structure.

I. Introduction

Since the publication of Itakura and Saito's two-multiplier lattice filter [1], lattice filters have been extensively studied and have given rise to a number of applications in linear prediction, communication, signal processing, and identification. The main feature of the lattice filter is that it is an orthogonalizing device which replaces the original signal process by a sequence of orthogonal residuals generating the same space. As a result the adaptation of every stage of a lattice filter is decoupled from the previous stages and convergence of the estimated reflection coefficients is faster than when the coefficients of autoregressive models are estimated using standard techniques. Also, the stability of a lattice filter can be checked by inspection and, in an adaptive implementation, the reflection coefficients can always be computed in such a way that stability of the shaping filter is guaranteed.

Because the lattice filter produces as its output one-step ahead and one-step backward residuals of the incoming observations, it is straightforward to use it as a one-step prediction filter and this has been one of its most obvious applications. As such it is nothing but a clever implementation of the Levinson algorithm [2], [3]. We show in this paper that the basic orthogonalizing property of the lattice filter also allows one to produce d-step forward and backward predictions of the observations with little extra computation. The basic idea is to use the sequence of one-step backward residuals as an orthogonal basis for the space of past observations, and to construct the d-step predicted estimates of the signal process $y(t)$ by projecting the future values of $y(t)$ on this basis. Similarly, backward d-step predicted estimates of $y(t)$ are constructed by projecting past values of $y(t)$ on the appropriate basis formed by orthogonalized one-step forward residuals. As can be guessed, our presentation will rely heavily on projection arguments in Hilbert space. Using similar ideas, a d-step filter has been derived independently by Reddy *et al.* [20], but the properties of the d-step filter are not studied in that paper.

The basic two-multiplier lattice filter generating one-step forward and backward residuals has given rise to a number of adaptive implementations.

A stochastic approximation scheme proposed by Griffiths [5] has been further developed and studied by Makhoul [9], Makhoul and Viswanathan [10], Griffiths and Medaugh [6], and Carter [8]. Goodwin and Sin [18] have suggested a gradient algorithm in which sensitivity lattice forms are added to the basic lattice filter for the computation of the gradient. Finally, Morf and his co-workers [12]–[16], and Shensa [17] have derived an "exact" recursive least-squares implementation, where the word "exact" means that, given the data $\{y(o), \cdots y(t-1)\}$, the output $\hat{y}(t|t-1,\cdots,t-k)$ of the kth order adaptive lattice filter coincides with the prediction $\hat{y}(t)$ obtained by fitting a kth order autoregressive model by least-squares to the same set of data. Morf's solution requires more memory and more computations per step, but it has the major advantage of a much faster convergence (in the stationary case), or a much faster tracking capability (in the case where the statistics of the signal change and where a forgetting factor is built into the algorithm). While the recursive least-squares lattice form has the convergence properties inherent to least-squares solutions for autoregressive models, convergence has not been proved for the other two schemes.

In this paper we present two adaptive implementations of the d-step lattice filter. First, a gradient algorithm is presented along the lines of those developed in [5] and [9] for the one-step lattice filter. We also present an "exact" recursive least-squares implementation of our d-step lattice filter along the lines of the one-step recursive least-squares implementations of Morf *et al.* [12]–[16]. Each of these two schemes can easily be adjusted to account for time-varying

Manuscript received November 3, 1980 and January 27, 1981; revised October 20, 1981 and March 15, 1982. Paper recommended by H. V. Poor, Past Chairman of the Estimation Committee. This work was supported by IRSIA and the Australian Research Grants Committee.

The authors were on leave at the University of Newcastle, Newcastle, N. S. W., Australia. They are with the Department of Electrical Engineering, Louvain University, Louvain-la-Neuve, Belgium.

parameters by the introduction of exponential weighting (or fading memory) factors.

The paper is organized as follows. In Section II we recall the basic two-multiplier lattice structure of Itakura and Saito together with some of its main orthogonality properties. We use the orthogonal sequences produced by this basic lattice filter to derive a d-step lattice–ladder filter in Section III, and we show its similarity with the one-step filter. The d-step filter also obeys a number of interesting orthogonality properties. By the addition of a ladder part it can be transformed into a d-step predictor for the observation process. So far the lattice and ladder recursions have been derived assuming that the observation process $y(t)$ is (wide-sense) stationary with known covariances. In Sections IV and V we give two adaptive implementations of the d-step filter (a gradient algorithm and a recursive least-squares scheme) for the case of unknown and/or time-varying statistics. Finally, in Section VI we show how the adaptive lattice schemes can be transformed into adaptive d-step ahead predictors by the addition of a ladder part. In the recursive least-squares algorithm we show that two slightly different predictors can be defined, and we discuss the connection between these two schemes.

II. The Basic Lattice Structure

In this section we first give the equations of the two-multiplier lattice filter that was introduced by Itakura and Saito [1], and has been further studied by a number of authors [4]–[20]. This lattice structure constitutes a whitening filter and provides all-pole models of increasing order for the signal process. In addition, one-step ahead predictions of increasing order for the signal process are obtained from this lattice form with almost no extra computations. A major advantage of the lattice formulation for the prediction or the modeling of signal processes is that the stability of the lattice filter can be checked by inspection. The material of this section is not new, but will be needed for the construction of the d-step prediction filter.

We shall consider a real valued (wide-sense) stationary vector random process $y(t)$, $t \in Z$, $y(t) \in \mathbb{R}^p$, and the Hilbert space H spanned by the components of the $y(t)$'s. We shall denote by Y_t^{t+k} for $k \geqslant 0$ the closed linear subspace of H spanned by the components of $\{y(t), y(t+1), \cdots, y(t+k)\}$; Y_t^{t+k} for $k<0$ will denote the empty space. Finally, for every element $y(\tau)$ in H, $E\{y(\tau)|Y_t^{t+k}\}$ will denote the orthogonal projection of $y(\tau)$ onto Y_t^{t+k}, i.e.,

$$E\{(y(\tau)-E\{y(\tau)|Y_t^{t+k}\})y^T(j)\}=0, \qquad j=t,\cdots,t+k.$$

With this definition $E\{y(t)|y(s)\} = E\{y(t)y^T(s)\} E\{y(s)y^T(s)\}^{-1}y(s)$.[1] In the Gaussian case, $E\{y(t)|y(s)\}$ is the conditional expectation of $y(t)$ given $y(s)$.

[1] We assume that y is a full rank process, so that the inverse exists.

We now introduce the following random processes associated with the process $y(t)$.

$$f_k(t) \triangleq y(t)-E\{y(t)|Y_{t-k}^{t-1}\} \tag{2.1a}$$

$$g_k(t) \triangleq y(t-k)-E\{y(t-k)|Y_{t-k+1}^{t}\} \tag{2.1b}$$

$$f_0(t)=g_0(t)=y(t). \tag{2.1c}$$

The variables $f_k(t)$ and $g_k(t)$ are called the forward and backward residuals of order k of the process $y(t)$. It can be shown that they can be computed by the following recursive formulas:

$$f_{k+1}(t)=f_k(t)-K_{k+1}^b g_k(t-1) \tag{2.2a}$$

$$g_{k+1}(t)=g_k(t-1)-K_{k+1}^f f_k(t) \tag{2.2b}$$

$$f_0(t)=g_0(t)=y(t) \tag{2.2c}$$

where

$$K_{k+1}^b=S_k\left(R_k^b\right)^{-1}, \quad K_{k+1}^f=S_k^T\left(R_k^f\right)^{-1} \tag{2.3}$$

$$S_k=E\{f_k(t)g_k^T(t-1)\} \tag{2.4}$$

$$R_k^b=E\{b_k(t)g_k^T(t)\}, \quad R_k^f=E\{f_k(t)f_k^T(t)\}. \tag{2.5}$$

The initial conditions are given by (2.1c). Equations (2.2) can be implemented as a lattice filter as shown in Fig. 1.

Properties of the Basic Lattice Filter

We list here some properties of the lattice filter which will be used later on. Properties 1–3 can be found, e.g., in [9] while 4 is easy to derive.

1) The reflection coefficient matrices K_{k+1}^b and K_{k+1}^f are such that they minimize trR_{k+1}^f and trR_{k-1}^b, respectively.

2) In the scalar case $K_k^b=K_k^f=K_k$, while $R_k^b=R_k^f=R_k$.

3) The forward and backward residuals satisfy the following orthogonality properties:

$$E\{f_k(t+k)f_j^T(t+j)\}=R_k^f\delta_{kj} \tag{2.6}$$

$$E\{g_k(t)g_j^T(t)\}=R_k^b\delta_{kj} \tag{2.7}$$

$$E\{g_k(t)y^T(t-i)\}=0 \qquad 0\leqslant i\leqslant k-1 \tag{2.8}$$

$$E\{f_k(t)y^T(t-i)\}=0 \qquad 1\leqslant i\leqslant k. \tag{2.9}$$

Two alternative expressions for S_k (and hence, for K_{k+1}^f and K_{k+1}^b) can be derived from (2.8) and (2.9)

$$S_k=E\{y(t)g_k^T(t-1)\}=E\{f_k(t)y^T(t-k-1)\}. \tag{2.10}$$

It also follows easily from (2.6), (2.7), and the definitions (2.1) that both $\{g_0(t), g_1(t), \cdots, g_{k-1}(t)\}$ and $\{f_0(t-k+1), f_1(t-k+2), \cdots, f_{k-1}(t)\}$ are orthogonal bases for Y_{t-k+1}^t.

4) Using $\{g_0(t), \cdots, g_{k-1}(t)\}$ as an orthogonal basis for Y_{t-k+1}^t leads to the following expression for $E\{y(t+1)|Y_{t-k+1}^t\}$:

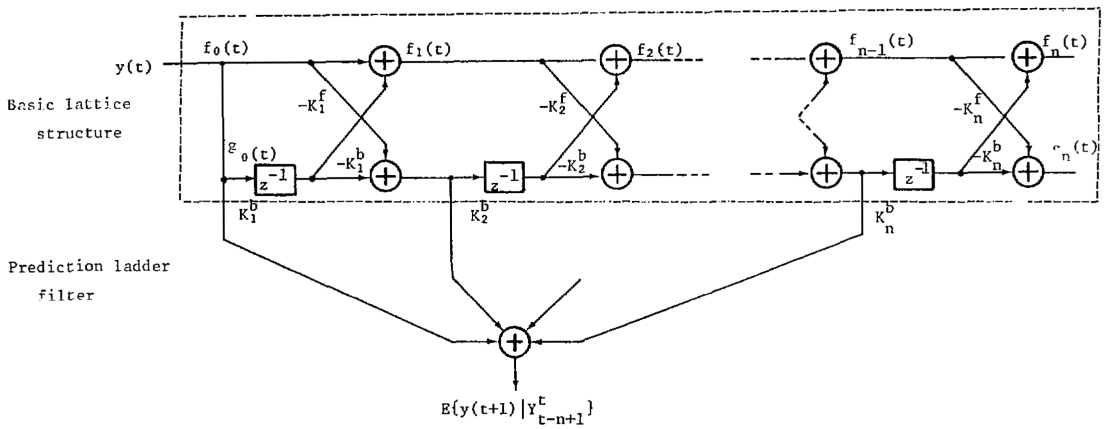


Fig. 1.

$$E\{y(t+1)|Y_{t-k+1}^t\}$$
$$= E\{y(t+1)|g_0(t), g_1(t), \cdots, g_{k-1}(t)\}$$
$$= \sum_{i=0}^{k-1} E\{y(t+1)|g_i(t)\}$$
$$= \sum_{i=0}^{k-1} E\{f_i(t+1)|g_i(t)\} \tag{2.11}$$
$$= \sum_{i=0}^{k-1} K_{i+1}^b g_i(t). \tag{2.12}$$

Equality (2.11) follows from the definition of $f_i(t+1)$ and the orthogonality property (2.8).

The expression (2.12) for the one-step ahead predicted estimate of $y(t+1)$ is interesting because all the terms in the summation are readily available from the lattice filter of Fig. 1. Therefore, the lattice filter is not only a whitening filter and a modeling filter for the $y(t)$ process; with an additional summation (which now constitutes a ladder filter) it also provides a one-step ahead predictor, for stationary processes with known covariances, as shown in Fig. 1. A corresponding expression can be obtained for the one-step backward predictor, by projecting $y(t-k)$ on the orthogonal basis $\{f_0(t-k+1), f_1(t-k+2), \cdots, f_{k-1}(t)\}$ and using (2.10)

$$E\{y(t-k)|Y_{t-k+1}^t\} = \sum_{i=1}^{k} E\{y(t-k)|f_{i-1}(t-k+i)\}$$
$$= \sum_{i=1}^{k} S_{i-1}^T (R_{i-1}^f)^{-1} f_{i-1}(t-k+i)$$
$$= \sum_{i=1}^{k} K_i^f f_{i-1}(t-k+i). \tag{2.13}$$

III. The *D*-Step Ahead Predictor in Lattice and Ladder Form

The same idea that was used in the last section to construct a one-step ahead predictor for $y(t+1)$ from an orthogonal transformation of Y_{t-k+1}^t using the backward residual sequence will be used now to construct a d-step ahead predictor. By the same argument as before we have

$$E\{y(t+d)|Y_{t-k+1}^t\}$$
$$= E\{y(t+d)|g_0(t), g_1(t), \cdots, g_{k-1}(t)\}$$
$$= \sum_{i=0}^{k-1} E\{y(t+d)|g_i(t)\} \tag{3.1}$$
$$= \sum_{i=0}^{k-1} K_{d,i+1}^b g_i(t) \tag{3.2}$$

where

$$K_{d,i+1}^b = E\{y(t+d)g_i^T(t)\}(R_i^b)^{-1} \triangleq S_{d,i}(R_i^b)^{-1}. \tag{3.3}$$

Notice that for $d=1$, we have

$$K_{1,i+1}^b = K_{i+1}^b, \qquad S_{1,i} = S_i. \tag{3.4}$$

We now define the *d-step forward residual of order k*

$$f_{d,k}(t+d) = y(t+d) - E\{y(t+d)|Y_{t-k+1}^t\}. \tag{3.5}$$

Note that $f_{1,k}(t) = f_k(t)$. With these d-step ahead residuals we can now derive an alternative expression for $S_{d,i}$ which will be similar to (2.4) in the case of one-step ahead predictions. Using the orthogonality conditions (2.8) we have

$$S_{d,i} = E\{y(t+d)g_i^T(t)\} = E\{f_{d,i}(t+d)g_i^T(t)\} \tag{3.6}$$
$$= E\{f_{d,i}(t+d)y^T(t-i)\}. \tag{3.7}$$

Finally, we derive a recursive relation for $f_{d,k}(t)$, where the recursion is on k, the order of the predictor. From (3.5) and (3.2) we have

$$f_{d,k+1}(t) = y(t) - E\{y(t)|Y_{t-d-k}^{t-d}\}$$
$$= y(t) - \sum_{i=0}^{k} K_{d,i+1}^b g_i(t-d). \tag{3.8}$$

Hence

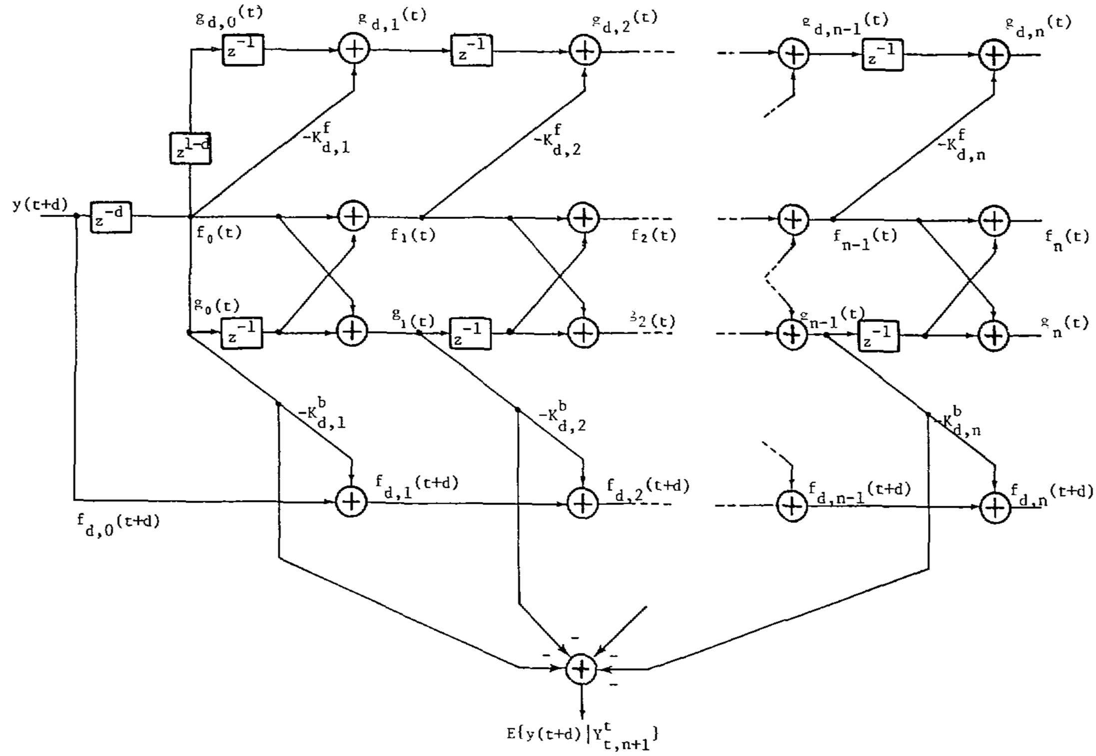


Fig. 2.

$$f_{d,k+1}(t)=f_{d,k}(t)-K^b_{d,k+1}g_k(t-d) \tag{3.9}$$

with initial condition

$$f_{d,0}(t)=y(t). \tag{3.10}$$

Note the similarity between (3.9) and (2.2a); for $d=1$ these two equations become identical. The d-step ahead predictor in lattice and ladder form is represented in Fig. 2. The basic lattice filter is used to generate the backward residuals. These are multiplied by the coefficient matrices $K^b_{d,i}$ to produce the d-step predictor for $E\{y(t+d)|Y^t_{t-k+1}\}$ in the ladder part of the filter. Another part of the ladder filter, which is not strictly necessary if only predicted estimates of y are required and the $K^b_{d,i}$ are available, constructs the prediction error residuals of increasing order; this part does not require any additional multiplication. The coefficient matrices $K^b_{d,i}$ are not provided by the lattice structure, but have to be either known or constructed. Either of the expressions (3.6) or (3.7), together with (3.3), can be used as a starting point to generate estimates of $K^b_{d,i}$, although some are preferable to others as we shall argue in Section IV, where we shall discuss adaptive implementations of d-step predictors.

D-step backward residuals of order k can be defined similarly

$$g_{d,k}(t)\triangleq y(t-d-k+1) - E\{y(t-d-k+1)|Y^t_{t-k+1}\} \tag{3.11}$$

with $g_{1,k}(t)=g_k(t)$.

$$g_{d,k+1}(t)=g_{d,k}(t-1)-K^f_{d,k+1}f_k(t) \tag{3.12}$$

$$=y(t-d-k)-\sum_{i=0}^{k}K^f_{d,i+1}f_i(t-k+i) \tag{3.13}$$

with $g_{d,0}(t)=y(t-d+1)$.

$$K^f_{d,i+1}=E\{y(t-d-i)f_i^T(t)\}(R^f_i)^{-1} \triangleq T_{d,i}(R^f_i)^{-1}. \tag{3.14}$$

Alternative expressions for $T_{d,i}$ are

$$T_{d,i}=E\{y(t-d-i)f_i^T(t)\}$$

$$=E\{g_{d,i}(t-1)f_i^T(t)\} \tag{3.15}$$

$$=E\{g_{d,i}(t-1)y^T(t)\}. \tag{3.16}$$

Finally, the d-step backward predictor for $y(t)$ can now be written as

$$E\{y(t-d-k+1)|Y^t_{t-k+1}\}=\sum_{i=1}^{k}K^f_{d,i}f_{i-1}(t-k+i). \tag{3.17}$$

In this last expression, we have used $\{f_0(t-k+1), f_1(t-k+2),\cdots,f_{k-1}(t)\}$ as a basis for Y^t_{t-k+1}. Note the complete symmetry between (3.9) and (3.12), (3.8) and (3.13), (3.6)-(3.7), and (3.15)-(3.16), (3.2) and (3.17). Finally, note that, for $d=1$, the expressions of Section III become identical to those of Section II with $f_{1,k}(t)=f_k(t)$, $g_{1,k}(t)=g_k(t)$, $S_{1,k}=S_k=T_{1,k}^T$, $K_{1,k}^f=K_k^f$, and $K_{1,k}^b=K_k^b$.

Expressions (3.9) and (3.12) provide recursive relations for $f_{d,k}(t)$ and $g_{d,k}(t)$, where the recursion is on k, the order of the predictor. We now establish a relation between $(d+1)$-step and d-step forward and backward residuals.

$$\begin{aligned} f_{d,k}(t) &= y(t)-E\{y(t)|Y_{t-d-k+1}^{t-d}\} \\ &= y(t)-E\{y(t)|Y_{t-d-k+1}^{t-d-1}\} \\ &\quad + E\{y(t)|Y_{t-d-k+1}^{t-d-1}\}-E\{y(t)|Y_{t-d-k+1}^{t-d}\} \\ &= f_{d+1,k-1}(t)-E\{y(t)|f_{k-1}(t-d)\} \\ &= f_{d+1,k-1}(t)-L_{d,k}^f f_{k-1}(t-d) \end{aligned} \tag{3.18}$$

$$L_{d,k}^f \triangleq M_{d,k-1}^f\left(R_{k-1}^f\right)^{-1} \tag{3.19}$$

with

$$\begin{aligned} M_{d,k-1}^f &\triangleq E\{y(t)f_{k-1}^T(t-d)\} \\ &= E\{f_{d+1,k-1}(t)f_{k-1}^T(t-d)\}. \end{aligned} \tag{3.20}$$

The last equality follows from the fact that $f_{d+1,k-1}(t)=y(t)-E\{y(t)|Y_{t-d-k+1}^{t-d-1}\}$, and this last term is orthogonal to $f_{k-1}(t-d)$. A symmetrical recursion can be obtained for the backward residuals

$$g_{d,k}(t)=g_{d+1,k-1}(t)-L_{d,k}^b g_{k-1}(t) \tag{3.21}$$

$$L_{d,k}^b \triangleq M_{d,k-1}^b\left(R_{k-1}^b\right)^{-1} \tag{3.22}$$

with

$$\begin{aligned} M_{d,k-1}^b &\triangleq E\{y(t-k-d+1)g_{k-1}^T(t)\} \\ &= E\{g_{d+1,k-1}(t)g_{k-1}^T(t)\}. \end{aligned} \tag{3.23}$$

The recursive formulas (3.18) and (3.21) allow the computation of $(d+1)$-step residuals from d-step residuals or vice-versa. However, they require the computation of the additional correlation matrices $M_{d,k}^f$ and $M_{d,k}^b$, which are not available in the d-step predictor lattice and ladder filter. In contrast, the recursive formulas (3.9) and (3.12) (with a recursion on k) use the coefficient matrices $K_{d,k}^f$ and $K_{d,k}^b$ which are used for the computation of d-step predicted estimates of y. Therefore, an alternative is to compute the $f_{d,k}(t)$ and $g_{d,k}(t)$ for all d using the formulas (3.9) and (3.12) separately for each d.

Properties of the d-Step Forward and Backward Prediction Filter

The basic lattice filter has many interesting orthogonality properties; they have been listed by Makhoul [9]. Similarly, we list here some orthogonality properties of the d-step lattice and ladder filter.

We shall first define the covariances of the d-step forward and backward residuals

$$\begin{aligned} E\{f_{d,k}(t)f_{d,k}^T(t)\} &= R_{d,k}^f, \\ E\{g_{d,k}(t)g_{d,k}^T(t)\} &= R_{d,k}^b. \end{aligned} \tag{3.24}$$

These residuals now have the following properties

$$E\{f_{d,k}(t+k)f_{d,j}^T(t+j)\}=0 \qquad \text{if } |k-j|\geqslant d \tag{3.25}$$

$$E\{g_{d,k}(t)g_{d,j}^T(t)\}=0 \qquad \text{if } |k-j|\geqslant d \tag{3.26}$$

$$E\{g_{d,k}(t)y^T(t-i)\}=0 \qquad 0\leqslant i\leqslant k-1 \tag{3.27}$$

$$E\{f_{d,k}(t)y^T(t-i)\}=0 \qquad d\leqslant i\leqslant d+k-1 \tag{3.28}$$

$$E\{f_{d,k}(t)g_j^T(t)\}=0 \qquad j\geqslant d+k \tag{3.29}$$

$$E\{f_{d,k}(t+d)g_j^T(t)\}=0 \qquad 0\leqslant j\leqslant k-1 \tag{3.30}$$

$$E\{f_{d,k}(t)f_{d,k}^T(t)\}=E\{f_{d,k}(t)y^T(t)\}=R_{d,k}^f \tag{3.31}$$

$$E\{g_{d,k}(t)g_{d,k}^T(t)\}=E\{g_{d,k}(t)y^T(t-d-k+1)\}=R_{d,k}^b \tag{3.32}$$

$$E\{g_{d,k}(t)g_j^T(t)\}=0 \qquad 0\leqslant j\leqslant k-1 \tag{3.33}$$

$$E\{f_{d,k}(t)f_{d,j}^T(t)\}=R_{d,\max(k,j)}^f \tag{3.34}$$

$$E\{g_{d,k}(t+k)g_{d,j}^T(t+j)\}=R_{d,\max(k,j)}^b. \tag{3.35}$$

We prove (3.30) as an example. $f_{d,k}(t+d)$ is given by (3.5), while $g_j(t)=y(t-j)-E\{y(t-j)|Y_{t-j+1}^t\}$. Therefore, $f_{d,k}(t+d)$ is orthogonal to Y_{t-k+1}^t, while $g_j(t)$ lies in Y_{t-j}^t. Hence, these two quantities are orthogonal provided $0\leqslant j\leqslant k-1$. All the other proofs follow similar arguments.

So far, we have presented the basic lattice filter and the d-step lattice–ladder prediction filter under the assumptions of a stationary process and known correlation functions. In most practical applications the correlation functions are not available and must be recursively estimated from the data. For the basic lattice filter, a number of adaptive schemes have been proposed in the last few years for the recursive computation of the reflection coefficients K_k^f and K_k^b from the data. These schemes can be broadly classified into gradient algorithms (see [5]–[10]) and least-squares algorithms (see [12]–[17]); each of these two classes of schemes can be adjusted to account for time varying parameters by the introduction of exponential weighting (or fading memory) factors. In the following two sections, we shall derive two different adaptive algorithms for a d-step ladder filter. For simplicity of notations, we shall assume that the observation process $y(t)$ is scalar.

IV. Adaptive Implementation of a D-Step Predictor in Lattice–Ladder Form: Gradient Algorithm

The purpose of this section is to show that the d-step prediction filter of Section III can be adaptively implemented using a gradient algorithm similar to that of Griffith [5] and Makhoul [9]. First, we describe the adaptive one-step lattice filter. Since $y(t)$ is assumed scalar, $K_k^f = K_k^b = K_k$. The various coefficients are then computed as follows. Given $f_k(\tau)$ and $g_k(\tau)$ up to time t, for $k = 1,\cdots,n$,

$$K_k(t) = \frac{2 \sum_{\tau=t_0+1}^{t} \beta^{t-\tau} f_{k-1}(\tau) g_{k-1}(\tau-1)}{R_{k-1}(t)} \tag{4.1}$$

where

$$R_{k-1}(t) = \sum_{\tau=t_0+1}^{t} \beta^{t-\tau}\left[f_{k-1}^2(\tau)+g_{k-1}^2(\tau-1)\right]. \tag{4.2}$$

Here β is a fading memory factor: $0 < \beta \leqslant 1$, $t_0 = t_0' + n - 1$, where t_0' is the initial observation time and n is the maximal order to be considered for the predictor. Using $K_k(t)$, new residuals $f_k(t+1)$ and $g_k(t+1)$ can be computed, for $k = 1,\cdots,n$, through

$$f_k(t+1) = f_{k-1}(t+1) - K_k(t) g_{k-1}(t) \tag{4.3a}$$

$$g_k(t+1) = g_{k-1}(t) - K_k(t) f_{k-1}(t+1) \tag{4.3b}$$

$$f_0(t+1) = g_0(t+1) = y(t+1). \tag{4.3c}$$

With these new residuals evaluated at $t+1$, one computes a new estimate of $K_k(t+1)$, $k = 1,\cdots,n$. With this particular fading memory algorithm, $K_k(t)$ can also be written as an update of $K_k(t-1)$:

$$K_k(t) = K_k(t-1) + \frac{f_{k-1}(t) g_k(t) + g_{k-1}(t-1) f_k(t)}{R_{k-1}(t)} \tag{4.4}$$

where

$$R_{k-1}(t) = \beta R_{k-1}(t-1) + f_{k-1}^2(t) + g_{k-1}^2(t-1). \tag{4.5}$$

The reflection coefficients $K_k(t)$ will always satisfy $|K_k(t)| < 1$, which guarantees the stability of the lattice filter.

We now propose similar adaptive expressions for $K_{d,k}^b$ and $K_{d,k}^f$ starting from the following exact definitions for the scalar case [see (3.6) and (3.15)]:

$$K_{d,k+1}^b = E\{f_{d,k}(t+d) g_k(t)\}(R_k^b)^{-1} \tag{4.6}$$

$$K_{d,k+1}^f = E\{g_{d,k}(t-1) f_k(t)\}(R_k^f)^{-1}. \tag{4.7}$$

Notice that, even in the scalar case, $K_{d,k}^b \neq K_{d,k}^f$.

The reason for using the above expressions for $K_{d,k+1}^b$ and $K_{d,k+1}^f$ rather than the other definitions involving $y(t)$ [see (3.6), (3.7), (3.15), (3.16)] is that (4.6) and (4.7) involve only residuals, which converge to a white noise process as the order of the model increases. Now it is well known from statistical theory that the variance of the error on the estimated cross-correlation function between two processes is smaller if each of these processes is a white noise process. The adaptive expressions are now as follows:

$$K_{d,k}^b(t) = \frac{\sum_{\tau=t_0+d}^{t} \beta^{t-\tau} f_{d,k-1}(\tau) g_{k-1}(\tau-d)}{R_{k-1}^b(t-d)} \tag{4.8}$$

where

$$R_{k-1}^b(t-d) \triangleq \sum_{\tau=t_0+d}^{t} \beta^{t-\tau} g_{k-1}^2(\tau-d) = \sum_{\tau=t_0}^{t-d} \beta^{t-d-\tau} g_{k-1}^2(\tau). \tag{4.9}$$

$R_{k-1}^b(t-d)$ and $K_{d,k}^b(t)$ can be computed recursively.

$$R_{k-1}^b(t-d) = \beta R_{k-1}^b(t-d-1) + g_{k-1}^2(t-d) \tag{4.10}$$

$$K_{d,k}^b(t) = K_{d,k}^b(t-1) + \frac{f_{d,k}(t) g_{k-1}(t-d)}{R_{k-1}^b(t-d)}. \tag{4.11}$$

Similarly,

$$K_{d,k}^f(t) = \frac{\sum_{\tau=t_0+d}^{t} \beta^{t-\tau} g_{d,k-1}(\tau-1) f_{k-1}(\tau)}{R_{k-1}^f(t)} \tag{4.12}$$

where

$$R_{k-1}^f(t) \triangleq \sum_{\tau=t_0+d}^{t} \beta^{t-\tau} f_{k-1}^2(\tau). \tag{4.13}$$

Also

$$R_{k-1}^f(t) = \beta R_{k-1}^f(t-1) + f_{k-1}^2(t) \tag{4.14}$$

$$K_{d,k}^f(t) = K_{d,k}^f(t-1) + \frac{g_{d,k}(t) f_{k-1}(t)}{R_{k-1}^f(t)}. \tag{4.15}$$

Given $K_{d,k}^b(t-1)$, $K_{d,k}^f(t-1)$, and the residuals up to time t, updated coefficients $K_{d,k}^b(t)$ and $K_{d,k}^f(t)$ can be computed using (4.11) and (4.15). With these new coefficients and the new observation $y(t+1)$, one can then compute new residuals using the recursive relations (4.3) for the one-step residuals and the following recursions for the d-step residuals for $k = 1,\cdots,n$:

$$g_{d,k}(t+1) = g_{d,k-1}(t) - K_{d,k}^f(t) f_{k-1}(t+1) \tag{4.16a}$$

$$f_{d,k}(t+1) = f_{d,k-1}(t+1) - K_{d,k}^b(t) g_{k-1}(t-d+1) \tag{4.16b}$$

$$g_{d,0}(t+1)=y(t-d+2) \tag{4.16c}$$
$$f_{d,0}(t+1)=y(t+1). \tag{4.16d}$$

It might appear that three separate recursive equations are needed to compute the energies $R_k^b(t)$, $R_k^f(t)$, and the total energy $R_k(t)$.

Actually, $R_k(t)$ can be expressed as a function of R_k^b and R_k^f, which reduces the number of computations

$$R_{k-1}(t)=R_{k-1}^f(t)+R_{k-1}^b(t-1)+\beta^t C_{k-1} \tag{4.17}$$

where

$$C_{k-1}=\sum_{\tau=t_0+1}^{t_0+d-1}\beta^{-\tau}f_{k-1}^2(\tau). \tag{4.18}$$

C_{k-1} is a constant term which accounts for the fact that the first term in $R_{k-1}^f(t)$ is $\beta^{t-t_0-d}f_{k-1}^2(t_0+d)$, while the corresponding first term in $R_{k-1}(t)$ is $\beta^{t-t_0-1}f_{k-1}^2(t_0+1)$. Notice that the last term of (4.17) is only important during the initialization phase. It can be dropped for all t such that $t-t_o-d+1>-4/_{1n\beta}$ because then the term $\beta^t C_{k-1}<0.01R_{k-1}^f(t)$. The summations in (4.1), (4.8)-(4.9), and (4.12)-(4.13) have been written under the assumption that $y(t_0-n+1)$ is the first available observation. Alternatively, these summations can all be made to start at $\tau=t_1$ with $t_1\geqslant t_0+d$. We describe the first few steps of the adaptive algorithm in Appendix I to demonstrate that all the computations can be handled causally and recursively from the available data.

V. Adaptive Implementation of a D-Step Predictor in Lattice–Ladder Form: Least-Squares Algorithm

We present now the adaptive implementation of the filter of Section III using a recursive least-squares procedure. As it has been said in the Introduction, this procedure, although leading to more computations than the gradient algorithm of Section IV, has the major advantages of being an "exact" one and having a much faster convergence or tracking capability. Our *d*-step recursive least-squares (RLS) filter is a generalization of the one-step RLS filter derived by Morf *et al.* [12]–[16].

Given the data $\{y(j),\ o\leqslant j\leqslant t\}$ the one-step and *d*-step forward and backward residuals of order k, defined in the previous sections, will have the following form:

$$f_k(t)=y(t)-\sum_{i=1}^{k}a_{k,i}(t)y(t-i)=A_k(t)^T\phi_k(t) \tag{5.1}$$

$$g_k(t)=y(t-k)-\sum_{i=1}^{k}b_{k,i}(t)y(t-k+i)=B_k(t)^T\phi_k(t) \tag{5.2}$$

$$f_{d,k}(t)=y(t)-\sum_{i=1}^{k}a_{d,k,i}(t)y(t-d-i+1)=A_{d,k}(t)^T\phi_{d+k-1}(t) \tag{5.3}$$

$$g_{d,k}(t)=y(t-d-k+1)-\sum_{i=1}^{k}b_{d,k,i}(t)y(t-k+i)=B_{d,k}(t)^T\phi_{d+k-1}(t) \tag{5.4}$$

where

$$A_k(t)^T=[1\quad -a_{k,1}(t)\quad\cdots\quad -a_{k,k}(t)] \tag{5.5}$$

$$B_k(t)^T=[-b_{k,k}(t)\quad\cdots\quad -b_{k,1}(t)\quad 1] \tag{5.6}$$

$$A_{d,k}(t)^T=[1\quad \overbrace{o\ \cdots\ o}^{d-1}\quad -a_{d,k,1}(t)\quad\cdots\quad -a_{d,k,k}(t)] \tag{5.7}$$

$$B_{d,k}(t)^T=\left[-b_{d,k,k}(t)\quad\cdots\quad -b_{d,k,1}(t)\quad \underbrace{o\ \cdots\ o}_{d-1}\quad 1\right] \tag{5.8}$$

$$\phi_k(t)^T=[y(t)\quad y(t-1)\quad\cdots\quad y(t-k)]. \tag{5.9}$$

We also define

$$Y_k(t)=\begin{bmatrix} y(0) & y(1) & \cdots & y(k) & \cdots & y(t)\\ 0 & y(0) & & & & \\ \vdots & \ddots & \ddots & \vdots & & \vdots\\ 0 & \cdots & 0 & y(0) & \cdots & y(t-k)\end{bmatrix} \tag{5.10}$$

and the sample covariance matrix

$$R_k(t)=Y_k(t)Y_k(t)^T. \tag{5.11}$$

It will be assumed throughout that $R_k(t)$ is nonsingular. The coefficient vectors $A_k(t)$, $B_k(t)$, $A_{d,k}(t)$, and $B_{d,k}(t)$ are defined at each time as the solutions of the following minimization problems: $\min A_k(t)^TR_k(t)A_k(t)$, $\min B_k(t)^TR_k(t)B_k(t)$, $\min A_{d,k}(t)^TR_{d+k-1}(t)A_{d,k}(t)$, $\min B_{d,k}(t)^TR_{d+k-1}(t)B_{d,k}(t)$. This amounts to minimizing the sum of the squares of the residuals from 0 to t, but all evaluated with the same coefficients $A_k(t)$ [resp. $B_k(t)$, $A_{d,k}(t)$, $B_{d,k}(t)$]. The minimization yields the following equations:

$$R_k(t)A_k(t)=\begin{array}{c}\left[\begin{array}{c}R_k^f(t)\\ \hline 0\\ \vdots\\ 0\end{array}\right]\begin{array}{l}1\\ \left.\begin{array}{l}\\ \\ \\ \end{array}\right\}k\end{array}\end{array},\quad R_k(t)B_k(t)=\left[\begin{array}{c}0\\ \vdots\\ 0\\ \hline R_k^b(t)\end{array}\right]\begin{array}{l}\left.\begin{array}{l}\\ \\ \\ \end{array}\right\}k\\ 1\end{array} \tag{5.12}$$

$$R_{d-k-1}(t)A_{d,k}(t)=\left[\begin{array}{c}R_{d,k}^f(t)\\ x\\ \vdots\\ x\\ 0\\ \vdots\\ 0\end{array}\right]\begin{array}{l}1\\ \left.\begin{array}{l}\\ \\ \\ \end{array}\right\}d-1\\ \left.\begin{array}{l}\\ \\ \\ \end{array}\right\}k\end{array} \tag{5.13a}$$

$$R_{d+k-1}(t)B_{d,k}(t)=\left[\begin{array}{c}0\\ \vdots\\ 0\\ x\\ \vdots\\ x\\ R_{d,k}^b(t)\end{array}\right]\begin{array}{l}\left.\begin{array}{l}\\ \\ \\ \end{array}\right\}k\\ \left.\begin{array}{l}\\ \\ \\ \end{array}\right\}d-1\\ 1\end{array}. \tag{5.13b}$$

The last k equations on the left of (5.12) and of (5.13a) define $A_k(t)$ and $A_{d,k}(t)$, while the first equation defines $R_k^f(t)$ and $R_{d,k}^f(t)$ and similarly for $B_k(t)$ and $B_{d,k}(t)$. $R_k^f(t)$, $R_{d,k}^f(t)$, $R_k^b(t)$, $R_{d,k}^b(t)$ are the sample covariances of the optimal residuals. The remainder of the section consists in finding recursive expressions for the coefficient vectors $A_k(t)$, $B_k(t)$, $A_{d,k}(t)$, $B_{d,k}(t)$ (recursions both in the order k and in the time t) without having to invert the matrix $R_k(t)$ or $R_{d+k-1}(t)$. These recursions will then lead to an implementation in lattice form. The recursive formulas are derived from the following recursions on $R_k(t)$:

$$R_k(t+1)=R_k(t)+\phi_k(t+1)\phi_k(t+1)^T \tag{5.14}$$

$$R_{k-1}(t)=\begin{array}{c}\begin{array}{c}1\\ k+1\end{array}\left[\begin{array}{c|c}X & X\\ \hline X & R_k(t-1)\end{array}\right]\\ \begin{array}{cc}1 & k+1\end{array}\end{array}=\begin{array}{c}\left[\begin{array}{c|c}R_k(t) & X\\ \hline X & X\end{array}\right]\begin{array}{c}k+1\\ 1\end{array}\\ \begin{array}{cc}k+1 & 1\end{array}\end{array} \tag{5.15}$$

$$R_{d+k}(t)=\begin{array}{c}\begin{array}{c}d\\ k+1\end{array}\left[\begin{array}{c|c}X & X\\ \hline X & R_k(t-d)\end{array}\right]\\ \begin{array}{cc}d & k+1\end{array}\end{array}=\begin{array}{c}\left[\begin{array}{c|c}R_k(t) & X\\ \hline X & X\end{array}\right]\begin{array}{c}k+1\\ d\end{array}\\ \begin{array}{cc}k+1 & d\end{array}\end{array} \tag{5.16}$$

where the X are submatrices of appropriate dimensions. We will also need two auxiliary quantities

a $(k+1)-$vector:

$$C_k(t)=R_k(t)^{-1}\phi_k(t) \tag{5.17}$$

a scalar:

$$\gamma_k(t)=\phi_k(t)^TR_k(t)^{-1}\phi_k(t)=\phi_k(t)^TC_k(t). \tag{5.18}$$

Note that $0\leqslant\gamma_k(t)<1$. Order and time update recursions for the one step (but not the d-step) prediction filter have been given in [13]–[17]. The formulas in [16] basically have the same complexity as the gradient method. Closely related recursive formulas have also been given in [21] in the context of Levinson predictors. A tutorial and self-contained derivation of both the one-step and the d-step predictor formulas is given in [19]. In order to keep this paper within reasonable limits, we will directly give the final formulas for the d-step filter. (See Appendix II for the derivation of these recursions.)

Order Update Recursions for the d-Step Lattice Filter

$$A_{d,k+1}(t)=\begin{bmatrix}A_{d,k}(t)\\ 0\end{bmatrix}-\begin{bmatrix}0\\ \vdots\\ 0\\ B_k(t-d)\end{bmatrix}\frac{S_{d,k}(t)}{R_k^b(t-d)} \tag{5.19}$$

$$R_{d,k+1}^f(t)=R_{d,k}^f(t)-\frac{T_{d,k}^*(t)S_{d,k}(t)}{R_k^b(t-d)} \tag{5.20}$$

$$f_{d,k-1}(t)=f_{d,k}(t)-K_{d,k+1}^b(t)g_k(t-d) \tag{5.21}$$

$$B_{d,k+1}(t)=\begin{bmatrix}0\\ B_{d,k}(t-1)\end{bmatrix}-\begin{array}{c}d\left\{\begin{array}{c}\\ \\ \\ \end{array}\right.\end{array}\!\!\begin{bmatrix}A_k(t)\\ 0\\ \vdots\\ 0\end{bmatrix}\frac{S_{d,k}^*(t)}{R_k^f(t)} \tag{5.22}$$

$$R_{d,k+1}^b(t)=R_{d,k}^b(t)-\frac{T_{d,k}(t)S_{d,k}^*(t)}{R_k^f(t)} \tag{5.23}$$

$$g_{d,k+1}(t)=g_{d,k}(t-1)-K_{d,k+1}^f(t)f_k(t) \tag{5.24}$$

where

$$S_{d,k}(t)=\left[\text{last row of } R_{d+k}(t)\right]\begin{bmatrix}A_{d,k}(t)\\ 0\end{bmatrix} \tag{5.25}$$

$$S_{d,k}^*(t)=\left[\text{first row of } R_{d+k}(t)\right]\begin{bmatrix}0\\ B_{d,k}(t-1)\end{bmatrix} \tag{5.26}$$

$$T_{d,k}^*(t)=\left[\text{first row of } R_{d+k}(t)\right]\begin{bmatrix}0\\ \vdots\\ 0\\ B_k(t-d)\end{bmatrix}\begin{array}{l}\left.\begin{array}{l}\\ \\ \end{array}\right\}d\\ \\ \end{array} \tag{5.27}$$

$$T_{d,k}(t)=\left[\text{last row of } R_{d+k}(t)\right]\begin{bmatrix}A_k(t)\\ 0\\ \vdots\\ 0\end{bmatrix}\begin{array}{l}\\ \left.\begin{array}{l}\\ \\ \end{array}\right\}d\end{array} \tag{5.28}$$

$$K_{d,k-1}^b(t)=\frac{S_{d,k}(t)}{R_k^b(t-d)},\quad K_{d,k+1}^f(t)=\frac{S_{d,k}^*(t)}{R_k^f(t)}. \tag{5.29}$$

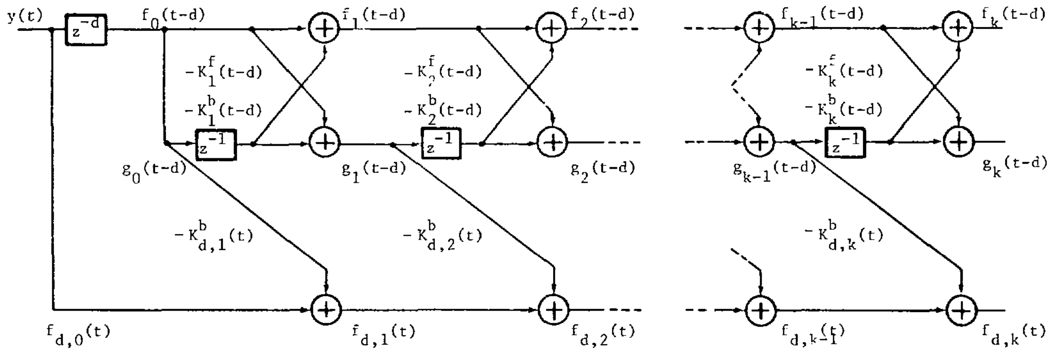


Fig. 3.

Equations (5.21) and (5.24) constitute the basic order recursions for the d-step forward and backward residuals. The recursions start with $f_{d,o}(t)=y(t)$ and $g_{d,o}(t)=y(t-d+1)$. Note that the recursion for $f_{d,k}(t)$ is obtained by adding a ladder part to the basic lattice filter, which is used to generate the one-step backward residuals $g_k(t)$ (see Fig. 3).

Time Update Recursions for the d-Step Lattice Filter

$$A_{d,k}(t+1)=A_{d,k}(t)-\begin{bmatrix}0\\ C_{d+k-2}(t)\end{bmatrix}\cdot\phi_{d+k-1}(t+1)^T A_{d,k}(t) \tag{5.30}$$

$$R^f_{d,k}(t+1)=R^f_{d,k}(t)+f_{d+k-1}(t)\cdot\phi_{d+k-1}(t+1)^T A_{d,k}(t). \tag{5.31}$$

$$B_{d,k}(t+1)=B_{d,k}(t)-\begin{bmatrix}C_{d+k-2}(t+1)\\ 0\end{bmatrix}\cdot\phi_{d+k-1}(t+1)^T B_{d,k}(t). \tag{5.32}$$

$$R^b_{d,k}(t+1)=R^b_{d,k}(t)+g_{d+k-1}(t+1)\cdot\phi_{d+k-1}(t+1)^T B_{d,k}(t). \tag{5.33}$$

$$S_{d,k}(t+1)=S_{d,k}(t)+g_{d+k-1}(t)\cdot\phi_{d+k-1}(t+1)^T A_{d,k}(t). \tag{5.34}$$

$$S^*_{d,k}(t+1)=S^*_{d,k}(t)+f_{d+k-1}(t+1)\cdot\phi_{d+k-1}(t+1)^T B_{d,k}(t). \tag{5.35}$$

The time update recursions for $A_{d,k}(t)$, $B_{d,k}(t)$, $S_{d,k}(t)$, $S^*_{d,k}(t)$, together with the order update recursions for $A_{d,k}(t)$ and $B_{d,k}(t)$ and the recursions of the basic one-step prediction filter constitute a complete set of recursions for the least squares computation of the d-step residuals. Notice that the recursions on $R^f_{d,k}(t)$ and $R^b_{d,k}(t)$ are only needed if the sum of squares of these residuals is desired.

VI. Adaptive Prediction

We have already shown in Sections II and III how the basic lattice filter and the d-step lattice–ladder filter can also provide one-step and d-step ahead predictors in the case of known statistics for the process [(2.12) and (3.2)].

When using a gradient type algorithm, the adaptive implementations of these predictors is straightforward. Given the data $y(\tau)$ up to time t, one can compute $f_k(\tau), b_k(\tau), K_k(\tau)$ up to time t, for $k=1$ to n. Hence, the prediction of $y(t+1)$ given $\{y(t),\cdots,y(t-k+1)\}$ will be given by

$$\hat{y}_{1,k}(t+1)=\sum_{i=1}^{k} K_i^b(t)g_{i-1}(t). \tag{6.1}$$

Similarly, the prediction of $y(t+d)$ given $\{y(t),\cdots,y(t-k+1)\}$ will be given by

$$\hat{y}_{d,k}(t+d)=\sum_{i=1}^{k} K_{d,i}^b(t)g_{i-1}(t). \tag{6.2}$$

In the case of the recursive least-squares procedure of Section V, two slightly different predictors can be defined. Indeed, it follows from the definition of $f_k(t+1)$ and from the repeated application of the basic lattice recursion with initial condition $f_o(t+1)=y(t+1)$ that

$$f_k(t+1)=y(t+1)-\sum_{i=1}^{k} a_{k,i}(t+1)y(t-i+1) \tag{6.3}$$

$$=y(t+1)-\sum_{i=1}^{k} K_i^b(t+1)g_{i-1}(t). \tag{6.4}$$

Hence,

$$\sum_{i=1}^{k} a_{k,i}(t+1)y(t-i+1)=\sum_{i=1}^{k} K_i^b(t+1)g_{i-1}(t). \tag{6.5}$$

Any one of the two expressions in (6.5) could be defined as the one-step ahead prediction of $y(t+1)$ given $\{y(t),\cdots,y(t-k+1)\}$, except that the coefficients $a_{k,i}(t+1)$ and $K_i^b(t+1)$ depend upon all observations up to time $t+1$, which would make this predictor noncausal. This suggests that a truly causal least-squares one step ahead predictor can be obtained by replacing the coefficients $a_{k,i}(t+1)$ [resp. $K_i^b(t+1)$] in (6.5) by $a_{k,i}(t)$ [resp. $K_i^b(t)$].

We shall define the one step ahead predictor as

$$\hat{y}_{1,k}(t+1)=\sum_{i=1}^{k} K_i^b(t)g_{i-1}(t). \tag{6.6}$$

It turns out that this predictor is slightly different from one that would have been defined as

$$y^*_{1,k}(t+1)=\sum_{i=1}^{k} a_{k,i}(t)y(t-i+1). \qquad (6.7)$$

If we define

$$\hat{f}_k(t+1)=A_k(t)^T\phi_k(t+1) \qquad (6.8)$$

$$\hat{g}_k(t+1)=B_k(t)^T\phi_k(t+1) \qquad (6.9)$$

then $y^*_{1,k}(t+1)$ can be expressed as

$$y^*_{1,k}(t+1)=y(t+1)-\hat{f}_k(t+1). \qquad (6.10)$$

Multiplying the order update recursion for $A_k(t)$ by $\phi_k^T(t+1)$ leads to the following recursion for $\hat{f}_k(t+1)$:

$$\hat{f}_{k-1}(t+1)=\hat{f}_k(t+1)-K^b_{k+1}(t)\hat{g}_k(t). \qquad (6.11)$$

From the repeated application of (6.11) with initial condition $\hat{f}_o(t+1)=y(t+1)$ it follows that

$$\hat{f}_{k+1}(t+1)=y(t+1)-\sum_{i=1}^{k} K^b_i(t)\hat{g}_{i-1}(t) \qquad (6.12)$$

and

$$y^*_{1,k}(t+1)=\sum_{i=1}^{k} K^b_i(t)\hat{g}_{i-1}(t) \qquad (6.13)$$

in which the $\hat{g}_{i-1}(t)$ are actually estimates of $g_{i-1}(t)$ obtained with coefficients computed at time $t-1$. [Compare (6.9) to (5.2).]

Expression (6.6) has been chosen because it uses the most recent estimates of the coefficients $K^b_i(t)$ and the backward residuals $g_{i-1}(t)$. Notice that in the stationary case expressions (6.6) and (6.7) converge to the same value as the number of observations increases, since the coefficient vectors $B_{i-1}(t)$ converge to constant values. In the nonstationary case however, or in the adaptation stages of the adaptive lattice filter, expression (6.6) is preferable.

The predictor (6.6) is a least-squares one step ahead predictor of order k. The estimate appears to be linear in $\{g_0(t),\cdots,g_{k-1}(t)\}$ or, equivalently, $\{y(t),\cdots,y(t-k+1)\}$; however, remember that the coefficients $K^b_i(t)$ are functions of the whole observation record $\{y(0),\cdots,y(t)\}$ [the same is also true for expression (6.1)].

Similarly two different d-step ahead predictors could be defined

$$\hat{y}_{d,k}(t+d)=\sum_{i=1}^{k} K^b_{d,i}(t)g_{i-1}(t) \qquad (6.14)$$

or

$$y^*_{d,k}(t+d)=\sum_{i=1}^{k} a_{d,k,i}(t)y(t-i+1). \qquad (6.15)$$

Again we choose expression (6.14) because it uses the most recent information for the computation of the coefficients.

VII. Concluding Remarks

We have constructed a d-step prediction filter in lattice and ladder form using the orthogonalizing property of the basic linear prediction lattice filter. This d-step prediction filter also has interesting orthogonality properties.

The equations for the d-step predictions and prediction errors assume known covariances. Two different adaptive implementations for the case of unknown process statistics are also given. In addition, we have shown how the d-step lattice–ladder filter can be used to generate d-step ahead predictions for the observation process.

Appendix I

Let t_0-n+1 be the initial observation time and let t_1 be the initial computation time with $t_1 \geqslant t_0+d$.

The startup of the gradient procedure of Section IV is then as follows:

$$f_0(t_1)=g_0(t_1)=y(t_1)$$

$$f_{d,0}(t_1)=y(t_1), \quad g_{d,0}(t_1)=y(t_1-d+1)$$

$$R^f_0(t_1)=\sum_{\tau=t_0+d}^{t_1}\beta^{t_1-\tau}y^2(\tau)$$

$$R^b_0(t_1-1)=\sum_{\tau=t_0}^{t_1-1}\beta^{t_1-1-\tau}y^2(\tau)$$

$$R^b_0(t_1-d)=\sum_{\tau=t_0}^{t_1-d}\beta^{t_1-d-\tau}y^2(\tau)$$

$$C_0=\sum_{\tau=t_0+1}^{t_0+d-1}\beta^{-\tau}y^2(\tau)$$

$$R_0(t_1)=R^f_0(t_1)+R^b_0(t_1-1)+\beta^{t_1}C_0$$

$$K_1(t_1)=\frac{2\sum_{\tau=t_0+1}^{t_1}\beta^{t_1-\tau}y(\tau)y(\tau-1)}{R_0(t_1)}$$

$$K^b_{d,1}(t_1)=\frac{\sum_{\tau=t_0+d}^{t_1}\beta^{t_1-\tau}y(\tau)y(\tau-d)}{R^b_0(t_1-d)}$$

$$K^f_{d,1}(t_1)=\frac{\sum_{\tau=t_0+d}^{t_1}\beta^{t_1-\tau}y(\tau-d)y(\tau)}{R^f_0(t_1)}$$

$$f_1(\tau+1)=y(\tau+1)-K_1(t_1)y(\tau), \qquad \tau=t_0,\cdots,t_1$$

$$g_1(\tau+1)=y(\tau)-K_1(t_1)y(\tau+1), \qquad \tau=t_0-1,\cdots,t_1$$

$$f_{d,1}(\tau+1)=y(\tau+1)-K^b_{d,1}(t_1)y(\tau-d+1), \qquad \tau=t_0+d-1,\cdots,t_1$$

$$g_{d,1}(\tau+1)=y(\tau-d+1)-K^f_{d,1}(t_1)y(\tau+1), \qquad \tau=t_0+d-2,\cdots,t_1.$$

Now all the quantities are available that are needed to compute $K_2(t_1)$, $K^b_{d,2}(t_1)$, $K^f_{d,2}(t_1)$ via (4.1), (4.8), and (4.12), and hence the residuals of order 2 via (4.3) and (4.16). The time update of $K_1(t_1+1)$, $K^b_{d,1}(t_1+1)$, $K^f_{d,1}(t_1+1)$ is performed via (4.4), (4.5), (4.10), (4.11), (4.14), (4.15); hence the time update residuals of order 1 can also be computed via (4.3) and (4.16).

Appendix II

Equation (5.19) is derived as follows. We have by (5.13)

$$R_{d+k}(t)A_{d,k+1}(t)=\begin{bmatrix} R^f_{d,k+1}(t) \\ x \\ \vdots \\ x \\ 0 \\ \vdots \\ 0 \end{bmatrix}\begin{matrix} 1 \\ \left.\begin{matrix} \\ \\ \\ \end{matrix}\right\} d-1 \\ \left.\begin{matrix} \\ \\ \\ \end{matrix}\right\} k+1 \end{matrix}. \tag{A2.1}$$

Using (5.15) we also have

$$R_{d+k}(t)\begin{bmatrix} A_{d,k}(t) \\ 0 \end{bmatrix}=\left[\begin{array}{c|c} R_{d+k-1}(t) & X \\ \hline X & X \end{array}\right]\begin{bmatrix} A_{d,k}(t) \\ 0 \end{bmatrix}=\begin{bmatrix} R^f_{d,k}(t) \\ 0 \\ \vdots \\ 0 \\ S_{d,k}(t) \end{bmatrix}. \tag{A2.2}$$

One can write

$$A_{d,k+1}(t)=\begin{bmatrix} A_{d,k}(t) \\ 0 \end{bmatrix}+\alpha\begin{bmatrix} 0 \\ \vdots \\ 0 \\ B_k(t-d) \end{bmatrix} \tag{A2.3}$$

and the problem is then to find the value of α. In order to satisfy (A2.1), α must be chosen such that

$$S_{d,k}(t)+\alpha\big[\text{last row of } R_{d+k}(t)\big]\begin{bmatrix} 0 \\ \vdots \\ 0 \\ B_k(t-d) \end{bmatrix}=0 \tag{A2.4}$$

but, by (5.16) and (5.12), we have

$$\big[\text{last row of } R_{d+k}(t)\big]\begin{bmatrix} 0 \\ \vdots \\ 0 \\ B_k(t-d) \end{bmatrix}=R^b_k(t-d), \tag{A2.5}$$

and hence we obtain (5.19).

Premultiplying (5.19) by $R_{d+k}(t)$ and using (A2.1) and (A2.2) yields (5.20) and premultiplying (5.19) by $\phi^T_{d+k}(t)$ yields (5.21). Equations (5.22)–(5.24) are obtained similarly. By (5.13) and (5.14)

$$R_{d+k-1}(t+1)A_{d,k}(t)=\begin{bmatrix} R^f_{d,k}(t) \\ x \\ \vdots \\ x \\ 0 \\ \vdots \\ 0 \end{bmatrix}+\phi_{d+k-1}(t+1)\cdot\phi_{d+k-1}(t+1)^T A_{d,k}(t) \tag{A2.6}$$

and by (5.15)

$$R_{d+k-1}(t+1)\begin{bmatrix} 0 \\ C_{d+k-2}(t) \end{bmatrix}=\left[\begin{array}{c|c} X & X \\ \hline X & R_{d+k-2}(t) \end{array}\right]\begin{bmatrix} 0 \\ C_{d+k-2}(t) \end{bmatrix}=\begin{bmatrix} X \\ \phi_{d+k-2}(t) \end{bmatrix}. \tag{A2.7}$$

Using (5.13), (A2.6), and (A2.7) and remembering that $A_{d,k}(t+1)$ is defined by the last k equations of (5.13), one gets (5.30). Premultiplying (5.30) by $R_{d+k-1}(t+1)$ and keeping only the first equation leads to

$$R^f_{d,k}(t+1)=\big[\text{first row of } R_{d+k-1}\big]A_{d,k}(t)-\big[\text{first row of } R_{d+k-1}(t+1)\big]\begin{bmatrix} 0 \\ C_{d+k-2}(t) \end{bmatrix}\cdot\phi_{d+k-1}(t+1)^T A_{d,k}(t). \tag{A2.8}$$

The first term on the right-hand side of (A2.8) yields, using (5.14) and (A2.6),

$$R^f_{d,k}(t)+y(t+1)\phi_{d+k-1}(t+1)^T A_{d,k}(t). \tag{A2.9}$$

The second term yields, using (5.12), (A2.1), and the time update recursion for $C_k(t)$

$$\big[-y(t+1)+f_{d+k-1}(t+1)\big]\phi^T_{d+k-1}(t+1)A_{d,k}(t). \tag{A2.10}$$

Combining these two terms we get (5.31). Equations (5.32) and (5.33) are obtained similarly. Finally, we derive the time update recursion (5.34). Using the definition of $S_{d,k}(t+1)$ and (5.30) we have

$$S_{d,k}(t+1)=\big[\text{last row of } R_{d+k}(t+1)\big]\cdot\left[\begin{bmatrix} A_{d,k}(t) \\ 0 \end{bmatrix}-\begin{bmatrix} 0 \\ C_{d+k-2}(t) \\ 0 \end{bmatrix}\phi_{d+k-1}(t+1)^T A_{d,k}(t)\right]. \tag{A2.11}$$

The first term on the right-hand side yields, using (5.14) and (A2.1),

$$S_{d,k}(t)+y_{t-d-k+1}\phi_{d+k-1}(t+1)^T A_{d,k}(t). \quad (A2.12)$$

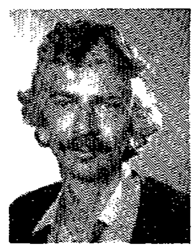
The second term yields, using (5.15), (A2.1), and the order recursion for $C_k(t)$

$$-y_{t-d-k+1}\phi_{d+k-1}(t+1)^T A_{d,k}(t) + g_{d+k-1}(t)\phi_{d+k-1}(t+1)^T A_{d,k}(t). \quad (A2.13)$$

Combining both terms yields (5.34). Again, (5.35) is obtained in a similar way.

References

[1] F. Itakura and S. Saito, "Digital filtering techniques for speech analysis and synthesis," in *Proc. 7th Int. Congress Acoust.*, Budapest, Hungary, 1971, paper 25 C-1.
[2] N. Levinson, "The Wiener RMS error criterion in filter design and prediction," *J. Math. Phys.*, vol. 25, pp. 261–278, 1947.
[3] A. Lindquist, "A new algorithm for optimal filtering of discrete time stationary processes," *SIAM J. Contr.*, vol. 12, pp. 736–746, Nov. 1974.
[4] A. H. Gray and J. Markel, "Digital lattice and ladder filter synthesis," *IEEE Trans. Audio Electroacoust.*, vol. AU-21, pp. 491–500, Dec. 1973.
[5] L. J. Griffiths, "A continuously-adaptive filter implemented as a lattice structure," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Hartford, CT, May 1977, pp. 683–686.
[6] L. J. Griffiths and R. S. Medaugh, "Convergence properties of an adaptive noise cancelling lattice structure," in *Proc. IEEE Conf. Decision Contr.*, San Diego, CA, Jan. 1979, pp. 1357–1361.
[7] B. W. Dickinson and J. Turner, "Reflection coefficient estimation using Cholesky decomposition," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-27, pp. 146–149, Apr. 1979.
[8] T. E. Carter, "Study of an adaptive structure for linear prediction analysis of speech," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, 1978, pp. 27–30.
[9] J. Makhoul, "A class of all-zero lattice digital filters: Properties and applications," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-26, pp. 304–314, Aug. 1978.
[10] J. Makhoul and R. Viswanathan, "Adaptive lattice methods for linear prediction," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, 1978, pp. 83–86.
[11] M. D. Srinath and M. M. Viswanathan, "Sequential algorithms for identification of parameters of an autoregressive process," *IEEE Trans. Automat. Contr.*, vol. AC-20, pp. 542–546, Aug. 1975.
[12] M. Morf, D. T. Lee, J. R. Nickolls, and A. Vieira, "A classification of algorithms for ARMA models and ladder realizations," in *Proc. 1977 IEEE Int. Conf. Acoust., Speech, Signal Processing*, Hartford, CT, 1977, pp. 13–19.
[13] M. Morf, A. Vieira, and D. T. Lee, "Ladder forms for identification and speech processing," in *Proc. IEEE Conf. Decision Contr.*, 1977, pp. 1074–1078.
[14] M. Morf and D. T. Lee, "Recursive least-squares ladder forms for fast parameter tracking," in *Proc. IEEE Conf. Decision Contr.*, 1978, pp. 1362–1367.
[15] D. T. Lee and M. Morf, "Recursive square-root ladder estimation algorithms," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Denver, CO, Apr. 1980, pp. 1005–1016.
[16] D. T. Lee, M. Morf, and B. Friedlander, "Recursive least squares ladder estimation algorithms," *IEEE Trans. Circuits Syst.*, vol. CAS-28, pp. 467–481, June 1981.
[17] M. J. Shensa, "Recursive least-squares lattice algorithms—A geometrical approach," *IEEE Trans. Automat. Contr.*, vol. AC-26, pp. 695–702, June 1981.
[18] G. C. Goodwin and K. S. Sin, *Adaptive Filtering, Prediction and Control*. Englewood Cliffs, NJ: Prentice-Hall, 1983.
[19] M. Gevers and V. Wertz, "A recursive least-squares *D*-step ahead predictor in ladder form," Int. Rep., Université Catholique de Louvain, Louvain-la-Neuve, Belgium, Dec. 1980.
[20] V. U. Reddy, B. Egardt, and T. Kailath, "Optimized lattice-form adaptive line enhancer for a sinusoidal signal in broad-band noise," *IEEE Trans. Circuits Syst.*, vol. CAS-28, pp. 542–550, June 1981.
[21] L. Ljung, M. Morf, and D. Falconer, "Fast calculation of gain matrices for recursive estimation schemes," *Int. J. Contr.*, vol. 27, pp. 1–19, 1978.



Michel R. Gevers (S'66–S'70–M'72) was born in Antwerp, Belgium, in 1945. He received the electrical engineering degree from Louvain University, Louvain-la-Neuve, Belgium, in 1968, and the Ph.D. degree from Stanford University, Stanford, CA, in 1972.

He went to Stanford University, where he was supported by a Harkness fellowship and then an ESRO/NASA fellowship, in 1969, following a one-year research assistantship in the Solid State Laboratory, Louvain University. Since 1972 he has been Assistant Professor and, subsequently, Professor at the Laboratoire d'Automatique et d'Analyse des Systèmes at Louvain University. He was head of the laboratory from 1976 to 1980. In 1980 he was on sabbatical leave at the University of Newcastle, N.S.W., Australia. His main research interests are in estimation, identification, stochastic processes, and multivariable system theory. He has done applied work on hydrological, biomedical, and industrial problems. He is the author or coauthor of about 40 papers and conference papers.

Vincent J. Wertz was born in Liège, Belgium, in 1955. He received the engineering degree in applied mathematics and the Ph.D. degree from Louvain University, Louvain-la-Neuve, Belgium, in 1978 and 1982, respectively.

Since September 1978, he has been a Research Assistant at the Laboratoire d'Automatique et d'Analyse des Systèmes, Louvain University. From July 1980 to January 1981, he was on leave at the Department of Electrical Engineering, University of Newcastle, N.S.W., Australia, where he was sponsored by a fellowship of IRSIA (Institut pour la Recherche Scientifique dans l'Industrie et l'Agriculture). From February 1981 to July 1981, he was an Assistant at the Centre Universitaire de Stif, Algeria. His main interests are in multivariable systems and in system identification. His Ph.D. dissertation will deal with parameterization and identifiability of multivariable systems.