A Recursive Procedure for ARMA Modeling

RANDOLPH L. MOSES, JAMES A. CADZOW, SENIOR MEMBER, IEEE, AND A. A. (LOUIS) BEEX

Abstract—**This paper presents a two-part fast recursive algorithm for ARMA modeling. The algorithm first obtains estimates of the p autoregressive coefficients from a set of p extended Yule-Walker equations. An exact recursive lattice algorithm for this estimator is then derived. The $q + 1$ numerator spectrum coefficients are then obtained by using one of the output data sequences of this lattice algorithm. The complete recursive algorithm is fast in the sense that $O(p + q)$ computations are required for each update. Moreover, an exponential forgetting factor is incorporated to facilitate tracking of time variations in the time series.**

I. Introduction

There are many applications in which it is desired to estimate the essential attributes from observations of a zero-mean, complex-valued wide sense stationary time series $\{x(n)\}$. This characterization is often adequately revealed through knowledge of its autocorrelation function

$$r_x(n) = E\{x(n+m)x^*(m)\} \qquad n = 0, \ +1, \ +2, \ \cdots \tag{1}$$

in which E and $*$ denote the operations of expectation and complex conjugation, respectively. The requisite characterization may also be made in the frequency domain through the power spectral density function

$$S_x(e^{j\omega}) = \sum_{n=-\infty}^{\infty} r_x(n) e^{-j\omega n}. \tag{2}$$

Frequently, this second-order statistical characterization provides all the information required for a given application (e.g., optimal Wiener filtering, one-step prediction, etc.).

A variety of parametric procedures have been proposed for estimating the power spectral density function from a finite set of data observations [2]-[10]. Considerable work has recently been focused on the autoregressive moving average (ARMA) model of order (p, q), where it is assumed that $\{x(n)\}$ satisfies

$$x(n) + \sum_{i=0}^{p} a_i x(n-i) = \sum_{k=0}^{q} b_k w(n-k) \tag{3}$$

and where $\{w(n)\}$ is zero-mean, unit variance white noise. The corresponding power spectral density function is given by

$$S_x(e^{j\omega}) = \frac{|b_0 + b_1 e^{-j\omega} + \cdots + b_q e^{-jq\omega}|^2}{|1 + a_1 e^{-j\omega} + \cdots + a_p e^{-jp\omega}|^2} = \frac{|B(e^{j\omega})|^2}{|A(e^{j\omega})|^2} \tag{4}$$

in which the a_k and b_k are referred to as the autoregressive (AR) and moving average (MA) coefficients, respectively. The power spectral density function may also be written as

$$S_x(e^{j\omega}) = \frac{C(e^{j\omega})}{|A(e^{j\omega})|^2} \tag{5a}$$

where

$$C(z) \triangleq B(z)B^*(1/z^*)$$
$$= c_{-q}z^{-q} + \cdots + c_{-1}z^{-1} + c_0 + c_1 z + \cdots + c_q z^q. \tag{5b}$$

We shall refer to the c_k coefficients as the numerator spectral (NS) coefficients. It is well known that any continuous spectral density can be approximated arbitrarily closely by this rational form if the order pair (p, q) is selected adequately large [1]. Thus, by imposing a rational form on the spectral model, we incur no real loss in spectral representation.

The problem of ARMA spectral estimation is concerned with estimating the coefficients of the power spectral density model in either (4) or (5) from a finite data record, say $x(1)$, $x(2)$, $\cdots$, $x(N)$. Most ARMA spectral estimation procedures available are block-processing algorithms; that is, they operate on the block of N data points to obtain a spectral estimate in a single computational effort. However, in many applications one desires to update the estimated coefficients of an ARMA spectral model as each new data point becomes available. Examples include radar and sonar signal processing, and applications involving nonstationary data where the coefficients should track changing signal characteristics (in nonstationary environments these coefficients no longer have the interpretation of being power spectral density coefficients for a stationary time series). For these types of applications a recursive coefficient estimation procedure is more suitable than a block processing one.

The growing interest in recursive algorithms has been met with the development of so-called fast recursive lattice algorithms. Many of these algorithms are based on the autoregressive (AR) model, which is the special case of (2) with $q = 0$ there [11]-[14]. ARMA algorithms have also been derived in the context of two-dimensional AR models [15], [21], [22]. However, these methods assume knowledge of the driving white noise process $\{w(k)\}$, or they must estimate it by some sort of bootstrapping procedure.

In this paper we present an alternative recursive procedure for ARMA spectral estimation. This procedure is composed of two parts. First, the autoregressive coefficients are estimated by using a fast recursive lattice algorithm. Then, a forward prediction error sequence generated by this lattice algorithm is used to recursively compute numerator spectrum coefficient

Manuscript received October 30, 1984; revised October 30, 1984. This work was supported in part by the Statistics and Probability Program of the Office of Naval Research under Contract N00014-0257.

R. L. Moses and A. A. Beex are with the Department of Electrical Engineering, Virginia Polytechnic Institute and State University, Blacksburg, VA 24061.

J. A. Cadzow is with the Department of Electrical and Computer Engineering, Arizona State University, Tempe, AZ 85287.

0096-3518/85/1000-1188$01.00 © 1985 IEEE

estimates. This recursive procedure is fast in the sense that only $O(p + q)$ computations per time update are required.

The outline of the paper is as follows. In Section II we derive a block processing estimator for the AR coefficients of an ARMA model. Section III develops a fast recursive lattice implementation of the latter AR coefficient estimator. A recursive numerator spectrum estimator is given in Section IV. Section V presents some numerical examples and Section VI concludes the paper.

II. Block Processing AR Coefficient Estimation

This section develops a computationally efficient block processing procedure for estimating the AR coefficients in an ARMA (p, q) model from the N data observations $x(1), \cdots, x(N)$. This procedure is based on the so-called extended Yule-Walker equations defined by multiplying both sides of (1) by $x^*(n - m)$ and taking the expected value to give

$$\sum_{m=1}^{p} a_k r_x(m-k) = -r_x(m), \qquad m > q. \tag{6}$$

It is important to note that the parameter m is restricted to exceed the numerator order q. (The Yule-Walker equations involve the moving average coefficients b_k in a nonlinear manner for $0 \leq m \leq q$).

A straightforward procedure for obtaining AR coefficient estimates can be found by expressing the first "p" Yule-Walker equations as a set of linear equations

$$\begin{bmatrix} r_x(q) & r_x(q-1) & \cdots & r_x(q-p+1) \\ r_x(q+1) & r_x(q) & \cdots & r_x(q-p+2) \\ \vdots & \vdots & & \vdots \\ r_x(q+p-1) & r_x(q+p-2) & \cdots & r_x(q) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{bmatrix} = - \begin{bmatrix} r_x(q+1) \\ r_x(q+2) \\ \vdots \\ r_x(q+p) \end{bmatrix} \tag{7a}$$

or

$$\boldsymbol{R}\boldsymbol{a} = -\boldsymbol{r}. \tag{7b}$$

If exact autocorrelations are given, (7b) can be used to solve for the AR coefficient vector $\boldsymbol{a}$. In practice however, it is necessary to compute appropriate autocorrelation estimates from the given set of time series observations. The estimates considered here are prewindow estimates, defined by the $(n - q) \times 1$ vectors $\boldsymbol{x}_n$ and $\boldsymbol{y}_n$ and the $(n - q) \times m$ matrices $X_{m,n}$ and $Y_{m,n}$

$$\boldsymbol{x}_n = [x(q+1)\; x(q+2) \cdots x(n)]^T \tag{8a}$$

$$\boldsymbol{y}_n = [y(q+1)\; y(q+2) \cdots y(n)]^T \tag{8b}$$

$$X_{p,N} = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ x(q+1) & 0 & & \cdot \\ x(q+2) & x(q+1) & & \cdot \\ & & & \cdot \\ \cdot & \cdot & & 0 \\ \cdot & \cdot & & x(q+1) \\ \cdot & \cdot & & \cdot \\ x(N-1) & x(N-2) & \cdots & x(N-p) \end{bmatrix} = [S\boldsymbol{x}_N \;\; S^2\boldsymbol{x}_N \cdots S^p\boldsymbol{x}_N] \tag{8c}$$

$$Y_{p,N} = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ y(q+1) & 0 & & \cdot \\ y(q+2) & y(q+1) & & \cdot \\ & & & \cdot \\ \cdot & \cdot & & 0 \\ \cdot & \cdot & & y(q+1) \\ \cdot & \cdot & & \cdot \\ y(N-1) & y(N-2) & \cdots & y(N-p) \end{bmatrix} = [S\boldsymbol{y}_N \;\; S^2\boldsymbol{y}_N \cdots S^p\boldsymbol{y}_N] \tag{8d}$$

where $y(k) = x(k - q)$, and where the shift operator S is defined by

$$S\boldsymbol{x}_n = [0 \quad x(q+1) \cdots x(n-1)]^T. \tag{9}$$

The autocorrelation matrix and vector in expression (7) are then estimated by

$$\boldsymbol{R}_N = Y_{p,N}^H X_{p,N} \tag{10a}$$

$$\boldsymbol{r}_N = Y_{p,N}^H \boldsymbol{x}_N \tag{10b}$$

where H is the Hermitian operator. The effect of the zeros in (8c) and (8d) is to bias the autocorrelation estimates that make up $\boldsymbol{R}_N$ and $\boldsymbol{r}_N$. Specific formulas for this bias can be found in [5], [16]. These biases are proportional to the fraction of data points set to zero and therefore vanish as $N \rightarrow \infty$. Heuristically, we expect the biases in $\boldsymbol{R}_N$ and $\boldsymbol{r}_N$ to cause a bias in $\boldsymbol{a}_{N,p}$, and for this reason find it undesirable. On the other hand, these padding zeros facilitate a corresponding decrease in the number of computations needed to solve for the AR coefficient estimates.

If $\boldsymbol{R}$ and $\boldsymbol{r}$ in (7) are replaced by $\boldsymbol{R}_N$ and $\boldsymbol{r}_N$, the corresponding AR coefficient vector estimate is

$$\boldsymbol{a}_{p,N} = -[\boldsymbol{R}_N]^{-1}\boldsymbol{r}_N \tag{11a}$$

$$= -[Y_{p,N}^H X_{p,N}]^{-1} Y_{p,N}^H \boldsymbol{x}_N. \tag{11b}$$

It can be shown [23] that if $\{x(k)\}$ is an ergodic ARMA (p_a, q_a) process, $p_a \leq p$ and $q_a \leq q$, and $\{w(k)\}$ is an independent, identically distributed sequence with finite fourth moments, then $\boldsymbol{a}_{p,N}$ in (11) is an asymptotically unbiased and consistent estimate of $\boldsymbol{a}$ in (7). In fact, $\sqrt{N}[a - a_{p,N}]$ is asymptotically normally distributed [23].

Equation (11) also exhibits some attractive computational features. First, the $p \times p$ matrix $Y_{p,N}^H X_{p,N}$ has displacement rank 3 (see [17], [18]), so (11) may be solved using only $O(3p^2)$ computations, as compared with $O(p^3)$ computations needed by conventional solution techniques. For large values of p, the computational savings can be significant. A second attractive feature is that equation (11) may be solved recursively (requiring $O(p)$ computations per update) via the algorithm discussed in the next section.

Note that when $q = 0$, the estimator in (11) reduces to the prewindow AR model coefficient estimator in [13], [19].

III. The Fast Recursive Algorithm

In this section we introduce a computationally fast recursive implementation of the procedure discussed in the previous section. This recursive method facilitates continual updating of the AR coefficient estimates as new data points become available. Thus at every time interval this method yields the same AR coefficients as those obtained by solving the system of equations (11). Moreover, the recursive method provides AR coefficient estimates for all model orders from 1 to p.

The recursive algorithm is based on updating elements of prediction error vectors. Equation (11) can be written

$$Y^H_{m,n} f^x_{m,n} = \mathbf{0} \tag{12}$$

where the $(n - q) \times 1$ vector $f^x_{m,n}$ is the so-called forward prediction error vector defined by

$$f^x_{m,n} = x_n + X_{m,n} a_{m,n}. \tag{13}$$

Note that the kth element of $f^x_{p,n}$ (for $k \geq q + m$) is

$$f^x_{m,n}(k) = x(k) + \sum_{i=1}^{m} a_{m,n}(i) x(k-i) \tag{14}$$

which can be thought of as the error resulting from a prediction of $x(k)$ by a linear combination of the m previous data $x(k - 1), x(k - 2), \cdots, x(k - m)$. To reinforce this prediction error interpretation, consider the mth order estimate of the vector x_n as being

$$\hat{x}_{m,n} = -X_{m,n} a_{m,n} \tag{15}$$

which in turn generates the forward prediction error vector

$$f^x_{m,n} = x_n - \hat{x}_{m,n}. \tag{16}$$

Upon substitution of expression (11) into (16), the estimate vector is given by

$$\begin{aligned} \hat{x}_{m,n} &= X_{m,n}[Y^H_{m,n} X_{m,n}]^{-1} Y^H_{m,n} x_n \\ &= P_{XY} x_n \end{aligned} \tag{17}$$

and the corresponding forward prediction error becomes

$$\begin{aligned} f^x_{m,n} &= \{I - X_{m,n}[Y^H_{m,n} X_{m,n}]^{-1} Y^H_{m,n}\} x_n \\ &= P^c_{XY} x_n. \end{aligned} \tag{18}$$

We have used here the compact $(n - q) \times (n - q)$ matrix product representations

$$P_{XY} = X_{m,n}[Y^H_{m,n} X_{m,n}]^{-1} Y^H_{m,n} \tag{19a}$$

$$P^c_{XY} = I - P_{XY}. \tag{19b}$$

Alternatively, the AR coefficient vector estimate $a_{m,n}$ in (11) can be associated with an auxiliary minimization problem involving the prediction error vector. Namely, this estimate minimizes the quadratic functional

$$g(a_{m,n}) = [f^x_{m,n}]^H W [f^x_{m,n}] \tag{20}$$

where W is the $(n - q) \times (n - q)$ positive semidefinite matrix

$$W = Y_{m,n} Y^H_{m,n} \tag{21}$$

We can also define the mth order backward prediction error vector for x_n by

$$b^x_{m,n-1} = S^{m+1} x_n + X_{m,n} \tilde{a}_{m,n}. \tag{22}$$

The subscript $n - 1$ is used to indicate that observed data only through time $n - 1$ appears in equation (22). It can be seen that the kth row of (22) represents a prediction of $x(k - m - 1)$ by a linear combination of the m most immediate future values $x(k - m), \cdots, x(k - 1)$. The resulting error in this backward prediction is $b^x_{m,n-1}(k)$. Proceeding in a manner similar to that for the forward prediction error vector, the $\tilde{a}_{m,n}$ vector is given by

$$\tilde{a}_{m,n} = -[Y^H_{m,n} X_{m,n}]^{-1} Y^H_{m,n} [S^{m+1} x_n]. \tag{23}$$

The vector estimate of $S^{m+1} x_n$ is specified by

$$S^{m+1} x_n = -X_{m,n} \tilde{a}_{m,n} = P_{XY}(S^{m+1} x_n) \tag{24}$$

and the corresponding backward prediction error vector is

$$b^x_{m,n-1} = P^c_{XY}(S^{m+1} x_n) \tag{25}$$

where P_{XY} and P^c_{XY} are given by (19). Alternatively, the $\tilde{a}_{m,n}$ vector is the one that minimizes the quadratic functional

$$g(a_{m,n}) = [b^x_{m,n-1}]^H W [b^x_{m,n-1}] \tag{26}$$

where W is defined in (21).

Prediction error vectors corresponding to the y_n vector can be defined similarly by

$$\begin{aligned} f^y_{m,n} &= y_n - Y_{m,n}[X^H_{m,n} Y_{m,n}]^{-1} X^H_{m,n} y_n \\ &= P^c_{YX} y_n \end{aligned} \tag{27}$$

and the mth order backward prediction error then is

$$\begin{aligned} b^y_{m,n-1} &= [S^{m+1} y_n] - Y_{m,n}[X^H_{m,n} Y_m]^{-1} X^H_{m,n} [S^{m+1} y] \\ &= P^c_{YX}(S^{m+1} y_n). \end{aligned} \tag{28}$$

We can gain more insight about the four prediction error vectors through a geometric interpretation. Consider the $(n - q)$-dimensional Hilbert space $H = \{C^{n-q}, \langle x, y \rangle = y^H x\}$. Note that the vectors x_n, $S^m x_n$, y_n and $S^m y_n$ and the columns of $X_{m,n}$ and $Y_{m,n}$ are all elements of H. The span of the columns of $X_{m,n}$ is a subspace of H which we denote by M_X. Similarly, M_Y is the subspace spanned by the m columns of $Y_{m,n}$.

Let us now consider the forward prediction of x_n. From (17) we see that the $\hat{x}_{m,n}$ is formed by operating on x_n by the matrix P_{XY}. P_{XY} is a linear operator that maps elements of H onto the subspace M_X. Also, it is evident from (19a) that $P^2_{XY} = P_{XY}$, so that P_{XY} is a projection operator onto the subspace M_X. P_{XY}

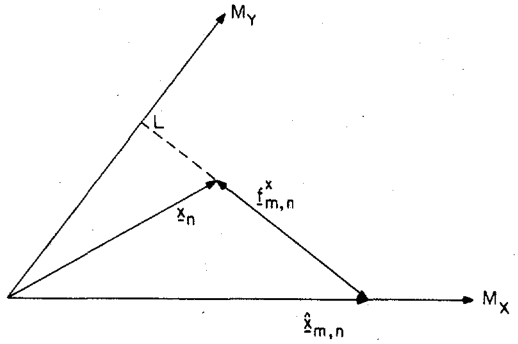


Fig. 1. Geometric relationship between x_n, $\hat{x}_{m,n}$, and $f^x_{m,n}$.

is not in general the orthogonal projection operator onto M_X, but instead projects in a direction orthogonal to M_Y. The relationship between $\boldsymbol{x}_n$, $\hat{\boldsymbol{x}}_{m,n}$, and $\boldsymbol{f}^x_{m,n}$ is depicted in Fig. 1, and the corresponding relationships for $\boldsymbol{b}^x_{m,n-1}$, $\boldsymbol{f}^y_{m,n}$, and $\boldsymbol{b}^y_{m,n-1}$ are similar.

Four scalars that will appear in the recursive algorithm are inner products of these prediction error vectors

$$\sigma_{m,n} = [\boldsymbol{b}^y_{m,n-1}]^H[\boldsymbol{f}^x_{m,n}] = [S^{m+1}\boldsymbol{y}_n]^H P^c_{XY}[\boldsymbol{x}_n] \tag{29}$$

$$\tau_{m,n} = [\boldsymbol{f}^y_{m,n}]^H[\boldsymbol{b}^x_{m,n-1}] = [\boldsymbol{y}_n]^H P^c_{XY}[S^{m+1}\boldsymbol{x}_n] \tag{30}$$

$$\mu_{m,n} = [\boldsymbol{f}^y_{m,n}]^H[\boldsymbol{f}^x_{m,n}] = [\boldsymbol{y}_n]^H P^c_{XY}[\boldsymbol{x}_n] \tag{31}$$

$$\omega_{m,n-1} = [\boldsymbol{b}^y_{m,n-1}]^H[\boldsymbol{b}^x_{m,n-1}] = [S^{m+1}\boldsymbol{y}_n]^H P^c_{XY}[S^{m+1}\boldsymbol{x}_n]. \tag{32}$$

In addition, a fifth scalar,

$$\gamma_{m,n} = [\boldsymbol{e}_n]^H P^c_{XY}[\boldsymbol{e}_n]. \tag{33}$$

In (29)–(33) we used the relations

$$[P^c_{XY}]^2 = P^c_{XY} \tag{34a}$$

$$[P^c_{YX}]^2 = P^c_{YX} \tag{34b}$$

which follow by noting that P^c_{XY} and P^c_{YX} are complements of projection operators and therefore are projection operators themselves.

The strategy behind the fast recursive AR estimator is to determine $f^x_{m,n}(n)$ for $m = 1, 2, \cdots, p$ and for each n. This is accomplished by finding order update equations for this prediction error. These update equations require that the other prediction error vectors, as well as σ, τ, μ, ω, and γ, be updated. Since all of these quantities can be defined in terms of P^c_{XY}, all necessary update equations can be obtained using a single update formula for P^c_{XY}. Such a formula is given in the following theorem.

Theorem 1 (Projection Operator Update Theorem): Let A and B be $n \times m$ matrices, and $\boldsymbol{a}$ and $\boldsymbol{b}$ be $n \times 1$ vectors. Define the augmented matrices $C = [A : \boldsymbol{a}]$ and $D = [B : \boldsymbol{b}]$. If $[B^H A]^{-1}$ and $[D^H C]^{-1}$ exist, then

$$P^c_{CD} = P^c_{AB} - P^c_{AB}\boldsymbol{a}[\boldsymbol{b}^H P^c_{AB}\boldsymbol{a}]^{-1}\boldsymbol{b}^H P^c_{AB} \tag{35}$$

where P^c_{AB} and P^c_{CD} are defined as in (19). The proof of this theorem is given in Appendix A.

It follows from (35) that for any vectors $\boldsymbol{c}$ and $\boldsymbol{d}$,

$$\boldsymbol{c}^H P^c_{CD}\boldsymbol{d} = \boldsymbol{c}^H P^c_{AB}\boldsymbol{d} - \frac{(\boldsymbol{c}^H P^c_{AB}\boldsymbol{a})(\boldsymbol{b}^H P^c_{AB}\boldsymbol{d})}{(\boldsymbol{b}^H P^c_{AB}\boldsymbol{a})}. \tag{36}$$

All necessary update equations may now be derived. Since only the last elements of prediction error vectors are needed, we will for notational convenience drop the "(n)," and use for example $f^x_{m,n}$ (without the boldface), to denote a last element of a prediction error vector. All necessary update equations are obtained by using $A = X_{m,n}$ and $B = Y_{m,n}$ in (36) with appropriate vectors for $\boldsymbol{a}$, $\boldsymbol{b}$, $\boldsymbol{c}$, and $\boldsymbol{d}$. The update equations are given below, and Table I outlines the particular vectors $\boldsymbol{a}$, $\boldsymbol{b}$, $\boldsymbol{c}$, and $\boldsymbol{d}$, used in (36) to derive them:

$$f^x_{m+1,n} = f^x_{m,n} - [b^x_{m,n-1}][\sigma_{m,n}]/[\omega_{m,n-1}] \tag{37}$$

$$f^y_{m+1,n} = f^y_{m,n} - [b^y_{m,n-1}][\tau_{m,n}]^*/[\omega_{m,n-1}]^* \tag{38}$$

$$b^x_{m+1,n} = b^x_{m,n-1} - [f^x_{m,n}][\tau_{m,n}]/[\mu_{m,n}] \tag{39}$$

$$b^y_{m+1,n} = b^y_{m,n-1} - [f^y_{m,n}][\sigma_{m,n}]^*/[\mu_{m,n}]^* \tag{40}$$

$$\mu_{m+1,n} = \mu_{m,n} - [\sigma_{m,n}][\tau_{m,n}]/[\omega_{m,n-1}] \tag{41}$$

$$\omega_{m+1,n} = \omega_{m,n-1} - [\sigma_{m,n}][\tau_{m,n}]/[\mu_{m,n}] \tag{42}$$

$$\sigma_{m,n} = \sigma_{m,n-1} + [b^y_{m,n-1}]^*[f^x_{m,n}]/[\gamma_{m,n}] \tag{43}$$

$$\tau_{m,n} = \tau_{m,n-1} + [f^y_{m,n}]^*[b^x_{m,n-1}]/[\gamma_{m,n}] \tag{44}$$

$$\mu_{m,n} = \mu_{m,n-1} + [f^y_{m,n}]^*[f^x_{m,n}]/[\gamma_{m,n}] \tag{45}$$

$$\omega_{m,n-1} = \omega_{m,n-2} + [b^y_{m,n-1}]^*[b^x_{m,n-1}]/[\gamma_{m,n}] \tag{46}$$

$$\gamma_{m+1,n} = \gamma_{m,n} - [b^y_{m,n-1}]^*[b^x_{m,n-1}]/[\omega_{m,n-1}] \tag{47}$$

$$\gamma_{m+1,n+1} = \gamma_{m,n} - [f^y_{m,n}]^*[f^x_{m,n}]/[\mu_{m,n}]. \tag{48}$$

The initial conditions for the updated parameters are obtained by considering their defining equations.

From (13), (22), (27), and (28) it can be seen that

$$f^x_{0,n} = b^x_{0,n} = x(n) \tag{49a}$$

$$f^y_{0,n} = b^y_{0,n} = y(n) = x(n-q). \tag{49b}$$

Also, since $\boldsymbol{y}_n = \boldsymbol{0}$ for $n \leq q$, it follows that

$$\sigma_{0,n} = \tau_{0,n} = \mu_{0,n} = \omega_{0,n} = 0 \quad \text{for } n \leq q. \tag{50}$$

Moreover, $\boldsymbol{f}^x_{m,q+m}$ and $\boldsymbol{b}^x_{m,q+m-1}$ are zero vectors since equations (13) and (22) are not overdetermined at that point. Since $\sigma_{m,p+m}$ and $\tau_{m,q+m}$ are formed by inner products involving these vectors, it follows that

$$\sigma_{m,q+m} = \tau_{m,q+m} = 0 \qquad m = 0, 1, \cdots, p-1. \tag{51}$$

Finally, for $m = 0$, we have

$$\gamma_{0,n} = 1. \tag{52}$$

The implementation of the algorithm is summarized in Table II.

The recursive algorithm may be implemented using the lattice filter shown in Fig. 2. It is seen here that the prediction

TABLE I
VECTORS a, b, c, AND d, USED IN (36) TO DERIVE THE UPDATE EQUATIONS

Equation	a	b	c	d
(37)	$S^{m+1}x_n$	$S^{m+1}y_n$	e_n	x_n
(38)	$S^{m+1}x_n$	$S^{m+1}y_n$	y_n	e_n
(39)	x_n	y_n	e_n	$S^{m+1}x_n$
(40)	x_n	y_n	$S^{m+1}y_n$	e_n
(41)	$S^{m+1}x_n$	$S^{m+1}y_n$	y_n	x_n
(42)	x_n	y_n	$S^{m+1}y_n$	$S^{m+1}x_n$
(43)	e_n	e_n	$S^{m+1}y_n$	x_n
(44)	e_n	e_n	y_n	$S^{m+1}x_n$
(45)	e_n	e_n	y_n	x_n
(46)	e_n	e_n	$S^{m+1}y_n$	$S^{m+1}x_n$
(47)	$S^{m+1}x_n$	$S^{m+1}y_n$	e_n	e_n
(48)	x_n	y_n	e_n	e_n

TABLE II
THE FAST RECURSIVE AR COEFFICIENT ESTIMATION ALGORITHM

```
1) At time $n = q$ set $\sigma_{m,n} = \tau_{m,n} = 0$, $m = 0, 1, \cdots, p - 1$.
2) Set:
       $f^x_{0,n} = b^x_{0,n} = x(n)$
       $f^y_{0,n} = b^y_{0,n} = y(n) = x(n-q)$
       $\gamma_{0,n} = 1$.
3) Set the maximum filter order: $p_{\max} = \min [p, n - q - 2]$. For each $m = 0, 1, \cdots, p_{\max} - 1$,
     a) Update $\sigma_{m,n}, \tau_{m,n}, \mu_{m,n}, \omega_{m,n}$ using (43)-(46)
     b) Update to $f^x_{m+1,n}, b^x_{m+1,n}, f^y_{m+1,n}, b^y_{m+1,n}$ using (37)-(40)
     c) Update to $\gamma_{m+1,n}$ using (47).
4) If $p_{max} < p$, add a filter order:
     a) Set $m = p_{\max}$
     b) Update to $\sigma_{m,n}, \tau_{m,n}$ using (43), (44)
     c) Update to $\mu_{m+1,n}, \omega_{m+1,n}$ using (41), (42)
     d) Update $f^x_{m+1,n}, b^x_{m+1,n}, f^y_{m+1,n}, b^y_{m+1,n}$ using (37)-(40).
5) Set $n = n + 1$ and go to Step 2.
```

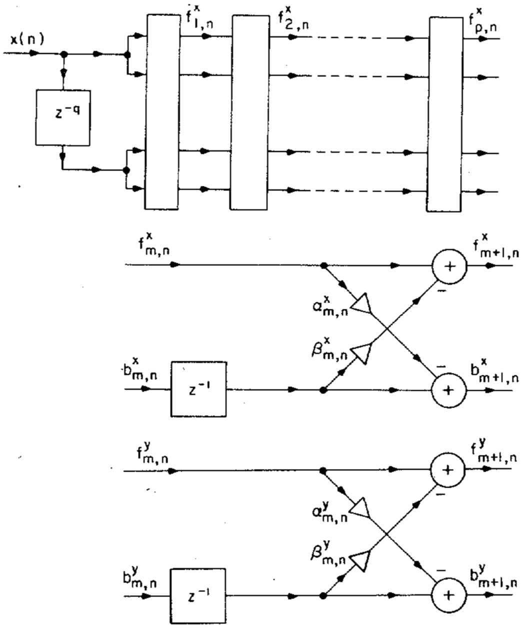


Fig. 2. Lattice filter for AR coefficient estimation.

errors are represented by signals propagating through the filter and the filter multipliers (often called reflection coefficients or partial correlation coefficients) are given by

$$\alpha^x_{m,n} = \tau_{m,n}/\mu_{m,n} \tag{53}$$

$$\beta^x_{m,n} = \sigma_{m,n}/\omega_{m,n-1} \tag{54}$$

$$\alpha^y_{m,n} = \sigma^*_{m,n}/\mu^*_{m,n} \tag{55}$$

$$\beta^y_{m,n} = \tau^*_{m,n}/\omega^*_{m,n-1}. \tag{56}$$

The conversion from reflection coefficients to autoregressive coefficients proceeds as follows. Define $A_{m,n}(z)$ as the transfer function at time n from $x(n)$ to $f^x_{m,n}$, and $\tilde{A}_{m,n}(z)$ as the transfer function at time n from $x(n)$ to $b^x_{m,n}$. Then

$$A_{0,n}(z) = \tilde{A}_{0,n}(z) = 1. \tag{57}$$

By taking z-transforms of (37) and (39) and by noting that $F^x_{m,n}(z) = A_{m,n}(z)$ and $B^x_{m,n}(z) = \tilde{A}_{m,n}(z)$, we find

$$A_{m,n}(z) = A_{m-1,n}(z) + z^{-1}\beta^x_{m-1,n}\tilde{A}_{m-1,n-1}(z) \tag{58a}$$

$$\tilde{A}_{m,n}(z) = \tilde{A}_{m-1,n-1}(z) + z^{-1}\alpha^x_{m-1,n}A_{m-1,n}(z). \tag{58b}$$

Given the α^x and β^x coefficients, (58) can be used to recursively compute $A_{m,n}(z)$ and $\tilde{A}_{m,n}(z)$ for $m = 1, 2, \cdots, p$. In this way we obtain autoregressive coefficient estimates for all autoregressive model orders $1, 2, \cdots, p$.

The number of computations required per update is of major importance in real time applications of recursive algorithms. From Table II it can be checked that $9p$ multiplies, $9p$ divides, and $9p$ adds are required to update the lattice algorithm. To convert from reflection coefficients to AR coefficients via (58) requires an additional $p^2 + p$ adds and $p^2 + p$ multiplies.

As a final note, when $q = 0$ (i.e., when an autoregressive model is chosen) the top and bottom halves of the filter in Fig. 2 become equivalent. If the redundancy is eliminated, this filter degenerates to the AR lattice filter described in [13].

Exponential Weighting of Time Updates

Many applications of recursive estimation algorithms require the ability to track time variations in the estimated coefficients. Such a tracking ability may be incorporated into this algorithm with only a minor increase in computations through the use of an exponential forgetting factor. In this case, data lags k times in the past are attenuated by λ^k for some $0 < \lambda \leq 1$. To effect this, R_N and r_N in (10) are replaced by

$$R_N = Y^H_{p,N}\Lambda X_{m,n} \tag{59a}$$

$$r_N = Y^H_{p,N}\Lambda x_N \tag{59b}$$

where $\Lambda = \text{diag}\,[\lambda^{n-q-1} \cdots \lambda\ 1]$. It should be noted that if $\lambda < 1$, the variance of $[a - a_{p,N}]$ no longer approaches zero as $N \to \infty$, so $a_{p,N}$ is no longer a consistent estimate of a.

Only the time update recursive formulas are modified by the addition of this exponential forgetting factor. Specifically, (43)-(46), are changed to

$$\sigma_{m,n} = \lambda\sigma_{m,n-1} + [b^y_{m,n-1}]^*[f^x_{m,n}]/[\gamma_{m,n}] \tag{60}$$

$$\tau_{m,n} = \lambda\tau_{m,n-1} + [f^y_{m,n}]^*[b^x_{m,n-1}]/[\gamma_{m,n}] \tag{61}$$

$$\mu_{m,n} = \lambda\mu_{m,n-1} + [f^y_{m,n}]^*[f^x_{m,n}]/[\gamma_{m,n}] \tag{62}$$

$$\omega_{m,n-1} = \lambda\omega_{m,n-2} + [b^y_{m,n-1}]^*[b^x_{m,n-1}]/[\gamma_{m,n}]. \tag{63}$$

The modified recursive algorithm then requires $2p + 1$ more multiplies per time update with the addition of the forgetting factor.

IV. Recursive Numerator Estimation

Now that a fast recursive AR coefficient estimator has been developed, there remains the problem of recursively estimating the numerator spectral coefficients in (5). To motivate the derivation of this estimator, define the filtered data sequence

$$f(n) = \sum_{i=0}^{p} a_i x(n-i). \tag{64}$$

From (5) and (64) it can be seen that $C(e^{j\omega})$ is the power spectral density function for $\{f(n)\}$, so that

$$c_k = E\{f(n+k)f^*(n)\}. \tag{65}$$

We can estimate c_k by approximating the expectation in (65) by a time average. Since exact AR coefficients are not known, $f(n)$ must also be estimated. But from (14) we see that $f^x_{p,n}$ is an estimate of $f(n)$, and this estimate is already computed in the recursive AR coefficient estimator. Thus, an estimate of c_k is given by

$$c_k(n) = \begin{cases} w_k/[n-q-k] \displaystyle\sum_{m=q+1}^{n-k} \lambda^{n-k-m}[f^x_{p,m+k}][f^x_{p,m}]^*, \\ \quad 0 \le k \le q \\ c^*_{-k}, \qquad -q \le k \le -1 \end{cases} \tag{66}$$

where λ is the exponential forgetting factor. The corresponding numerator power spectral density estimate is

$$\hat{C}(e^{j\omega}) = \sum_{k=-q}^{q} c_k(n)e^{-j\omega k}. \tag{67}$$

Note that this estimator is suboptimal in the sense that $f^x_{p,m}$ implicitly uses the AR coefficients based on m rather than n data points. Heuristically we expect $f^x_{p,m}$ to be less accurate for smaller m, and may, therefore, want to weight them accordingly. This can be realized by using $\lambda < 1$ (and possibly different from λ used in the AR parameter estimation). The authors found that, even from stationary process realizations, using $\lambda < 1$ in (66) often provides more accurate NS coefficient estimates than using $\lambda = 1$.

The windowing function w_k is incorporated to provide the ability of ensuring that the estimated $c_k(n)$ sequence corresponds to a nonnegative spectrum estimate. One possible choice for w_k is the linear tapered window [20]

$$w_k = \begin{cases} 1 - \alpha k/[q+1], & -q \le k \le q \\ 0, & \text{otherwise.} \end{cases}$$

If $\alpha = 0$, no windowing is used. If $\alpha = 1$, then w_k is a full triangular window, and nonnegative numerator spectrum estimates from (66) and (67) are guaranteed. However, when $\alpha = 1$, a rather severe bias is imposed on the numerator spectral estimates. In many instances, $\alpha \ll 1$ may be chosen such that the numerator spectral estimate is nonnegative, and also less biased than when $\alpha = 1$ [20].

Equation (66) can be recursively updated by

$$c_k(n) = \{\lambda(n-1-q-k)c_k(n-1) \\ + [f^x_{p,n}][f^x_{p,n-k}]^* w_k\}/(n-q-k). \tag{69}$$

Alternatively, $\tilde{c}_k(n) \triangleq (n - q - k)c_k(n)$ can be updated instead, leading to the simpler update formula:

$$\tilde{c}_k(n) = \lambda\tilde{c}_k(n-1) + [f^x_{p,n}][f^x_{p,n-k}]^* w_k. \tag{70}$$

The recursive updates in (69) require $q + 1$ adds, $4(q + 1)$ multiplies, and $q + 1$ divides per update. If (70) is used instead, only $q + 1$ adds and $3(q + 1)$ multiplies are needed. If $\lambda = 1$ or if $w_k = 1$, the number of multiplies decreases accordingly. In any case, this recursive numerator algorithm requires $O(q)$ computations per update.

It should be noted that one reason this numerator spectral coefficient estimator requires so few computations per update is that the MA coefficients are not explicitly estimated. In order to derive MA coefficient estimates from the NS coefficient estimates, a spectral factorization of $\hat{C}(e^{j\omega})$ must be performed.

V. Numerical Example

In order to provide an indication of the effectiveness of the recursive ARMA algorithm, we present two examples. The first example is the spectral estimation of the ARMA (4, 2) process with

$$\begin{array}{llll} a_1 = & 0.4424 & & b_0 = 1. \\ a_2 = & -0.0203 & & b_1 = 0. \\ a_3 = & 0.4164 & & b_2 = 1. \\ a_4 = & 0.8853 & & \end{array}$$

Autoregressive and numerator spectral coefficients were recursively obtained using the algorithm presented herein with $p = 4$, $q = 2$, $\lambda = 1$ in the AR parameter estimator, $\lambda = 0.96$ in the NS parameter estimator, and $\alpha = 0.1$. Spectral estimates corresponding to the coefficient estimates at 100 data point intervals are shown in Fig. 3(a). Fig. 3(b) gives similar estimates obtained using the recursive AR estimator in [13] with $p = 18$. We see that the ARMA spectral estimates are much closer to the exact spectrum than the AR spectral estimates. Moreover, 120 computations per update are needed in the ARMA method, whereas 324 computations per update are needed for the AR estimator. Other recursive ARMA lattice algorithms [15], [21], [22] require 3 to 5 times the number of computations per update required by this ARMA method.

The second example is concerned with tracking the instantaneous frequency of an FM modulated signal in noise. The time

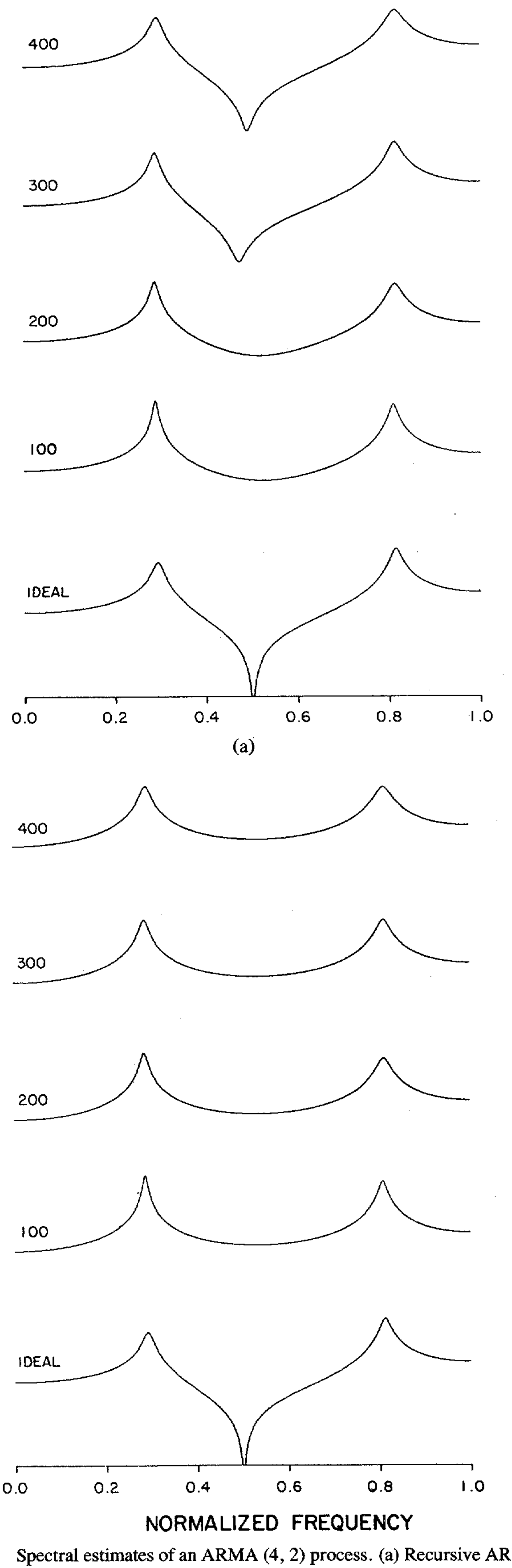


Fig. 3. Spectral estimates of an ARMA (4, 2) process. (a) Recursive ARMA (4, 2) method. (b) Recursive AR (18) method.

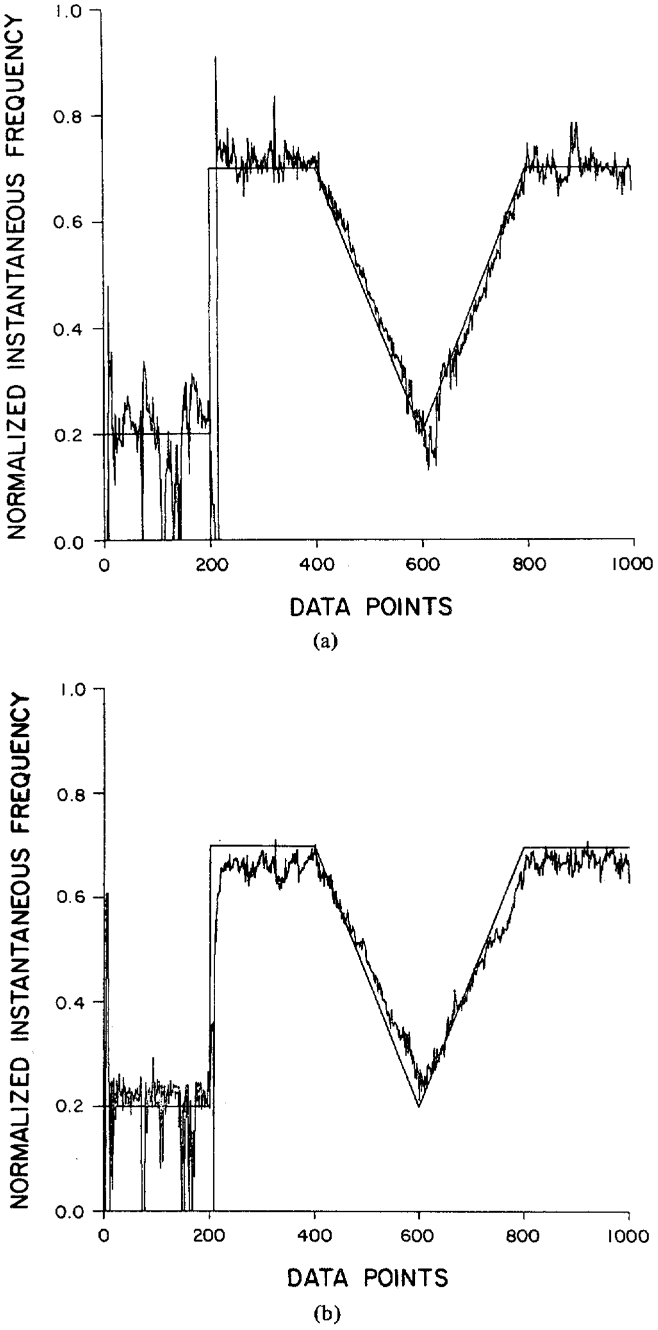


Fig. 4. Frequency tracking example. (a) Estimate from ARMA lattice. (b) Estimates from AR lattice.

series is a single sinusoid with amplitude of 3.2 embedded in additive unit variance white noise. The recursive ARMA algorithm with $p = q = 2$, and the recursive AR algorithm with $p = 2$ were used to estimate the instantaneous frequency of this sinusoid. Fig. 4 shows these estimates for $\lambda = 0.9$ (the true instantaneous frequency is also shown there). Of the two methods, the ARMA method exhibits less bias in the frequency estimates and more accurately estimates the frequency slew rate.

VI. Conclusions

We have presented a computationally fast recursive algorithm for estimating the parameters of a power spectral density parametrization associated with an ARMA time series model. The autoregressive coefficient estimates are based on the solution of a set of extended Yule–Walker equations. These estimates are under appropriate conditions asymptotically unbiased and consistent. A recursive lattice filter implementation of this coefficient estimator is derived which requires $O(p)$ computations per time update. Moreover, an exponential forgetting factor can be implemented which facilitates the tracking of variations in the data. The numerator spectral coefficients are then estimated from a forward prediction error sequence generated by the AR lattice. The latter are updated recursively in $O(q)$ computations, which can also be implemented with an exponential forgetting factor.

The recursive ARMA algorithm requires about $9p + q$ adds, $9p + 3q$ multiplies, and $9p$ divides per time update; a number that compares favorably with other ARMA lattice algorithms. In the present ARMA spectral estimator however, moving average coefficient estimates are not obtained explicitly (some sort of spectral factorization must be performed to get them). For applications in which the MA coefficients perse are not needed, this algorithm can be both efficient and effective.

We have not yet discussed the convergence properties of the c_k estimates. It can be shown that when $\lambda = 1$, and under conditions similar to those needed for AR coefficient convergence, each $c_k(n)$ coefficient converges to $w_k c_k$ (as $n \to \infty$) with probability 1. Asymptotic normality of $\sqrt{n}[w_k c_k - c_k(n)]$ can also be established. Moreover, the $c_k(n)$ coefficients may converge faster if $\lambda < 1$ is used in the NS estimator until the AR coefficient estimates become reasonably close to their convergence points. A paper detailing these results is currently in preparation.

Appendix A

Proof of Theorem 1

Define M_A as the subspace of the Hilbert space $\{C^n, \langle x, y\rangle = x^H y\}$ spanned by the columns of A. Similarly define M_B, M_C, and M_D. Note that since $[B^H A]^{-1}$ and $[D^H C]^{-1}$ exist, M_A and M_B have dimension p and M_C and M_D have dimension $p + 1$. Also define $M_A^\perp$, $M_B^\perp$, $M_C^\perp$, and $M_D^\perp$ as the corresponding orthogonal complements. The vector $\boldsymbol{a}$ can be split into a component in M_A and a component in $M_B^\perp$. This second component along with the columns of A spans M_C:

$$M_C = M_A \oplus \text{span } \{P^c_{AB}\boldsymbol{a}\} = M_A \oplus M_x \tag{A1}$$

where

$$\boldsymbol{x} = P^c_{AB}\boldsymbol{a} \tag{A2}$$

Analogously,

$$M_D = M_B \oplus \text{span } \{P^c_{BA}\boldsymbol{b}\} = M_B \oplus M_y \tag{A3}$$

where

$$\boldsymbol{y} = P^c_{BA}\boldsymbol{b}. \tag{A4}$$

Since P_{AB} maps orthogonally to M_B, P^c_{AB} maps onto $M_B^\perp$. Therefore, $\boldsymbol{x} \in M_B^\perp$. Similarly, $\boldsymbol{y} \in M_A^\perp$. In fact, for any suitable matrices A and B, the projection operator P_{AB} defines a decomposition of the Hilbert space C^n into two (not necessarily orthogonal) subspaces M_A and $M_B^\perp$. Therefore, if $\boldsymbol{z}$ is any vector in C^n,

$$\boldsymbol{z} = P_{AB}\boldsymbol{z} + P^c_{AB}\boldsymbol{z} \tag{A5}$$

where $P_{AB}\boldsymbol{z} \in M_A$ and $P^c_{AB}\boldsymbol{z} \in M_B$.

Equation (35) is established by first showing that

$$P_{CD} = P_{AB} + P_{xy} \tag{A6}$$

and by then performing some simple algebraic manipulations on (A6). To show (A6), note that P_{CD} defines the decomposition of C^n into two subspaces as follows.

$$C^n = M_C \oplus M_D^\perp \tag{A7}$$

By using equation (A1) we obtain

$$C^n = M_A \oplus M_x \oplus M_D^\perp. \tag{A8}$$

Let $\boldsymbol{z} = \boldsymbol{f} + \boldsymbol{g} + \boldsymbol{h}$, where $\boldsymbol{f} \in M_A$, $\boldsymbol{g} \in M_x$ and $\boldsymbol{h} \in M_D^\perp$. Since $(\boldsymbol{f} + \boldsymbol{g}) \in M_C$ and $\boldsymbol{h} \in M_D^\perp$, it follows that

$$\boldsymbol{f} + \boldsymbol{g} = P_{CD}\boldsymbol{z}. \tag{A9}$$

We claim that $\boldsymbol{f} = P_{AB}\boldsymbol{z}$ and $\boldsymbol{g} = P_{xy}\boldsymbol{z}$. To show $\boldsymbol{f} = P_{AB}\boldsymbol{z}$, we must show $P_{AB}(\boldsymbol{g} + \boldsymbol{h}) = \boldsymbol{0}$. Since P_{AB} projects onto M_A along $M_B^\perp$, and $g \in M_B^\perp$, it follows that $P_{AB}\boldsymbol{g} = \boldsymbol{0}$. Also, $M_B \subseteq M_D$, so $M_D^\perp \subseteq M_B^\perp$. Since $\boldsymbol{h} \in M_D^\perp$, $\boldsymbol{h} \in M_B^\perp$, so $P_{AB}\boldsymbol{h} = \boldsymbol{0}$. Therefore, $\boldsymbol{f} = P_{AB}\boldsymbol{z}$.

To show $\boldsymbol{g} = P_{xy}\boldsymbol{z}$, we must show $P_{xy}(\boldsymbol{f} + \boldsymbol{h}) = \boldsymbol{0}$. Recall that P_{xy} projects onto M_x along $M_y^\perp$. Since $\boldsymbol{y} \in M_D$, $M_D^\perp \subseteq M_y^\perp$, so $P_{xy}\boldsymbol{h} = \boldsymbol{0}$. Moreover, $\boldsymbol{y} \in M_A^\perp$, so $M_A \subseteq M_y^\perp$ and $P_{xy}\boldsymbol{f} = \boldsymbol{0}$. Therefore, $\boldsymbol{g} = P_{xy}\boldsymbol{z}$. Equation (A9) becomes

$$\boldsymbol{f} + \boldsymbol{g} = P_{CD}\boldsymbol{z} = P_{AB}\boldsymbol{z} + P_{xy}\boldsymbol{z} \tag{A10}$$

Since $\boldsymbol{z}$ is arbitrary, (A6) follows immediately.

Now, since $\boldsymbol{x} \in M_B^\perp$, $P_{AB}P_{xy} = 0$. Therefore,

$$\begin{aligned} P^c_{CD} &= I - P_{CD} \\ &= I - P_{AB} - P_{xy} \\ &= (I - P_{AB})(I - P_{xy}) \\ &= P^c_{AB}(I - P_{xy}). \end{aligned} \tag{A11}$$

Writing out P_{xy} and substituting from equations (A2) and (A4) yields

$$P^c_{CD} = P^c_{AB} - P^c_{AB}(P^c_{AB}\boldsymbol{a})[(P^c_{BA}\boldsymbol{b})^H(P^c_{AB}\boldsymbol{a})]^{-1}(P^c_{BA}\boldsymbol{b})^H. \tag{A12}$$

Finally, by applying the properties $[P^c_{BA}]^H = P^c_{AB}$ and $(P^c_{AB})^2 = P^c_{AB}$ to (A12), (35) is obtained. This completes the proof.

References

[1] L. M. Koopmans, *The Spectral Analysis of Time Series.* New York: Academic, 1962.

[2] G. Box and G. Jenkins, *Time Series Analysis: Forecasting and Control.* San Francisco, CA: Holden-Day, 1976.

[3] J. A. Cadzow, "ARMA spectral estimation: An efficient closed-form

procedure," in *Proc. RADC Spectrum Estimation Workshop*, Oct. 1979, pp. 81-97.

[4] ——, "High performance spectral estimation—A new ARMA method," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-28, pp. 524-529, Oct. 1980.

[5] ——, "Spectral estimation: An overdetermined rational model equation approach," in *Proc. IEEE*, vol. 70, pp. 907-939, Sept. 1982.

[6] W. Gersh, "Estimation of the autoregressive parameters of mixed autoregressive moving-average time series," *IEEE Trans. Automat. Contr.*, vol. AC-15, pp. 583-588, Oct. 1970.

[7] D. Graupe, D. J. Krause, and J. B. Moore, "Identification of autoregressive moving-average parameters of time series," *IEEE Trans. Automat. Contr.*, vol. AC-20, pp. 104-107, Feb. 1975.

[8] J. F. Kinkel, J. Perl, L. Scharf, and A. Stubberud, "A note on covariance-invariant digital filter design and autoregressive-moving average spectral estimation," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-27, pp. 200-202, Apr. 1979.

[9] M. Kaveh, "High resolution spectral estimation for noisy signals," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-27, pp. 286-287, June 1979.

[10] P. R. Gutowski, E. A. Robinson, and S. Treitel, "Spectral estimation: Fact or fiction," *IEEE Trans. Geosci. Electron.*, vol. GE-16, pp. 80-84, Apr. 1966.

[11] E. Satorius and S. Alexander, "Channel equalization using adaptive lattice algorithms," *IEEE Trans. Commun.*, vol. COM-27, pp. 899-905, June 1979.

[12] D. Falconer and L. Ljung, "Application of fast Kalman estimation," *IEEE Trans. Commun.*, vol. COM-26, pp. 1439-1446, Oct. 1976.

[13] D. Lee and M. Morf, "Recursive square root ladder estimation algorithms," in *Proc. 1980 IEEE Int. Conf. Acoust., Speech, Signal Processing*, Denver, CO, pp. 1005-1017.

[14] B. Friedlander, "Recursive algorithms for ladder forms," SCI Tech. Mem., 1980.

[15] ——, "Recursive algorithms for pole-zero ladder forms," SCI Tech. Mem., 1980.

[16] J. A. Cadzow and R. L. Moses, "An adaptive ARMA spectral estimator," in *Proc. Spectral Estimation Workshop*, Hamilton, Ont., Aug. 17-18, 1981.

[17] T. Kailath, S. Y. Kung, and M. Morf, "Displacement ranks of matrices and linear equations," *J. Math. Anal. Appl.*, vol. 68, pp. 395-407, Apr. 1979.

[18] B. Friedlander, M. Morf, T. Kailath, and L. Ljung, "New inversion formulas for matrices classified in terms of their distance from Toeplitz matrices," *Lin. Alg. Appl.*, vol. 27, pp. 31-60, 1979.

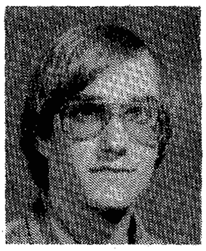
[19] M. Morf, B. Dickinson, T. Kailath, and A. Vieira, "Efficient solution of covariance equations for linear prediction," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-25, pp. 429-433, Oct. 1977.

[20] S. Bruzzone and M. Kaveh, "On some suboptimum ARMA spectral estimators," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-28, pp. 753-755, Dec. 1980.

[21] B. Friedlander, "Lattice filters for adaptive procedures," *Proc. IEEE*, vol. 70, no. 8, pp. 829-867, Aug. 1982.

[22] B. Friedlander and S. Maitra, "Speech deconvolution by recursive ARMA lattice filters," in *Proc. 1981 Int. Conf. Acoust., Speech, Signal Processing*, Atlanta, GA, Mar. 1981, pp. 343-346.

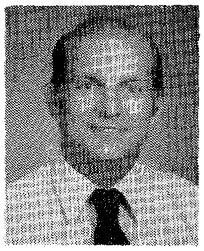
[23] T. Söderström and P. Stoica, "Comparison of some instrumental variable methods—Consistency and accuracy aspects," *Automatica*, vol. 17, pp. 101-105, 1981.



Randolph L. Moses received the B.S., M.S., and Ph.D. degrees in electrical engineering from Virginia Polytechnic Institute and State University, Blacksburg, in 1979, 1980, and 1984, respectively.

In Summer 1983 he was a SCEEE Summer Faculty Research Fellow at Rome Air Development Center, and in Summer 1984 he served as Visiting Assistant Professor at VPI & SU. Since August 1984 he has been with the Eindhoven University of Technology, The Netherlands as a NATO Postdoctoral Fellow. His current research interests include parametric time series analysis, system identification, and model reduction.

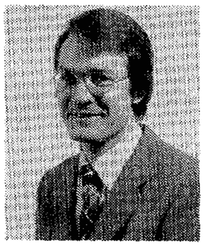
Dr. Moses is a member of Eta Kappa Nu, Tau Beta Pi, Phi Kappa Phi, and Sigma Xi.



James A. Cadzow (S'57-M'60-SM'75) was born in Niagara Falls, NY, on January 3, 1936. He received the B.S. and M.S. degrees in electrical engineering from the State University of New York, Buffalo, in 1958 and 1963, respectively, and the Ph.D. degree from Cornell University, Ithaca, NY in 1964.

From 1958 to 1963 he was associated with the USARDL, Fort Monmouth, NJ, Bell Aerosystems, Buffalo, NY, and Cornell Aeronautical Laboratories, Buffalo, NY. He was Professor of Electrical Engineering at S.U.N.Y. Buffalo from 1964 to 1977 and at Virginia Polytechnic Institute, Blacksburg, from 1977 to 1981. In 1981, he joined Arizona State University, Tempe, where he is presently a Research Professor. He served as a Visiting Professor of Electrical Engineering at Stanford University, Stanford, CA, from 1968 to 1969, and was a Visiting Professor and National Institutes of Health Fellow at the Department of Biomedical Engineering, Duke University, Durham, NC during 1972 and 1973. He is the author of *Discrete-Time Systems* (Englewood Cliffs, NJ: Prentice-Hall, 1973) and coauthor of the texts *Discrete-Time and Computer Control Systems* (Englewood Cliffs, NJ: Prentice-Hall, 1970) and *System, Signals and Transforms* (Prentice-Hall, 1985). His research interests are in communication and control theory, digital signal processing, and system modeling.

Dr. Cadzow is a member of Sigma Xi and Phi Kappa Phi.



A. A. (Louis) Beex was born in Veldhoven, The Netherlands, in 1949. He received the "Ingenieur" degree in 1974 from Eindhoven University of Technology, Eindhoven, The Netherlands, and the Ph.D. degree in 1979 from Colorado State University, Fort Collins, both are in electrical engineering.

From 1976 to 1978 he was a Staff Research Engineer at Starkey Laboratories, Minneapolis, MN, working on digital signal processing applications. Since 1979 he has been a faculty member in the Department of Electrical Engineering at Virginia Polytechnic Institute and State University, Blacksburg, VA, currently as Associate Professor. He has served as a Consultant to Colorado State University, Fort Collins, the Naval Surface Weapons Center, Dahlgren, VA, and Rome Air Development Center, Rome, NY. His activities and interests are covered by stochastic/digital signal processing and system identification.