Multichannel ARMA Modeling by Least Squares Circular Lattice Filtering

Mrityunjoy Chakraborty and Surendra Prasad, *Member, IEEE*

***Abstract*— This paper makes an attempt to develop least squares lattice algorithms for the ARMA modeling of a linear, slowly time-varying, multichannel system employing scalar computations only. Using an equivalent scalar, periodic ARMA model and a *circular* delay operator, the signal set for each channel is defined in terms of *circularly* delayed input and output vectors corresponding to that channel. The orthogonal projection of each current output vector on the subspace spanned by the corresponding signal set is then computed in a manner that allows independent AR and MA order recursions. The resulting lattice algorithm can be implemented in a parallel architecture employing one processor per channel with the data flowing amongst them in a circular manner. The evaluation of the ARMA parameters from the lattice coefficients follows the usual step-up algorithmic approach but requires, in addition, the circulation of certain variables across the processors since the signal sets become linearly dependent beyond certain stages. The proposed algorithm can also be used to estimate a process from two correlated, multichannel processes adaptively allowing the filter orders for both the processes to be chosen independently of each other. This feature is further exploited for ARMA modeling a given multichannel time series with unknown, white input.**

I. Introduction

LEAST squares lattice (LSL) algorithms have attracted a great deal of interest in recent years in the fields of adaptive filtering, estimation, control, and system identification. The advantages of these algorithms, like higher convergence rates, order recursive structures, and better numerical properties, have been well recognized. Most of these algorithms can be extended to the multichannel case directly by replacing the scalar variables of the single channel LSL algorithm by appropriate matrix and vector quantities. The resulting algorithms, however, turn out to be too slow to be of much practical interest because they need to depend heavily on computationally expensive matrix operations like multiplication, inversion, Cholesky factorization, etc. This problem has been studied by various authors over the past few years, and methods have been suggested [1]–[4] to overcome this by scalarizing the multichannel LSL algorithms in an appropriate manner.

In this paper, we have proposed a scalar LSL algorithm for identifying a slowly time-varying, linear, d-channel system, which is either known *a priori* to be, or, being modeled as a d-variate ARMA system, which is given by

$$\boldsymbol{x}(k)+\sum_{i=1}^{p}\boldsymbol{A}(i)\boldsymbol{x}(k-i)=\boldsymbol{w}(k)+\sum_{j=1}^{q}\boldsymbol{B}(j)\boldsymbol{w}(k-j) \quad (1)$$

where $\boldsymbol{A}(i)$'s and $\boldsymbol{B}(j)$'s are, respectively, the $d\times d$ AR and MA coefficients, $\boldsymbol{x}(k)$ and $\boldsymbol{w}(k)$ are, respectively, the d-variate output and input processes and p and q are the AR and the MA orders, respectively. The parameters $\boldsymbol{A}(i)$ and $\boldsymbol{B}(j)$ are assumed to be varying with time slowly, and it is required to estimate and track them adaptively by an appropriate least squares minimization procedure. The classical approach [5], [6] to this problem assumes the AR and the MA orders to be identical and replaces (1) by a $2d$-channel AR model of the joint output/input process $[\boldsymbol{x}^t(k),\boldsymbol{w}^t(k)]^t$, which can be identified by using the scalarized, multichannel LSL algorithms [1]–[4]. Because only those models having identical AR and MA orders are generated, this approach requires the introduction of a total of $|p-q|$ additional AR or MA terms with zero coefficients in (1), depending on whether $p>q$ or otherwise. These additional terms, however, lead to an unnecessary increase in the computational requirements. A more elegant approach to this problem, in the single channel case, has been suggested recently by Karlsson and Hayes [7], where they have identified the prediction errors that provide a complete characterization of an ARMA process in the same manner as the forward and the backward prediction errors characterize an AR process. These prediction errors can be order updated in a manner that results in separate AR and MA order update blocks. The cascading of these blocks gives rise to an ARMA lattice filter that evaluates ARMA models of all possible orders up to any chosen maximum order. (A similar work using algebraic formulation has been presented in [8]). In a direct d-channel extension of this approach, the prediction errors are, however, given by $d\times 1$ vectors, and the PARCOR coefficients and the residual energy variables are obtained as $d\times d$ matrices. Updating of these quantities then requires multiplication and inversion of a set of $d\times d$ matrices in each stage of the lattice and for each input sample, which gives rise to an enormous increase in the overall computational cost. In this paper, we propose a computationally efficient multichannel extension of the Karlsson-Hayes approach, which involves purely scalar operations whose computations can be mapped on a pipelined d-processor architecture.

The derivation is based on an alternative realization of $\boldsymbol{x}(k)$ in terms of an equivalent scalar, periodic ARMA model

Manuscript received February 12, 1992; revised December 10, 1993. The associate editor coordinating the review of this paper and approving it for publication was Prof. Andrew Yagle.

M. Chakraborty is with the Department of Electrical Engineering, Indian Institute of Technology, New Delhi, India.

S. Prasad is with the Department of Electronics and Electrical Communication Engineering, Indian Institute of Technology, Kharagpur, India.

IEEE Log Number 9403264.

1053-587X/94$04.00 © 1994 IEEE

developed in [9], which we briefly describe in Section II. A LSL algorithm is then developed in Section III to identify this equivalent scalar model that permits independent order recursions in both the AR and the MA orders. The resulting lattice filter, which is introduced as the "least squares circular ARMA lattice (LSCAL) filter," is made up of a cascade of certain basic building blocks, each of which exhibits identical lattice sections for all the channels, pipelined in a circular manner. Two applications of the proposed LSCAL algorithm are considered next in Section IV—one in the context of estimation of a process $z(k)$ from two correlated multichannel processes $\boldsymbol{x}(k)$ and $\boldsymbol{u}(k)$ and the other in the context of ARMA modeling of a multichannel time series with unknown white input. Notations similar to those used by Karlsson and Hayes [7] have been followed throughout this presentation for the convenience of readers who may be familiar with their work.

II. Review of the Scalar Representation of a Vector ARMA Process

Assume that $\boldsymbol{w}(k)$ has a positive definite covariance matrix $\boldsymbol{Q}$, i.e., $E[\boldsymbol{w}(k)\boldsymbol{w}^H(k)] = \boldsymbol{Q}$ (the superscript 'H' denotes Hermitian transposition). Then, the Cholesky factorization of $\boldsymbol{Q}$ gives $\boldsymbol{Q} = \boldsymbol{LDL}^H$, where $\boldsymbol{L}$ is a unit lower triangular matrix, and $\boldsymbol{D}$ is a diagonal matrix with real, positive diagonal entries. Introducing a new input vector $\boldsymbol{u}(k)$ in (1), where $\boldsymbol{u}(k) = \boldsymbol{L}^{-1}\boldsymbol{w}(k)$, we obtain

$$\boldsymbol{L}^{-1}\boldsymbol{x}(k) + \sum_{i=1}^{p}\boldsymbol{L}^{-1}\boldsymbol{A}(i)\boldsymbol{x}(k-i) = \boldsymbol{u}(k) + \sum_{j=1}^{q}\boldsymbol{L}^{-1}\boldsymbol{B}(j)\boldsymbol{L}\boldsymbol{u}(k-j). \tag{2}$$

Next, consider two scalar processes $y(n)$ and $v(n)$ related to $\boldsymbol{x}(k)$ and $\boldsymbol{w}(k)$ by $y(r+(k-1)d) = x_r(k)$, $v(r+(k-1)d) = u_r(k)$, $r = 1, 2, \ldots, d$. Replacing the components of $\boldsymbol{x}(k)$ and $\boldsymbol{u}(k)$ in (2) by the corresponding elements of $y(n)$ and $v(n)$, respectively, and equating the rows of the L.H.S. and the R.H.S. of the matrix equation (2) separately, we find that $y(n)$ corresponds to a periodically stationary ARMA process, which is given by

$$y(n) + \sum_{i=1}^{p_n} a_n(i)y(n-i) = v(n) + \sum_{j=1}^{q_n} b_n(j)v(n-j) \tag{3}$$

where $a_n(i) = a_{n+ld}(i), b_n(j) = b_{n+ld}(j), p_n = p_{n+ld}, q_n = q_{n+ld}$ for any integer l, and $y(n)$ and $v(n)$ are two periodically stationary processes, i.e., $E[y(n+m)y^*(n)] = E[y(n+ld+m)y^*(n+ld)]$ and $E[v(n+m)v^*(n)] = E[v(n+ld+m)v^*(n+ld)]$ (the superscript '*' denotes complex conjugation). The parameters $a_n(i)$'s and $b_n(j)$'s in (3) are given by the elements of the matrices $\boldsymbol{L}^{-1}, \boldsymbol{L}^{-1}\boldsymbol{A}(I)$'s and $\boldsymbol{L}^{-1}\boldsymbol{B}(J)\boldsymbol{L}$ s in (2), $I = 1, 2, \ldots, p$, $J = 1, 2, \ldots, q$, and the index 'n' is related to the index 'k' as $n = r + (k-1)d, r = 1, 2, \ldots, d$. If $\boldsymbol{w}(k)$ is given to be white, $v(n)$ becomes a periodically stationary white process. In fact, it has been shown in [9] that if (1) is the innovation representation of the multivariate ARMA process, then $v(n)$ turns out to be the innovation (i.e., infinite order prediction error) sequence corresponding to $y(n)$. The relationship between the two models (1) and (3) is one-to-one, i.e., given (3), one can always get back the multichannel model (1) by a reverse procedure.

III. Development of the LSCAL Algorithm

A. Basic LSCAL Recursions

As stated earlier, our objective is to develop a LSL algorithm for identifying the equivalent scalar, periodic ARMA model (3). Using the periodicity of (3), this can be formulated as d simultaneous least squares estimation problems. For convenience, a real-valued, prewindowed data set is considered, i.e., the data available at index k is given by $\{\boldsymbol{x}(l), \boldsymbol{u}(l) \mid 1 \le l \le k\}$, and it is assumed that $\boldsymbol{x}(l) = \boldsymbol{u}(l) = 0, l \le 0$. The algorithm is then developed using orthogonal projection operations in the Hilbert space H_k of all real $k \times 1$ vectors with the following inner product: $\langle \boldsymbol{x}, \boldsymbol{y}\rangle = \frac{1-\lambda}{1-\lambda^k}\sum_{i=1}^{k}\lambda^{k-i}\boldsymbol{x}(i)\boldsymbol{y}(i), \boldsymbol{x}, \boldsymbol{y} \in H_k, 0 < \lambda < 1$ ('λ' is the usual forgetting factor). The two notations $\|\boldsymbol{a}\|$ and $\langle \boldsymbol{y}\rangle$ are used in this paper to denote the norm of a vector $\boldsymbol{a}$ and the subspace spanned by the *single* element $\boldsymbol{y}$, respectively.

Let us define the present output vector $\boldsymbol{x}_r(k)$ and the input vector $\boldsymbol{u}_r(k)$ for the rth channel $r = 1, 2, \ldots, d$ as follows:

$$\boldsymbol{x}_r(k) = [x_r(1), x_r(2), \ldots, x_r(k)]^t,\ r = 1, 2, \ldots, d,$$
$$\boldsymbol{u}_r(k) = [u_r(1), u_r(2), \ldots, u_r(k)]^t,\ r = 1, 2, \ldots, d.$$

In addition, define the circular delay operator D^{-1} [1], which takes a vector $\boldsymbol{g}_r(k)$, where $\boldsymbol{g}_r(k)$ is either $\boldsymbol{x}_r(k)$ or $\boldsymbol{u}_r(k), r = 1, 2, \ldots, d$ and produces a circularly shifted version of that in the following manner:

$$D^{-1}\boldsymbol{g}_1(k) = [0, g_d(1), g_d(2), \ldots, g_d(k-1)]^t \tag{4a}$$
$$D^{-1}\boldsymbol{g}_r(k) = \boldsymbol{g}_{r-1}(k),\ r = 2, 3, \ldots, d \tag{4b}$$

and

$$D^{-1}\begin{bmatrix} 0 \\ \boldsymbol{g}_r(k)\end{bmatrix} = \begin{bmatrix} 0 \\ D^{-1}\boldsymbol{g}_r(k)\end{bmatrix} \tag{4c}$$

Using these definitions, the data matrix $\boldsymbol{D}_{k,i,j,r},\ i, j = 1, 2, \ldots,\ r = 1, 2, \ldots, d$ is formed as follows:

$$\boldsymbol{D}_{k,i,j,r} = [\boldsymbol{U}_{k,j,r} \mid \boldsymbol{X}_{k,i,r}] \tag{5}$$

where the submatrices $\boldsymbol{X}_{k,i,r}$ and $\boldsymbol{U}_{k,j,r}$ are given by

$$\boldsymbol{X}_{k,i,r} = [D^{-1}\boldsymbol{x}_r(k), D^{-2}\boldsymbol{x}_r(k), \ldots, D^{-i}\boldsymbol{x}_r(k)], \tag{6}$$
$$\boldsymbol{U}_{k,i,r} = [\boldsymbol{u}_r(k), D^{-1}\boldsymbol{u}_r(k), \ldots, D^{-(j-1)}\boldsymbol{u}_r(k)]. \tag{7}$$

(The two indices i and j indicate the number of output and input columns, respectively). The rth channel signal set $S_{k,i,j,r},\ r = 1, 2, \ldots, d$, at time index k and for the (i, j)th order is next defined as the collection of the columns of $\boldsymbol{D}_{k,i,j,r}$. Let $\delta_{k,i,j,r}$ be the subspace of H_k spanned by $S_{k,i,j,r}$. The orthogonal projection of $\boldsymbol{x}_r(k)$ on $\delta_{k,i,j,r}$, say $\hat{\boldsymbol{x}}_{k,i,j,r}(k)$, will then be given by a linear combination of the elements of $S_{k,i,j,r}$, $r = 1, 2, \ldots, d$ in the following transversal filter form:

$$\hat{\boldsymbol{x}}_{k,i,j,r}(k) = -\sum_{m=1}^{i} A^{f}_{k,i,j,r}(m) D^{-m} \boldsymbol{x}_r(k) + \sum_{l=0}^{j-1} B^{f}_{k,i,j,r}(l) D^{-l} \boldsymbol{u}_r(k). \tag{8}$$

From the derivation of the scalar, periodic ARMA model (3) discussed in Section II, it is easy to see that $A^{f}_{k,i,j,r}(m)$'s and $B^{f}_{k,i,j,r}(l)$'s provide the linear least squares estimates of $a_n(m)$'s and $b_n(l)$'s, respectively, for $n = r + (k-1)d, p_n = i$ and $q_n = j - 1$ in (3), based on data up to the present index k (The superscript 'f' indicates that $A^{f}_{k,i,j,r}(m)$'s and $B^{f}_{k,i,j,r}(l)$'s are associated with a forward projection operation). In the least squares modeling problem, we are interested in obtaining this orthogonal projection in an order recursive manner. More specifically, we want to evaluate the estimates $A^{f}_{k,i,j,r}(m)$ and $B^{f}_{k,i,j,r}(l)$ at each index k for all i, j, $1 \le i \le M_r, 1 \le j \le N_r$, where (M_r, N_r) is any specified maximum order for the rth channel or equivalently for the nth index in (3), where $n = r + (k-1)d$. To meet this objective, we first derive a lattice filter by introducing a set of prediction errors that provide a complete characterization of the scalar, periodic ARMA process $y(n)$. For this purpose, we apply the approach of Karlsson and Hayes [7] to (3).

Let $\boldsymbol{P}_{k,i,j,r}$ and $\boldsymbol{P}^{\perp}_{k,i,j,r}$ denote, respectively, the orthogonal projection and projection error operators associated with $\delta_{k,i,j,r}$, i.e., $\boldsymbol{P}^{\perp}_{k,i,j,r} = \boldsymbol{I} - \boldsymbol{P}_{k,i,j,r}$. As the signal set for each channel consists of both input and output vectors, we will be dealing with two kinds of projection errors—one associated with the input and the other with the output vectors These are termed as the input projection errors and the output projection errors, respectively. We then define the (i,j)th-order *output forward projection error* for the rth channel $r = 1, 2, \ldots, d$ as

$$\boldsymbol{e}_{k,i,j,r} = \boldsymbol{P}^{\perp}_{k,i,j,r} \boldsymbol{x}_r(k). \tag{9}$$

Similarly, the (i,j)th-order *output backward projection error* $\boldsymbol{b}_{k,i,j,r}$ and the *input backward projection error* $\boldsymbol{c}_{k,i,j,r}, r = 1, 2, \ldots, d$ can be defined as

$$\boldsymbol{b}_{k,i,j,r} = \boldsymbol{P}^{\perp}_{k,i,j,r} D^{-(i+1)} \boldsymbol{x}_r(k), \tag{10}$$

$$\boldsymbol{c}_{k,i,j,r} = \boldsymbol{P}^{\perp}_{k,i,j,r} D^{-j} \boldsymbol{u}_r(k). \tag{11}$$

It is easy to see that the order recursive computation of $\boldsymbol{b}_{k,i,j,r}$ and $\boldsymbol{c}_{k,i,j,r}$ results in Gram–Schmidt orthogonalization of the columns of $\boldsymbol{D}_{k,i,j,r}$ with $\boldsymbol{b}_{k,i,j,r}$ extending the orthogonalization from $\boldsymbol{D}_{k,i,j,r}$ to $\boldsymbol{D}_{k,i+1,j,r}$ and $\boldsymbol{c}_{k,i,j,r}$ from $\boldsymbol{D}_{k,i,j,r}$ to $\boldsymbol{D}_{k,i,j+1,r}$ by generating new orthogonal components.

Next, we consider the reduced subspace $\delta^{-}_{k,i,j-1,r}$ spanned by the set $\{D^{-1}\boldsymbol{x}_r(k), \ldots, D^{-i}\boldsymbol{x}_r(k), D^{-1}\boldsymbol{u}_r(k), \ldots, D^{-(j-1)}\boldsymbol{u}_r(k)\}$, (the superscript "-" on $\delta^{-}_{k,i,j-1,r}$ indicates that the current input $\boldsymbol{u}_r(k)$ for the rth channel has not been included). Let $\boldsymbol{P}^{-}_{k,i,j-1,r}$ and $\boldsymbol{P}^{-\perp}_{k,i,j-1,r}$, respectively, denote the orthogonal projection and projection error operators associated with $\delta^{-}_{k,i,j-1,r}$ i.e., $\boldsymbol{P}^{-\perp}_{k,i,j-1,r} = \boldsymbol{I} - \boldsymbol{P}^{-}_{k,i,j-1,r}$. We can then define three more projection errors corresponding to $\delta^{-}_{k,i,j-1,r}$ on analogous lines. These are as follows:

$$\boldsymbol{f}^{-}_{k,i,j-1,r} = \boldsymbol{P}^{-\perp}_{k,i,j-1,r} \boldsymbol{u}_r(k), \tag{12}$$

$$\boldsymbol{b}^{-}_{k,i,j-1,r} = \boldsymbol{P}^{-\perp}_{k,i,j-1,r} D^{-(i+1)} \boldsymbol{x}_r(k), \tag{13}$$

$$\boldsymbol{c}^{-}_{k,i,j-1,r} = \boldsymbol{P}^{-\perp}_{k,i,j-1,r} D^{-j} \boldsymbol{u}_r(k). \tag{14}$$

The six error vectors $\boldsymbol{e}_{k,i,j,r}, \boldsymbol{b}_{k,i,j,r}, \boldsymbol{c}_{k,i,j,r}, \boldsymbol{f}^{-}_{k,i,j-1,r}$, $\boldsymbol{b}^{-}_{k,i,j-1,r}$ and $\boldsymbol{c}^{-}_{k,i,j-1,r}$ for each channel form a complete set in the sense that their order updating can be accomplished by using the definitions (9)–(14) only, and no additional projection error needs to be defined. The order update relations are derived by using the following result [10] for the order updating of an orthogonal projection: Suppose we have a set of vectors $\boldsymbol{U} = \{\boldsymbol{u}_i \mid i = 1, 2, \ldots, p\}$, and $\{\boldsymbol{U}\}$ represents the subspace spanned by $\boldsymbol{U}$, with $\boldsymbol{P}_U$ and $\boldsymbol{P}^{-\perp}_U$ being, respectively, the associated orthogonal projection and projection error operators. If an additional vector $\boldsymbol{v}$ is appended to $\boldsymbol{U}$, then the subspace $\{\boldsymbol{U}, \boldsymbol{v}\}$ can be decomposed as $\{\boldsymbol{U}, \boldsymbol{v}\} = \{\boldsymbol{U}\} \oplus \langle \boldsymbol{P}^{\perp}_U \boldsymbol{v} \rangle$. For any vector $\boldsymbol{z}$, we then have $\boldsymbol{P}_{U,v}\boldsymbol{z} = \boldsymbol{P}_U \boldsymbol{z} + \frac{\langle \boldsymbol{P}^{\perp}_U \boldsymbol{v}, \boldsymbol{z} \rangle}{\|\boldsymbol{P}^{\perp}_U \boldsymbol{v}\|^2} \boldsymbol{P}^{\perp}_U \boldsymbol{v}$. Noting that $\boldsymbol{z} = \boldsymbol{P}_U \boldsymbol{z} + \boldsymbol{P}^{\perp}_U \boldsymbol{z}, \langle \boldsymbol{P}^{\perp}_U \boldsymbol{v}, \boldsymbol{P}_U \boldsymbol{z} \rangle = 0$ and $\boldsymbol{P}^{\perp}_{U,v} = \boldsymbol{I} = \boldsymbol{P}_{U,v}$, the projection error $\boldsymbol{P}^{\perp}_{U,v}\boldsymbol{z}$ can be obtained as

$$\boldsymbol{P}^{\perp}_{U,v}\boldsymbol{z} = \boldsymbol{P}^{\perp}_U \boldsymbol{z} - \frac{\langle \boldsymbol{P}^{\perp}_U \boldsymbol{v}, \boldsymbol{P}^{\perp}_U \boldsymbol{z} \rangle}{\|\boldsymbol{P}^{\perp}_U \boldsymbol{v}\|^2} \boldsymbol{P}^{\perp}_U \boldsymbol{v}. \tag{15}$$

The order update relations for the projection errors, which are defined in (9)–(14), can be written down using the general form (15) and are listed in Table I, where for each projection error, the corresponding expressions for $\{\boldsymbol{U}\}$, $\boldsymbol{P}^{\perp}_U \boldsymbol{v}$, and $\boldsymbol{z}$ are also shown. These relations involve a set of multiplier coefficients (which are popularly called "reflection coefficients") given by $\alpha_{k,r}(i,j)$, $\beta_{k,r}(i,j)$, $\gamma_{k,r}(i,j)$, $\mu_{k,r}(i,j)$, $\alpha'_{k,r}(i,j)$, $\beta'_{k,r}(i,j)$ $\gamma'_{k,r}(i,j)$, $\mu'_{k,r}(i,j)$. Each of these reflection coefficients corresponds to the multiplier term $\frac{\langle P^{\perp}_U v, P^{\perp}_U z \rangle}{\|P^{\perp}_U v\|^2}$ in (15) and can be evaluated by substituting $\boldsymbol{P}^{\perp}_U \boldsymbol{v}$ and $\boldsymbol{P}^{\perp}_U \boldsymbol{z}$ by the corresponding expressions given in Table I. Since only the current components of the projection error vectors are to be computed by the lattice filter, the order update relations are given not in the vectorized form (17) but in a scalar form by taking the kth entries of the L.H.S. and R.H.S. of (15). Note that the two prediction errors $e_{k,i,j,r}(k)$ and $f^{-}_{k,i,j,r}(k)$ can be updated separately in the AR and the MA orders (i.e., in the indices i and j, respectively).

B. LSCAL Building Blocks

The order update relations derived above can be used to construct the basic building blocks of the LSCAL filter. We begin by considering the prediction error $e_{k,i,j,r}(k)$. Ideally, $e_{k,i,j,r}(k) = 0$ for $i = p_n$, and $j = q_n + 1$, (p_n, q_n) being the true order of the equivalent scalar, periodic model (3) at index n, where $n = r + (k-1)d$. This implies that in order to identify the correct order of the system followed by the evaluation of the system parameters for that order, it is first required to order update $e_{k,i,j,r}(k)$ in both the indices i and j. Assume that

TABLE I
ORDER UPDATE EQUATIONS FOR THE PREDICTION ERRORS. ONLY THE kTH COMPONENTS ARE CONSIDERED. THE INDEX r RANGES FROM 1 TO d.

Order update relation	$\{U\}$	$P_U^{\perp} v$	z
$b_{k,i,j,r}(k) = b^{-}_{k,i,j-1,r}(k) - \alpha_{k,r}(i,j-1) f^{-}_{k,i,j-1,r}(k)$	$\delta^{-}_{k,i,j-1,r}$	$f^{-}_{k,i,j-1,r}$	$D^{-(i+1)} x_r(k)$
$c_{k,i,j,r}(k) = c^{-}_{k,i,j-1,r}(k) - \beta_{k,r}(i,j-1) f^{-}_{k,i,j-1,r}(k)$	$\delta^{-}_{k,i,j-1,r}$	$f^{-}_{k,i,j-1,r}$	$D^{-j} u_r(k)$
For $r = 2, 3, \ldots, d$, $b^{-}_{k,i,j-1,r}(k) = b_{k,i-1,j-1,r-1}(k) - \gamma_{k,r-1}(i-1,j-1)\, e_{k,i-1,j-1,r-1}(k)$	$\delta_{k,i-1,j-1,r-1}$	$e_{k,i-1,j-1,r-1}$	$D^{-(i+1)} x_r(k)$
For $r = 1$, $b^{-}_{k,i,j-1,1}(k) = b_{k-1,i-1,j-1,d}(k-1) - \gamma_{k-1,d}(i-1,j-1)\, e_{k-1,i-1,j-1,d}(k-1)$	$\delta_{k-1,i-1,j-1,d}$	$e_{k-1,i-1,j-1,d}$	$D^{-(i+1)} x_1(k)$
For $r = 2, 3, \ldots, d$, $c^{-}_{k,i,j-1,r}(k) = c_{k,i-1,j-1,r-1}(k) - \mu_{k,r-1}(i-1,j-1)\, e_{k,i-1,j-1,r-1}(k)$	$\delta_{k,i-1,j-1,r-1}$	$e_{k,i-1,j-1,r-1}$	$D^{-j} u_r(k)$
For $r = 1$, $c^{-}_{k,i,j-1,1}(k) = c_{k-1,i-1,j-1,d}(k-1) - \mu_{k-1,d}(i-1,j-1)\, e_{k-1,i-1,j-1,d}(k-1)$	$\delta_{k-1,i-1,j-1,d}$	$e_{k-1,i-1,j-1,d}$	$D^{-j} u_1(k)$
$e_{k,i,j,r}(k) = e_{k,i-1,j,r}(k) - \gamma'_{k,r}(i-1,j)\, b_{k,i-1,j,r}(k)$	$\delta_{k,i-1,j,r}$	$b_{k,i-1,j,r}$	$x_r(k)$
$e_{k,i,j,r}(k) = e_{k,i,j-1,r}(k) - \mu'_{k,r}(i,j-1)\, c_{k,i,j-1,r}(k)$	$\delta_{k,i,j-1,r}$	$c_{k,i,j-1,r}$	$x_r(k)$
$f^{-}_{k,i,j,r}(k) = f^{-}_{k,i-1,j,r}(k) - \alpha'_{k,r}(i-1,j)\, b^{-}_{k,i-1,j,r}(k)$	$\delta^{-}_{k,i-1,j,r}$	$b^{-}_{k,i-1,j,r}$	$u_r(k)$
$f^{-}_{k,i,j,r}(k) = f^{-}_{k,i,j-1,r}(k) - \beta'_{k,r}(i,j-1)\, c^{-}_{k,i,j-1,r}(k)$	$\delta^{-}_{k,i,j-1,r}$	$c^{-}_{k,i,j-1,r}$	$u_r(k)$

for $r = 1, 2, \ldots, d, e_{k,i,j,r}(k)$ has been computed recursively up to the $\{(i-1), (j-1)\}$th order, i.e., $e_{k,i-1,j-1,r}(k)$ is available for all the channels, and it is required to obtain the same for the $\{i, j\}$th order. This can be done in two ways, namely, a) generate $e_{k,i,j-1,r}(k)$ first followed by $e_{k,i,j,r}(k)$, i.e., AR updating first followed by MA updating, and b) generate $e_{k,i-1,j,r}(k)$ first followed by $e_{k,i,j,r}(k)$, i.e., MA updating first followed by AR updating. For a two-channel model, methods a) and b) are illustrated in Fig. 1(a) and (b), respectively, where order updating of the other input signals, i.e., $c^{-}_{k,i,j-2,r}(k), b_{k,i-1,j-1,r}(k), f^{-}_{k,i,j-2,r}(k)$ in the case of Fig. 1(a) and $b^{-}_{k,i-1,j-1,r}(k), c_{k,i-1,j-1,r}(k), f^{-}_{k,i-1,j-1,r}(k)$ in the case of Fig. 1(b) are also carried out. This makes both Fig. 1(a) and (b) compatible for cascading.

Both the AR and the MA update blocks are structurally identical except for the multiplier coefficients and the signals appearing at various nodes. Each block has identical lattice sections for all the channels, implying that the lattice recursions for various channels can be carried out in parallel by employing one processor for each channel. The processors are pipelined *circularly* in the following manner: For $r = 2, 3, \ldots, d, b^{-}_{k,i,j-1,r}(k)$ and $c^{-}_{k,i,j-1,r}(k)$ are computed by the $(r-1)$th processor (i.e., processor employed for the $(r-1)$th channel) and transmitted to the rth processor and for $r = 1, b^{-}_{k+1,i,j-1,1}(k+1)$ and $c^{-}_{k+1,i,j-1,1}(k+1)$ are computed by the dth processor and transmitted through a delay to channel 1.

Both the AR-MA configuration of Fig. 1(a) and the MA-AR configuration of Fig. 1(b) can be used for orthogonalizing the columns of $\boldsymbol{D}_{k,i,j,r}$ in the Gram–Schmidt manner. In the case of Fig. 1(a), the orthogonalization is extended from $\boldsymbol{D}_{k,i,j-1,r}$ to $\boldsymbol{D}_{k,i+1,j,r}$ via the generation of the orthogonal components $\boldsymbol{c}_{k,i,j-1,r}$ and $\boldsymbol{b}_{k,i,j,r}$. In the case of Fig. 1(b), similar extension of orthogonalization from $\boldsymbol{D}_{k,i-1,j,r}$ to $\boldsymbol{D}_{k,i,j+1,r}$ results from the generation of $\boldsymbol{b}_{k,i-1,j,r}$ and $\boldsymbol{c}_{k,i,j,r}$.

Both Fig. 1(a) and (b) assume the availability of $e_{k,i-1,j-1,r}(k)$ at their inputs and generate, respectively, the following two sets of output forward prediction errors: $\{e_{k,i,j-1,r}(k), e_{k,i,j,r}(k)\}$ and $\{e_{k,i-1,j,r}(k), e_{k,i,j,r}(k)\}$. In general, the set of output forward prediction errors generated at the various stages of the order recursive computation depends on the order in which the elements of the signal set $S_{k,i,j,r}$ are selected. Suppose that the scalar, periodic ARMA model (3) has an order (p_n, q_n) at index n, where $n = r + (k-1)d$, implying that the order recursive evaluation of $e_{k,i,j,r}(k)$ has

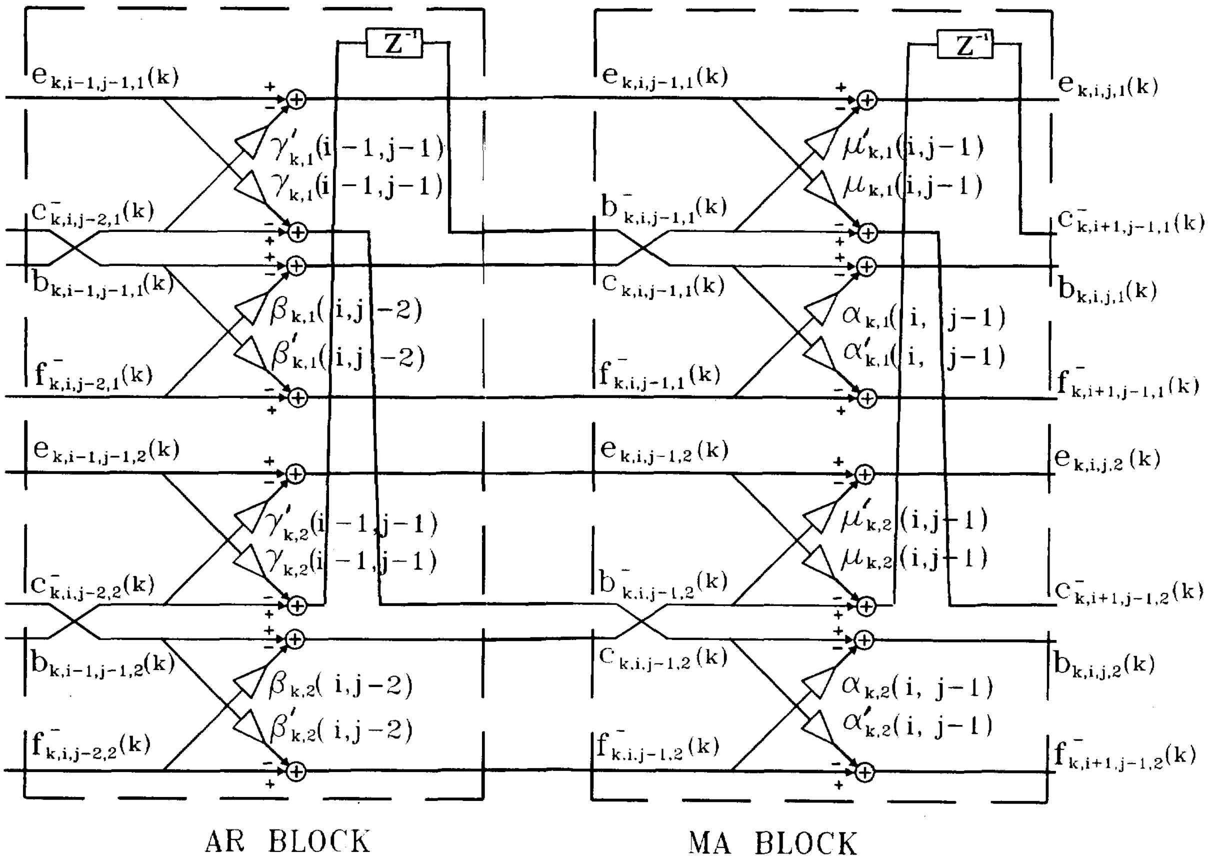


Fig. 1. (a) AR-MA recursions.

to be carried out up to the term $e_{k,p_n,q_n+1,r}(k)$. Then, the order in which the elements of $S_{k,p_n,q_n+1,r}$ may be selected is described by the sequences S1, S2, and S3 (from left to right) as given below:

S1: $\{D^{-1}\boldsymbol{x}_r(k), \boldsymbol{u}_r(k)\}, \ldots, \{D^{-p_n}\boldsymbol{x}_r(k), D^{-q_n}\boldsymbol{u}_r(k)\}$, if $p_n = q_n + 1$,

S2: $D^{-1}\boldsymbol{x}_r(k), \ldots, D^{-(p_n-q_n-1)}\boldsymbol{x}_r(k)$, $\{D^{-(p_n-q_n)}\boldsymbol{x}_r(k), \boldsymbol{u}_r(k)\}, \ldots,$ $\{D^{-p_n}\boldsymbol{x}_r(k), D^{-q_n}\boldsymbol{u}_r(k)\}$, if $p_n > q_n + 1$,

S3: $\boldsymbol{u}_r(k), \ldots, D^{-(q_n-p_n)}\boldsymbol{u}_r(k)$, $\{\boldsymbol{x}_r(k), D^{-(q_n-p_n+1)}\boldsymbol{u}_r(k)\}, \ldots,$ $\{D^{-p_n}x_r(k), D^{-q_n}u_r(k)\}$, if $p_n < q_n + 1$

where a sequence of the form $\{\boldsymbol{y}_1, \boldsymbol{z}_1\}, \{\boldsymbol{y}_2, \boldsymbol{z}_2\}, \ldots, \{\boldsymbol{y}_t, \boldsymbol{z}_t\}$ implies two possible orderings: a) $\boldsymbol{y}_1, \boldsymbol{z}_1, \boldsymbol{y}_2, \boldsymbol{z}_2, \ldots, \boldsymbol{y}_t, \boldsymbol{z}_t$ and b) $\boldsymbol{z}_1, \boldsymbol{y}_1, \boldsymbol{z}_2, \boldsymbol{y}_2, \ldots, \boldsymbol{z}_t, \boldsymbol{y}_t$. This implies that in the case of S1, two different sequences for $e_{k,i,j,r}(k)$, each terminating at $e_{k,p_n,q_n+1,r}(k)$, are possible, and they are as follows: a) $e_{k,0,0,r}(k)$, $e_{k,1,0,r}(k)$, $e_{k,1,1,r}(k)$, $e_{k,2,1,r}(k)$, $e_{k,2,2,r}(k)$, $\ldots$, b) $e_{k,0,0,r}(k)$, $e_{k,0,1,r}(k)$, $e_{k,1,1,r}(k)$, $e_{k,1,2,r}(k)$, $e_{k,2,2,r}(k), \ldots$. In other words, one can update either in the manner of "AR-MA-AR-MA-...," or "MA-AR-MA-AR-...." In the cases of S2 and S3, however, the first $(p_n - q_n - 1)$ and $(q_n - p_n + 1)$ steps, respectively, involve AR and MA updating only, resulting in the following sequences: c) $e_{k,0,0,r}, e_{k,1,0,r}, e_{k,2,0,r}, \ldots,$ and d) $e_{k,0,0,r}$, $e_{k,0,1,r}, e_{k,0,2,r}, \ldots, (e_{k,0,0,r}(k) = x_r(k))$.

We therefore need to modify the AR and MA recursions for the following two special cases: a) AR0 recursions where j is held at 0 and i is incremented and b) MA0 recursions where i is held at zero and j is incremented. As seen above, cases a) and b) arise, respectively, in the first $(p_n - q_n - 1)$ and $(q_n - p_n + 1)$ stages of the sequences S2 and S3. Noting that for $j = 0, \boldsymbol{b}^-_{k,i,0,r} = \boldsymbol{b}_{k,i,0,r}, \boldsymbol{b}_{k,0,0,r} = D^{-1}\boldsymbol{x}_r(k), \boldsymbol{f}^-_{k,0,0,r} = \boldsymbol{u}_r(k), \boldsymbol{c}_{k,i,0,r} = \boldsymbol{f}^-_{k,i,0,r}$ and identifying those sections of the AR-MA recursions where j remains fixed while i gets incremented, the AR0 building blocks can be derived. For a two-channel model, this is shown in Fig. 1(c). Note that the ith order AR0 lattice, while orthogonalizing the columns of $\boldsymbol{D}_{k,i+1,0,r}$, generates the orthogonal vectors $\boldsymbol{b}_{k,0,0,r}, \boldsymbol{b}_{k,1,0,r}, \ldots, \boldsymbol{b}_{k,i,0,r}$ for $r = 1, 2, \ldots, d$.

Similarly, for $i = 0$, we have $\boldsymbol{c}^-_{k,0,0,r} = D^{-1}\boldsymbol{u}_r(k)$. In addition, $\boldsymbol{c}^-_{k,0,j,r} = \boldsymbol{c}_{k,0,j,r-1}$ for $r = 2, 3, \ldots, d$, and $\boldsymbol{c}^-_{k,0,j,1} = \left[0, \boldsymbol{c}^t_{k-1,0,j,d}\right]^t$ for $r = 1$. Identifying those sections of Fig. 1(b), where i remains fixed while j gets incremented, we obtain the MA0 building blocks as shown for a two-channel model in Fig. 1(d). The jth-order MA0 lattice pro-

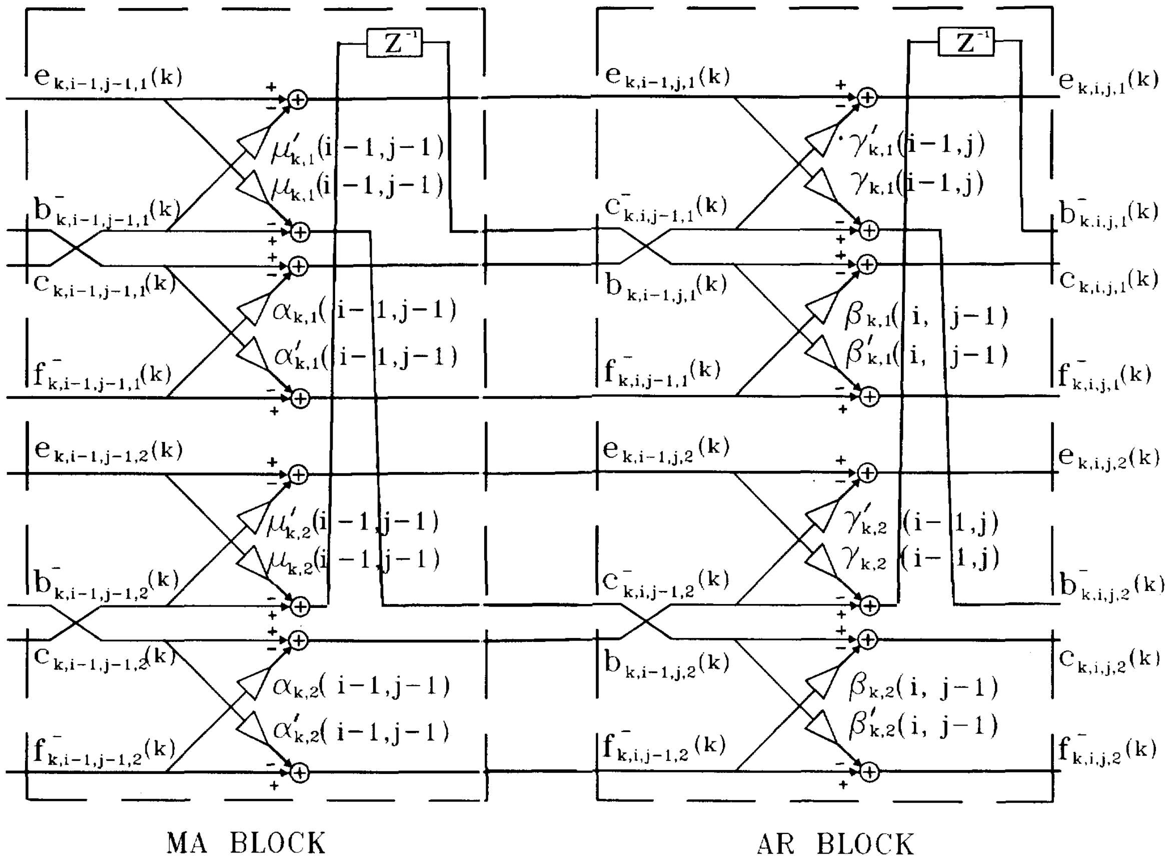


Fig. 1. (b) MA-AR recursions.

duces the orthogonal vectors $\boldsymbol{c}_{k,0,0,r}, \boldsymbol{c}_{k,0,1,r}, \ldots, \boldsymbol{c}_{k,0,2,r}$ for $r = 1, 2, \ldots, d$.

Note that an AR0 block can be directly interfaced to a MA block input by using the relation $\boldsymbol{c}_{k,i,0,r} = \boldsymbol{f}^-_{k,i,0,r}$. In the case of MA0-AR interface, we modify the MA0 block into a MA block by noting that for $r = 2, 3, \ldots, d, \boldsymbol{b}^-_{k,0,j,r} = \boldsymbol{e}_{k,0,j,r-1}$ and for $r = 1, \boldsymbol{b}^-_{k,0,j,1} = \left[0, \boldsymbol{e}^t_{k-1,0,j,d}\right]^t$. This is shown in Fig. 1(e) and can be connected to an AR block in the usual MA-AR manner.

The LSCAL algorithm is listed in Table II for the case having M AR0 stages followed by N MA-AR stages (other cases can be handled in a similar manner). The algorithm recursively updates the lattice parameters, namely, the PARCOR coefficients and the residual energy variables, which are given, respectively, by the numerators and denominators of the reflection coefficients. The PARCOR coefficients are defined as $\rho^-_{fb}(k,i,j,r) = \langle \boldsymbol{f}^-_{k,i,j-1,r}, \boldsymbol{b}^-_{k,i,j-1,r}\rangle$, $\rho^-_{fc}(k,i,j,r) = \langle \boldsymbol{f}^-_{k,i,j-1,r}, \boldsymbol{c}^-_{k,i,j-1,r}\rangle$, $\rho^-_{be}(k,i,j,r) = \langle \boldsymbol{b}_{k,i,j,r}, \boldsymbol{e}_{k,i,j,r}\rangle$, $\rho^-_{ce}(k,i,j,r) = \langle \boldsymbol{c}_{k,i,j,r}, \boldsymbol{e}_{k,i,j,r}\rangle$, and the energy variables are denoted by $\epsilon \boldsymbol{f}^-_{k,i,j,r}, \epsilon \boldsymbol{e}_{k,i,j,r}, \epsilon \boldsymbol{b}_{k,i,j,r}, \epsilon \boldsymbol{b}^-_{k,i,j,r}, \epsilon \boldsymbol{c}_{k,i,j,r}, \epsilon \boldsymbol{c}^-_{k,i,j,r}$ (ϵ is the norm-square functional on H_k, i.e., $\epsilon \boldsymbol{a} = \|\boldsymbol{a}\|^2, \forall \boldsymbol{a}$ belonging to H_k). The algorithm also uses two angle parameters [10], $\eta^-_{k,i,j,r}$ and $\eta_{k,i,j,r}$ and are given by $\eta^-_{k,i,j,r} = \langle \boldsymbol{\pi}(k), \boldsymbol{P}^{-\perp}_{k,i,j,r}\boldsymbol{\pi}(k)\rangle$ and $\eta_{k,i,j,r} = \langle \boldsymbol{\pi}(k), \boldsymbol{P}^{\perp}_{k,i,j,r}\boldsymbol{\pi}(k)\rangle$, where $\boldsymbol{\pi}(k)$ is the $k \times 1$ *pinning vector*, i.e., $\boldsymbol{\pi}(k) = [0, 0, \ldots, 1]^t$. The update relations for these variables and the initialization of the algorithm are not discussed here. These follow directly from [10] and are given in [11].

C. The Complete LSCAL Filter

Here, we develop the complete LSCAL filter using the building blocks discussed above. First, consider the case where the AR order is greater than the MA order in (1), i.e., $p > q$. It is easy to verify that the AR and MA orders of the equivalent scalar ARMA model (3) at index n, where $n = r + (k-1)d$ are given by $p_n = pd + (r-1)$ and $q_n = qd + (r-1)$. Thus, for $d \geq 2$ in (1), we have $p_n > q_n + 1$ if $p > q$. In other words, the first $(p_n - q_n - 1)$ stages of the LSCAL filter for the rth channel will involve AR0 recursions only. However, the important thing to notice here is that the quantity $(p_n - q_n - 1) = (p-q)d - 1 = \bar{p}$ (say) is a constant and independent of r, i.e., same for all the channels. The corresponding recursions can, therefore, be carried out for all the channels together by cascading $\bar{p}$ AR0 blocks generating $e_{k,\bar{p},0,r}(k)$ at the output for $r = 1, 2, \ldots, d$ simultaneously.

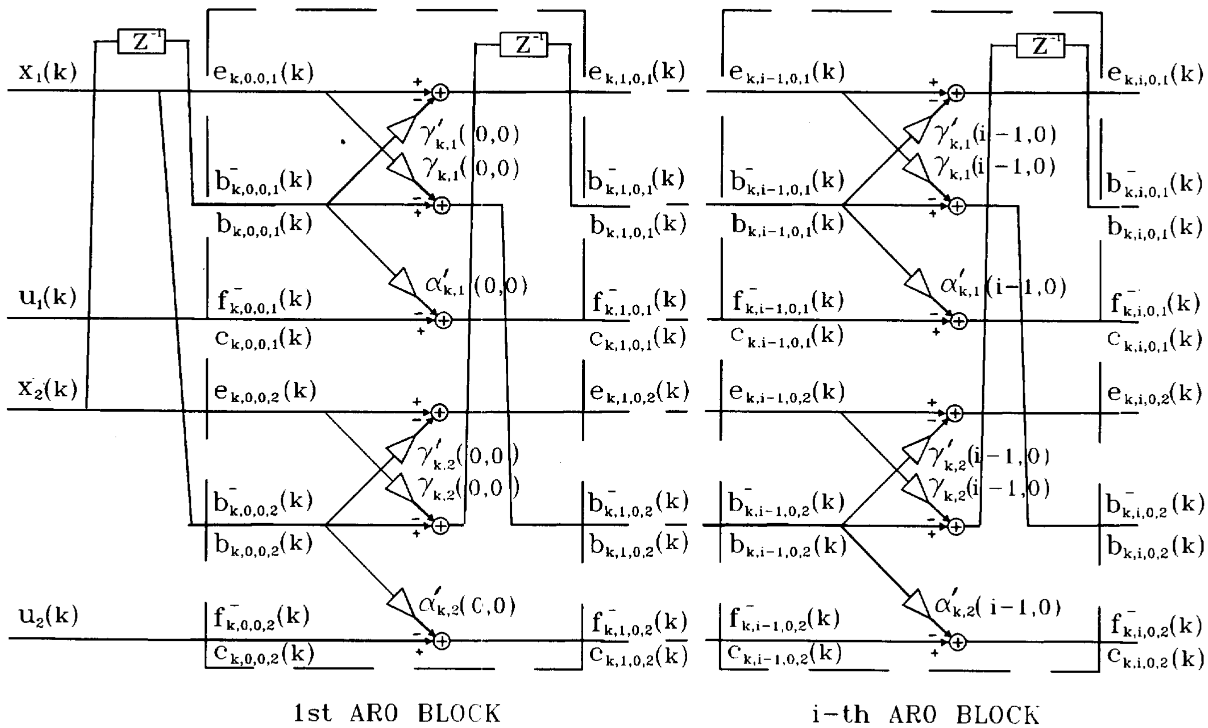


Fig. 1. (c) AR0 recursions.

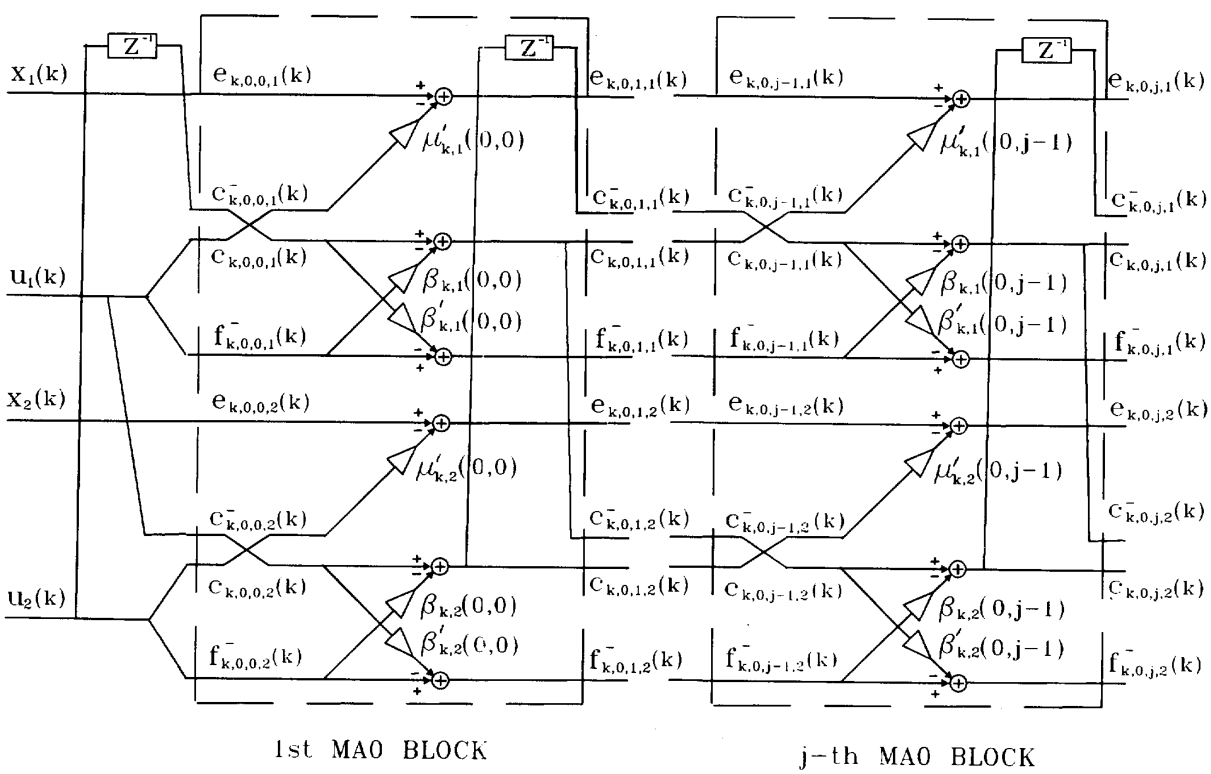


Fig. 1. (d) MA0 recursions.

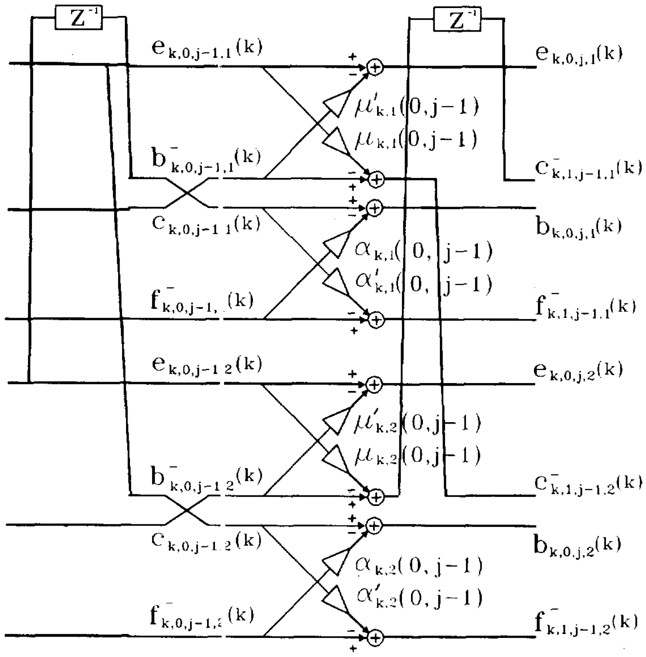


Fig. 1. (e) Modifying a MA0 block for MA0-AR interfacing.

The remaining $(p_n + q_n + 1 - \bar{p}) = 2(qd + r)$ stages are realized by cascading AR and MA blocks as shown in Fig. 2, where the 'ARMA' block is either an AR-MA or an MA-AR combination. Clearly, the total number of ARMA blocks required to complete the order recursive procedure for all the channels is $\max\{(qd + r) \mid r = 1, 2, \ldots, d\} = (q + 1)d$. In addition, observe that the quantity $\bar{p}$ is a function of $(p - q)$, and therefore, for all models with orders $(p+m, q+m)$, where m is an integer $(m \geq -q)$, the first $\bar{p}$ stages remain the same, i.e., a cascade of $\bar{p}$ AR0 blocks. On the other hand, if q is kept fixed but p is varied, then every increment or decrement in p results in the addition or deletion of more AR0 blocks. Similarly, when $q \geq p$, we have $p_n < q_n + 1$. Therefore, the first $(q_n - p_n + 1) = ((q - p)d + 1) = \bar{q}$ (say) stages consist of MA0 blocks only, and this remains unchanged for all models with orders $(p+m, q+m)(p \geq -m)$. The remaining $(p_n+q_n+1-\bar{q}) = 2(pd+r-1)$ stages are realized by cascading $(pd+d-1)$, i.e., $\max\{(pd+r-1) \mid r = 1, 2, \ldots, d\})$ ARMA blocks. In addition, every increment or decrement in q with p fixed adds or deletes d more MA0 blocks. The LSC AL filter shown in Fig. 2 thus evaluates ARMA models of all possible orders. Once the true orders are established, the recursions corresponding to other branches can be discarded.

One important thing to be noticed is that while for the rth channel the total number of elements in the corresponding signal set is $(p_n + q_n + 1)$, where (p_n, q_n) is the true order of the scalar model at index $n = r + (k-1)d$, the associated lattice recursions, however, involve only $[(p_n + q_n + 1) - (r - 1)]$ stages. This is because, for true choice of the model order (p_n, q_n), the signal set $S_{k,p_n,q_n+1,r}$ forms a linearly dependent set with the dimension of the subspace $\delta_{k,p_n,q_n+1,r}$, spanned by its elements, reducing to $[(p_n + q_n + 1) - (r - 1)]$. This is proved in the following proposition, where, for convenience, we have replaced p_n and q_n by p_r and q_r, respectively, using the periodicity of the model orders.

Proposition 1: The dimension of $\delta_{k,p_r,q_r+1,r} = (p_r + q_r + 1) - (r - 1), r = 1, 2, \ldots, d$.

Proof: We prove the result by induction. Assume that the result is true up to $r = m$, where m is an integer such that $1 \leq m \leq (d-1)$. We will show that the result is true for $r = (m+1)$ as well. Consider the signal set $S_{k,p_{m+1},q_{m+1}+1,m+1}$, for the $(m+1)$th channel, given by

$$\begin{aligned} &S_{k,p_{m+1},q_{m+1}+1,m+1} \\ &\quad = \{D^{-1}\boldsymbol{x}_{m+1}(k), D^{-2}\boldsymbol{x}_{m+1}(k), \ldots, D^{-p_{m+1}}\boldsymbol{x}_{m+1}(k), \\ &\qquad \boldsymbol{u}_{m+1}(k), D^{-1}\boldsymbol{u}_{m+1}(k), \ldots, D^{-q_{m+1}}\boldsymbol{u}_{m+1}(k)\}. \end{aligned} \quad (16)$$

From the definition (4b) and the fact that $p_{m+1} = p_m + 1, q_{m+1} = q_m + 1$, we have

$$\begin{aligned} &S_{k,p_{m+1},q_{m+1}+1,m+1} \\ &\quad = S_{k,p_m,q_m+1,m} \bigcup \{\boldsymbol{x}_m(k)\} \bigcup \{\boldsymbol{u}_{m+1}(k)\}. \end{aligned} \quad (17)$$

However, as p_m and q_m are, respectively, the true AR and MA orders of the equivalent scalar, periodic ARMA model (3) for the index $n = m + (k-1)d, \boldsymbol{x}_m(k)$ will be given by a linear combination of the elements belonging to $S_{k,p_m,q_m}\widehat{+1, m}$. This implies that the space spanned by the set $S_{k,p_m,q_m}\widehat{+1, m} \cup \{x_m(k)\}$ will be the same as $\delta_{k,p_m,q_m+1,m}$, and therefore, its dimension, by assumption, will be $[(p_m + q_m + 1) - (m - 1)]$. From this and (17),

TABLE II

LSCAL Algorithm for the Case Having M AR0 Recursive Stages Followed by N MA-AR Blocks. (The "do-par–end-par" loop indicates the operations that can be done in parallel. δ is a very small positive quantity (of the order of 0.001) to be supplied to avoid divisions by zero.)

Initialization

$X_d(0) = 0;$

do-par, $r = 1, 2, \ldots, d,$

$\epsilon e_{0,0,0,r} = \epsilon f^-_{0,0,0,r} = \epsilon b^-_{0,0,0,r} = \delta;$

end-par

For $i = 0$ *to* $(M-1)$ *begin*

do-par, $r = 1, 2, \ldots, d,$

$\rho_{be}(0,i,0,r) = \rho^-_{fb}(0,i,0,r) = 0;$

end-par

$b^-_{0,i,0,d}(0) = e_{0,i,0,d}(0) = 0; \; \epsilon b^-_{0,i,0,d} = \epsilon e_{0,i,0,d} = \delta;$

end

For $i = M$ *to* $(M+N-1)$ *begin*

do-par, $r = 1, 2, \ldots, d,$

$\rho_{ce}(0,i,i-M,r) = \rho^-_{fb}(0,i,i-M,r) = \rho_{be}(0,i,i-M+1,r)$

$= \rho^-_{fc}(0,i+1,i-M,r) = 0;$

end-par

$c_{0,i,i-M,d}(0) = e_{0,i,i-M,d}(0) = b_{0,i,i-M+1,d}(0) = e_{0,i,i-M+1,d}(0) = 0;$

$\epsilon c_{0,i,i-M,d} = \epsilon e_{0,i,i-M,d} = \epsilon b_{0,i,i-M+1,d} = \epsilon e_{0,i,i-M+1,d} = \delta;$

end

Beginning of iteration

0-th stage :

do-par, $r = 1, 2, \ldots, d,$

$e_{k,0,0,r}(k) = x_r(k); \; \epsilon e_{k,0,0,r} = \lambda . \epsilon e_{k-1,0,0,r} + x_r^2(k);$

$f^-_{k,0,0,r}(k) = u_r(k); \; \epsilon f^-_{k,0,0,r} = \lambda . \epsilon f^-_{k-1,0,0,r} + u_r^2(k);$

$\eta_{k,0,0,r} = \eta^-_{k,0,0,r} = 1$

For $r = 2, 3, \ldots, d,$

$b^-_{k,0,0,r}(k) = x_{r-1}(k); \; \epsilon b^-_{k,0,0,r} = \lambda . \epsilon b^-_{k-1,0,0,r} + x_{r-1}^2(k);$

For $r = 1,$

$b^-_{k,0,0,1}(k) = x_d(k-1); \; \epsilon b^-_{k,0,0,1} = \lambda . \epsilon b^-_{k-1,0,0,1} + x_d^2(k-1);$

end-par

First M AR0 recursions :

For $i = 0$ *to* $(M-1)$ *begin*

do-par, $r = 1, 2, \ldots, d,$

$$\eta^-_{k,i+1,0,r} = \eta_{k,i+1,0,r} = \eta_{k,i,0,r} - \frac{\{b^-_{k,i,0,r}(k)\}^2}{\epsilon b^-_{k,i,0,r}};$$

$$\rho_{be}(k,i,0,r) = \lambda . \rho_{be}(k-1,i,0,r) + \frac{b^-_{k,i,0,r}(k) . e_{k,i,0,r}(k)}{\eta_{k,i,0,r}};$$

$$\rho^-_{fb}(k,i,0,r) = \lambda . \rho^-_{fb}(k-1,i,0,r) + \frac{b^-_{k,i,0,r}(k) . f^-_{k,i,0,r}(k)}{\eta^-_{k,i,0,r}};$$

$$e_{k,i+1,0,r}(k) = e_{k,i,0,r}(k) - \frac{\rho_{be}(k,i,0,r)}{\epsilon b^-_{k,i,0,r}} . b^-_{k,i,0,r}(k);$$

$$f^-_{k,i+1,0,r}(k) = f^-_{k,i,0,r}(k) - \frac{\rho^-_{fb}(k,i,0,r)}{\epsilon b^-_{k,i,0,r}} . b^-_{k,i,0,r}(k);$$

$$\epsilon e_{k,i+1,0,r} = \epsilon e_{k,i,0,r} - \frac{\{\rho_{be}(k,i,0,r)\}^2}{\epsilon b^-_{k,i,0,r}};$$

$$\epsilon f^-_{k,i+1,0,r} = \epsilon f^-_{k,i,0,r} - \frac{\{\rho^-_{fb}(k,i,0,r)\}^2}{\epsilon b^-_{k,i,0,r}};$$

For $r = 2, 3, \ldots, d,$

$$b^-_{k,i+1,0,r}(k) = b^-_{k,i,0,r-1}(k) - \frac{\rho_{be}(k,i,0,r-1)}{\epsilon e_{k,i,0,r-1}} . e_{k,i,0,r-1}(k);$$

$$\epsilon b^-_{k,i+1,0,r} = \epsilon b^-_{k,i,0,r-1} - \frac{\{\rho_{be}(k,i,0,r-1)\}^2}{\epsilon e_{k,i,0,r-1}};$$

For $r = 1,$

$$b^-_{k,i+1,0,1}(k) = b^-_{k-1,i,0,d}(k-1) - \frac{\rho_{be}(k-1,i,0,d)}{\epsilon e_{k-1,i,0,d}} . e_{k-1,i,0,d}(k-1);$$

$$\epsilon b^-_{k,i+1,0,1} = \epsilon b^-_{k-1,i,0,d} - \frac{\{\rho_{be}(k-1,i,0,d)\}^2}{\epsilon e_{k-1,i,0,d}};$$

end-par

end

do-par, $r = 1, 2, \ldots, d,$

$c_{k,M,0,r}(k) = f^-_{k,M,0,r}(k); \; \epsilon c_{k,M,0,r} = \epsilon f^-_{k,M,0,r};$

end-par

Next N MA-AR blocks :

For $i = M$ *to* $(M+N-1)$ *begin*

MA-Block :

do-par, $r = 1, 2, \ldots, d,$

$$\eta_{k,i,i-M+1,r} = \eta_{k,i,i-M,r} - \frac{\{c_{k,i,i-M,r}(k)\}^2}{\epsilon c_{k,i,i-M,r}};$$

$$\eta^-_{k,i+1,i-M,r} = \eta^-_{k,i,i-M,r} - \frac{\{b^-_{k,i,i-M,r}(k)\}^2}{\epsilon b^-_{k,i,i-M,r}};$$

$$\rho_{ce}(k,i,i-M,r) = \lambda . \rho_{ce}(k-1,i,i-M,r) + \frac{c_{k,i,i-M,r}(k) . e_{k,i,i-M,r}(k)}{\eta_{k,i,i-M,r}};$$

$$\rho^-_{fb}(k,i,i-M,r) = \lambda . \rho^-_{fb}(k-1,i,i-M,r) + \frac{f^-_{k,i,i-M,r}(k) . b^-_{k,i,i-M,r}(k)}{\eta^-_{k,i,i-M,r}};$$

$$e_{k,i,i-M+1,r}(k) = e_{k,i,i-M,r}(k) - \frac{\rho_{ce}(k,i,i-M,r)}{\epsilon c_{k,i,i-M,r}} c_{k,i,i-M,r}(k);$$

$$b_{k,i,i-M+1,r}(k) = b^-_{k,i,i-M,r}(k) - \frac{\rho^-_{fb}(k,i,i-M,r)}{\epsilon f^-_{k,i,i-M,r}} . f^-_{k,i,i-M,r}(k);$$

$$f^-_{k,i+1,i-M,r}(k) = f^-_{k,i,i-M,r}(k) - \frac{\rho^-_{fb}(k,i,i-M,r)}{\epsilon b^-_{k,i,i-M,r}} . b^-_{k,i,i-M,r}(k);$$

$$\epsilon e_{k,i,i-M+1,r} = \epsilon e_{k,i,i-M,r} - \frac{\{\rho_{ce}(k,i,i-M,r)\}^2}{\epsilon c_{k,i,i-M,r}};$$

it then follows that the dimension of $\delta_{k,p_{m+1},q_{m+1}+1,m+1} = [(p_m+q_m+1)-(m-1)]+1$, which, after some manipulations, is seen to be equal to $[(p_{m+1}+q_{m+1}+1)-((m+1)-1)]$. This proves that the result is valid for $r = (m+1)$ as well.

TABLE II *(cont'd)*

$$\epsilon b_{k,i,i-M+1,r} = \epsilon b^{-}_{k,i,i-M,r} - \frac{\{\rho^{-}_{fb}(k,i,i-M,r)\}^2}{\epsilon f^{-}_{k,i,i-M,r}};$$

$$\epsilon f^{-}_{k,i+1,i-M,r} = \epsilon f^{-}_{k,i,i-M,r} - \frac{\{\rho^{-}_{fb}(k,i,i-M,r)\}^2}{\epsilon b^{-}_{k,i,i-M,r}};$$

For $r = 2, 3, \ldots, d,$

$$c^{-}_{k,i+1,i-M,r}(k) = c_{k,i,i-M,r-1}(k) - \frac{\rho_{ce}(k,i,i-M,r-1)}{\epsilon e_{k,i,i-M,r-1}} \cdot e_{k,i,i-M,r-1}(k);$$

$$\epsilon c^{-}_{k,i+1,i-M,r} = \epsilon c_{k,i,i-M,r-1} - \frac{\{\rho_{ce}(k,i,i-M,r-1)\}^2}{\epsilon e_{k,i,i-M,r-1}};$$

For $r = 1,$

$$c^{-}_{k,i+1,i-M,1}(k) = c_{k-1,i,i-M,d}(k-1) - \frac{\rho_{ce}(k-1,i,i-M,d)}{\epsilon e_{k-1,i,i-M,d}} \cdot e_{k-1,i,i-M,d}(k-1);$$

$$\epsilon c^{-}_{k,i+1,i-M,1} = \epsilon c_{k-1,i,i-M,d} - \frac{\{\rho_{ce}(k-1,i,i-M,d)\}^2}{\epsilon e_{k-1,i,i-M,d}};$$

end-par

AR-Block :

do-par, $r = 1, 2, \ldots, d,$

$$\eta_{k,i+1,i-M+1,r} = \eta_{k,i,i-M+1,r} - \frac{\{b_{k,i,i-M+1,r}(k)\}^2}{\epsilon b_{k,i,i-M+1,r}};$$

$$\eta^{-}_{k,i+1,i-M+1,r} = \eta^{-}_{k,i+1,i-M,r} - \frac{\{c^{-}_{k,i+1,i-M,r}(k)\}^2}{\epsilon c^{-}_{k,i+1,i-M,r}};$$

$$\rho_{be}(k,i,i-M+1,r) = \lambda \cdot \rho_{be}(k-1,i,i-M+1,r) + \frac{b_{k,i,i-M+1,r}(k) \cdot e_{k,i,i-M+1,r}(k)}{\eta_{k,i,i-M+1,r}};$$

$$\rho^{-}_{fc}(k,i+1,i-M,r) = \lambda \cdot \rho^{-}_{fc}(k-1,i+1,i-M,r) + \frac{f^{-}_{k,i+1,i-M,r}(k) \cdot c^{-}_{k,i+1,i-M,r}(k)}{\eta^{-}_{k,i+1,i-M,r}};$$

$$e_{k,i+1,i-M+1,r}(k) = e_{k,i,i-M+1,r}(k) - \frac{\rho_{be}(k,i,i-M+1,r)}{\epsilon b_{k,i,i-M+1,r}} \cdot b_{k,i,i-M+1,r}(k);$$

$$c_{k,i+1,i-M+1,r}(k) = c^{-}_{k,i+1,i-M,r}(k) - \frac{\rho^{-}_{fc}(k,i+1,i-M,r)}{\epsilon f^{-}_{k,i+1,i-M,r}} \cdot f^{-}_{k,i+1,i-M,r}(k);$$

$$f^{-}_{k,i+1,i-M+1,r}(k) = f^{-}_{k,i+1,i-M,r}(k) - \frac{\rho^{-}_{fc}(k,i+1,i-M,r)}{\epsilon c^{-}_{k,i+1,i-M,r}} \cdot c^{-}_{k,i+1,i-M,r}(k);$$

$$\epsilon e_{k,i+1,i-M+1,r} = \epsilon e_{k,i,i-M+1,r} - \frac{\{\rho_{be}(k,i,i-M+1,r)\}^2}{\epsilon b_{k,i,i-M+1,r}}.$$

$$\epsilon c_{k,i+1,i-M+1,r} = \epsilon c^{-}_{k,i+1,i-M,r} - \frac{\{\rho^{-}_{fc}(k,i+1,i-M,r)\}^2}{\epsilon f^{-}_{k,i+1,i-M,r}};$$

$$\epsilon f^{-}_{k,i+1,i-M+1,r} = \epsilon f^{-}_{k,i+1,i-M,r} - \frac{\{\rho^{-}_{fc}(k,i+1,i-M,r)\}^2}{\epsilon c^{-}_{k,i+1,i-M,r}};$$

For $r = 2, 3, \ldots, d,$

$$b^{-}_{k,i+1,i-M+1,r}(k) = b_{k,i,i-M+1,r-1}(k) - \frac{\rho_{be}(k,i,i-M+1,r-1)}{\epsilon e_{k,i,i-M+1,r-1}} \cdot e_{k,i,i-M+1,r-1}(k);$$

$$\epsilon b^{-}_{k,i+1,i-M+1,r} = \epsilon b_{k,i,i-M+1,r-1} - \frac{\{\rho_{be}(k,i,i-M+1,r-1)\}^2}{\epsilon e_{k,i,i-M+1,r-1}};$$

For $r = 1,$

$$b^{-}_{k,i+1,i-M+1,1}(k) = b_{k-1,i,i-M+1,d}(k-1) - \frac{\rho_{be}(k-1,i,i-M+1,d)}{\epsilon e_{k-1,i,i-M+1,d}} \cdot e_{k-1,i,i-M+1,d}(k-1);$$

$$\epsilon b^{-}_{k,i+1,i-M+1,1} = \epsilon b_{k-1,i,i-M+1,d} - \frac{\{\rho_{be}(k-1,i,i-M+1,d)\}^2}{\epsilon e_{k-1,i,i-M+1,d}};$$

end-par
end
end

Finally, to initialize the induction, consider the case when $r = 1$. It is easy to verify by inspection that the elements of the corresponding signal set $S_{k,p_1,q_1}\widehat{+1,1}$ are not related by any linear relationship amongst themselves, implying that the dimension of $\delta_{k,p_1,q_1}\widehat{+1,1} = (p_1 + q_1 + 1)$, which is the same as $[(p_r + q_r + 1) - (r - 1)]$ for $r = 1$. This completes the proof. □

It is also easy to verify by inspection that the first $[(p_r + q_r + 1) - (r - 1)]$ elements of the sequences S1, S2, or S3 are not related by any linear relationship and thus form a linearly independent set. From this and Proposition 1, it then follows that the lattice recursions for the rth channel terminate (i.e., $e_{k,i,j,r}(k)$ reaches zero value) at the $[(p_r + q_r + 1) - (r - 1)]$th stage for true choice of (p_n, q_n). (This analysis, however, implicitly assumes that the unknown system conforms exactly to the model (1), i.e., it assumes estimation errors due to imperfect modeling to be zero, or negligibly small, at least asymptotically). The implications of this on the estimation of the ARMA parameters is examined below.

D. Order Recursive Estimation of ARMA Parameters

As seen from (8), the order recursive estimation of the AR and the MA parameters of (3) requires the $A^f_{k,i,j,r}(m)$'s and $B^f_{k,i,j,r}(l)$'s to be updated in both AR and MA orders. We derive a step-up like approach for this purpose by expressing the other associated orthogonal projections in the manner of

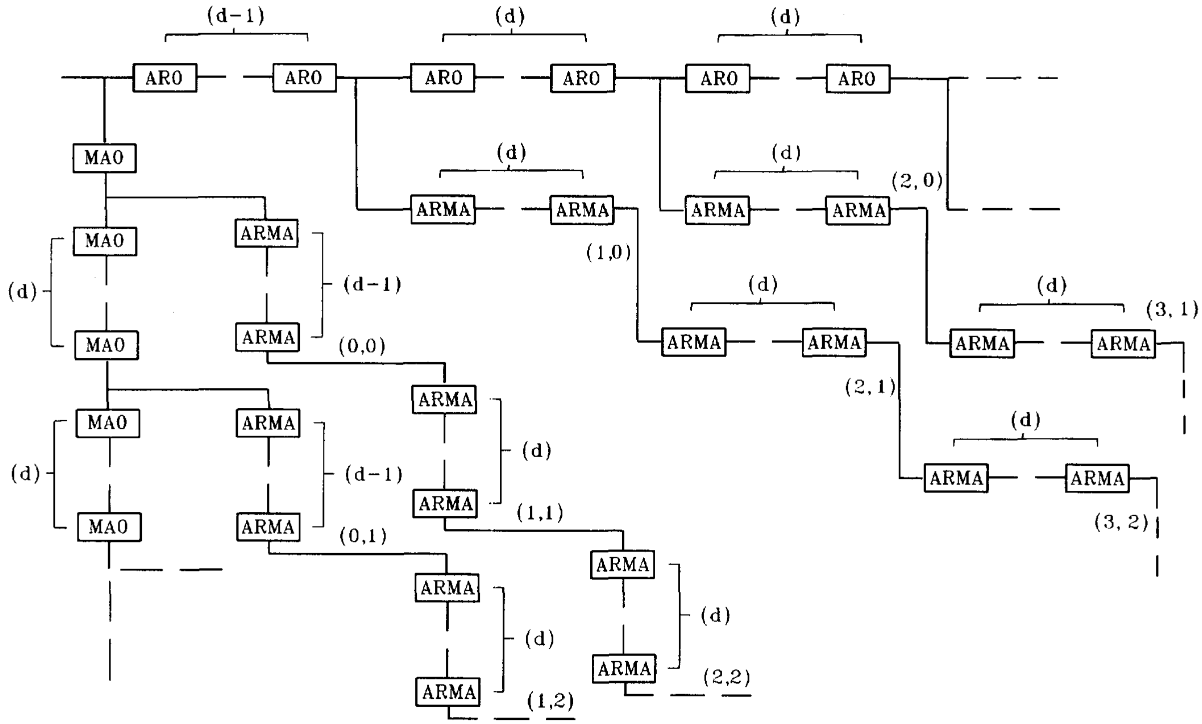


Fig. 2. Complete LSCAL filter. The 'ARMA' blocks indicate either AR-MA or MA-AR configurations. The pair $(i,j), i,j = 0,1,2,\ldots$ denotes the order of the model being evaluated at a particular stage of the LSCAL filter.

(8) (i.e., in transversal forms) as follows:

$$\boldsymbol{P}^{-}_{k,i,j-1,r}\boldsymbol{u}_r(k) = -\sum_{m=1}^{i} C^{f}_{k,i,j-1,r}(m)D^{-m}\boldsymbol{x}_r(k) + \sum_{l=1}^{j-1} D^{f}_{k,i,j-1,r}(l)D^{-l}\boldsymbol{u}_r(k), \tag{18}$$

$$\boldsymbol{P}_{k,i,j,r}D^{-(i+1)}\boldsymbol{x}_r(k) = -\sum_{m=1}^{i} A^{b}_{k,i,j,r}(m)D^{-m}\boldsymbol{x}_r(k) + \sum_{l=0}^{j-1} B^{b}_{k,i,j,r}(l)D^{-l}\boldsymbol{u}_r(k), \tag{19}$$

$$\boldsymbol{P}_{k,i,j,r}D^{-j}\boldsymbol{u}_r(k) = -\sum_{m=1}^{i} C^{b}_{k,i,j,r}(m)D^{-m}\boldsymbol{x}_r(k) + \sum_{l=0}^{j-1} D^{b}_{k,i,j,r}(l)D^{-l}\boldsymbol{u}_r(k), \tag{20}$$

$$\boldsymbol{P}^{-}_{k,i,j-1,r}D^{-(i+1)}\boldsymbol{x}_r(k) = -\sum_{m=1}^{i} \overline{A}_{k,i,j-1,r}(m)D^{-m}\boldsymbol{x}_r(k) + \sum_{l=1}^{j-1} \overline{B}_{k,i,j-1,r}(l)D^{-l}\boldsymbol{u}_r(k), \tag{21}$$

$$\boldsymbol{P}^{-}_{k,i,j-1,r}D^{-j}\boldsymbol{u}_r(k) = -\sum_{m=1}^{i} \overline{C}_{k,i,j-1,r}(m)D^{-m}\boldsymbol{x}_r(k) + \sum_{l=1}^{j-1} \overline{D}_{k,i,j-1,r}(l)D^{-l}\boldsymbol{u}_r(k). \tag{22}$$

(The superscript 'b' on $A^b_{k,i,j,r}(m), B^b_{k,i,j,r}(l), C^b_{k,i,j,r}(m), D^b_{k,i,j,r}(l)$ indicates that these are associated with backward projection operations.) The transversal parameters in (8) and in (18)–(22) are updated using the following general approach: Suppose $\boldsymbol{P}_{U,v}\boldsymbol{z}$ in (15) is to be expressed in a transversal filter form as $\boldsymbol{P}_{U,v}\boldsymbol{z} = \sum_{i=1}^{p} c_{U,v}(i)\boldsymbol{u}_i + K_{U,v}\boldsymbol{v}$ given the transversal filters: $\boldsymbol{P}_U\boldsymbol{z} = \sum_{i=1}^{p} c_U(i)\boldsymbol{u}_i, \boldsymbol{P}_U\boldsymbol{v} = \sum_{i=1}^{p} d_U(i)\boldsymbol{u}_i$. Noting that $\boldsymbol{P}^{\perp}_U\boldsymbol{v} = \boldsymbol{v} - \boldsymbol{P}_U\boldsymbol{v}$, we may then write $\boldsymbol{P}_{U,v}\boldsymbol{z}$ as

$$\boldsymbol{P}_{U,v}\boldsymbol{z} = \boldsymbol{P}_U\boldsymbol{z} + \frac{\langle \boldsymbol{P}^{\perp}_U\boldsymbol{z}, \boldsymbol{P}^{\perp}_U\boldsymbol{v}\rangle}{\|\boldsymbol{P}^{\perp}_U\boldsymbol{v}\|^2}(\boldsymbol{v} - \boldsymbol{P}_U\boldsymbol{v}). \tag{23}$$

Replacing $\boldsymbol{P}_{U,v}\boldsymbol{z}$ on the L.H.S. and $\boldsymbol{P}_U\boldsymbol{z}$ and $\boldsymbol{P}_U\boldsymbol{v}$ on the R.H.S. of (23) by the respective transversal filter forms and equating the coefficients of the $\boldsymbol{u}_i$'s and $\boldsymbol{v}$ on both sides, the "higher order" coefficients $c_{U,v}(i)$ and $K_{U,v}$ are obtained in terms of the "lower order" coefficients $c_U(i)$ and $d_U(i)$. The update relations of the transversal filters defined in (8) and (18)–(22), based on this principle, are listed in Table III. The order-recursive computation of the estimates $A^f_{k,i,j,r}(m)$ and $B^f_{k,i,j,r}(l)$, however, can not be carried out up to the $(p_r + q_r + 1)$th stage for true choice of the model orders since, as seen earlier, the signal set $S_{k,i,j,r}$, in such cases, forms a linearly dependent set for all (i, j) in the sequence S1, S2 or, S3 such that $i + j > [(p_r + q_r + 1) - (r - 1)]$ implying that the projection $\boldsymbol{P}_{k,i,j,r}\boldsymbol{x}_r(k)$ will be obtained as a linear combination of the first $[(p_r + q_r + 1) - (r - 1)]$ elements of the sequence (note that in (23), $\boldsymbol{P}_u v$ cannot be expressed by the R.H.S if $\boldsymbol{v}$ lies in the subspace $\{U\}$ since we then have $\|\boldsymbol{P}_U^\perp \boldsymbol{v}\| = 0$; instead, we will have $\boldsymbol{P}_{U,v}\boldsymbol{z} = \boldsymbol{P}_U\boldsymbol{z}$). More specifically, we will have

$$\boldsymbol{P}_{k,p_r,q_r+1,r}\boldsymbol{x}_r(k) = -\sum_{i=1}^{t_r} A^f_{k,t_r,s_r+1,r}(i)D^{-i}\boldsymbol{x}_r(k) + \sum_{j=0}^{s_r} B^f_{k,t_r,s_r+1,r}(j)D^{-j}\boldsymbol{u}_r(k), \tag{24}$$

where t_r and s_r are two integers such that $t_r \le p_r, s_r \le q_r$, and $t_r + s_r + 1 = [(p_r + q_r + 1) - (r - 1)]$ (for $r = 1, t_r = p_r$ and $s_r = q_r$). Now, from the discussion of Section II, it is easy to see that the first $(r - 1)$ MA coefficients of the scalar model (3) at index $n = r + (k - 1)d$ are identically zero, i.e., $b_n(j) = 0, j = 1, 2, \ldots, (r - 1)$. Define the modified signal set $\tilde{S}_{k,i,j,r} \subset S_{k,i,j,r}$ as $\tilde{S}_{k,i,j,r} = \{D^{-1}\boldsymbol{x}_r(k), \ldots, D^{-i}\boldsymbol{x}_r(k), \boldsymbol{u}_r(k), D^{-r}\boldsymbol{u}_r(k), \ldots, D^{-(j-1)}\boldsymbol{u}_r(k)\}$. By inspection, it can be verified that the elements of $\tilde{S}_{k,p_r,q_r+1,r}$ are not related by any linear relationship, i.e., $\tilde{S}_{k,p_r,q_r+1,r}$ is a linearly independent set. Further, the number of elements in $\tilde{S}_{k,p_r,q_r+1,r}$ is $[(p_r + q_r + 1) - (r - 1)]$, which is also the dimension of the space $\delta_{k,p_r,q_r+1,r}$ meaning that $\tilde{S}_{k,p_r,q_r+1,r}$ forms a basis of $\delta_{k,p_r,q_r+1,r}$. Clearly, to estimate the parameters of (3) at the index $n = r+(k-1)d$, it is required to express $P_{k,p_r,q_r+1,r}x_r(k)$ as a linear combination of the elements of $\tilde{S}_{k,p_r,q_r+1,r}$ and *not* of $S_{k,t_r,s_r+1,r}$ as given by (24). Unfortunately, the fact that the first $(r - 1)$ MA coefficients are zero cannot be incorporated in the lattice recursions directly since, as the development of the LSCAL algorithm shows, the elements of the signal set are required to be chosen in the manner of S1, S2, or S3, and thus, the first $(r - 1)$ terms $D^{-1}\boldsymbol{u}_r(k), D^{-2}\boldsymbol{u}_r(k), \ldots, D^{-(r-1)}\boldsymbol{u}_r(k)$ cannot be skipped.

The problem, however, can be solved by expressing each of these terms in (24) as a linear combination of the basis vectors belonging to $\tilde{S}_{k,p_r,q_r+1,r}$. First, observe that for the first channel, the ARMA parameter estimates are available directly at the end of the $(p_1 + q_1 + 1)$th step since $\tilde{S}_{k,p_1,q_1+1,1} \equiv S_{k,p_1,q_1+1,1} \equiv S_{k,t_1,s_1+1,1}$. Suppose, the estimates $\hat{a}_r(i)$ and $\hat{b}_r(j)$ (i.e., $\hat{a}_n(i)$ and $\hat{b}_n(j)$ for $n = r + (k - 1)d$) have been evaluated up to the $(m - 1)$th channel, $1 < m \le d$, i.e., for $r = 1, 2, \ldots, m - 1, \boldsymbol{P}_{k,p_r,q_r+1,r}\boldsymbol{x}_r(k)$ has been obtained in terms of the elements of $\tilde{S}_{k,p_r,q_r+1,r}$ as

$$P_{k,p_r,q_r+1,r}\boldsymbol{x}_r(k) = -\sum_{i=1}^{p_r} \hat{a}_r(i)D^{-i}\boldsymbol{x}_r(k) + \hat{b}_r(0)\boldsymbol{u}_r(k) + \sum_{j=r}^{q_r} \hat{b}_r(j)D^{-j}\boldsymbol{u}_r(k). \tag{25}$$

In addition, assume that the estimates $\hat{a}_r(i)$ and $\hat{b}_r(j)$ $r = 1, 2, \ldots, m - 1$ have been transmitted by the rth channel processor to the mth channel processor. Using the fact that for true choice of (p_r, q_r), $\boldsymbol{P}_{k,p_r,q_r+1,r}\boldsymbol{x}_r(k) = x_r(k)$, (25) is then rewritten as

$$\boldsymbol{u}_r(k) = \frac{1}{\hat{b}_r(0)}[\boldsymbol{x}_r(k) + \sum_{i=1}^{p_r} \hat{a}_r(i)D^{-i}\boldsymbol{x}_r(k) - \sum_{j=r}^{q_r} \hat{b}_r(j)D^{-j}\boldsymbol{u}_r(k)]. \tag{26}$$

From the definition (4a)–(4c) of the operator $D^{-1}, \boldsymbol{u}_r(k)$ and $\boldsymbol{x}_r(k)$ in (26) can be recognized as $\boldsymbol{u}_r(k) = D^{-(m-r)}\boldsymbol{u}_m(k), \boldsymbol{x}_r(k) = D^{-(m-r)}\boldsymbol{x}_m(k)$. In general, any $D^{-i}\boldsymbol{x}_r(k)$ and $D^{-j}\boldsymbol{u}_r(k)$ in (26) can be replaced, respectively, by $D^{-(m-r+i)}\boldsymbol{x}_m(k)$ and $D^{-(m-r+j)}\boldsymbol{u}_m(k)$. In addition since $p_r = pd + (r - 1)$ and $q_r = qd + (r - 1)$, we have $D^{-p_r}\boldsymbol{x}_r(k) = D^{-p_r}D^{-(m-r)}\boldsymbol{x}_m(k) = D^{-p_m}\boldsymbol{x}_m(k)$, and similarly, $D^{-q_r}\boldsymbol{u}_r(k) = D^{-q_m}\boldsymbol{u}_m(k)$. Making these substitutions in (26), we observe that (26) expresses $D^{-(m-r)}\boldsymbol{u}_m(k), r = 1, 2, \ldots, (m - 1)$ as linear combinations of the elements of the set $\{D^{-(m-r)}\boldsymbol{x}_m(k), \ldots, D^{-p_m}\boldsymbol{x}_m(k), D^{-m}\boldsymbol{u}_m(k), \ldots, D^{-q_m} u_m(k)\}$, which is a proper subset of $\tilde{S}_{k,p_m,q_m+1,m}$. Replacing the $(m - 1)$ terms $D^{-(m-r)}\boldsymbol{u}_r(k), r = 1, 2, \ldots, (m - 1)$ in (24) by their respective representation in terms of the elements of $\tilde{S}_{k,p_m,q_m+1,m}, \boldsymbol{P}_{k,p_m,q_m+1,1}\boldsymbol{x}_m(k)$ is then obtained in the desired transversal form (25).

IV. Applications of the LSCAL Algorithm

A. Joint Process Estimation Using Two Correlated Processes

The LSCAL algorithm derived above is suited to system identification problems. It is also possible to use this for estimating a process $z(k)$ from two correlated, d-channel processes $\boldsymbol{x}(k)$ and $\boldsymbol{u}(k)$ adaptively. A (p, q)th order estimator is given by

$$\hat{z}_{k,p,q}(k) = \sum_{i=0}^{p} \boldsymbol{g}^t_{k,p,q,i}\boldsymbol{x}(k - i) + \sum_{j=0}^{q} \boldsymbol{h}^t_{k,p,q,i}\boldsymbol{u}(k - j) \tag{27}$$

where $\hat{z}_{k,p,q}(k)$ is the least squares estimate of $z(k)$ based on data up to the index k, and $\boldsymbol{g}_{k,p,q,i} = [g_{k,p,q,i}(1), \ldots, g_{k,p,q,i}(d)]^t$ and $\boldsymbol{h}_{k,p,q,j} = [h_{k,p,q,j}(1), \ldots, h_{k,p,q,j}(d)]^t$ are the ith and jth coefficients of the two transversal filters processing $\boldsymbol{x}(k)$ and $\boldsymbol{u}(k)$, respectively. The purpose is to evaluate $\hat{z}_{k,p,q}(k)$ for all (p, q) up to a given maximum order. This is achieved by projecting the vector $\boldsymbol{z}(k) = [z(1), \ldots, z(k)]^t$ orthogonally on the subspace $\boldsymbol{\Lambda}_{k,p,q}$ spanned by the set $\{\boldsymbol{x}_d(k), D^{-1}\boldsymbol{x}_d(k), \ldots, D^{-(pd+(d-1))}\boldsymbol{x}_d(k), \boldsymbol{u}_d(k), D^{-1}\boldsymbol{u}_d(k), \ldots, D^{-(qd+(d-1))}\boldsymbol{u}_d(k)\}$, which can be seen to be equivalent to $\{x_d(k)\} \cup S_{k,p_d,q_d+1,d}$, where $p_d = pd + (r - 1), q_d = qd + (r - 1)$. As explained earlier, the LSCAL algorithm orthogonalizes each channel signal set by generating the orthogonal components $\boldsymbol{b}_{k,i,j,r}$ and $\boldsymbol{c}_{k,i,j,r}$ in a specific order. Selecting

TABLE III
UPDATE RELATIONS FOR THE TRANSVERSAL FILTERS

AR Updating of $A^f_{k,i,j,r}(m)$, $B^f_{k,i,j,r}(l)$, $C^f_{k,i,j,r}(m)$, $D^f_{k,i,j,r}(l)$:

1(a).
$$A^f_{k,i+1,j,r}(m) = A^f_{k,i,j,r}(m) - \gamma'_{k,r}(i,j)\,A^b_{k,i,j,r}(m),\ m = 1,2,..,i.$$
$$= -\gamma'_{k,r}(i,j),\ m = i+1.$$
$$B^f_{k,i+1,j,r}(l) = B^f_{k,i,j,r}(l) - \gamma'_{k,r}(i,j)\,B^b_{k,i,j,r}(l),\ l = 0,1,..,j-1.$$

2(a).
$$C^f_{k,i+1,j,r}(m) = C^f_{k,i,j,r}(m) - \alpha'_{k,r}(i,j)\,\overline{A}_{k,i,j,r}(m),\ m = 1,2,..,i.$$
$$= -\alpha'_{k,r}(i,j),\ m = i+1.$$
$$D^f_{k,i+1,j,r}(l) = D^f_{k,i,j,r}(l) - \alpha'_{k,r}(i,j)\,\overline{B}_{k,i,j,r}(l),\ l = 1,2,..,j.$$

MA Updating of $A^f_{k,i,j,r}(m)$, $B^f_{k,i,j,r}(l)$, $C^f_{k,i,j,r}(m)$, $D^f_{k,i,j,r}(l)$:

1(b).
$$A^f_{k,i,j+1,r}(m) = A^f_{k,i,j,r}(m) - \mu'_{k,r}(i,j)\,C^b_{k,i,j,r}(m),\ m = 1,2,..,i.$$
$$B^f_{k,i,j+1,r}(l) = B^f_{k,i,j,r}(l) - \mu'_{k,r}(i,j)\,D^b_{k,i,j,r}(l),\ l = 0,1,..,j-1.$$
$$= \mu'_{k,r}(i,j),\ l = j.$$

2(b).
$$C^f_{k,i,j+1,r}(m) = C^f_{k,i,j,r}(m) - \beta'_{k,r}(i,j)\,\overline{C}_{k,i,j,r}(m),\ m = 1,2,..,i.$$
$$D^f_{k,i,j+1,r}(l) = D^f_{k,i,j,r}(l) - \beta'_{k,r}(i,j)\,\overline{D}_{k,i,j,r}(l),\ l = 1,2,..,j,$$
$$= \beta'_{k,r}(i,j),\ l = j+1.$$

Updating other parameters :

3.
$$A^b_{k,i,j,r}(m) = \overline{A}_{k,i,j-1,r}(m) - \alpha_{k,r}(i,j-1)\,C^f_{k,i,j-1,r}(m),\ m = 1,2,..,$$
$$B^b_{k,i,j,r}(l) = \alpha_{k,r}(i,j-1),\ l = 0,$$
$$= \overline{B}_{k,i,j-1,r}(l) - \alpha_{k,r}(i,j-1)\,D^f_{k,i,j-1,r}(l),\ l = 1,2,..,j-1.$$

4.
$$C^b_{k,i,j,r}(m) = \overline{C}_{k,i,j-1,r}(m) - \beta_{k,r}(i,j-1)\,C^f_{k,i,j-1,r}(m),\ m = 1,2,..,i$$
$$D^b_{k,i,j,r}(l) = \beta_{k,r}(i,j-1),\ l = 0,$$
$$= \overline{D}_{k,i,j-1,r}(l) - \beta_{k,r}(i,j-1)\,D^f_{k,i,j-1,r}(l),\ l = 1,2,..,j-1.$$

5. *For* $r = 2, 3, \ldots, d,$
$$\overline{A}_{k,i+1,j,r}(m) = -\gamma_{k,r-1}(i,j),\ m = 1,$$
$$= A^b_{k,i,j,r-1}(m-1) - \gamma_{k,r-1}(i,j)\,A^f_{k,i,j,r-1}(m-1),\quad m = 2,3,..,i+1.$$
$$\overline{B}_{k,i+1,j,r}(l) = B^b_{k,i,j,r-1}(l-1) - \gamma_{k,r-1}(i,j)\,B^f_{k,i,j,r-1}(l-1),\quad l = 1,2,..,j.$$

For $r = 1,$
$$\overline{A}_{k,i+1,j,1}(m) = -\gamma_{k-1,d}(i,j),\ m = 1,$$
$$= A^b_{k-1,i,j,d}(m-1) - \gamma_{k-1,d}(i,j)\,A^f_{k-1,i,j,d}(m-1),\quad m = 2,3,..,i+1.$$
$$\overline{B}_{k,i+1,j,1}(l) = B^b_{k-1,i,j,d}(l-1) - \gamma_{k-1,d}(i,j)\,B^f_{k-1,i,j,d}(l-1),\quad l = 1,2,..,j.$$

6. *For* $r = 2, 3, \ldots, d,$
$$\overline{C}_{k,i+1,j,r}(m) = -\mu_{k,r-1}(i,j),\ m = 1,$$
$$= C^b_{k,i,j,r-1}(m-1) - \mu_{k,r-1}(i,j)\,A^f_{k,i,j,r-1}(m-1),\quad m = 2,3,..,i+1.$$
$$\overline{D}_{k,i+1,j,r}(l) = D^b_{k,i,j,r-1}(l-1) - \mu_{k,r-1}(i,j)\,B^f_{k,i,j,r-1}(l-1),\quad l = 1,2,..,j.$$

For $r = 1,$
$$\overline{C}_{k,i+1,j,1}(m) = -\mu_{k-1,d}(i,j),\ m = 1,$$
$$= C^b_{k-1,i,j,d}(m-1) - \mu_{k-1,d}(i,j)\,A^f_{k-1,i,j,d}(m-1),\quad m = 2,3,..,i+1.$$
$$\overline{D}_{k,i+1,j,1}(l) = D^b_{k-1,i,j,d}(l-1) - \mu_{k-1,d}(i,j)\,B^f_{k-1,i,j,d}(l-1),\quad l = 1,2,..,j.$$

the vectors $\boldsymbol{b}_{k,i,j,d}$ and $\boldsymbol{c}_{k,i,j,d}$ generated by the dth channel recursions, together with $\boldsymbol{e}_{k,p_d,q_d+1,d}$ generated at the $(p_d + q_d + 1)$th stage, we obtain an orthogonal basis for $\Lambda_{k,p,q}.\hat{z}_{k,p,q}(k)$ can then be evaluated by projecting $z(k)$ along each orthogonal component, adding the individual projections and taking the kth component.

The above joint process estimator (JPE) can be used when one is required to identify (1) with a nonidentity coefficient matrix $\boldsymbol{B}(0)$. It is easy to verify that the identification for the rth channel, $r = 1, 2, \ldots, d$ can be carried out by feeding the signals $x_r(k-1)$ and $w_r(k)$ at the $\boldsymbol{x}$ and $\boldsymbol{u}$ input terminals, respectively, of a $(p-1, q)$th-order LSCAL JPE and applying $x_r(k)$ as $z(k)$.

B. ARMA Modeling with Unknown, White Input

In applications concerning time series modeling, it is often required to fit an ARMA model of the type (1) to a $d \times 1$ process $\boldsymbol{x}(k)$. The input $\boldsymbol{w}(k)$, which is a white process in this case, is usually not available, which introduces nonlinearity in the problem via the product of the unknown terms $\boldsymbol{B}(j)$ and $\boldsymbol{w}(k-j)$. The ARMA parameters cannot then be estimated optimally by linear least squares methods. An approximate but practically useful approach [5], [6] is to employ a bootstrap technique to estimate the present input sample from the available data and then feed it back to the input of the lattice. Here, we propose a bootstrap procedure to generate the input sample estimate $\hat{\boldsymbol{u}}_r(k)$ using the LSCAL JPE. (For this purpose, we consider a generalized JPE, where one is given a set of processes $z^r(k), r = 1,2,\ldots,d$, where each is estimated in the form of (27), generating $\hat{z}^r_{k,p,q}(k), r = 1,2,\ldots,d$.) First, observe that $v(n)$ in (3), under white input conditions, is the prediction error associated with the linear prediction of $y(n)$ from $y(n-i)$'s and $v(n-j)$'s, $i = 1,2,\ldots,p_n, j = 1,2,\ldots,q_n$. In addition, note that the inner product $\langle \boldsymbol{z}_1, \boldsymbol{z}_2\rangle, \boldsymbol{z}_1, \boldsymbol{z}_2 \in H_k$ provides an unbiased, k-sample estimate of the correlation between the corresponding random variables $\boldsymbol{z}_1, \boldsymbol{z}_2$, and it approaches the true correlation as $k \to \infty$. Define the set $\bar{S}_{k,p_r,q_r,r} \equiv \{D^{-i}\boldsymbol{x}_r(k), D^{-j}\boldsymbol{u}_r(k) | i = 1,2,\ldots,p_r; j = r, r+1,\ldots,q_r\}$, and let $\bar{\boldsymbol{P}}^{\perp}_{k,p_r,q_r,r}$ be the orthogonal projection error operator associated with the subspace $\bar{\delta}_{k,p_r,q_r,r}$ spanned by $\bar{S}_{k,p_r,q_r,r}$. For large k, we can then write $\boldsymbol{u}_r(k) \approx \bar{\boldsymbol{P}}^{\perp}_{k,p_r,q_r,r}\boldsymbol{x}_r(k) = \hat{\boldsymbol{u}}_r(k)$ (say). To show how the present input estimate $\hat{u}_r(k)$ (i.e., the kth entry of $\hat{\boldsymbol{u}}_r(k)$) can be obtained, let us consider the case $r = 1$ first. It is easy to see that the set $\bar{S}_{k,p_1,q_1,1}$ is given by the collection of the columns of the following matrix:

$$\left[\frac{\boldsymbol{0}}{\boldsymbol{x}_d(k-1), \boldsymbol{D}_{k-1,\tilde{p}_d,\tilde{q}_d,d}}\right]$$

where $\tilde{p}_d = (p-1)d + (d-1), \tilde{q}_d = (q-1)d + (d-1)$, and the data matrix $\boldsymbol{D}_{k,i,j,r}$ is defined in (5). Clearly, $\hat{u}_1(k)$ can be obtained by using a $(p-1, q-1)$th order LSCAL JPE, which is described above, in the following manner: Apply $\boldsymbol{x}_r(k-1)$'s and $\boldsymbol{u}_r(k-1)$'s at the x and u input terminals, $x_1(k)$ as $z^1(k)$, and subtract the output $\hat{z}^1_{k,p-1,q-1}$ from $x_1(k)$. For $r = 2,3,\ldots,d$, it is easy to see that $\bar{S}_{k,p_r,q_r,r} = \bar{S}_{k,p_{r-1},q_{r-1},r-1} \cup \{\boldsymbol{x}_{r-1}(k)\}$. Consequently, $\bar{\delta}_{k,p_r,q_r,r} = \langle \hat{\boldsymbol{u}}_{r-1}(k)\rangle \otimes \bar{\delta}_{k,p_{r-1},q_{r-1},r-1}$, which provides a recursive relation on $\bar{\delta}_{k,p_r,q_r,r}$ in the index r. Carrying out this recursion on the R.H.S. further until $r = 1$, we obtain

$$\bar{\delta}_{k,p_r,q_r,r} = \langle \hat{\boldsymbol{u}}_{r-1}(k)\rangle \otimes \langle \hat{\boldsymbol{u}}_{r-2}(k)\rangle \otimes \cdots \otimes \langle \hat{\boldsymbol{u}}_1(k)\rangle \otimes \bar{\delta}_{k,p_1,q_1,1}. \tag{28}$$

Consequently

$$\bar{\boldsymbol{P}}_{k,p_r,q_r,r}\boldsymbol{x}_r(k) = \sum_{i=1}^{r-1} \tau_{k,r,i}\hat{\boldsymbol{u}}_i(k) + \bar{\boldsymbol{P}}_{k,p_1,q_1,1}\boldsymbol{x}_r(k), \quad r = 2,3,\ldots,d \tag{29}$$

where $\bar{\boldsymbol{P}}_{k,p_r,q_r,r} = \boldsymbol{I} - \bar{\boldsymbol{P}}^{\perp}_{k,p_r,q_r,r}$, and $\tau_{k,r,i} = \frac{\langle \boldsymbol{x}_r(k), \hat{\boldsymbol{u}}_i(k)\rangle}{\|\hat{\boldsymbol{u}}_i(k)\|^2}, i = 1,2,\ldots,(r-1)$. Note that the current entry of $\bar{\boldsymbol{P}}_{k,p_1,q_1,1}\boldsymbol{x}_r(k)$ is given by the output $\hat{z}^r_{k,p-1,q-1}(k)$ of the same JPE with $x_r(k)$ replacing $z^r(k)$. Evaluating the R.H.S. of (29) for the kth index and subtracting from $x_r(k)$, we obtain $\hat{u}_r(k)$. Note that this scheme, while evaluating $\hat{u}_r(k), d \geq r \geq 2$, makes use of not only the joint process estimator but also the estimated input samples up to the previous channel, viz., $\hat{u}_i(k), i = 1,2,\ldots,(r-1)$. To make the above procedure self-sustaining, $\hat{u}_r(k)$'s are fed back to the input of the LSCAL filter through unit delays to play the role of the actual input at the next index of time. The initial values of input estimates can be taken as zero as suggested in [5]. If the model orders are not known *a priori*, the recursions, in the initial phase, need to be carried out up to a large order until actual order is established.

V. Discussion and Conclusion

A least squares lattice algorithm has been presented for identifying a multichannel ARMA model that is computationally efficient in that it employs scalar operations only and is amenable to pipelined implementation. For a (p,q)th-order model, the proposed algorithm involves computations of $O[(p+q+1)d](O[p+q+1)^2d^2]$ for computing the ARMA parameters) per processor [11], which is d times less than the order of computation required by a direct multichannel extension of the Karlsson–Hayes algorithm [7]. Further computational advantages arise from the ability of the algorithm to compute ARMA models of any specified order (p,q) directly, without requiring p and q to be identical. For instance, if the d-channel ARMA model is converted into a $2d$-channel AR model by assuming $p = q$ [5], [6] and is then identified by, say, the method of [1], then, depending on whether $p > q$ or otherwise, one needs to employ a total of $[15(p-q)d^2 - 3d]$ and $[14(q-p)d^2 + 2d]$ extra additions and $[32(p-q)d^2 - 6d]$ and $[30(q-p)d^2 + 4d]$ extra multiplications, respectively, as compared with the proposed method (note that each figure is directly proportional to the difference $|p-q|$).

A computer simulation study of the proposed LSCAL algorithm was carried out on a two-channel ARMA model for two different time-varying environments. In the former, the parameters were held fixed up to a certain fixed index of time, after which they were changed abruptly to another set of values. In the latter, the parameters were allowed to vary linearly but slowly with time. In both cases, the estimates latched on to the parameter trajectories after an initial convergence time of 8–10 samples. Whereas in the former, the steady-state values of the estimation errors were zero, in the latter, because of continuous time variation, nonzero estimation errors were present, although with sufficiently small magnitude (within 5%). In addition, long and extensive simulations (which are not presented here for lack of space) indicated no evidence of numerical instability.

The proposed method, as seen above, is free of the overparameterization problem that characterizes the conventional ARMA modeling methods [5], [6], as the latter force p and q to be identical. Some extent of overparameterization, and, consequently, certain excess computations, may still come up in situations where the model to be identified has different orders for different channels. This is because the proposed algorithm assumes identical orders for all the channels. In such contexts, the recently proposed identification algorithm of Glentis *et al.* [12] may be more useful. Their method, like the algorithm proposed here, employs scalar operations only, but, in addition, permits independent order recursions for all the channels.

References

[1] T. Kawase, H. Sakai, and H. Tokumaru, "Recursive least squares circular lattice and escalator estimation algorithms," *IEEE Trans. Acoust. Speech Signal Processing*, vol. ASSP-31, no. 1, pp. 228–231, Feb. 1983.

[2] F. Ling and J. G. Proakis, "A generalized multichannel least squares lattice algorithm based on sequential processing stages," *IEEE Trans. Acoust. Speech Signal Processing,* vol. ASSP-32, no. 32, pp. 381–390, Apr. 1984.

[3] H. Lev-Ari, "Modular architectures for adaptive multichannel lattice algorithms," *IEEE Trans. Acoust. Speech Signal Processing,* vol. ASSP-35, no. 4, pp. 543–552, Apr. 1987.

[4] P. S. Lewis, "QR-based algorithms for multichannel adaptive least squares lattice filters," *IEEE Trans. Acoust. Speech Signal Processing,* vol. 38, no. 3, pp. 421–432, Mar. 1990.

[5] D. T. L. Lee, B. Friedlander, and M. Morf, "Recursive ladder algorithms for ARMA modeling," *IEEE Trans. Automat. Contr.,* vol. AC-27, no. 4, pp. 753–764, Aug. 1981.

[6] A. Benveniste and C. Chaure, "AR and ARMA identification algorithm of Levinson type: An innovation approach," *IEEE Trans. Automat. Contr.,* vol. AC-26, no. 6, pp. 1243–1261, Dec. 1981.

[7] E. Karlsson and M. H. Hayes, "Least squares ARMA modeling of linear time-varying systems: Lattice filter structures and fast RLS algorithms," *IEEE Trans. Acoust., Speech, Signal Processing,* vol. ASSP-35, pp. 994–1014, July 1987.

[8] S. Karaboyas, N. Kalouptsidis, and C. Caroubalos, "Efficient ARX identification algorithms with full parallelism," *IEEE Trans. Acoust., Speech, Signal Processing,* vol. 38, no. 11, pp. 1902–1913, Nov. 1990.

[9] M. Chakraborty and S. Prasad, "Multivariate ARMA modeling by scalar algorithms," *IEEE Trans. Signal Processing,* vol. 41, no. 4, pp. 1692–1697, Apr. 1993.

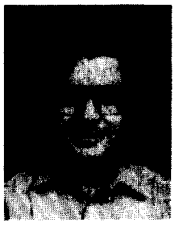
[10] D. T. L. Lee, M. Morf, and B. Friedlander, "RLS ladder estimation algorithms," *IEEE Trans. Acoust., Speech, Signal Processing,* vol. ASSP-29, pp. 1362–1367, June 1981.

[11] M. Chakraborty, "Multivariate ARMA modeling by scalar algorithms," Ph.D. Thesis, Dept. of Elect. Eng., Indian Inst. of Technol. New Delhi, Feb. 1993.

[12] G. A. Glentis and N. Kalouptsidis, "Fast adaptive algorithms for multichannel filtering and system identification," *IEEE Trans. Signal Processing,* vol. 40, pp. 2433–2458, Oct. 1992.

Mrityunjoy Chakraborty received the B.E. degree in electronics and telecommunication engineering from Jadavpur University, Calcutta, India, in 1983 and the M.Tech. degree in electrical engineering from the Indian Institute of Technology, Kanpur, India, in 1985. He submitted his Ph.D. thesis on multichannel ARMA modeling and filtering to the Department of Electrical Enginnering, Indian Institute of Technology, New Delhi, India, in February 1993, where he is presently working as a senior scientific officer.

Earlier, he held a faculty position in the Department of Electrical and Electronic Engineering, Birla Institute of Technology and Science, Pilani, India, during 1986–1987. His research interests are in the areas of digital and statistical signal processing, including adaptive filtering, fast algorithms for signal processing, spectrum estimation and modeling, computational linear algebra, and multiresolution signal processing.



Surendra Prasad (M'75) received the B.Tech degree in electronics and electrical communication engineering from the Indian Institute of Technology, Kharagpur, India, in 1969 and the M.Tech. and Ph.D. degrees in electrical communication engineering from the Indian Institute of Technology, New Delhi, in 1971 and 1974, respectively.

He has been with the Indian Institute of Technology, New Delhi, since 1971, where he is presently a Professor of Electrical Engineering and coordinator of the Research Programme in Telematics. He was a visiting research fellow at the Loughborough University of Technology, Loughborough, England, during 1976–1977, where he was involved in developing algorithms for adaptive array processing for HF arrays. He was also a visiting faculty member at the Pennsylvania State University, University Park, during 1985–1986. His teaching and research interests include communications and statistical and digital signal processing. He has been a consultant to a number of government agencies and industry in these and related areas. Currently, he is engaged in research in various aspects of statistical signal processing, including among others, single and multichannel ARMA filtering and modeling, high-resolution spectrum estimation and array processing, higher order spectrum estimation and its applications to problems in digital communication, and underwater acoustics and seismic data processing. He has published more than sixty papers in these areas in reputed journals. He edited a special issue of the *Journal of I.E.T.E* in March–April 1989 in the area of statistical signal processing.

Dr. Prasad was the recipient of the Vikram Sarabhai Research Award in Electronics and Telecommunications for the year 1987 and the Shanti Swarup Bhatnagar Award for Engineering Sciences for 1988. He was a Co-Chairperson for the "Indo-US Workshop in One and Two Dimensions" held in New Delhi in November 1989. He is a Fellow of the Indian National Academy of Engineering and the Indian National Academy of Science.