AD-A101 870 VIRGINIA POLYTECHNIC INST AND STATE UNIV BLACKSBURG D--ETC F/G 12/1
AN ADAPTIVE ARMA SPECTRAL ESTIMATOR, PART 2.(U)
JUN 81 J A CADZOW, R L MOSES N00014-80-C-0303
UNCLASSIFIED 10 NL

END
DATE FILMED
8-81
DTIC

AN ADAPTIVE ARMA SPECTRAL ESTIMATOR: PART 2*

James A. Cadzow and Randolph L. Moses
Department of Electrical and Computer Engineering
Arizona State University
Tempe, Arizona 85281

Abstract - This paper presents a recursive algorithmic implementation of the prewindowed high performance method of ARMA spectral modeling as described in Part 1. This algorithm provides updates of the ARMA models optimal autoregressive coefficients (in actuality prediction errors) as each new data point becomes available. The algorithm is computationally fast in the sense that it requires O(p) multiplications and additions for each update. It is shown that this fast recursive algorithm may be implemented using a lattice filter arrangement, and it therefore exhibits several of the "nice" properties associated with lattice type algorithms such as numerical robustness and good convergence properties.

I. INTRODUCTION

In Part 1 of this paper we described an algorithm for obtaining an estimate of the power spectral density associated with a given time series $\{x(n)\}$. In particular, the following ARMA (p,q) spectral density model as hypothesized

$$S_x(\omega) = \left| \frac{b_0 + b_1 e^{-j\omega} + \ldots + b_q e^{-jq\omega}}{1 + a_1 e^{-j\omega} + \ldots + a_p e^{-jp\omega}} \right|^2 \qquad (1)$$

A closed form algorithm for estimating this model's a_i and b_j parameters was then developed in which the finite set of time series observations

$$x(1),\ x(2),\ \ldots,\ x(n) \qquad (2)$$

were used in the parameter selection process. This so called "high performance" algorithm operates on a data block of length n to obtain the model's coefficients in a single computational effort. It is therefore called a block processing algorithm.

There are many situations, however, in which a block processing algorithm for spectral estimation is not an appropriate tool. In a variety of applications, the data measurement is an ongoing process and it is therefore desirable to recursively update the autoregressive and moving average parameters as each new data point becomes available. This capability is of particular importance in those cases where one wishes to adaptively model the spectrum of a long, ongoing time series. Algorithms with this recursive updating capability are called "recursive algorithms".

Recently, several fast recursive spectral estimation algorithms have been developed [1]-[7]. Most of these algorithms are based on so called least-squares AR spectral estimation methods [8], although a few ARMA algorithms have also been developed [4] & [6]. These algorithms seek to minimize a prediction error vector in order to obtain the desired spectral estimates.

In this paper we develop a more effective ARMA algorithm that efficiently computes the optimal autoregressive coefficients by recursively updating a set of prediction error elements as each new data point is observed. This recursive algorithm is based on the prewindowed high performance method as described in Part 1, and, therefore is predicated on the approximation of the ARMA model's underlying Yule-Walker equations. Although similar algorithms may be derived for the unmodified and for the two other modified versions of the high performance method, the recursive algorithm based on the prewindow version is a bit easier to derive and is characterized by a "fast start-up" capability in that spectral estimates are possible with as few as two data points. The recursive algorithm herein presented is computationally efficient in the sense that O(p) multiplications and additions are required to update the "necessary" parameters as each new data point is observed. This paper's recursive algorithm was originally developed in [9]. A more straightforward derivation is herein presented which provides a greater degree of insight. In addition, a lattice filter implementation of this algorithm is developed. Moreover, because of this ladder-type implementation, this algorithm is characterized by several other nice properties associated with ladder algorithms such as numerical robustness and good convergence properties [10], [11].

II. THE PREDICTION ERROR VECTORS

The recursive update equations herein presented do not explicitly update the ARMA model's autoregressive coefficients in obtaining optimal updated spectral estimates. Instead, a set of "equivalent" parameters known as prediction errors are updated. In this section, we discuss the relationship between the prediction errors and autoregressive coefficients.

As outlined in Part 1 of this paper, the optimal p^{th} order set of autoregressive coefficients for the prewindowed version of the high performance method are obtained by solving the following system of p linear equations in p unknowns

$$[Y^+_{n,p}\ X_{n,p}]\underline{a}_p + Y^+_{n,p}\,\underline{x}_n = \underline{\theta} \qquad (3)$$

where

$$\underline{x}_n = [x(1),\ x(2),\ \ldots,\ x(n)]' \qquad (4a)$$

$$\underline{y}_n = S^q \underline{x}_n \qquad (4b)$$

$$X_{n,p} = [S\underline{x}_n : S^2\underline{x}_n : \ldots : S^p\underline{x}_n] \qquad (4c)$$

$$Y_{n,p} = [S\underline{y}_n : S^2\underline{y}_n : \ldots : S^p\underline{y}_n] \qquad (4d)$$

in which $\underline{\theta}$ is the zero vector and S is the down shift operator.[1] Here the dagger symbol (+) denotes complex conjugate transposition and the prime symbol (') denotes transposition. The subscripts p and n explicitly indicate that the denominator order of the spectral model of equation (1) is p and that n data points are available. Whenever this explicit information is not needed, we will use $\underline{x}$, $\underline{y}$, X, and Y in place of $\underline{x}_n$, $\underline{y}_n$, $X_{n,p}$, and $Y_{n,p}$, respectively.

It is recalled from Part 1, that the high performance ARMA modeling approach is predicated on approximating t

*This work was supported in part by the Office of Naval Research, Statistics and Probability Program under Contract N00014-80-C-0303.

[1] In this paper, matrices are denoted by capital letters (e.g. X), vectors are denoted by underlined lower case English letters (e.g. $\underline{x}$) and scalars are denoted by lower case Greek letters (e.g. α). Moreover, the down shift operator S is defined by

$$S\underline{x}_n = [0, x(1), x(2), \ldots, x(n-1)]'$$

Yule-Walker equations where t≥p. Upon examination of expression (3), it is apparent that we have here restricted t = p. This restriction is required in order to facilitate the development of the fast recursive algorithm. Unfortunately, by requiring t = p, the spectral estimation performance suffers in comparison to that achieved with larger values of t. As the data length n increases, however, this performance degradation diminishes and typically is of an insignificant nature. This is indeed fortunate since it is precisely for long data length cases that the recursive algorithm would most likely be utilized.

Under the assumption that $Y^{\dagger}X$ is nonsingular, the optimal autoregressive coefficient vector which satisfies expression (3) is given by

$$\underline{a}_p^{\circ} = -[Y_{n,p}^{\dagger} X_{n,p}]^{-1} Y_{n,p}^{\dagger} \underline{x}_n \tag{5}$$

In what is to follow, it is beneficial to interpret this autocorrelation coefficient selection procedure from a prediction error viewpoint. Namely, we may reformulate expression (3) as

$$Y_{n,p}^{\dagger} \underline{f}_{p,n}^{x} = \underline{\theta} \tag{6}$$

in which $\underline{f}_{p,n}^{x}$ is the so-called forward prediction error as specified by

$$\underline{f}_{p,n}^{x} = \underline{x}_n + X_{n,p} \underline{a}_p \tag{7}$$

It is referred to as the forward prediction error since its k^{th} component can be interpreted as being the error resulting from a prediction of the element x(k) by a linear combination of the p most recent time series elements x(k-1), x(k-2),. . ., x(k-p).

The optimal autoregressive coefficient vector (5) can be then associated with an auxiliary minimization problem involving the prediction error vector. Namely, it is readily shown that this optimal vector minimizes the following quadratic functional

$$g(\underline{a}_p) = \underline{f}_{p,n}^{x\,\dagger} W \underline{f}_{p,n}^{x} \tag{8}$$

where W is the n×n positive semidefinite matrix specified by

$$W = Y_{n,p} Y_{n,p}^{\dagger} \tag{9}$$

To reinforce this prediction error interpretation, let us define the following estimate of vector $\underline{x}_n$

$$\hat{\underline{x}}_n = -X_{n,p} \underline{a}_p \tag{10}$$

which in turn generates the forward predicting error

$$\underline{f}_{p,n}^{x} = \underline{x}_n - \hat{\underline{x}}_n \tag{11}$$

Upon substitution of expression (5) into (10) the optimum forward prediction error vector is given by

$$\hat{\underline{x}}_n^{\circ} = X_{n,p}[Y_{n,p}^{\dagger} X_{n,p}]^{-1} Y_{n,p}^{\dagger} \underline{x}_n$$

$$= P_{XY} \underline{x}_n \tag{12a}$$

while the minimizing forward prediction error for this selection becomes

$$\underline{f}_{p,n}^{x\,\circ} = \left(I - X_{n,p}[Y_{n,p}^{\dagger} X_{n,p}]^{-1} Y_{n,p}^{\dagger}\right) \underline{x}_n \tag{12b}$$

$$= P_{XY}^{c} \underline{x}_n$$

We have here used the compact matrix product representations

$$P_{XY} = X_{n,p}[Y_{n,p}^{\dagger} X_{n,p}] Y_{n,p}^{\dagger} \tag{13a}$$

$$P_{XY}^{c} = I - P_{XY} \tag{13b}$$

Since we are only interested in the optimal $\hat{\underline{x}}_n$ and $\underline{f}_{p,n}^{x}$, we will drop the "°" symbol and assume that $\hat{\underline{x}}_n$ and $\underline{f}_{p,n}^{x}$ are the optimal ones as given by equation (12)

We may also define the delayed backward prediction error vector for $\underline{x}_n$ by [2]

$$\underline{d}_{p,n}^{x} = S^{p+1} \underline{x}_n + X_{n,p} \underline{\tilde{a}}_p \tag{14}$$

It can be seen that the k^{th} row of equation (14) represents a prediction of x(k-p-1) by a linear combination of the p most immediate future values x(k-p), x(k-p+1),. . .,x(k-1). The resulting error in this backward prediction is $d_{p,n}^{x}(k)$. In this case the optimum $\underline{\tilde{a}}_p$ vector is the one that minimizes the quadratic function

$$\tilde{g}(\underline{\tilde{a}}_p) = [\underline{d}_{p,n}^{x}]^{\dagger} W \underline{d}_{p,n}^{x} \tag{15}$$

where W is defined in equation (9)

In this case the optimal $\tilde{a}$ is given by

$$\underline{\tilde{a}}_p^{\circ} = -[Y_{n,p}^{\dagger} X_{n,p}]^{-1} Y_{n,p}^{\dagger} (S^{p+1} \underline{x}_n) \tag{16}$$

In a similar manner to the forward prediction error case the optimal estimate of $S^{p+1}\underline{x}_n$ is specified by

$$\widehat{S^{p+1}\underline{x}_n}^{\circ} = -X_{n,p} \underline{\tilde{a}}_p^{\circ} = P_{XY}(S^{p+1} \underline{x}_n) \tag{17a}$$

and the optimal delayed backward prediction error vector is

$$\underline{d}_{p,n}^{x\,\circ} = S^{p+1} \underline{x}_n - \widehat{S^{p+1}\underline{x}_n}^{\circ} = P_{XY}^{c}(S^{p+1} \underline{x}_n) \tag{17b}$$

where P_{XY} and P_{XY}^{c} are given by equation (13). Furthermore, it can be shown that the optimal $\underline{\tilde{a}}_p$ as given by equation (16) also arises by approximating p Yule-Walker equations in a manner similar to the approximation given by equation (3) for $\underline{a}_p$.

It is clear that the forward prediction error vector $\underline{f}_{p,n}^{x}$ and the autoregressive coefficient vector $\underline{a}_p$ are interchangeable in the sense that one can always be found from the other using equation (7). Similarly $\underline{d}_{p,n}^{x}$ and $\underline{\tilde{a}}_p$ are interchangeable since one can always be found from the other using equation (14). It is also true that the 2p elements of $\underline{a}_p$ and $\underline{\tilde{a}}_p$ are interchangeable with the 2p elements $f_{1,n}^{x}(n)$, $f_{2,n}^{x}(n)$,..., $f_{p,n}^{x}(n)$ and $d_{1,n}^{x}(n)$, $d_{2,n}^{x}(n)$, ..., $d_{p,n}^{x}(n)$ (i.e., the n^{th} elements of the 2p prediction error vectors $\underline{f}_{1,n}^{x}$, $\underline{f}_{2,n}^{x}$,..., $\underline{f}_{p,n}^{x}$ and $\underline{d}_{1,n}^{x}$, $\underline{d}_{2,n}^{x}$,...,$\underline{d}_{p,n}^{x}$). We will show this last fact in Section VIII, where we will also see that the prediction errors lead to a lattice filter structure which is related to the autoregressive coefficient vectors.

In the fast recursive algorithm, the autoregressive coefficient vectors $\underline{a}_p$ and $\underline{\tilde{a}}_p$ are not directly updated. Instead, the prediction error elements $f_{1,n}^{x}(n)$,..., $f_{p,n}^{x}(n)$, and $d_{1,n}^{x}(n)$,..., $d_{p,n}^{x}(n)$ are updated. Since these elements are interchangeable with the autoregressive coefficients, there is no information lost in updating only the prediction error elements. However, the prediction error elements may be updated in a computationally efficient manner, requiring O(p) multiplications and additions for the update. Moreover, as we shall see later, the 2p prediction error elements enable us to find all of the $\underline{a}_m$ and $\underline{\tilde{a}}_m$ vectors for ARMA denominator orders from 1 to p. It is for these reasons that we choose to update the

[2] We use the term "delayed" because although the subscript n appears, x(n) is never used in (14). The undelayed backward prediction error vector will be discussed in a later section.

prediction error elements.

We may also obtain additional autoregressive coefficient estimates similar to $\underline{a}$ and $\underline{\tilde{a}}$ by considering the prediction error vectors associated with the vector $\underline{y}_n$ given in equation (4b). Specifically, the optimum forward prediction error corresponding to $\underline{y}_n$ is defined as

$$\underline{f}^{y}_{p,n}{}^{\circ} = \underline{y}_n + Y_{n,p}\,\underline{c}^{\circ}_p \tag{18a}$$

where

$$\underline{c}^{\circ}_p = -[X^{\dagger}_{n,p}\,Y_{n,p}]^{-1}X^{\dagger}_{n,p}\,\underline{y}_n \tag{18b}$$

and $X_{n,p}$ $Y_{n,p}$ and $\underline{y}_n$ are defined in equation (4). As in earlier cases, $\underline{c}^{\circ}_p$ can be found by approximating p Yule-Walker equations or, equivalently, by minimizing the quadratic functional

$$h(\underline{c}_p) = [\underline{f}^{y}_{p,n}]^{\dagger}[X_{n,p}\,X^{\dagger}_{n,p}][\underline{f}^{y}_{p,n}] \tag{19}$$

Corresponding to this optimal autoregressive coefficient vector $\underline{c}^{\circ}_p$ we may also define the estimate

$$\hat{\underline{y}}_n{}^{\circ} = -Y_{n,p}\,\underline{c}^{\circ}_p = Y_{n,p}[X^{\dagger}_{n,p}Y_{n,p}]^{-1}X^{\dagger}_{n,p}\underline{y}_n$$
$$= P_{YX}\,\underline{y}_n \tag{20}$$

which, in turn, gives rise to the optimum error vector

$$\underline{f}^{y}_{p,n}{}^{\circ} = P^{c}_{YX}\,\underline{y}_n$$

where P_{YX} and P^{c}_{YX} are defined as in equation (13)

Finally, the optimal delayed backward prediction error vector for $\underline{y}_n$ is defined by

$$\underline{d}^{y}_{p,n}{}^{\circ} = S^{p+1}\underline{y}_n + Y_{n,p}\,\tilde{\underline{c}}^{\circ}_p \tag{21a}$$

where

$$\tilde{\underline{c}}^{\circ}_p = -[X^{\dagger}_{n,p}Y_{n,p}]^{-1}X^{\dagger}_{n,p}(S^{p+1}\underline{y}_n) \tag{21b}$$

Again, we can define the predicted value of $S^{p+1}\underline{y}$ by

$$\widehat{S^{p+1}\underline{y}}_n{}^{\circ} = Y_{n,p}[X^{\dagger}_{n,p}Y_{n,p}]^{-1}X^{\dagger}_{n,p}(S^{p+1}\underline{y}) = P_{YX}(S^{p+1}\underline{y}) \tag{22}$$

and it follows that

$$\underline{d}^{y}_{p,n}{}^{\circ} = P^{c}_{YX}\,(S^{p+1}\underline{y}_n) \tag{23}$$

Just as for the $\underline{x}_n$ vector, the 2_p entities $\underline{f}^{y}_{1,n}(n),\ldots,\underline{f}^{y}_{p,n}(n)$, and $\underline{d}^{y}_{1,n}(n)\ldots,\underline{d}^{y}_{p,n}(n)$ can be efficiently updated and enable us to determine the optimal $\underline{c}_m$ and $\tilde{\underline{c}}_m$ coefficient vectors for all ARMA model denominator orders from 1 to p.[3]

III. THE HILBERT SPACE SETTING

The problem of recursively updating the prediction error vectors in the fast algorithm can be more easily understood by casting the problem in a Hilbert space setting. Consider the n dimensional complex Euclidian space

$$H = C^n = C \times C \times \ldots \times C \tag{24}$$

with the standard vector inner product defined by

$$\langle \underline{x},\underline{y}\rangle = \underline{x}^{\dagger}\underline{y} = \sum_{i=1}^{n} x^{*}(i)\,y(i) \tag{25}$$

[3] Throughout the remainder of the paper, the "°" symbol will be dropped and the prediction error vectors $\underline{f}^{x}$, $\underline{d}^{x}$, $\underline{f}^{y}$, and $\underline{d}^{y}$ are assumed to be the optimal ones.

We note that the n×1 vectors $\underline{x}_n$, $S^{m}\underline{x}_n$, $\underline{y}_n$, and $S^{m}\underline{y}_n$ are all elements of H. Moreover, the p columns of matrix $X_{n,p}$ are also elements of H. The set of all linear combinations of these p elements is a subspace of H, which we denote by M_X. Similarly, M_Y is the subspace spanned by the p columns of $Y_{n,p}$.

Let us now consider the forward prediction of $\underline{x}_n$. From equation (12a) we see that $\underline{x}_n$ is formed by a matrix multiplication involving $\underline{x}_n$. The matrix P_{XY} is seen to be a linear operator on the Hilbert space H. It is apparent that P_{XY} maps elements of H into elements in the subspace M_X, that is

$$P_{XY}\colon H \to M_X \tag{26}$$

Also, it is evident from equation (13) that $P_{XY}{}^2 = P_{XY}$ so that the operator P_{XY} is a projection operator onto the subspace M_X. In general, P_{XY} is not the orthogonal projection operator onto subspace M_X. Instead, the associated direction of projection is determined by the matrix $Y_{n,p}$. It can be seen from equation (6) that the direction of projection of P_{XY} is orthogonal to M_Y. Thus, P_{XY} is the projection operator onto the subspace M_X along $M_Y{}^{\perp}$ (the orthogonal complement of M_Y).

With these thoughts in mind, we can provide a simple geometric interpretation to the four error vectors described in the last section. In particular, the geometric relationship between $\underline{x}_n$, $\hat{\underline{x}}_n$, and $\underline{f}^{x}_{p,n}$ is depicted in Figure 1.

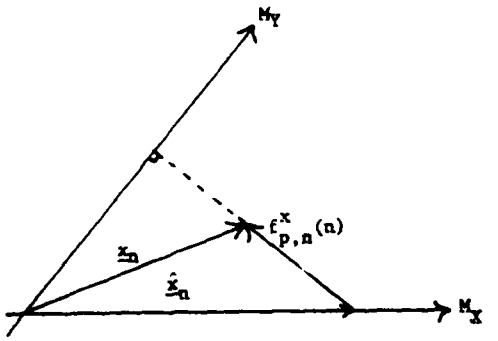

Figure 1: Geometric Relationship Between $\underline{x}_n$, $\hat{\underline{x}}_n$, and the optimal prediction error is $\underline{f}^{x}_{p,n}$

The vector $\hat{\underline{x}}_n$ is seen to be that projection of $\underline{x}_n$ onto M_X that is orthogonal to M_Y. We note from Figure 1 that

$$\underline{f}^{x}_{p,n} \perp M_Y \tag{27a}$$

or, equivalently, that

$$\langle \underline{f}^{x}_{p,n},\, S^{m}\underline{y}_n\rangle = 0, \quad m = 1,2,\ldots,p \tag{27b}$$

The geometric relationships for $\underline{d}^{x}_{p,n}$, $\underline{f}^{y}_{p,n}$, and $\underline{d}^{y}_{p,n}$ are similar to Figure 1.

Since P_{XY} and P_{YX} are projection operators, so are their complements P^{c}_{XY} and P^{c}_{YX}. It follows that

$$(P^{c}_{XY})^2 = P^{c}_{XY} \tag{28a}$$

$$(P^{c}_{YX})^2 = P^{c}_{YX} \tag{28b}$$

We can also see from equation (13) that P_{XY} and P_{YX} are strongly related, namely

$$P_{XY} = [P_{YX}]^{+} \tag{29a}$$

and $$P^{c}_{XY} = [P^{c}_{YX}]^{+} \tag{29b}$$

In addition to the four prediction error vectors, there are four inner products that are useful in deriving the fast algorithm. These complex-valued scalars are defined as:

$$\sigma_{p,n} \triangleq [\underline{f}^{x}_{p,n}]^{+}[\underline{d}^{y}_{p,n}] = [\underline{x}_n]^{\dagger}P^{c}_{XY}[S^{p+1}\underline{y}_n] \tag{30}$$

$$\tau_{p,n} \triangleq [\underline{d}^x_{p,n}]^\dagger [\underline{f}^y_{p,n}] = [S^{p+1}\underline{x}_n]^\dagger P^c_{XY}[\underline{y}_n] \quad (31)$$

$$\mu_{p,n} \triangleq [\underline{f}^x_{p,n}]^\dagger [\underline{f}^y_{p,n}] = [\underline{x}_n]^\dagger P^c_{XY}[\underline{y}_n] \quad (32)$$

$$\nu_{p,n} \triangleq [\underline{d}^x_{p,n}]^\dagger [\underline{d}^y_{p,n}] = [S^{p+1}\underline{x}_n]^\dagger P^c_{XY}[S^{p+1}\underline{y}_n] \quad (33)$$

IV. THE PROJECTION OPERATOR THEOREM

From the results of the last section it is apparent that the various prediction error vectors and scalars are all described by the operators P_{XY} and P_{YX}. As a new data point x(n+1) becomes available, we desire to update the prediction errors vectors and scalars in a computationally efficient manner. Because the operators P_{XY} and P_{YX} are used repeatedly and their structures change as new data points become available, we prefer to update them and then obtain updated error vectors by applying these updated projection operators. Recursive update equations for the projection operators P_{XY} and P_{YX}, are readily obtained by appealing to the following theorem.

Theorem 1. (Projection Operator Theorem) Let A and B be n×m matrices. Furthermore, consider the augmented matrices $\bar{A} = [A : \underline{a}]$ and $\bar{B} = [B : \underline{b}]$ in which $\underline{a}$ and $\underline{b}$ are n×1 vectors. If $[A^\dagger B]^{-1}$ and $[\bar{A}^\dagger \bar{B}]^{-1}$ exist, then the associated projection operator corresponding to the augmented matrices is given by

$$P^c_{\bar{A}\bar{B}} = P^c_{AB} - P^c_{AB}\,\underline{a}[\underline{b}^\dagger P^c_{AB}\,\underline{a}]^{-1}\underline{b}^\dagger P^c_{AB} \quad (34)$$

where $$P_{AB} = A[B^\dagger A]^{-1}B^\dagger \quad (35a)$$

$$P^c_{AB} = I - P_{AB} \quad (35b)$$

The theorem may be straightforwardly proven by writing $[\bar{A}\ \bar{B}]$ in terms of Schur complements and performing some matrix algebra. Alternatively, the theorem may be proven using Hilbert space concepts.

Using this projection operator theorem we may now obtain all the necessary equations for the fast recursive algorithm. Two types of recursive equations are of interest. First, equations are needed that provide the m+1st order prediction error vectors in terms of the mth order errors. These are called order update equations. Second, equations are needed that enable us to update the prediction errors as a new data point becomes available. These equations are referred to as the time update recursions.

These two sets of equations are derived below.

V. ORDER UPDATE RECURSIONS

In this section the order update equations for $\underline{f}^x$, $\underline{f}^y$, $\underline{d}^x$, $\underline{d}^y$, μ and ν are derived by making use of the projection operator theorem.

Consider first the error vector $\underline{f}^x_{m+1,n}$ associated with the optimum m+1st order autoregressive coefficients. Here, m can take on any value in the range 0, 1, ..., p - 1, where p is the desired autoregressive coefficient order. From equations (4) and (12b), we see that

$$\underline{f}^x_{m+1,n} = P^c_{\bar{X}\bar{Y}}\,\underline{x}_n \quad (36a)$$

where $$\bar{X} = X_{n,m+1} = [X_{n,m} : S^{m+1}\underline{x}_n] \quad (36b)$$

$$\bar{Y} = Y_{n,m+1} = [Y_{n,m} : S^{m+1}\underline{y}_n] \quad (36c)$$

Applying the projection operator theorem to (36a) with A = X, B = Y, $\underline{a} = S^{m+1}\underline{x}$, and $\underline{b} = S^{m+1}\underline{y}$, we have

$$\underline{f}^x_{m+1,n} = P^c_{XY}\,\underline{x} - P^c_{XY}(S^{m+1}\underline{x})[(S^{m+1}\underline{y})^\dagger P^c_{XY}(S^{m+1}\underline{x})]^{-1}(S^{m+1}\underline{y})^\dagger P^c_{XY}\,\underline{x}$$

$$\underline{f}^x_{m+1,n} = \underline{f}^x_{m,n} - \frac{\tau^*_{m,n}}{\nu^*_{m,n}}\,\underline{d}^x_{m,n} \quad (37)$$

As mentioned earlier, to implement the recursive algorithm we only need the error element at time n, that is $f^x_{m+1,n}(n)$. From equation (37) we see that

$$f^x_{m+1,n}(n) = f^x_{m,n}(n) - \frac{\tau^*_{m,n}}{\nu^*_{m,n}}\,d^x_{m,n}(n) \quad (38)$$

Equation (38) is the desired order update equation for f^x.

In a similar manner, the order update equation for f^y is found to be

$$\underline{f}^y_{m+1,n} = P^c_{\bar{Y}\bar{X}}\,\underline{y}_n \quad (39)$$

where $\bar{X}$ and $\bar{Y}$ are defined in equation (36). Applying the projection operator theorem yields

$$\underline{f}^y_{m+1,n} = \underline{f}^y_{m,n} - \frac{\tau_{m,n}}{\nu_{m,n}}\,\underline{d}^y_{m,n} \quad (40)$$

The nth component of equation (40) is the desired order update equation for the forward y prediction error, that is

$$f^y_{m+1,n}(n) = f^y_{m,n}(n) - \frac{\tau_{m,n}}{\nu_{m,n}}\,d^y_{m,n}(n) \quad (41)$$

The order update equations for the delayed backward prediction error vectors may be similarly derived. These equations are, however, not as useful as the combined order and time update equations. The combined update equations give $\underline{d}^x_{m+1,n+1}$ and $\underline{d}^y_{m+1,n+1}$ in terms of $\underline{d}^x_{m,n}$ and $\underline{d}^y_{m,n}$. In deriving the combined order and time update equation for the delayed backward $\underline{x}_n$ estimate, we first note that from equation (17b)

$$\underline{d}^x_{m+1,n+1} = P^c_{\bar{X}\bar{Y}}(S^{m+1}\underline{x}_n) \quad (42a)$$

where $$\bar{X} = X_{n+1,m+1} = \begin{bmatrix} \theta' & \vdots & 0 \\ \cdots & \vdots & \cdots \\ X_{n,m} & \vdots & \underline{x}_n \end{bmatrix} \quad (42b)$$

$$\bar{Y} = Y_{n+1,m+1} = \begin{bmatrix} \theta' & \vdots & 0 \\ \cdots & \vdots & \cdots \\ Y_{n,m} & \vdots & \underline{y}_n \end{bmatrix} \quad (42c)$$

We now apply the projection operator theorem to equation (42a) with $A = \bar{X}$ and $B = \bar{Y}$ in equation (34). After some simple algebraic manipulation, we get

$$\underline{d}^x_{m+1,n+1} = \begin{bmatrix} 0 \\ \cdots \\ \underline{d}^x_{m,n} \end{bmatrix} - \frac{\tau^*_{m,n}}{\mu^*_{m,n}} \begin{bmatrix} 0 \\ \cdots \\ \underline{f}^x_{m,n} \end{bmatrix} \quad (43)$$

The n+1st component of equation (43) yields the desired update equation

$$d^x_{m+1,n+1}(n+1) = d^x_{m,n}(n) - \frac{\tau^*_{m,n}}{\mu^*_{m,n}}\,f^x_{m,n}(n) \quad (44)$$

The delayed backward time and order update equation for $\underline{y}_n$ is derived in a similar manner. The details are omitted, but it is readily shown that

$$d^y_{m+1,n+1}(n+1) = d^y_{m,n}(n) - \frac{\sigma_{m,n}}{\mu_{m,n}}\,f^y_{m,n}(n) \quad (45)$$

Finally, the order update equations for the scalars $\mu_{m,n}$ and $\nu_{m,n}$ are derived. From equation (32)

$$\mu_{m+1,n} = \underline{y}_n^\dagger P^c_{\bar{X}\bar{Y}}\,\underline{x}_n \quad (46a)$$

where $$\bar{X} = X_{n,m+1} = [X_{n,m} : (S^{m+1}\underline{x}_n)] \quad (46b)$$

$$\bar{Y} = Y_{n,m+1} = [Y_{n,m} : (S^{m+1}\underline{y}_n)] \quad (46c)$$

Applying the projection operator theorem gives the desired result

$$\mu_{m+1,n} = \underline{y}^{\dagger} P^{c}_{XY}\, \underline{x}$$

$$- \underline{y}^{\dagger} P^{c}_{XY}(S^{m+1}\underline{x})[(S^{m+1}\underline{y})^{\dagger} P^{c}_{XY}(S^{m+1}\underline{x})]^{-1}(S^{m+1}\underline{y})^{\dagger} P^{c}_{XY}\, \underline{x}$$

$$\mu_{m+1,n} = \mu_{m,n} - \frac{\sigma_{m,n}\tau_{m,n}}{\nu^{*}_{m,n}} \tag{47}$$

Also, from equation (33)

$$\nu_{m+1,n+1} = (S^{m+1}\underline{y}_n)^{\dagger} P^{c}_{\bar{X}\bar{Y}}(S^{m+1}\underline{x}_n) \tag{48}$$

where $\bar{X}$ and $\bar{Y}$ are defined in equation (46). Applying the projection operator theorem yields the combined order and time update equation for ν.

$$\nu_{m+1,n+1} = \nu_{m,n} - \frac{\sigma_{m,n}\tau_{m,n}}{\mu^{*}_{m,n}} \tag{49}$$

VI. TIME UPDATE EQUATIONS

The remaining recursive equations update the forward and delayed backward errors as a new data point is obtained. For this reason these equations are called time update equations.

When a new data point becomes available, the effect on the prediction error vectors is to append a row to the bottom of their defining matrix equation [see, for example, equation (7)]. Appending a row to the matrices $X_{n,m}$ and $Y_{n,m}$ does not seem to fit in the framework of the projection operator theorem, in which columns are appended to $X_{n,m}$ and $Y_{n,m}$. It turns out, however, that we can accomplish the task of annihilating a row in the error vector matrix equation by appending to $X_{n,m}$ and $Y_{n,m}$ the $n^{\underline{th}}$ basis vector defined by

$$\underline{e}_n = [0 \;\; 0 \;\ldots\; 0 \;\; 1]' \tag{50}$$

To see how this works, let us consider as an example the forward prediction error vector for $\underline{x}_n$. If we append the $n\times 1$ vector $\underline{e}_n$ to $X_{n,m}$ in equation (7) we have

$$\begin{bmatrix} \tilde{f}^{x}_{m,n}(1) \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \tilde{f}^{x}_{m,n}(n) \end{bmatrix} = \begin{bmatrix} x(1) \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ x(n) \end{bmatrix} + \left[\begin{array}{ccc:c} 0 & \cdot\cdot\cdot & 0 & 0 \\ x(1) & \cdot\cdot\cdot & 0 & 0 \\ \cdot & \cdot\cdot\cdot & x(1) & \vdots \\ \cdot & & \cdot & \\ \cdot & & \cdot & 0 \\ x(n-1) & x(n-m) & & 1 \end{array}\right] \left[\begin{array}{c} a_m(1) \\ a_m(2) \\ \vdots \\ a_m(m) \\ \xi \end{array}\right] \tag{51}$$

where $\tilde{f}^{x}$ is used instead of $\underline{f}^{x}$ to indicate the presence of the $\underline{e}_n$ vector. The optimal $\tilde{f}^{x}_{m,n}$ vector is given by

$$\tilde{f}^{x}_{m,n} = P^{c}_{\bar{X}\bar{Y}}\, \underline{x}_n \tag{52a}$$

where

$$\bar{X} = [X_{n,m} \vdots \underline{e}_n] \tag{52b}$$

$$\bar{Y} = [Y_{n,m} \vdots \underline{e}_n] \tag{52c}$$

From equation (27b) we know that

$$< \tilde{f}^{x}_{m,n} \,,\, S^{i}\underline{y}_n > = 0 \quad , \quad i = 1,2,\ldots,m \tag{53a}$$

$$< \tilde{f}^{x}_{m,n} \,,\, \underline{e}_n > = 0 \tag{53b}$$

Equation (53b) is satisfied only if we force $\tilde{f}^{x}_{m,n}(n) = 0$. This can always be done because the scalar ξ appears only in the last row of equation (51). In particular, $\tilde{f}^{x}_{m,n}(n) = 0$ if we choose

$$\xi = -\sum_{i=1}^{m} a_m(i)x(n-i) \tag{54}$$

Since $\tilde{f}^{x}_{m,n}(n) = 0$, equation (53a) is seen to depend only on the first n-1 components of the vectors. It is easily seen, then, that $\tilde{f}^{x}_{m,n}(k)$ for k=1,2,...,n-1 are determined using the first n-1 rows of (51) in such a manner that the vector $[\tilde{f}^{x}_{m,n}(1),\ldots,\tilde{f}^{x}_{m,n}(n-1)]'$ is orthogonal to the first n-1 components of each column of $Y_{n,m}$. But this is exactly the problem of determining the forward prediction error based on n-1 data points. Thus, we see that

$$\underline{\tilde{f}}^{x}_{m,n} = \begin{bmatrix} \underline{f}^{x}_{m,n-1} \\ 0 \end{bmatrix} \tag{55}$$

Similar arguments show that this time annihilation property also holds for $\underline{f}^{y}_{m,n}$, $\underline{d}^{x}_{m,n}$, and $\underline{d}^{y}_{m,n}$ with resulting formulas similar to equation (55). We finally note that the scalars $\sigma_{m,n}$, $\tau_{m,n}$, $\mu_{m,n}$, and $\nu_{m,n}$ are formed as inner products of the prediction error vectors. Since the last element of $\tilde{f}^{x}_{m,n}$ (or $\tilde{f}^{y}_{m,n}$, $\tilde{d}^{x}_{m,n}$, $\tilde{d}^{y}_{m,n}$) is zero, it follows that

$$\sigma_{m,n-1} = [\tilde{f}^{x}_{m,n}]^{\dagger}[\tilde{d}^{y}_{m,n}] \tag{56}$$

and similarly for $\tau_{m,n}$, $\mu_{m,n}$, and $\nu_{m,n}$.

With these thoughts in mind we are in a position to derive time update equations for σ, τ, μ, and ν. First, let us define the augmented matrices

$$\bar{X} = [X_{n,m} \vdots \underline{e}_n] \tag{57a}$$

$$\bar{Y} = [Y_{n,m} \vdots \underline{e}_n] \tag{57b}$$

Then it follows that

$$\sigma_{m,n-1} = [\tilde{f}^{x}_{m,n}]^{\dagger}[\tilde{d}^{y}_{m,n}] = \underline{x}_n^{\dagger} P^{c}_{\bar{X}\bar{Y}}(S^{m+1}\underline{y}_n) \tag{58}$$

where $\bar{X}$ and $\bar{Y}$ are defined by equation (57). Application of the projection operator theorem yields

$$\sigma_{m,n-1} = \underline{x}_n^{\dagger}\left(P^{c}_{XY} - P^{c}_{XY}\,\underline{e}_n[\underline{e}_n^{\dagger} P^{c}_{XY}\underline{e}_n]^{-1}\underline{e}_n^{\dagger} P^{c}_{XY}\right)(S^{m+1}\underline{y}_n)$$

$$\sigma_{m,n-1} = \sigma_{m,n} - f^{x}_{m,n}{}^{*}(n)[1-\gamma_{m,n}]^{-1} d^{y}_{m,n}(n) \tag{59}$$

where $1 - \gamma_{m,n} \triangleq \underline{e}_n^{\dagger} P^{c}_{XY}\, \underline{e}_n$ (60)

By rewriting equation (59) we arrive at the desired time update equation

$$\sigma_{m,n} = \sigma_{m,n-1} + \frac{[f^{x}_{m,n}(n)]^{*}[d^{y}_{m,n}(n)]}{1-\gamma_{m,n}} \tag{61}$$

The time update equation for τ is found in a similar manner by using $\bar{X}$ for A, $\bar{Y}$ for B, $S^{m+1}\underline{x}_n$ for $\underline{a}$, and $\underline{y}_n$ for $\underline{b}$ in equation (34) to yield

$$\tau_{m,n} = \tau_{m,n-1} + \frac{[d^{x}_{m,n}(n)]^{*}[f^{y}_{m,n}(n)]}{1-\gamma_{m,n}} \tag{62}$$

The update equations for μ and ν are found to be

$$\mu_{m,n} = \mu_{m,n-1} + \frac{[f^{x}_{m,n}(n)]^{*}[f^{y}_{m,n}(n)]}{1-\gamma_{m,n}} \tag{63}$$

$$\nu_{m,n} = \nu_{m,n-1} + \frac{[d^{x}_{m,n}(n)]^{*}[d^{y}_{m,n}(n)]}{1-\gamma_{m,n}} \tag{64}$$

by using arguments analogous to those used in deriving equations (61) and (62).

Finally, update equations for $\gamma_{m,n}$ are needed. To obtain an order update equation, we note that

$$1 - \gamma_{m+1,n} = \underline{e}_n^{\dagger} P^c_{\bar{X}\bar{Y}} \underline{e}_n \tag{65a}$$

where

$$\bar{X} = [X_{n,m} \vdots (S^{m+1}\underline{x}_n)] \tag{65b}$$

$$\bar{Y} = [Y_{n,m} \vdots (S^{m+1}\underline{y}_n)] \tag{65c}$$

Applying the projection operator theorem,

$$1 - \gamma_{m+1,n} = \underline{e}_n^{+} P^c_{XY} \underline{e}_n - \underline{e}_n^{\dagger} P^c_{XY}(S^{m+1}\underline{x})[(S^{m+1}\underline{y})^{\dagger} P^c_{XY}(S^{m+1}\underline{x})]^{-1}(S^{m+1}\underline{y}_n)^{\dagger} P^c_{XY} \underline{e}_n$$

$$= 1 - \gamma_{m,n} - \frac{[d^x_{m,n}(n)]^*[d^y_{m,n}(n)]}{\nu^*_{m,n}} \tag{66}$$

or, equivalently

$$\gamma_{m+1,n} = \gamma_{m,n} + \frac{[d^x_{m,n}(n)]^*[d^y_{m,n}(n)]}{\nu^*_{m,n}} \tag{67}$$

To obtain the time and order update equation for γ we note that

$$1 - \gamma_{m+1,n+1} = \underline{e}_{n+1}^{\dagger} P^c_{\bar{X}\bar{Y}} \underline{e}_{n+1} \tag{68a}$$

where

$$\bar{X} = X_{n+1,m+1} = \begin{bmatrix} 0 & \vdots & \underline{\theta}' \\ \cdots & \vdots & \cdots \\ \underline{x}_n & \vdots & X_{n,m} \end{bmatrix} \tag{68b}$$

$$\bar{Y} = Y_{n+1,m+1} = \begin{bmatrix} 0 & \vdots & \underline{\theta}' \\ \cdots & \vdots & \cdots \\ \underline{y}_n & \vdots & Y_{n,m} \end{bmatrix} \tag{68c}$$

Since the first element of $\underline{e}_{n+1}$ is zero, it follows that

$$1 - \gamma_{m+1,n+1} = \underline{e}_n^{+} P^c_{\bar{X}\bar{Y}} \underline{e}_n \tag{69a}$$

where

$$\bar{X} = [X_{n,m} \vdots \underline{x}_n] \tag{69b}$$

$$\bar{Y} = [Y_{n,m} \vdots \underline{y}_n] \tag{69c}$$

Applying the projection operator theorem yields

$$1 - \gamma_{m+1,n+1} = 1 - \gamma_{m,n} - \frac{[f^x_{m,n}(n)]^*[f^y_{m,n}(n)]}{\mu^*_{m,n}}$$

or

$$\gamma_{m+1,n+1} = \gamma_{m,n} + \frac{[f^x_{m,n}(n)]^*[f^y_{m,n}(n)]}{\mu^*_{m,n}} \tag{70}$$

These six time update equations (61), (62), (63), (64), (67), and (70) along with the six order update equations (38), (41), (44), (45), (47) and (49) obtained in the last section comprise the fast recursive algorithm.

VII. SUMMARY OF THE ALGORITHM

At this point all necessary equations for implementing the fast recursive algorithm have been derived. In this section initial conditions are discussed and a procedure for implementing the recursive equations is given.

The implementation of the algorithm is conceptually easier if we replace the delayed backward error quantities by (undelayed) backward error quantities.

The (undelayed) backward errors are defined by

$$\underline{b}^x_{p,n} = S^p \underline{x}_n + [S^0\underline{x}_n \vdots S^1\underline{x}_n \vdots \cdots \vdots S^{p-1}\underline{x}_n]\underline{\tilde{a}}_p \tag{71}$$

$$\underline{b}^y_{p,n} = S^p \underline{y}_n + [S^0\underline{y}_n \vdots S^1\underline{y}_n \vdots \cdots \vdots S^{p-1}\underline{y}_n]\underline{\tilde{c}}_p \tag{72}$$

where $\underline{\tilde{a}}_p$ and $\underline{\tilde{c}}_p$ are $p\times 1$ autoregressive coefficient vectors. By comparing equations (71) and (72) with equations (14) and (21), it is readily seen that the (optimal) backward and delayed backward error vectors are related by

$$\underline{d}^x_{m,n} = \begin{bmatrix} 0 \\ \cdots \\ \underline{b}^x_{m,n-1} \end{bmatrix} \tag{73a}$$

and

$$\underline{d}^y_{m,n} = \begin{bmatrix} 0 \\ \cdots \\ \underline{b}^y_{m,n-1} \end{bmatrix} \tag{73b}$$

The $n^{\underline{th}}$ components of equations (73) yield the desired results

$$d^x_{m,n}(n) = b^x_{m,n-1}(n-1) \tag{74a}$$

$$d^y_{m,n}(n) = b^y_{m,n-1}(n-1) \tag{74b}$$

Thus, using equation (74) we can replace the delayed backward error terms in the recursive equations by backward error terms.

It is also helpful to replace ν by another scalar defined by

$$\omega_{m,n} = [\underline{b}^x_{m,n}]^{\dagger}[\underline{b}^y_{m,n}] \tag{75}$$

By considering the defining equation for ν, equation (33), along with equation (73) it is a simple matter to show that

$$\omega_{m,n-1} = \nu_{m,n} \tag{76}$$

Table 1 summarizes the complete set of update equations with $\underline{d}^x$, $\underline{d}^y$, and ν replaced by $\underline{b}^x$, $\underline{b}^y$, and ω.

The initial conditions for the various quantities in Table 1 are obtained by considering the defining equations for these quantities. For example, from equation (24) we see that if $m = 0$ then $\underline{f}^x_{0,n} = \underline{x}_n$ or

$$f^x_{0,n}(n) = x(n) \tag{77}$$

From equation (71), it also follows that

$$b^x_{0,n}(n) = x(n) \tag{78}$$

Similarly, we can show that

$$f^y_{0,n}(n) = y(n) = x(n-q) \tag{79}$$

$$b^y_{0,n}(n) = y(n) = x(n-q) \tag{80}$$

Furthermore, since $\underline{y}_n$ is the zero vector for $n \le q$, it follows from equations (30) - (33) that

$$\sigma_{0,n} = \tau_{0,n} = \mu_{0,n} = \omega_{0,n} = 0 \text{ for } n \le q \tag{81}$$

It is clear from Table 1 that no order update equations exist from σ and τ. Therefore, initial conditions are needed for $\sigma_{p,n}$ and $\tau_{p,n}$ for each p and some corresponding n. A little thought will convince one that $f^x_{p,q+p}(q+p) = 0$ by using an argument similar to the time annihilation argument presented in Section VI. Moreover, we can also show that $d^x_{p,q+p}(q+p) = 0$. by using this type of argument. Thus,

$$\sigma_{m,q+m} = \tau_{m,q+m} = 0 \quad \text{for } m = 0,1,\ldots,p \tag{82}$$

$$f^x_{m+1,n}(n) = f^x_{m,n}(n) - b^x_{m,n-1}(n-1)\,\sigma^*_{m,n}/\omega^*_{m,n-1} \quad \text{(T-1)}$$

$$f^y_{m+1,n}(n) = f^y_{m,n}(n) - b^y_{m,n-1}(n-1)\,\tau_{m,n}/\omega_{m,n-1} \quad \text{(T-2)}$$

$$b^x_{m+1,n}(n) = b^x_{m,n-1}(n-1) - f^x_{m,n}(n)\,\tau^*_{m,n}/\mu^*_{m,n} \quad \text{(T-3)}$$

$$b^y_{m+1,n}(n) = b^y_{m,n-1}(n-1) - f^y_{m,n}(n)\,\sigma_{m,n}/\mu_{m,n} \quad \text{(T-4)}$$

$$\mu_{m+1,n} = \mu_{m,n} - \sigma_{m,n}\tau_{m,n}/\omega_{m,n-1} \quad \text{(T-5)}$$

$$\omega_{m+1,n} = \omega_{m,n-1} - \sigma_{m,n}\tau_{m,n}/\mu_{m,n} \quad \text{(T-6)}$$

$$\sigma_{m,n} = \sigma_{m,n-1} + [f^x_{m,n}(n)]^*[b^y_{m,n-1}(n-1)]/(1-\gamma_{m,n}) \quad \text{(T-7)}$$

$$\tau_{m,n} = \tau_{m,n-1} + [b^x_{m,n-1}(n-1)]^*[f^y_{m,n}(n)]/(1-\gamma_{m,n}) \quad \text{(T-8)}$$

$$\mu_{m,n} = \mu_{m,n-1} + [f^x_{m,n}(n)]^*[f^y_{m,n}(n)]/(1-\gamma_{m,n}) \quad \text{(T-9)}$$

$$\omega_{m,n} = \omega_{m,n-1} + [b^x_{m,n}(n)]^*[b^y_{m,n}(n)]/(1-\gamma_{m,n}) \quad \text{(T-10)}$$

$$\gamma_{m+1,n} = \gamma_{m,n} + [b^x_{m,n-1}(n-1)]^*[b^y_{m,n-1}(n-1)]/\omega^*_{m,n-1} \quad \text{(T-11)}$$

$$\gamma_{m+1,n+1} = \gamma_{m,n} + [f^x_{m,n}(n)]^*[f^y_{m,n}(n)]/\mu^*_{m,n} \quad \text{(T-12)}$$

Table 1: Summary of Update Equations

where p is the desired (maximum) autoregressive coefficient order.

Finally, when $m = 0$,

$$\gamma_{0,n} = 0 \quad (83)$$

Although other initial conditions may be obtained, these initial conditions are the only ones needed to implement the algorithm.

The implementation of the update formulas can be divided into three parts. First, for $n \le q$ the vector $\underline{\gamma}_n$ is the zero vector, so no operations are performed. For $q+1 \le n \le q+p+1$, the maximum order m that can be used is $n-q-1$. In this time interval, as a new data point arrives not only are time updates performed, but also the model order is increased. For $n > q+p+1$, the model order remains at p and the time updates only are performed.

The implementation of the algorithm for $n \ge q+1$ is summarized below.

As the new data point becomes available:

1) Set $n \leftarrow n+1$

2) These quantities are available from the last iteration:

$f^x_{m,n-1}(n-1), f^y_{m,n-1}(n-1), b^x_{m,n-1}(n-1), b^y_{m,n-1}(n-1)$ for $m = 0, 1, \ldots, \min[p, n-q-2]$

$\tau_{m,n-1}, \tau_{m,n-1}, \mu_{m,n-1}, \omega_{m,n-1}$ for $m = 0, 1, \ldots, \min[p, n-q-2]$

3) New initial conditions:

$$f^x_{0,n}(n) = x(n)$$
$$b^x_{0,n}(n) = x(n)$$
$$f^y_{0,n}(n) = x(n-q)$$
$$b^y_{0,n}(n) = x(n-q)$$
$$\gamma_{0,n} = 0$$

If $n \le p+q+2$, $\sigma_{n-q-1,n-1} = \tau_{n-q-1,n-1} = 0$

4) For each $m = 0,1,\ldots,\min[p-1,n-q-3]$ find

$\sigma_{m,n}, \tau_{m,n}, \mu_{m,n}, \omega_{m,n}$ using (T-7)-(T-10)

$f^x_{m+1,n}(n), f^y_{m+1,n}(n), b^y_{m+1,n}(n), b^y_{m+1,n}(n)$ using (T-1)-(T-4)

$\gamma_{m+1,n}$ using (T-11)

5) For $m = \min[p, n-q-2]$ find

$\sigma_{m,n}, \tau_{m,n}, \mu_{m,n}, \omega_{m,n}$ using (T-7)-(T-10)

6) If $n < p+q+1$ we need to add a filter order.

Set $m \leftarrow n-q-2$.

Find:

$f^x_{m+1,n}(n), f^y_{m+1,n}(n), b^x_{m+1,n}(n), b^y_{m+1,n}(n)$ using (T-1)-(T-4)

$\gamma_{m+1,n}$ using (T-1)

$\mu_{m+1,n}, \omega_{m+1,n}$ using (T-5), (T-6)

Set $m \leftarrow n-q-1$.

Find:

$\sigma_{m,n}, \tau_{m,n}$ using (T-7), (T-8) with $\sigma_{m,n-1} = \tau_{m,n-1} = 0$

At this point we are ready for the next data point to arrive.

It is clear from the above summary that O(p) multiplications and additions are required to update the prediction errors. More specifically, in the time update mode [i.e., when $n > p+q+2$ so no filter orders need to be added] 14p multiplications and 10p additions are performed per update. In the time and order update mode, [i.e., when $n \le p+q+2$], 17p multiplications and 13p additions are performed. It should be noted that this computational requirement may be significantly reduced by using a normalized lattice form similar to that in [2].

VIII. THE LATTICE STRUCTURE

The fast recursive algorithm lends itself to a digital filter structure known as the lattice filter. This structure can be seen by considering equations (T-1) - (T-4). It is clearly seen that the filter that produces the four outputs $f^x_{m,n}(n)$, $f^y_{m,n}(n)$, $b^x_{m,n}(n)$ and $b^y_{m,n}(n)$ is given in Figure 2, where

$$\alpha^x_{p,n} = \frac{\tau^*_{m,n}}{\mu^*_{m,n}} \tag{84}$$

$$\beta^x_{m,n} = \frac{\sigma^*_{m,n}}{\omega^*_{m,n-1}} \tag{85}$$

$$\alpha^y_{m,n} = \frac{\sigma_{m,n}}{\mu_{m,n}} \tag{86}$$

$$\beta^y_{m,n} = \frac{\tau_{m,n}}{\omega_{m,n-1}} \tag{87}$$

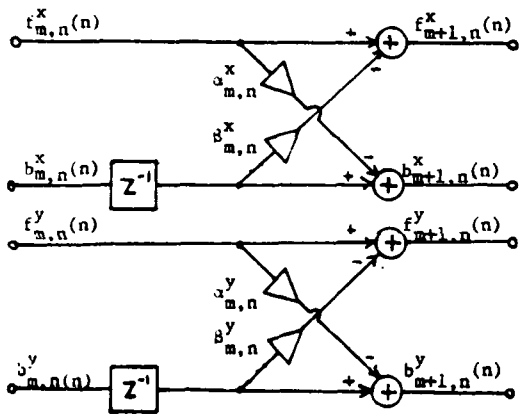

Figure 2:

Filter Realization of Error Order Update Equations

Thus, the entire p^{th} order filter is given by p stages of filter shown in Figure 2. This is depicted in Figure 3

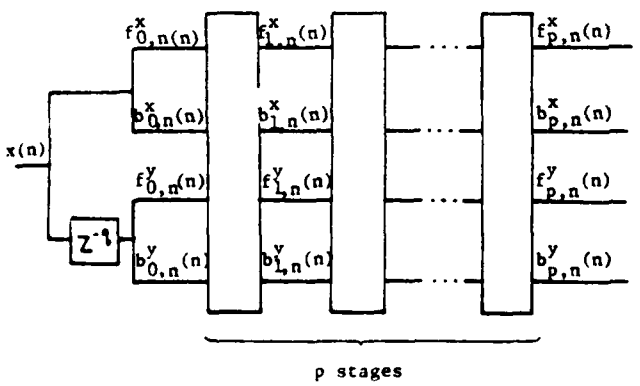

Figure 3:

Realization of the p^{th} Order Filter

In the startup phase of the algorithm [i.e., when the filter order is less than the desired order] the filter of Figure 3 begins with one stage, and successive stages are added as new data points arrive. Alternately, all p stages may be in place at the beginning; however, the four multiplier sections of each unused stage is set to zero until that stage is to be used.

The lattice filter structure at Figure 3 nicely depicts the relationship between the autoregressive coefficients and the prediction errors. To see this relationship, let us denote the transfer functions from x(n) to $f^x_{m,n}(n)$ and from x(n) to $b^x_{m,n}(n)$ by $F^x_m(z)$ and $B^x_m(z)$, respectively. Let us also define the autoregressive coefficient transfer functions

$$A_m(z) = 1 + a_m(1)z^{-1} + \ldots + a_m(m)z^{-m} \tag{88}$$

$$\tilde{A}_m(z) = \tilde{a}_m(0) + \tilde{a}_m(1)z^{-1} + \ldots + \tilde{a}_m(m-1)z^{-m+1} + z^{-m} \tag{89}$$

where the a_i and $\tilde{a}_i$ coefficients are defined by equations (5) and (71) respectively. A little thought will convince one that the transfer functions are related by

$$F^x_m(z) = A_m(z) \tag{90}$$

$$B^x_m(z) = \tilde{A}_m(z) \tag{91}$$

With this thought in mind we are now able to derive equations that relate the error elements to the autoregressive coefficients. From Figure 2 it is clear that

$$f^x_{m,n}(n) = f^x_{m-1,n}(n) + \beta^x_{m-1,n} b^x_{m-1,n-1}(n-1) \tag{92}$$

$$b^x_{m,n}(n) = \alpha^x_{m-1,n} f^x_{m-1,n}(n) + b^x_{m-1,n-1}(n-1) \tag{93}$$

By taking z transforms of equations (92) and (93) and using equations (90) and (91) we get

$$A_m(z) = A_{m-1}(z) + z^{-1}\beta^x_{m-1,n}\tilde{A}_{m-1}(z) \tag{94}$$

$$\tilde{A}_m(z) = z^{-1}\alpha^x_{m-1,n}A_{m-1}(z) + \tilde{A}_{m-1}(z) \tag{95}$$

Furthermore, we can see from Figure 3 that

$$A_0(z) = \tilde{A}_0(z) = 1 \tag{96}$$

and from equations (94) and (95) that

$$a_m(m) = \beta^x_{m,n} \tag{97}$$

$$\tilde{a}_m(0) = \alpha^x_{m,n} \tag{98}$$

Given the error elements $f^x_{m,n}(n)$ and $b^x_{m,n}(n)$ for $m = 1,2,\ldots,p$ we can use equations (92) and (93) to get the $\alpha^x_{m,n}$ and $\beta^x_{m,n}$ elements, then use equations (94)-(95) to get the $a_m(k)$ and $\tilde{a}_m(k)$ elements for $m = 1,2,\ldots,p$ and $k = 0,1,2,\ldots,m$. Thus, we are able to obtain the autoregressive coefficients for all model orders from 1 to p.

Similarly, given the $a_p(k)$ and $\tilde{a}_p(k)$ coefficients for $k = 0,1,\ldots,p$, we can use (94), (95), (97), and (98) to get the $\alpha^x_{p,n}$ and $\beta^x_{p,n}$ and then the $p-1^{st}$ order autoregressive coefficients by working down from order p to order 0. Then, using (92) and (93) we can obtain the desired error elements.

Thus, using equations (92)-(98) we are able to convert back and forth between the lattice error elements and the optimal autoregressive coefficients.

As a final note, when $q = 0$ (i.e., when an autoregressive model is chosen) it can be seen in Figure 3 that the top half and bottom half of the filter are equivalent. In this case, only one half is needed, and the filter in Figure 3 degenerates to the AR lattice filter described in [1].

IX. CONCLUSIONS

In this paper we have presented a recursive algorithm for obtaining the autoregressive coefficients of an ARMA model. The recursive algorithm is based on the prewindowed version of the high performance method of ARMA spectral estimation as described in Part 1. The recursive algorithm is computationally fast, requiring $O(p)$ additions and multiplications to update the parameters. Moreover, the algorithm can be implemented using a lattice filter structure offering numerical robustness and nice convergence properties associated with lattice type algorithms.

We have not yet discussed the problem of recursively estimating the moving average coefficients in the ARMA model. We do not at this time have such an algorithm. However, it is worth noting that the moving average information is present in the output prediction error sequences, and the utilization of this information for moving average coefficient estimation is currently under study. Another area under study is the use of various normalization procedures to effect a decrease in computational requirements and in sensitivity.

Finally, we note that the recursive algorithm presented here is based on approximating a set of p Yule-Walker equations. It has been shown in Part 1 of this paper that for short data lengths, improved spectral estimates result from using more than p Yule-Walker equations. For those cases in which the amount of data is small, a fast recursive algorithm based on the approximation of $t > p$ Yule-Walker equations would often prove useful. Such an algorithm is currently being pursued.

REFERENCES

[1] M. Morf and D. Lee, "Recursive Least Squares Ladder forms for Fast Parameter Tracking", Proc. 1978 IEEE Conf. D. & C., San Diego, CA, pp. 1362-1367.

[2] D. Lee and M. Morf, "Recursive Square Root Ladder Estimation Algorithms", Proc. 1980 IEEE Conf. Acoustics, Speech, and Signal Processing, Denver, CO, pp. 1005-1017.

[3] B. Friedlander, "Recursive Algorithms for Ladder Forms", SCI Technical Memorandum, 1980.

[4] B. Friedlander, "Recursive Algorithms for Pole-Zero Ladder Forms", SCI Technical Memorandum, 1980.

[5] J. Makhoul, "Stable and Efficient Lattice Methods for Linear Prediction, IEEE Trans. Acoust., Speech, Signal Proc., Vol. ASSP-25, No.5, October 1977, pp. 423-428.

[6] D. Lee, M. Morf, and B. Friedlander, "Recursive Least Squares Ladder Estimation Algorithms", IEEE Trans. Ckts. Sys., Vol. CAS-28, No.6, June 1981, pp. 467-481.

[7] B. Porat, B. Friedlander, and M. Morf, "Square Root Covariance Ladder Algorithms, Proc. 1981 IEEE Conf. Acoust. Speech, Sig. Proc., Atlanta, GA, pp. 877-880.

[8] M. Morf, B. Dickenson, T. Kailath, and A. Vieira, "Efficient Solution of Covariance Equations for Linear Prediction", IEEE Trans. Acoust. Speech, and Sig. Proc., Vol. ASSP-25, October 1977, pp. 429-433.

[9] K. Ogino, "Computationally Fast Algorithms for ARMA Spectral Estimation", Ph.D. Dissertation, Virginia Polytechnic Institute and State University, June 1981.

[10] E. Satorius and S. Alexander, "Channel Equalization Using Adaptive Lattice Algorithms", IEEE Trans. Comm, Vol. COMM-27, No. 6, June 1979, pp. 899-905.

[11] D. Falconer and L. Ljung, "Application of Fast Kalman Estimation to Adaptive Equalization, IEEE Trans. Comm. Vol. COMM-26, No. 10, October 1978, pp. 1439-1446.

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: 10 ✓	2. GOVT ACCESSION NO.: AD-A101 870
3. RECIPIENT'S CATALOG NUMBER: 9	
4. TITLE (and Subtitle): An Adaptive ARMA Spectral Estimator. Part 2. 6	5. TYPE OF REPORT & PERIOD COVERED: Conference Paper
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): B. A. Cadzow (Randolph) L. Moses 15	8. CONTRACT OR GRANT NUMBER(s): N00014-80-C-0303 ✓
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Virginia Tech, Department of Electrical Engineering, Blacksburg, VA 24061	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 12 10
11. CONTROLLING OFFICE NAME AND ADDRESS: Office of Naval Research (Code 436), Statistics and Probability Program, Arlington, VA 22217	12. REPORT DATE: 11 Jun 81
	13. NUMBER OF PAGES: 9
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	15. SECURITY CLASS. (of this report): Unclassified
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

APPROVED FOR PUBLIC RELEASE: DISTRIBUTION UNLIMITED.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

ARMA Spectral Estimate, Recursive time-update algorithm, Projection operator, Lattice structure.

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

A recursive time-update algorithmic implementation of the ARMA spectral estimator as described in Part 1 is presented. The computational efficiency of this algorithm is comparable to that achieved by recently developed fast AR LMS methods.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

S/N 0102-LF-014-6601

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

411 262