TECHNICAL
RESEARCH
REPORT

SYSTEMS
RESEARCH
CENTER

*Supported by the
National Science Foundation
Engineering Research Center
Program (NSFD CD 8803012),
Industry and the University*

A Levinson-Type Algorithm for A Class of Non-Toeplitz Systems with Applications to Multichannel IIR Filtering

by J. Pan and W.S. Levine

TR 92-65

A Levinson-Type Algorithm for A Class of Non-Toeplitz Systems with Applications to Multichannel IIR Filtering*

Jiqin Pan and William S. Levine†

Systems Research Center and Electrical Engineering Department

University of Maryland, College Park, MD 20742

(Telephone: (301) 405-3654)

Manuscript: May 30, 1992

The EDICS number: 4.3.1 (Adaptive Filters and Signal Processing — Algorithms)

Abstract

A Levinson-type recursion for a class of non-Toeplitz systems of linear equations is demonstrated. A complete solution is expressed as a linear combination of a partial solution and three auxiliary solutions. The class of systems possesses a special structure in that the coefficient matrices can be partitioned into four block Toeplitz submatrices. The number of multiplications and additions required to compute an n-dimensional solution is $\mathbf{O}(n^2)$. The recursion is then applied to multichannel IIR filtering. Specifically, a lattice structure is established for linear minimum mean square error predictors having independently and arbitrarily specified numbers of poles and zeros. Next the recursion is used to develop a fast time and order recursive algorithm for ARX system identification, producing parameter estimates of a family of ARX models. The algorithm preserves consistency of the well-known recursive least-squares algorithm.

*This research was supported by the Systems Research Center under National Science Foundation Grant CDR 8803012.

†Author to whom correspondence should be addressed.

1 Introduction

Levinson's recursion has been applied, directly or indirectly, to many problems [1,2,3]. Maximum entropy spectral analysis, geophysical signal analysis, and speech processing are a few examples. As reviewed in [3], a large number of important recursive algorithms are based on Levinson's recursion, which expresses a complete solution to a Toeplitz system of linear equations as a linear combination of a partial (forward) solution and an auxiliary (backward) solution. When a linear prediction problem is associated with a Toeplitz system, all the subsystems along the principle diagonal of the system are equal. This fact results in the usual form of Levinson's recursion [4]. When the system is no longer Toeplitz, this is not necessarily true and Levinson's recursion cannot be applied in its usual form. To overcome this difficulty, an algebraic approach for solving Toeplitz-like systems was developed based on the displacement expression of the coefficient matrices [5,6]. Very recently, a Levinson-type algorithm for solving a general system of linear equations was established by Porsani and Ulrych in a further effort to alleviate the difficulty in solving non-Toeplitz systems [3].

A special class of non-Toeplitz systems is encountered in IIR filtering, spectral estimation for ARMA processes and two-dimensional signal processing, where the associated coefficient matrices are usually non-Toeplitz, but possess a special form $\begin{bmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{bmatrix}$ with the property that all four submatrices are of the block-Toeplitz structure [7,8]. That is, for each submatrix, its block-elements are "constant along any diagonal." For this reason, we call this class of non-Toeplitz systems block-Toeplitz submatrix systems (BTSSs). In several important cases which will be described later [9], there is no explicit displacement representation for the coefficient matrices of the BTSS involved and consequently the algebraic approach developed in [5,6] cannot be applied efficiently. Porsani and Ulrych's algorithm is applicable in these cases; however, the computational complexity of their algorithm is high.

Exploiting the block-Toeplitz structure of each submatrix makes it possible to establish a Levinson-type recursion. This is done in Section 2. The recursion linearly combines a partial solution with *three* auxiliary solutions and these partial and auxiliary solutions are of lower order than the complete solution. In Section 3 the Levinson-type recursion developed in this paper is applied to several IIR filtering problems. Specifically, it is applied to linear minimum mean square error (LMMSE) IIR filtering, leading to a flexible lattice structure for LMMSE pole-zero predictors. Next, in Section 4, a fast time and order recursive algorithm for ARX system identification is developed. A useful feature of the algorithm is that the solution of an identification problem for an ARX model with a certain number of poles and zeros contains all parameter estimates of ARX models having less poles and zeros. The paper is concluded in Section 5.

2 Algorithm for Solving the BTSS

A. Preliminary

For the sake of simplicity, we express scalar quantities, column vectors, matrices, and matrix arrays in lowercase, boldface lowercase, uppercase, and boldface uppercase letters, respectively, where a matrix array designates an array of matrix (block) elements of the same dimension. To distinguish a zero number from a zero matrix, we denote a zero matrix by $\emptyset$. The class of non-Toeplitz systems of linear equations considered in this paper is described below

$$[\ I\ \mathbf{A}_{p,q}\ \mathbf{C}_{p,q}\ ]\mathcal{W}_{p,q} = [R_{p,q}\ \mathbf{0}_p\ \mathbf{0}_{q+1}], \tag{1}$$

where the coefficient matrix $\mathcal{W}_{p,q}$ is usually not Toeplitz, but possesses a special structure in that it can be partitioned into

$$\mathcal{W}_{p,q} = \begin{bmatrix} \mathbf{V}_{p,p} & \mathbf{\Delta}_{p,q} \\ \mathbf{\Lambda}_{q,p} & \mathbf{S}_{q,q} \end{bmatrix} \tag{2}$$

and all the four submatrices are block-Toeplitz matrices described as:

$$\begin{aligned} \mathbf{V}_{p,p} &= [V_{j-i}]_{i,j=0}^{p}, & \mathbf{\Delta}_{p,q} &= [\Delta_{j-i}]_{i=0,j=0}^{p,q}, \\ \mathbf{\Lambda}_{q,p} &= [\Lambda_{j-i}]_{i=0,j=0}^{q,p}, & \mathbf{S}_{q,q} &= [S_{j-i}]_{i,j=0}^{q}, \end{aligned} \tag{3}$$

where the scalars $p \geq 0$ and $q \geq -1$ can be independently and arbitrarily specified and $V_i \in \mathcal{R}^{m\times m}$, $\Delta_i \in \mathcal{R}^{m\times l}$, $\Lambda_i \in \mathcal{R}^{l\times m}$, and $S_i \in \mathcal{R}^{l\times l}$, where m and l are two given positive integers. It is because of this special structure that we call this class of systems block-Toeplitz submatrix systems (BTSSs). The subscript "p, q" appearing in Eqns. (1) and (2) is used to emphasize that the number of matrix elements along the principle diagonals of matrix arrays $\mathbf{V}_{p,p}$ and $\mathbf{S}_{q,q}$ has a great impact on the desired solution.

All matrices appearing in Eqn.(1) are known except $[\mathbf{A}_{p,q}\ \mathbf{C}_{p,q}]$ and $R_{p,q}$. The matrix I stands for an identity matrix and $\mathbf{0}_j, j = p$ or $q+1$, is defined as $\mathbf{0}_j \triangleq [\ \underbrace{\emptyset\ \cdots\ \emptyset}_{j}\ ]$. It seems strange that the unknowns of Eqn.(1) occur on both the left and right hand sides of Eq.(1). However, *the matrix array* $[\mathbf{A}_{p,q}\ \mathbf{C}_{p,q}]$ *does not depend on* $R_{p,q}$ *and* $R_{p,q}$ *is uniquely determined once* $[\mathbf{A}_{p,q}\ \mathbf{C}_{p,q}]$ *is known.* Thus, Eqn.(1) can be considered as a variant of conventional systems of linear equations.

The primary problem to be addressed here is how to efficiently compute $[\mathbf{A}_{p,q}\ \mathbf{C}_{p,q}]$ by exploiting the special structure possessed by the system of linear equations. Sometimes, computing $R_{p,q}$ is also of interest. For instance, in linear prediction, the matrix $R_{p,q}$ may represent the prediction error variance. Before trying to find a solution to the problem, let us consider two motivational examples in which BTSSs occur.

Example 1 (LMMSE IIR filtering): IIR filters provide potential performance improvements over their FIR counterparts in many applications because the poles of an IIR filter can offer superior system modeling ability [10]. Many techniques for designing IIR filters are based on the solution of the following prototype linear prediction problem: Let $\mathbf{y}_n$ and $\mathbf{u}_n$ be two discrete-time zero-mean stationary processes. We are interested in predicting the current value of $\mathbf{y}_n$ from past measurements of $\mathbf{y}_n$ and current and past measurements of $\mathbf{u}_n$. A linear predictor having p poles and q zeros is of the form:

$$\mathbf{y}_{n|n-1}(p,q) = -\sum_{i=1}^{p} A_i^{p,q}\mathbf{y}_{n-i} - \sum_{j=0}^{q} C_j^{p,q}\mathbf{u}_{n-j},$$

where $\mathbf{y}_{n|n-1}(p,q)$ is the one-step prediction of $\mathbf{y}_n$ at time n given $\{\mathbf{y}_i\}_{n-p}^{n-1}$ and $\{\mathbf{u}_j\}_{n-q}^{n}$. Define the output regression vector $\mathbf{y}_{n-p}^{n-1}$ and the input regression vector $\mathbf{u}_{n-q}^{n}$ as

$$\mathbf{y}_{n-p}^{n-1} \triangleq (\mathbf{y}_{n-1}^T \cdots \mathbf{y}_{n-p}^T)^T \quad \text{and} \quad \mathbf{u}_{n-q}^{n} \triangleq (\mathbf{u}_{n}^T \cdots \mathbf{u}_{n-q}^T)^T. \tag{4}$$

Thus, the above linear predictor has the compact form:

$$\mathbf{y}_{n|n-1}(p,q) = -\mathbf{A}_{p,q}\mathbf{y}_{n-p}^{n-1} - \mathbf{C}_{p,q}\mathbf{u}_{n-q}^{n}, \tag{5}$$

where the predictor coefficients $[\mathbf{A}_{p,q}\ \mathbf{C}_{p,q}] = [\ A_1^{p,q}\ \cdots\ A_p^{p,q}\ \ C_0^{p,q}\ \cdots\ C_q^{p,q}\ ]$. The linear predictor (5) can be determined by minimizing the mean-square (prediction) error

$$\mathrm{trace}(\mathrm{E}[(\mathbf{y}_n - \mathbf{y}_{n|n-1}(p,q))(\mathbf{y}_n - \mathbf{y}_{n|n-1}(p,q))^T])$$

when the second-order statistics of the process $\begin{pmatrix}\mathbf{y}_n \\ \mathbf{u}_n\end{pmatrix}$ are known or estimated. Consequently, the linear predictor becomes an LMMSE (linear minimum mean square error) or optimal least squares predictor. The linear predictor (5) can be considered as an IIR filter, where $\mathbf{y}_n$ and $\mathbf{u}_n$ are $m-$dimensional desired signals and $l-$dimensional filter inputs, respectively.

Applying the orthogonality principle, we obtain the parameters $\mathbf{A}_{p,q}$ and $\mathbf{C}_{p,q}$ of an LMMSE predictor (5) as the solution to a Yule-Walker equation identical to (1), where the Yule-Walker matrix $\mathcal{W}_{p,q}$ is symmetric and its block elements are equal to

$$V_i = \mathrm{E}[\mathbf{y}_n\mathbf{y}_{n-i}^T], \quad S_i = \mathrm{E}[\mathbf{u}_n\mathbf{u}_{n-i}^T], \quad \Delta_i = \mathrm{E}[\mathbf{y}_n\mathbf{u}_{n-i}^T], \quad \text{and} \quad \Lambda_i = \mathrm{E}[\mathbf{u}_n\mathbf{y}_{n-i}^T]. \tag{6}$$

The matrix $R_{p,q}$ on the right side of Eqn.(1) is equal to the prediction error variance of the LMMSE predictor:

$$R_{p,q} = \mathrm{E}[(\mathbf{y}_n - \mathbf{y}_{n|n-1}(p,q))(\mathbf{y}_n - \mathbf{y}_{n|n-1}(p,q))^T].$$

Example 2 (ARX Identification): System identification is of major importance in adaptive signal processing and spectral estimation [8,10]. The solution to an identification problem depends on the model which is used to model signals. In designing adaptive IIR filters and performing ARMA spectral estimation via input-output identification approach, an ARX model (autoregressive model with exogenous inputs) is often used.

In general, an ARX model has the following form

$$\mathbf{y}_n + \mathbf{A}_{p,q}\mathbf{y}_{n-p}^{n-1} + \mathbf{C}_{p,q}\mathbf{u}_{n-q}^{n} = \mathbf{e}_{n,p,q}, \tag{7}$$

where $\mathbf{e}_{n,p,q}$ is model noise and $[\mathbf{A}_{p,q}\ \mathbf{C}_{p,q}] = [\ A_1^{p,q}\ \cdots\ A_p^{p,q}\ \ C_0^{p,q}\ \cdots\ C_q^{p,q}\ ]$. The vectors $\mathbf{y}_{n-p}^{n-1}$ and $\mathbf{u}_{n-q}^{n}$ are as described in (4). The elements of the vectors, $\mathbf{y}_{n-i}, i = 0, \cdots, p$, and $\mathbf{u}_{n-j}, j = 0, \cdots, q$, are output and input signals of an unknown system. The least squares parameter estimate of an ARX model is equal to the the matrix array $[\mathbf{A}_{N,p,q}\ \mathbf{C}_{N,p,q}]$ that minimizes the accumulated weighted least squares error

$$\mathrm{trace}(\sum_{n=1}^{N} \lambda^{N-n}\mathbf{e}_{n,p,q}\mathbf{e}_{n,p,q}^T), \tag{8}$$

where $\lambda, 0 < \lambda \leq 1$, is the forgetting factor. It is well known that when the data are windowed via the autocorrelation method [7], the solution minimizing the quantity described in (8) satisfies a symmetric Yule-Walker equation, which is a symmetric block-Toeplitz submatrix system. This example will be further considered in Section 4.

B. Algorithm for Solving Block-Toeplitz Submatrix Systems

We intend to build a computation procedure to determine the desired solution $[\ \mathbf{A}_{p,q}\ \mathbf{C}_{p,q}\ ]$ from knowledge of a partial solution of lower order: $[\ \mathbf{A}_{p-1,q}\ \mathbf{C}_{p-1,q}\ ]$ or $[\ \mathbf{A}_{p,q-1}\ \mathbf{C}_{p,q-1}\ ]$. As will be shown later, such a procedure provides a fast solution to Eqn.(1). Moreover, as a by-product, it provides many partial solutions to Eqn.(1) which have lower order than the desired complete solution. These partial solutions can be quite useful in many applications.

To establish the desired procedure, we deliberately introduce an auxiliary system of linear equations described by

$$\begin{bmatrix} {}^2\mathbf{A}_{p,q} & I & \vdots & {}^2\mathbf{C}_{p,q} & \emptyset \\ \emptyset & {}^3\mathbf{A}_{p,q} & \vdots & I & {}^3\mathbf{C}_{p,q} \\ \multicolumn{2}{c}{{}^4\mathbf{A}_{p,q}} & \vdots & {}^4\mathbf{C}_{p,q} & I \end{bmatrix} \mathcal{W}_{p,q} = \begin{bmatrix} \mathbf{0}_p & {}^2R_{p,q} & \vdots & \mathbf{0}_q & N^2_{p,q} \\ {}^3M_{p,q} & \mathbf{0}_p & \vdots & {}^3R_{p,q} & \mathbf{0}_q \\ \multicolumn{2}{c}{\mathbf{0}_{p+1}} & \vdots & \mathbf{0}_q & {}^4R_{p,q} \end{bmatrix}. \tag{9}$$

Note that all the matrices in Eqn. (9), except the coefficient matrix $\mathcal{W}_{p,q}$, identity matrices, and zero matrices, are unknown. We call the unknowns on the left hand side: $[\,{}^2\mathbf{A}_{p,q}\ \ {}^2\mathbf{C}_{p,q}]$, $[\,{}^3\mathbf{A}_{p,q}\ \ {}^3\mathbf{C}_{p,q}]$, and $[\,{}^4\mathbf{A}_{p,q}\ \ {}^4\mathbf{C}_{p,q}]$ the auxiliary solutions. Eq. (9) is analogous to Eq. (1) in that the unknowns (auxiliary solutions) on the left hand side are independent of the unknowns on the right hand side and the unknowns on the right hand side are uniquely determined once the auxiliary solutions are known. By introducing some extra unknown matrices and exploiting the Toeplitz structure of $\mathbf{V}_{p,p}$, $\mathbf{\Delta}_{p,q}$, $\mathbf{\Lambda}_{q,p}$, and $\mathbf{S}_{q,q}$, we can merge Eqns. (1) and (9) into a combined equation which has either of two forms. They are

p-form: for any $p \geq 0, q \geq 0$,

$$\begin{bmatrix} I & \mathbf{A}_{p,q} & \emptyset & \vdots & \multicolumn{2}{c}{\mathbf{C}_{p,q}} \\ \emptyset & {}^2\mathbf{A}_{p,q} & I & \vdots & \emptyset & {}^2\mathbf{C}_{p,q} \\ \emptyset & {}^3\mathbf{A}_{p,q} & \emptyset & \vdots & I & {}^3\mathbf{C}_{p,q} \\ \multicolumn{2}{c}{{}^4\mathbf{A}_{p,q}} & \emptyset & \vdots & {}^4\mathbf{C}_{p,q} & I \end{bmatrix} \mathcal{W}_{p+1,q} = \begin{bmatrix} R_{p,q} & \mathbf{0}_p & M^1_{p,q} & \vdots & \multicolumn{2}{c}{\mathbf{0}_{q+1}} \\ {}^2M_{p,q} & \mathbf{0}_p & {}^2R_{p,q} & \vdots & {}^2N_{p,q} & \mathbf{0}_q \\ {}^3M_{p,q} & \mathbf{0}_p & M^3_{p,q} & \vdots & {}^3R_{p,q} & \mathbf{0}_q \\ \multicolumn{2}{c}{\mathbf{0}_{p+1}} & M^4_{p,q} & \vdots & \mathbf{0}_q & {}^4R_{p,q} \end{bmatrix} \tag{10}$$

and

$p+1$-form: for any $p \geq 0, q \geq 0$,

$$\begin{bmatrix} I & \mathbf{A}_{p+1,q} & \vdots & \multicolumn{2}{c}{\mathbf{C}_{p+1,q}} \\ {}^2\mathbf{A}_{p+1,q} & I & \vdots & {}^2\mathbf{C}_{p+1,q} & \emptyset \\ \emptyset & {}^3\mathbf{A}_{p+1,q} & \vdots & I & {}^3\mathbf{C}_{p+1,q} \\ \multicolumn{2}{c}{{}^4\mathbf{A}_{p+1,q}} & \vdots & {}^4\mathbf{C}_{p+1,q} & I \end{bmatrix} \mathcal{W}_{p+1,q} = \begin{bmatrix} R_{p+1,q} & \mathbf{0}_{p+1} & \vdots & \multicolumn{2}{c}{\mathbf{0}_{q+1}} \\ \mathbf{0}_{p+1} & {}^2R_{p+1,q} & \vdots & \mathbf{0}_q & N^2_{p+1,q} \\ {}^3M_{p+1,q} & \mathbf{0}_{p+1} & \vdots & {}^3R_{p+1,q} & \mathbf{0}_q \\ \multicolumn{2}{c}{\mathbf{0}_{p+2}} & \vdots & \mathbf{0}_q & {}^4R_{p+1,q} \end{bmatrix} \tag{11}$$

where all the matrices in Eqns. (10) and (11), except the coefficient matrix $\mathcal{W}_{p+1,q}$, identity matrices, and zero matrices, are unknown. The unknowns on the left hand side are independent of the unknowns on the right hand side and the unknowns on the right hand side are uniquely determined as long as the unknowns on the left hand side are known. We note that Eqn.(10) shares the same coefficient matrix with Eqn.(11) and equal systems of linear equations have equal solutions. This inspires us to introduce a weighting matrix such that the right hand side of Eqn.(11) is equal to the product between the weighting matrix and the right hand side of Eqn.(10). In fact, this desired weighting matrix exists and can be obtained based on the solution to Eqn. (10). The determination of this weighting matrix takes about $\mathbf{O}(p+q)$ flops. For details, see the Appendix. Thus, the desired recursion is built and schematically expressed below:

$$\begin{bmatrix} I & \mathbf{A}_{p+1,q} & \vdots & \multicolumn{2}{c}{\mathbf{C}_{p+1,q}} \\ {}^2\mathbf{A}_{p+1,q} & I & \vdots & {}^2\mathbf{C}_{p+1,q} & \emptyset \\ \emptyset & {}^3\mathbf{A}_{p+1,q} & \vdots & I & {}^3\mathbf{C}_{p+1,q} \\ \multicolumn{2}{c}{{}^4\mathbf{A}_{p+1,q}} & \vdots & {}^4\mathbf{C}_{p+1,q} & I \end{bmatrix} = \mathcal{G}_{p,q} \begin{bmatrix} I & \mathbf{A}_{p,q} & \emptyset & \vdots & \multicolumn{2}{c}{\mathbf{C}_{p,q}} \\ \emptyset & {}^2\mathbf{A}_{p,q} & I & \vdots & \emptyset & {}^2\mathbf{C}_{p,q} \\ \emptyset & {}^3\mathbf{A}_{p,q} & \emptyset & \vdots & I & {}^3\mathbf{C}_{p,q} \\ \multicolumn{2}{c}{{}^4\mathbf{A}_{p,q}} & \emptyset & \vdots & {}^4\mathbf{C}_{p,q} & I \end{bmatrix}, \tag{12}$$

where $\mathcal{G}_{p,q}$ is a matrix of proper dimension, which is determined, through an operation of $\mathbf{O}(p+q)$ flops, based on the partial solution and auxiliary solutions of order (p,q).

The above equality presents a recursion in the variable p. To build up the recursion in the variable q, we need another version of the combined equations that are obtained by introducing some extra unknown matrices and exploiting the Toeplitz structure of $\mathbf{V}_{p,p}$, $\mathbf{\Delta}_{p,q}$, $\mathbf{\Lambda}_{q,p}$, and $\mathbf{S}_{q,q}$. The combined equation has the following two forms.

q-form: for any $p \geq 0, q \geq 0$,

$$\begin{bmatrix} I & \mathbf{A}_{p,q} & \vdots & \multicolumn{2}{c}{\mathbf{C}_{p,q}} & \emptyset \\ {}^2\mathbf{A}_{p,q+1} & I & \vdots & \multicolumn{2}{c}{{}^2\mathbf{C}_{p,q+1}} & \emptyset \\ \emptyset & {}^3\mathbf{A}_{p,q} & \vdots & I & {}^3\mathbf{C}_{p,q} & \emptyset \\ \emptyset & {}^4\mathbf{A}_{p-1,q} & \vdots & \emptyset & {}^4\mathbf{C}_{p-1,q} & I \end{bmatrix} \mathcal{W}_{p,q+1} = \begin{bmatrix} R_{p,q} & \mathbf{0}_p & \vdots & \multicolumn{2}{c}{\mathbf{0}_{q+1}} & N^1_{p,q} \\ \mathbf{0}_p & {}^2R_{p,q+1} & \vdots & \multicolumn{2}{c}{\mathbf{0}_{q+1}} & N^2_{p,q+1} \\ {}^3M_{p,q} & \mathbf{0}_p & \vdots & {}^3R_{p,q} & \mathbf{0}_q & N^3_{p,q} \\ {}^4M_{p-1,q} & \mathbf{0}_p & \vdots & {}^4N_{p-1,q} & \mathbf{0}_q & {}^4R_{p-1,q} \end{bmatrix}, \tag{13}$$

and

$q+1$-form: for any $p \geq 0, q \geq 0$,

$$\begin{bmatrix} I & \mathbf{A}_{p,q+1} & \vdots & \multicolumn{2}{c}{\mathbf{C}_{p,q+1}} \\ {}^2\mathbf{A}_{p,q+2} & I & \vdots & \multicolumn{2}{c}{{}^2\mathbf{C}_{p,q+2}} \\ \emptyset & {}^3\mathbf{A}_{p,q+1} & \vdots & I & {}^3\mathbf{C}_{p,q+1} \\ {}^4\mathbf{A}_{p-1,q+1} & \emptyset & \vdots & {}^4\mathbf{C}_{p-1,q+1} & I \end{bmatrix} \mathcal{W}_{p,q+1} = \begin{bmatrix} R_{p,q+1} & \mathbf{0}_p & \vdots & \multicolumn{2}{c}{\mathbf{0}_{q+2}} \\ \mathbf{0}_p & {}^2R_{p,q+2} & \vdots & \multicolumn{2}{c}{\mathbf{0}_{q+2}} \\ {}^3M_{p,q+1} & \mathbf{0}_p & \vdots & {}^3R_{p,q+1} & \mathbf{0}_{q+1} \\ \mathbf{0}_p & M^4_{p-1,q+1} & \vdots & \mathbf{0}_{q+1} & {}^4R_{p-1,q+1} \end{bmatrix}, \tag{14}$$

where all matrices in Eqns. (13) and (14) are unknown except the coefficient matrix $\mathcal{W}_{p,q+1}$, identity matrices, and zero matrices. Note that the unknowns on the left hand side are independent of the unknowns on the right hand side and the unknowns on the right hand side are determined as long as the unknowns on the left hand side are known. In addition, Eqn.(13) shares the same coefficient matrix with Eqn.(14) and equal systems of linear equations have equal solutions. Thus, we can follow the idea for deriving the recursion in the variable p and obtain the following recursion:

$$\begin{bmatrix} I & \mathbf{A}_{p,q+1} & \vdots & \mathbf{C}_{p,q+1} & \\ {}^2\mathbf{A}_{p,q+2} & I & \vdots & {}^2\mathbf{C}_{p,q+2} & \\ \emptyset & {}^3\mathbf{A}_{p,q+1} & \vdots & I & {}^3\mathbf{C}_{p,q+1} \\ {}^4\mathbf{A}_{p-1,q+1} & \emptyset & \vdots & {}^4\mathbf{C}_{p-1,q+1} & I \end{bmatrix} = \mathcal{H}_{p,q} \begin{bmatrix} I & \mathbf{A}_{p,q} & \vdots & & \mathbf{C}_{p,q} & \emptyset \\ {}^2\mathbf{A}_{p,q+1} & I & \vdots & & {}^2\mathbf{C}_{p,q+1} & \emptyset \\ \emptyset & {}^3\mathbf{A}_{p,q} & \vdots & I & {}^3\mathbf{C}_{p,q} & \emptyset \\ \emptyset & {}^4\mathbf{A}_{p-1,q} & \vdots & \emptyset & {}^4\mathbf{C}_{p-1,q} & I \end{bmatrix} \tag{15}$$

where $\mathcal{H}_{p,q}$ is a matrix of proper dimension, which is determined from the partial solution and auxiliary solutions of order (p,q). As shown in the Appendix, the determination takes $\mathbf{O}(p+q)$ flops.

We emphasize that Eqns.(12) and (15) provide a *flexible* recursion for solving a BTSS because the desired solution of either order $(p+1,q)$ or order $(p,q+1)$ can be recursively determined from the partial and auxiliary solutions of order (p,q). It is due to this feature that we call the recursion the order-recursive algorithm (ORA). Another feature of the ORA is that all operations involved can be performed in parallel except the inner-product operations appearing in calculating the unknowns on the right hand side of Eqns. (10) and (13). When the order is increased from (p,q) to $(p+1,q)$ or $(p,q+1)$, the number of required addition and multiplication operations, denoted by N_r, is of order $p+q+1$. As a result, the computational complexity of the ORA for solving a BTSS of order (p^*,q^*) is $\mathbf{O}((q^*+1)^2+p^*(p^*+q^*+1))$. A Levinson-type algorithm for solving general non-Toeplitz systems was developed by Porsani and Ulrych [3]. Considering block-Toeplitz submatrix systems, in their algorithm, N_r is $\mathbf{O}((p+q+1)^2)$.

It follows from Eqn. (10) that when $q=-1$, the coefficient matrix becomes a block-Toeplitz matrix, i.e., $\mathcal{W}_{p+1,-1} = \mathbf{V}_{p+1,p+1}$. This shows to that ${}^3\mathbf{A}_{p,-1} = \mathbf{0}_p$ and ${}^4\mathbf{A}_{p,-1} = \mathbf{0}_p$ as well as all the matrices ${}^3M_{p,-1}$, $M^3_{p,-1}$, and $M^4_{p,-1}$ are equal to zero matrices. As a result, the Levinson-Wiggins-Whittle-Robinson algorithm [4] can be derived, based on the relation between Eqn.(10) and Eqn. (11), in the same way as we derive the ORA. In fact, the expression of their algorithm is contained in Eqn.(11) with $q=-1$.

C. The Inverse and the Determinant of the Coefficient Matrix

The ORA itself does not need any knowledge of either the inverse or the determinant of a coefficient matrix. As a by-product, however, the ORA can be used to determine both the inverse and the determinant of a *symmetric* coefficient matrix. Starting from Eqn.(9) we can write

$$[{}^2\mathbf{A}_{p,-1} \;\; I \quad \mathbf{0}_{q+1}]\mathcal{W}_{p,q} = [\mathbf{0}_p \;\; {}^2R_{p,-1} \;\; * \cdots * \,]$$

and

$$[{}^4\mathbf{A}_{p,q}\ {}^4\mathbf{C}_{p,q}\ I]\mathcal{W}_{p,q} = [\mathbf{0}_{p+1}\ \mathbf{0}_q\ {}^4R_{p,q}],$$

where the asterisks $*$ represent matrices of proper dimension about which we do not care. By exploiting the Toeplitz structure of the four submatrices of the coefficient matrix $\mathcal{W}_{p,q}$, we have that for any $i = 0, \cdots, p$,

$$[{}^2\mathbf{A}_{i,-1}\ I\quad \mathbf{0}_{p-i}\ \mathbf{0}_{q+1}]\mathcal{W}_{p,q} = [\mathbf{0}_i\ {}^2R_{i,-1}\quad * \cdots *\ ] \tag{16}$$

and for any $j = 0, \cdots, q$,

$$[{}^4\mathbf{A}_{p,j}\ {}^4\mathbf{C}_{p,j}\ I\quad \mathbf{0}_{q-j}]\mathcal{W}_{p,q} = [\mathbf{0}_{p+1}\ \mathbf{0}_j\ {}^4R_{p,j}\ \ * \cdots *\ ]. \tag{17}$$

Now suppose that the coefficient matrix $\mathcal{W}_{p,q}$ is symmetric. Thus, given (backward) solutions to Eqns.(16) and (17), we have the following factorization of the coefficient matrix $\mathcal{W}_{p,q}$:

$$\mathcal{F}_{p,q}\mathcal{W}_{p,q}\mathcal{F}_{p,q}^T = \begin{bmatrix} {}^2R_{0,-1} & & & & & & & \\ & {}^2R_{1,-1} & & & & & \mathbf{0} & \\ & & \ddots & & & & & \\ & & & {}^2R_{p,-1} & & & & \\ & & & & {}^4R_{p,0} & & & \\ & \mathbf{0} & & & & {}^4R_{p,1} & & \\ & & & & & & \ddots & \\ & & & & & & & {}^4R_{p,q} \end{bmatrix}, \tag{18}$$

where

$$\mathcal{F}_{p,q} = \begin{bmatrix} I & & & & & & & \\ \boxed{{}^2\mathbf{A}_{1,-1}} & I & & & & \mathbf{0} & & \\ \vdots & & \ddots & & & & & \\ \multicolumn{3}{c}{\boxed{{}^2\mathbf{A}_{p,-1}}} & I & & & & \\ \multicolumn{4}{c}{\boxed{{}^4\mathbf{A}_{p,0}}} & I & & & \\ \multicolumn{4}{c}{\boxed{{}^4\mathbf{A}_{p,1}}} & \boxed{{}^4\mathbf{C}_{p,0}} & I & & \\ \multicolumn{4}{c}{\vdots} & \vdots & & \ddots & \\ \multicolumn{4}{c}{\boxed{{}^4\mathbf{A}_{p,q}}} & \multicolumn{3}{c}{\boxed{{}^4\mathbf{C}_{p,q}}} & I \end{bmatrix}.$$

Denote by $\mathcal{D}_{p,q}$ the diagonal matrix. The factorization expressed in Eqn.(18) is of the following compact form:

$$\mathcal{F}_{p,q}\mathcal{W}_{p,q}\mathcal{F}_{p,q}^T = \mathcal{D}_{p,q} \tag{19}$$

and subsequently, the inverse of the coefficient matrix $\mathcal{W}_{p,q}$ is of form:

$$[\mathcal{W}_{p,q}]^{-1} = \mathcal{F}^T_{p,q}[\mathcal{D}_{p,q}]^{-1}\mathcal{F}_{p,q}. \tag{20}$$

Note that $\det(\mathcal{F}_{p,q}) = 1$. Eqn.(21) produces

$$\det(\mathcal{W}_{p,q}) = (\prod_{i=0}^{p} \det({}^2R_{i,-1}))(\prod_{j=0}^{q} \det({}^4R_{p,j})). \tag{21}$$

D. Numerical Example

An overall expression of the ORA is presented in Eqns. (12), (15), (38), (39), (42), and (43). For simplicity, however, no schematic presentation of the ORA is provided in this paper. Readers may refer to reference [9] for such a detailed presentation. To make up for this drawback, we present an example verifying the ORA.

Consider a Toeplitz submatrix system expressed as in Eqn. (1) with the following coefficient matrix,

$$\mathcal{W}_{1,2} = \left[\begin{array}{cc|ccc} 8.7510 & -6.7377 & 1.0187 & -1.9350 & 1.4684 \\ -6.7377 & 8.7510 & -0.0346 & 1.0187 & -1.9350 \\ \hline 1.0187 & -0.0346 & 1.0000 & -0.0183 & 0.0046 \\ -1.9350 & 1.0187 & -0.0183 & 1.0000 & -0.0183 \\ 1.4684 & -1.9350 & 0.0046 & -0.0183 & 1.0000 \end{array}\right].$$

The solution process of the ORA can be illustrated as follows.

(1) Calculation of the initial condition at order $(0,0)$:

The partial solution and auxiliary solutions at order $(0,0)$ are obtained based on (10) and expressed below:

$$\left[\begin{array}{ccc|cc} 1 & \mathbf{A}_{0,0} & 0 & \multicolumn{2}{c}{\mathbf{C}_{0,0}} \\ 0 & {}^2\mathbf{A}_{0,0} & 1 & 0 & {}^2\mathbf{C}_{0,0} \\ 0 & {}^3\mathbf{A}_{0,0} & 0 & 1 & {}^3\mathbf{C}_{0,0} \\ \multicolumn{2}{c}{{}^4\mathbf{A}_{0,0}} & \emptyset & {}^4\mathbf{C}_{0,0} & 1 \end{array}\right] = \left[\begin{array}{cc|c} 1 & 0 & -1.0187 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ -0.1164 & 0 & 1 \end{array}\right],$$

where, as shown in (40), $\mathbf{A}_{0,0}, {}^2\mathbf{A}_{0,0}, {}^3\mathbf{A}_{0,0}, {}^2\mathbf{C}_{0,0}, {}^3\mathbf{C}_{0,0}$, and ${}^4\mathbf{C}_{0,0}$ are *fictitious* matrices. The reason why we keep them in the above expression is to have precise and consistent description of the ORA. Consequently, the matrix arrays before and after the dotted line on the left hand side have two columns and one column, respectively.

(2) Calculation of the solutions at order $(1,0)$:

Using the order recursion in the variable p, described in Eqn. (12), yields the following:

$$\left[\begin{array}{cccc} 1 & \mathbf{A}_{1,0} & \vdots & \mathbf{C}_{1,0} \\ {}^2\mathbf{A}_{1,0} & 1 & \vdots & {}^2\mathbf{C}_{1,0} \quad 0 \\ 0 & {}^3\mathbf{A}_{1,0} & \vdots & 1 \quad {}^3\mathbf{C}_{1,0} \\ \multicolumn{2}{c}{{}^4\mathbf{A}_{1,0}} & \vdots & {}^4\mathbf{C}_{1,0} \quad 1 \end{array}\right] = \left[\begin{array}{cccc} 1 & 0.7660 & \vdots & -0.9922 \\ 0.7699 & 1 & \vdots & 0 \\ 0 & 0.0040 & \vdots & 1 \\ -0.2784 & -0.2104 & \vdots & 1 \end{array}\right],$$

where the matrix array after the dotted line on the left hand side has only one column because ${}^2\mathbf{C}_{1,0}, {}^3\mathbf{C}_{1,0}$, and ${}^4\mathbf{C}_{1,0}$ are fictitious matrices.

(3) Calculation of the solutions at order $(1,1)$:

Now, we switch the order increment direction from the variable p to the variable q. The partial solution and auxiliary solutions are updated through Eqn. (15):

$$\left[\begin{array}{ccccc} 1 & \mathbf{A}_{1,1} & \vdots & \multicolumn{2}{c}{\mathbf{C}_{1,1}} \\ {}^2\mathbf{A}_{1,1} & 1 & \vdots & {}^2\mathbf{C}_{1,1} & 0 \\ \emptyset & {}^3\mathbf{A}_{1,1} & \vdots & 1 & {}^3\mathbf{C}_{1,1} \\ {}^4\mathbf{A}_{0,1} & \emptyset & \vdots & {}^4\mathbf{C}_{0,1} & 1 \end{array}\right] = \left[\begin{array}{ccccc} 1 & 0.6159 & \vdots & -0.9738 & 1.2897 \\ 0.8690 & 1 & \vdots & -0.8506 & 0 \\ 0 & 0.0021 & \vdots & 1 & 0.0162 \\ 0.2485 & 0 & \vdots & -0.2348 & 1 \end{array}\right].$$

(4) Calculation of the solutions at order $(1,2)$:

Using Eqn.(15) again, we obtain the desired solution and the other auxiliary solutions:

$$\left[\begin{array}{ccccc} 1 & \mathbf{A}_{1,2} & \vdots & \multicolumn{2}{c}{\mathbf{C}_{1,2}} \\ {}^2\mathbf{A}_{1,1} & 1 & \vdots & {}^2\mathbf{C}_{1,2} & 0 \\ 0 & {}^3\mathbf{A}_{1,2} & \vdots & 1 & {}^3\mathbf{C}_{1,2} \\ {}^4\mathbf{A}_{0,2} & 0 & \vdots & {}^4\mathbf{C}_{0,2} & 1 \end{array}\right] = \left[\begin{array}{cccccc} 1 & 0.4980 & \vdots & -0.9736 & 1.4011 & -0.4746 \\ 1.1761 & 1 & \vdots & -1.1408 & 1.2361 & 0 \\ 0 & 0.0019 & \vdots & 1 & 0.0163 & -0.0006 \\ -0.3537 & 0 & \vdots & 0.3436 & -0.6598 & 1 \end{array}\right].$$

The determinant and the inverse of the matrix $\mathcal{W}_{0,2}$ are dependent on the backward solutions and the quantities ${}^2R_{0,-1}$ and ${}^4R_{0,j}, j = 0,1,2$. These quantities are obtained during the computation process of the ORA:

$${}^2R_{0,-1} = 8.7510, {}^4R_{0,0} = 0.8814, {}^4R_{0,1} = 0.5235, {}^4R_{0,2} = 0.4943.$$

The determinant is now determined as

$$\det(\mathcal{W}_{0,2}) = 8.7510 \cdot 0.8814 \cdot 0.5235 \cdot 0.4943 = 1.9959.$$

The inverse of the matrix $\mathcal{W}_{0,2}$ is equal to

$$[\mathcal{W}_{0,2}]^{-1} = \mathcal{F}^T \cdot \begin{bmatrix} 8.7510 & 0 & 0 & 0 \\ 0 & 0.8814 & 0 & 0 \\ 0 & 0 & 0.5235 & 0 \\ 0 & 0 & 0 & 0.4943 \end{bmatrix}^{-1} \cdot \mathcal{F}$$

$$= \begin{bmatrix} 0.5006 & -0.4894 & 0.9467 & -0.7156 \\ -0.4894 & 1.4787 & -0.9072 & 0.6952 \\ 0.9467 & -0.9072 & 2.7908 & -1.3349 \\ -0.7156 & 0.6952 & -1.3349 & 2.0231 \end{bmatrix},$$

where

$$\mathcal{F} = \begin{bmatrix} 1.0000 & 0 & 0 & 0 \\ -0.1164 & 1.0000 & 0 & 0 \\ 0.2485 & -0.2348 & 1.0000 & 0 \\ -0.3537 & 0.3436 & -0.6598 & 1.0000 \end{bmatrix}.$$

3 Application to LMMSE IIR filtering

As discussed in Example 1, many IIR filtering techniques are based on LMMSE predictors:

$$\mathbf{y}_{n|n-1}(p,q) = -\mathbf{A}_{p,q}\mathbf{y}_{n-p}^{n-1} - \mathbf{C}_{p,q}\mathbf{u}_{n-q}^{n} \tag{22}$$

with predictor coefficients $[\ \mathbf{A}_{p,q} \ \ \mathbf{C}_{p,q}\ ]$ minimizing

$$\mathrm{trace}(\mathrm{E}[(\mathbf{y}_n - \mathbf{y}_{n|n-1}(p,q))(\mathbf{y}_n - \mathbf{y}_{n|n-1}(p,q))^T]).$$

The predictor coefficients $[\ \mathbf{A}_{p,q} \ \ \mathbf{C}_{p,q}\ ]$ are equal to the solution to a Yule-Walker equation of the form described in Eqns.(1) - (2) and (6), where the unknown $R_{p,q}$ on the right hand side of Eqn.(1) can be interpretated as the prediction error variance of the LMMSE predictor:

$$R_{p,q} = \mathrm{E}[(\mathbf{y}_n - \mathbf{y}_{n|n-1}(p,q))(\mathbf{y}_n - \mathbf{y}_{n|n-1}(p,q))^T]. \tag{23}$$

Obviously, the Yule-Walker equation in this case is a block-Toeplitz submatrix system with a *symmetric* coefficient matrix. Therefore, the desired predictor coefficients can be determined recursively by using the ORA described in (12) and (15). As a by-product of the ORA, the matrix arrays $[{}^2\mathbf{A}_{p,q}, {}^2\mathbf{C}_{p,q}]$, $[{}^3\mathbf{A}_{p,q}, {}^3\mathbf{C}_{p,q}]$, and $[{}^4\mathbf{A}_{p,q}, {}^4\mathbf{C}_{p,q}]$ and the matrices ${}^2R_{p,q}$, ${}^3R_{p,q}$, and ${}^4R_{p,q}$ are produced in the solution process. In fact, it can be verified through Eqn.(10) that these matrix arrays are equal to coefficients of some useful LMMSE forward/backward predictors. Specifically, these predictors are expressed below:

$$\mathbf{y}_{n-p|n}(p,q) = -{}^2\mathbf{A}_{p,q}\mathbf{y}^n_{n-p+1} - {}^2\mathbf{C}_{p,q}\mathbf{u}^n_{n-q+1}, \tag{24}$$

$$\mathbf{u}_{n|n-1}(p,q) = -{}^3\mathbf{A}_{p,q}\mathbf{y}^{n-1}_{n-p} - {}^3\mathbf{C}_{p,q}\mathbf{u}^{n-1}_{n-q}, \tag{25}$$

and

$$\mathbf{u}_{n-q|n}(p,q) = -{}^4\mathbf{A}_{p,q}\mathbf{y}^n_{n-p} - {}^4\mathbf{C}_{p,q}\mathbf{u}^n_{n-q+1}, \tag{26}$$

where the prediction error variances of these predictors are equal to

$$\begin{aligned} {}^2R_{p,q} &= \mathrm{E}[(\mathbf{y}_{n-p} - \mathbf{y}_{n-p|n}(p,q))(\mathbf{y}_{n-p} - \mathbf{y}_{n-p|n}(p,q))^T] \\ {}^3R_{p,q} &= \mathrm{E}[(\mathbf{u}_n - \mathbf{u}_{n|n-1}(p,q))(\mathbf{u}_n - \mathbf{u}_{n|n-1}(p,q))^T] \\ {}^4R_{p,q} &= \mathrm{E}[(\mathbf{u}_{n-q} - \mathbf{u}_{n-q|n}(p,q))(\mathbf{u}_{n-q} - \mathbf{u}_{n-q|n}(p,q))^T]. \end{aligned} \tag{27}$$

Obviously, the LMMSE predictors in (24) – (26) represent, respectively, the backward predictor of $\mathbf{y}_{n-p}$, the forward predictor of $\mathbf{u}_n$, and the backward predictor of $\mathbf{u}_{n-q}$.

A. Lattice Filter

The prediction errors of the LMMSE predictors in (22) and (24) – (26) are expressed below

$$\begin{aligned}
{}^f\mathbf{e}_{n,p,q} &\triangleq \mathbf{y}_n - \mathbf{y}_{n|n-1}(p,q) = \mathbf{y}_n + [\mathbf{A}_{p,q}\ \mathbf{C}_{p,q}]\begin{pmatrix}\mathbf{y}^{n-1}_{n-p}\\ \mathbf{u}^n_{n-q}\end{pmatrix}; \quad {}^f\mathbf{e}_{n,0,0} = \mathbf{y}_n - \Delta_0 S_0^{-1}\mathbf{u}_n,\\
{}^b\mathbf{e}_{n,p,q} &\triangleq \mathbf{y}_{n-p} - \mathbf{y}_{n-p|n}(p,q) = \mathbf{y}_{n-p} + [{}^2\mathbf{A}_{p,q}\ {}^2\mathbf{C}_{p,q}]\begin{pmatrix}\mathbf{y}^n_{n-p+1}\\ \mathbf{u}^n_{n-q+1}\end{pmatrix}; \quad {}^b\mathbf{e}_{n,0,0} = \mathbf{y}_n,\\
{}^f\mathbf{v}_{n,p,q} &\triangleq \mathbf{u}_n - \mathbf{u}_{n|n-1}(p,q) = \mathbf{u}_n + [{}^3\mathbf{A}_{p,q}\ {}^3\mathbf{C}_{p,q}]\begin{pmatrix}\mathbf{y}^{n-1}_{n-p}\\ \mathbf{u}^{n-1}_{n-q}\end{pmatrix}; \quad {}^f\mathbf{v}_{n,0,0} = \mathbf{u}_n,\\
{}^b\mathbf{v}_{n,p,q} &\triangleq \mathbf{u}_{n-q} - \mathbf{u}_{n-q|n}(p,q) = \mathbf{u}_{n-q} + [{}^4\mathbf{A}_{p,q}\ {}^4\mathbf{C}_{p,q}]\begin{pmatrix}\mathbf{y}^n_{n-p}\\ \mathbf{u}^n_{n-q+1}\end{pmatrix}; \quad {}^b\mathbf{v}_{n,0,0} = \mathbf{u}_n - \Lambda_0 V_0^{-1}\mathbf{y}_n.
\end{aligned} \tag{28}$$

Inserting (12) and (15) into (28) and then using the expression of the weighting matrices, presented in Eqns. (37) and (41), yield the desired lattice recursion in either of two forms:

p-form: for any $p \geq 0$ and $q \geq 0$,

$$\begin{aligned}
{}^f\mathbf{e}_{n,p+1,q} &= {}^f\mathbf{e}_{n,p,q} + {}^2\mathcal{K}_{p,q}{}^b\mathbf{e}_{n-1,p,q} + {}^3\mathcal{K}_{p,q}{}^f\mathbf{v}_{n,p,q} \\
{}^b\mathbf{e}_{n,p+1,q} &= {}^b\mathbf{e}_{n-1,p,q} + {}^1\mathcal{L}_{p,q}{}^f\mathbf{e}_{n,p,q} + {}^3\mathcal{L}_{p,q}{}^f\mathbf{v}_{n,p,q} + {}^4\mathcal{L}_{p,q}{}^b\mathbf{v}_{n,p,q} \\
{}^f\mathbf{v}_{n,p+1,q} &= {}^f\mathbf{v}_{n,p,q} + {}^2\mathcal{M}_{p,q}{}^b\mathbf{e}_{n-1,p,q} \\
{}^b\mathbf{v}_{n,p+1,q} &= {}^b\mathbf{v}_{n,p,q} + {}^2\mathcal{N}_{p,q}{}^b\mathbf{e}_{n,p+1,q}
\end{aligned} \tag{29}$$

q-form: for any $p \geq 0$ and $q \geq 0$,

$$\begin{aligned}
{}^f\mathbf{e}_{n,p,q+1} &= {}^f\mathbf{e}_{n,p,q} + {}^3\mathcal{P}_{p,q}{}^f\mathbf{v}_{n,p,q} + {}^4\mathcal{P}_{p,q}{}^b\mathbf{v}_{n-1,p-1,q} \\
{}^b\mathbf{e}_{n,p,q+1} &= {}^b\mathbf{e}_{n,p,q} + {}^4\mathcal{Q}_{p,q}{}^b\mathbf{v}_{n,p-1,q} \\
{}^f\mathbf{v}_{n,p,q+1} &= {}^f\mathbf{v}_{n,p,q} + {}^4\mathcal{S}_{p,q}{}^b\mathbf{v}_{n-1,p-1,q} \\
{}^b\mathbf{v}_{n,p-1,q+1} &= {}^b\mathbf{v}_{n-1,p-1,q} + {}^1\mathcal{T}_{p,q}{}^f\mathbf{e}_{n,p,q} + {}^2\mathcal{T}_{p,q}{}^b\mathbf{e}_{n,p,q+1} + {}^3\mathcal{T}_{p,q}{}^f\mathbf{v}_{n,p,q}.
\end{aligned} \tag{30}$$

Thus, an LMMSE predictor having p poles and q zeros can be implemented in a lattice structure. The lattice structure can vary in the way the numbers of predictor poles and zeros are increased. One of the lattice structures is depicted in Figure 1, which is composed of two kinds of lattice cells: p-form and q-form. These lattice cells are drawn in Figure 2, where the operator denoted by z represents the one-step time delay operator.

We remark that the lattice recursion expressed in (29) and (30) differs from Markel and Gray's [11] in at least two aspects: (1) the desired signals and filter inputs receive symmetric treatment and (2) the lattice LMMSE pole-zero predictor is built as a successive expansion of lattice LMMSE pole-zero predictors of *lower* order. The first property is desired in cases where measurements of filter input are corrupted by noise [12]. The second property can be particularly useful in situations where filter order should be determined in terms of filter performance [4,9].

B. Orthogonal Properties of Backward Prediction Errors

The lattice filter of Figure 1 exhibits some useful correlation properties between the forward and backward prediction errors. One of the properties is the orthogonality of the backward prediction errors, which is particularly useful in joint-process estimation [7].

The expression of block elements of the Yule-Walker matrix $\mathcal{W}_{p,q}$ in (6) shows that

$$\mathcal{W}_{p,q} = \mathrm{E}[\begin{pmatrix} \mathbf{y}^n_{n-p} \\ \mathbf{u}^n_{n-q} \end{pmatrix} (\mathbf{y}^{n^T}_{n-p} \quad \mathbf{u}^{T}_{n-q})].$$

Substituting this expression into (18) and then using the definition of backward prediction errors

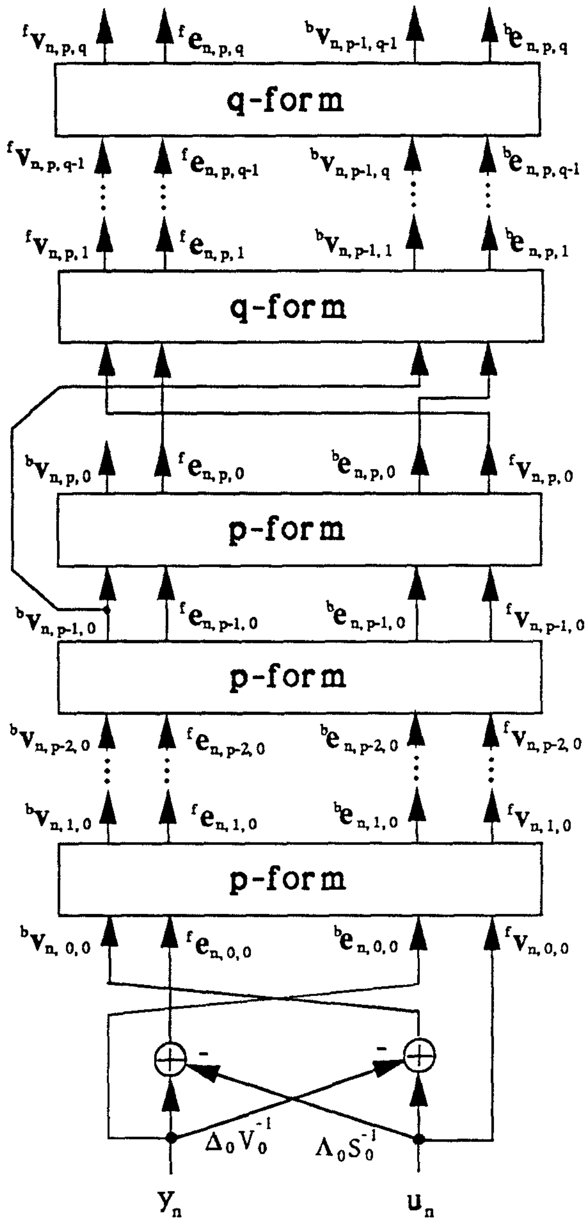

Figure 1: Lattice LMMSE pole-zero predictor

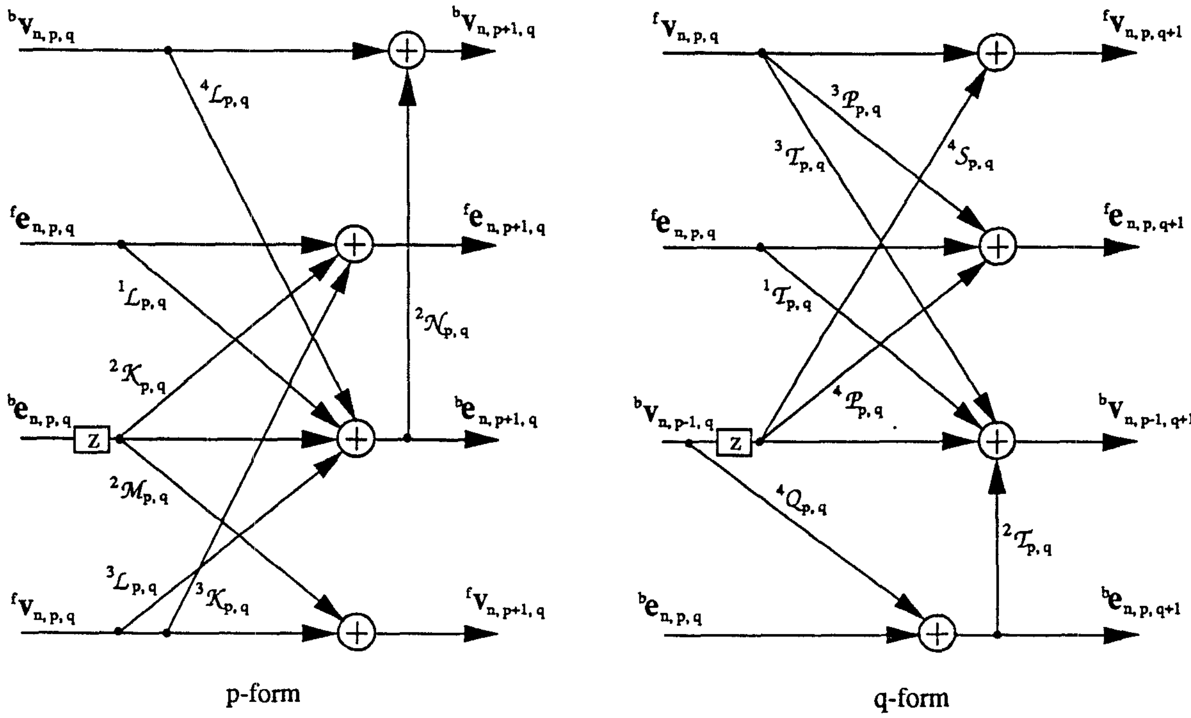

Figure 2: Lattice LMMSE pole-zero predictor

$^{b}\mathbf{e}_{n,p,q}$ and $^{b}\mathbf{v}_{n,p,q}$, we have

$$
\mathrm{E}\left[\begin{pmatrix} ^{b}\mathbf{e}_{n,0,-1} \\ ^{b}\mathbf{e}_{n,1,-1} \\ \vdots \\ ^{b}\mathbf{e}_{n,p,-1} \\ ^{b}\mathbf{v}_{n,p,0} \\ \vdots \\ ^{b}\mathbf{v}_{n,p,q} \end{pmatrix}\begin{pmatrix} ^{b}\mathbf{e}_{n,0,-1} \\ ^{b}\mathbf{e}_{n,1,-1} \\ \vdots \\ ^{b}\mathbf{e}_{n,p,-1} \\ ^{b}\mathbf{v}_{n,p,0} \\ \vdots \\ ^{b}\mathbf{v}_{n,p,q} \end{pmatrix}^T\right] = \begin{bmatrix} ^{2}R_{0,-1} & & & & & & & \\ & ^{2}R_{1,-1} & & & & \mathbf{0} & & \\ & & \ddots & & & & & \\ & & & ^{2}R_{p,-1} & & & & \\ & & & & ^{4}R_{p,0} & & & \\ & \mathbf{0} & & & & ^{4}R_{p,1} & & \\ & & & & & & \ddots & \\ & & & & & & & ^{4}R_{p,q} \end{bmatrix}. \quad (31)
$$

The backward prediction errors $^{b}\mathbf{e}_{n,p,-1}$ are not directly available in the lattice predictor of Figure 1. However, this does not cause any problem because they can be easily and efficiently determined by using a well-known lattice all-pole predictor [13].

4 Application to ARX Identification

A fast identification algorithm for multivariate ARX models having independently and arbitrarily specified number of poles and zeros has been developed by Karaboyas and Kalouptsidis [14].

Applying the results developed in Section 2, a fast time and order recursive algorithm (TORA) is presented in this section. The difference between the proposed algorithm and the aforementioned algorithm lies in the way the TORA updates parameter estimates recursively either in the number of ARX model poles or in the number of zeros. This parameter updating scheme leads to a particularly useful feature in that the solution to an identification problem for an ARX model with p poles and q zeros contains in it all parameter estimates of ARX models having smaller numbers of poles and zeros.

A. Time and Order Recursive Algorithm

The ARX identification problem considered here is described in Example 2. It is well known that when the data are windowed via the autocorrelation method [7], the least-squares parameter estimate of an ARX model satisfies the following system of linear equations, which is a symmetric block-Toeplitz submatrix system:

$$[\, I \ \mathbf{A}_{N,p,q} \ \mathbf{C}_{N,p,q} \,] \begin{bmatrix} \mathbf{V}_{N,p,p} & \boldsymbol{\Delta}_{N,p,q} \\ \boldsymbol{\Delta}^T_{N,q,p} & \mathbf{S}_{N,q,q} \end{bmatrix} = [R_{N,p,q} \ \mathbf{0}_p \ \mathbf{0}_{q+1}]. \tag{32}$$

The submatrices, $\mathbf{V}_{N,p,p}, \boldsymbol{\Delta}_{N,q,p}$, and $\mathbf{S}_{N,q,q}$, have the expression presented in (2) except for the time-varying elements. Specifically, in this case, they are equal to

$$\begin{aligned}
V_{N,i} &= \delta(i)\frac{1}{Nr_0}I + \frac{1}{N}\sum_{n=1}^{N}\lambda^{N-n}\mathbf{y}_n\mathbf{y}^T_{n-i}, \quad i = 0,\cdots,p, \quad N \geq 1 \\
S_{N,j} &= \delta(j)\frac{1}{Nr_0}I + \frac{1}{N}\sum_{n=1}^{N}\lambda^{N-n}\mathbf{u}_n\mathbf{u}^T_{n-j}, \quad j = 0,\cdots,q, \quad N \geq 1 \\
\Delta_{N,i} &= \begin{cases} \frac{1}{N}\sum_{n=1}^{N}\lambda^{N-n}\mathbf{y}_n\mathbf{u}^T_{n-i}, & i = 0,\cdots,q, \\ \frac{1}{N}\sum_{n=1}^{N}\lambda^{N-n}\mathbf{y}_{n+i}\mathbf{u}^T_{n}, & i = -1,\cdots,-p, \end{cases} \quad N \geq 1,
\end{aligned} \tag{33}$$

where $\delta(i)$ is the δ-function which is used to prevent the coefficient matrix in (32) from being singular. It is evident that the (matrix) elements expressed in (33) can be updated recursively in time:

$$\begin{aligned}
V_{N,i} &= \frac{N-1}{N}\lambda V_{N-1,i} + \frac{1}{N}\mathbf{y}_n\mathbf{y}_{n-i} + \delta(i)\delta(N-1)\frac{1}{r_0}I, \quad i = 0,\cdots,p, \quad N \geq 1 \\
S_{N,j} &= \frac{N-1}{N}\lambda S_{N-1,j} + \frac{1}{N}\mathbf{u}_n\mathbf{u}_{n-j} + \delta(j)\delta(N-1)\frac{1}{r_0}I, \quad j = 0,\cdots,q, \quad N \geq 1 \\
\Delta_{N,i} &= \frac{N-1}{N}\lambda\Delta_{N-1,i} + \begin{cases} \frac{1}{N}\mathbf{y}_n\mathbf{u}^T_{n-i}, & i = 0,\cdots,q, \\ \frac{1}{N}\mathbf{y}_{n+i}\mathbf{u}^T_{n}, & i = -1,\cdots,-p, \end{cases} \quad N \geq 1.
\end{aligned} \tag{34}$$

Also, due to the Toeplitz structure of the submatrices in Eqn. (32), the *fast* order-recursive algorithm developed in Section 2 can be used to determine the solution to Eqn. (32) at each time instant N. This leads to an efficient algorithm for ARX system identification, which we call the time and order recursive algorithm (TORA).

To evaluate the performance of the TORA, we compare the TORA with the recursive least-squares algorithm (RLS) in terms of consistency. Denote by $\tilde{\theta}_N$ and $\hat{\theta}_N$ the solutions minimizing the quantity expressed in (8) with $\lambda = 1$, which are obtained, respectively, by using the TORA and RLS. Then, it is shown in [9,15] that the difference between these two solutions has the following relation:

$$\tilde{\theta}_n - \hat{\theta}_n = -\frac{1}{r_0}\hat{\theta}_0 \hat{P}_n + (\frac{1}{r_0}\hat{\theta}_0 \hat{P}_n - \hat{\theta}_n) M_n \{\hat{P}_n M_n + I\}^{-1} \hat{P}_n, \qquad n \geq 0, \tag{35}$$

where $\hat{\theta}_0$ is the initial estimate for the RLS algorithm and $\frac{1}{r_0}I$ represents the initial gain matrix. The matrix M_n is a uniformly bounded matrix provided the input and output signals are uniformly bounded. The matrix $\hat{P}_n$ is the gain matrix associated with the RLS solution, which is defined as

$$\hat{P}_n \triangleq [\frac{1}{r_0} I + \sum_{n=1}^{N} \begin{pmatrix} \mathbf{y}_{n-p}^{n-1} \\ \mathbf{u}_{n-q}^{n} \end{pmatrix} \begin{pmatrix} \mathbf{y}_{n-p}^{n-1} \\ \mathbf{u}_{n-q}^{n} \end{pmatrix}^T]^{-1}.$$

Suppose that $\{\mathbf{y}_n\}_{n=1}^{\infty}$ and $\{\mathbf{u}_n\}_{n=1}^{\infty}$ represent two uniformly bounded sequences of output and input signals. By evaluating the identity (35), we have the following observation: If the gain matrix satisfies $||\hat{P}_n|| \to 0$, as $n \to \infty$, and the RLS parameter estimates are uniformly bounded: $\exists k > 0$ such that $||\hat{\theta}_n|| < k < \infty$ for $\forall n \geq 1$, then the TORA parameter estimates converge to the corresponding RLS parameter estimates with the convergence rate:

$$||\hat{\theta}_n - \tilde{\theta}_n|| \leq \mathbf{O}(||\hat{P}_n||). \tag{36}$$

The first assumption in the above statement is a necessary condition for the consistency of least-squares parameter estimation [4]. The second is automatically guaranteed by the convergence of least-squares parameter estimates. As a result, for uniformly bounded input and output signals, *which are not necessarily stationary signals*, the TORA preserves the consistency of the RLS algorithm.

B. Simulation Example

To further illustrate the performance of the TORA, we design an adaptive IIR filter by minimizing the accumulated squared error between the filter output and a desired response. The filter coefficients are updated by using the TORA through the equation-error formulation [10]. For comparison, a similar design is performed by using the RLS algorithm.

The desired signals, which are drawn in the upper plot of Figure 3, are generated by the following ARX system:

$$y_n + 0.7 \cdot y_{n-1} - 0.4975 \cdot y_{n-2} - 0.8483 \cdot y_{n-3} - u_n = w_n,$$

where the input u_n is a sinusoidal signal with two frequencies, $u_n = sin(3n) + sin(0.1n)$. The model noise w_n is a zero mean pseudo white noise with a variance of 0.5. The filter coefficients are drawn in the middle plot of Figure 3. As predicted in the previous discussion, the difference between filter coefficients updated by the TORA and the RLS algorithm converges to zero as more and more data is available. In fact, the convergence rate is very fast in this example; the difference becomes very small (less than 5%) after 100 iterations. We next compare the average error energy: the accumulated squared error between the filter output and the desired response divided by N. As shown in the lower plot of Figure 3, after 200 iterations, the average error energy of the adaptive filter updated by using the TORA is less than 2 dB higher than that of the corresponding adaptive filter updated by using the RLS algorithm. This indicates that the TORA is applicable to adaptive signal processing.

5 Conclusions

In this paper an order recursive algorithm (ORA) has been established for solving a class of non-Toeplitz systems. The systems have a special structure in that their coefficient matrices can be partitioned into four block-Toeplitz submatrices. In the simplest case, where the coefficient matrices are Toeplitz, the ORA reduces to the well-known Levinson algorithm. Compared with the approach recently proposed by Porsani and Ulrych [3], the ORA demonstrates superiority in computational efficiency. The ORA possesses high parallelism and generates both the desired solution and partial solutions of lower order.

Some applications which have importance in multichannel IIR filtering have been considered. Specifically, the ORA has been applied to linear prediction through zero-pole filters leading to a lattice structure for LMMSE zero-pole predictors. In keeping with conventional lattice filters, mutually orthogonal backward prediction errors are produced in the structure. Moreover, the desired signals and filter (predictor) inputs receive symmetric treatment in the structure. Another aspect of the structure is that an LMMSE predictor having independently and arbitrarily specified number of poles and zeros can be implemented as a successive expansion of LMMSE predictors with a lower number of poles and zeros. This, plus the prediction error variances which are produced in the determination of lattice predictor coefficients, may be of help in signal modeling.

The ORA has also been applied to system identification for ARX systems leading to the *fast* time and order recursive algorithm (TORA). Mathematically, the TORA is equivalent to a least-

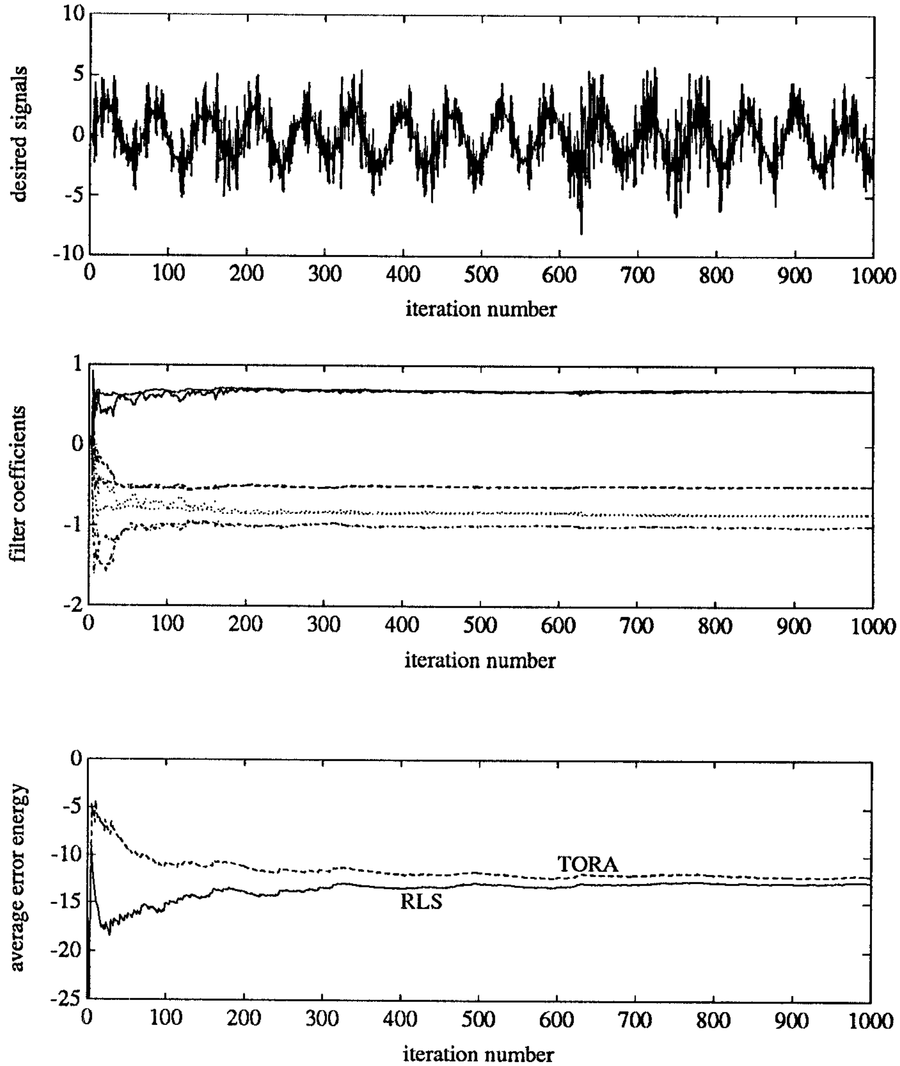

Figure 3: The comparison between the adaptive filters adjusted by using the TORA and the RLS algorithm. The "jumpier" curves in the middle plot represent the filter coefficients determined by using the TORA.

squares algorithm for ARX system identification when the involved signals are windowed through the correlation method. Computationally, it generates least-squares parameter estimates for *a group of ARX models rather than a single ARX model.* In addition, the TORA preserves the consistency of the well-known recursive least-squares algorithm for uniformly bounded signals. Note that the signals are not necessarily stationary.

The class of non-Toeplitz systems considered here includes the Yule-Walker equations appearing in two-dimensional AR systems when the region of support is a narrow band having two horizontal or vertical elements [8]. Further research is required to extend the ORA to the case where general Yule-Walker equations appearing in two-dimensional AR systems can be accommodated.

Appendix

Weighting Matrices: The definition of weighting matrix $\mathcal{G}_{p,q}$ comes from the observation that the same systems of linear equations have the same solutions and Eqn. (10) shares the same coefficient matrix with Eqn. (11). Premultiplying both sides of Eqn. (10) by matrix $\mathcal{G}_{p,q}$ yields a new system of linear equations. Thus, the new system has the same solution as Eqn. (11) if *both the left and right hand sides* of the new system are equal to their counterparts in Eqn. (11). Note that the unknowns on the left hand side of Eqn. (11) do not depend on the unknowns on the right hand side. So, it suffices to choose weighting matrix $\mathcal{G}_{p,q}$ such that (1) the right side of the new system is equal to its counterpart in Eqn. (11) and (2) all known matrices (i.e., the zero and identity matrices) on the left hand side of Eqn. (11) are equal to their counterparts on the left hand side of the new system.

To meet these two requirements, we define weighting matrix $\mathcal{G}_{p,q}$ as follows:

$$\mathcal{G}_{p,q} \triangleq \begin{bmatrix} I & {}^2\mathcal{K}_{p,q} & {}^3\mathcal{K}_{p,q} & \emptyset \\ {}^1\mathcal{L}_{p,q} & I & {}^3\mathcal{L}_{p,q} & {}^4\mathcal{L}_{p,q} \\ \emptyset & {}^2\mathcal{M}_{p,q} & I & \emptyset \\ {}^2\mathcal{N}_{p,q}{}^1\mathcal{L}_{p,q} & {}^2\mathcal{N}_{p,q} & {}^2\mathcal{N}_{p,q}{}^3\mathcal{L}_{p,q} & I+{}^2\mathcal{N}_{p,q}{}^4\mathcal{L}_{p,q} \end{bmatrix} \tag{37}$$

satisfying

$$\mathcal{G}_{p,q} \left[\begin{array}{ccc:cc} R_{p,q} & \mathbf{0}_p & M^1_{p,q} & \multicolumn{2}{c}{\mathbf{0}_{q+1}} \\ {}^2M_{p,q} & \mathbf{0}_p & {}^2R_{p,q} & {}^2N_{p,q} & \mathbf{0}_q \\ {}^3M_{p,q} & \mathbf{0}_p & M^3_{p,q} & {}^3R_{p,q} & \mathbf{0}_q \\ \multicolumn{2}{c}{\mathbf{0}_{p+1}} & M^4_{p,q} & \mathbf{0}_q & {}^4R_{p,q} \end{array}\right] = \left[\begin{array}{cc:cc} R_{p+1,q} & \mathbf{0}_{p+1} & \multicolumn{2}{c}{\mathbf{0}_{q+1}} \\ \mathbf{0}_{p+1} & {}^2R_{p+1,q} & \mathbf{0}_q & N^2_{p+1,q} \\ {}^3M_{p+1,q} & \mathbf{0}_{p+1} & {}^3R_{p+1,q} & \mathbf{0}_q \\ \multicolumn{2}{c}{\mathbf{0}_{p+2}} & \mathbf{0}_q & {}^4R_{p+1,q} \end{array}\right] \tag{38}$$

and

$${}^1\mathcal{L}_{p,q}C^{p,q}_q + {}^3\mathcal{L}_{p,q}{}^3C^{p,q}_q + {}^4\mathcal{L}_{p,q} = -{}^2C^{p,q}_1, \tag{39}$$

where $C_q^{p,q}, {}^3C_q^{p,q}$, and ${}^2C_1^{p,q}$ are matrix elements of the desired solution and auxiliary solutions:

$$\begin{aligned}
\mathbf{A}_{p,q} &= [\, A_1^{p,q} \;\cdots\; A_p^{p,q} \,], & \mathbf{C}_{p,q} &= [\, C_0^{p,q} \;\cdots\; C_q^{p,q} \,]; \\
{}^2\mathbf{A}_{p,q} &= [\, {}^2A_p^{p,q} \;\cdots\; {}^2A_1^{p,q} \,], & {}^2\mathbf{C}_{p,q} &= [\, {}^2C_q^{p,q} \;\cdots\; {}^2C_1^{p,q} \,]; \\
{}^3\mathbf{A}_{p,q} &= [\, {}^3A_1^{p,q} \;\cdots\; {}^3A_p^{p,q} \,], & {}^3\mathbf{C}_{p,q} &= [\, {}^3C_1^{p,q} \;\cdots\; {}^3C_q^{p,q} \,]; \\
{}^4\mathbf{A}_{p,q} &= [\, {}^4A_p^{p,q} \;\cdots\; {}^4A_0^{p,q} \,], & {}^4\mathbf{C}_{p,q} &= [\, {}^4C_q^{p,q} \;\cdots\; {}^4C_1^{p,q} \,].
\end{aligned} \tag{40}$$

Eqns. (38) and (39) can be easily solved and the explicit expression of the solution to these two equations can be obtained when the formula for the inverse of block matrices [16] is applied. When coefficient matrix $\mathcal{W}_{p,q}$ is symmetric and nonsingular, Eqns. (38) and (39) have a unique solution. For details, see reference [9]

Similarly, we can define another weighting matrix $\mathcal{H}_{p,q}$ as follows:

$$\mathcal{H}_{p,q} = \begin{bmatrix} I & \emptyset & {}^3\mathcal{P}_{p,q} & {}^4\mathcal{P}_{p,q} \\ {}^4\mathcal{Q}_{p,q+1}{}^1\mathcal{T}_{p,q} & I + {}^4\mathcal{Q}_{p,q+1}{}^2\mathcal{T}_{p,q} & {}^4\mathcal{Q}_{p,q+1}{}^3\mathcal{T}_{p,q} & {}^4\mathcal{Q}_{p,q+1} \\ \emptyset & \emptyset & I & {}^4\mathcal{S}_{p,q} \\ {}^1\mathcal{T}_{p,q} & {}^2\mathcal{T}_{p,q} & {}^3\mathcal{T}_{p,q} & I \end{bmatrix} \tag{41}$$

satisfying

$$\mathcal{H}_{p,q} \left[\begin{array}{cc:ccc} R_{p,q} & \mathbf{0}_p & \mathbf{0}_{q+1} & & N_{p,q}^1 \\ \mathbf{0}_p & {}^2R_{p,q+1} & \mathbf{0}_{q+1} & & N_{p,q+1}^2 \\ {}^3M_{p,q} & \mathbf{0}_p & {}^3R_{p,q} & \mathbf{0}_q & N_{p,q}^3 \\ {}^4M_{p-1,q} & \mathbf{0}_p & {}^4N_{p-1,q} & \mathbf{0}_q & {}^4R_{p-1,q} \end{array} \right] = \left[\begin{array}{cc:cc} R_{p,q+1} & \mathbf{0}_p & \multicolumn{2}{c}{\mathbf{0}_{q+2}} \\ \mathbf{0}_p & {}^2R_{p,q+2} & \multicolumn{2}{c}{\mathbf{0}_{q+2}} \\ {}^3M_{p,q+1} & \mathbf{0}_p & {}^3R_{p,q+1} & \mathbf{0}_{q+1} \\ \mathbf{0}_p & M_{p-1,q+1}^4 & \mathbf{0}_{q+1} & {}^4R_{p-1,q+1} \end{array} \right] \tag{42}$$

and

$${}^2\mathcal{T}_{p,q} + {}^1\mathcal{T}_{p,q}A_p^{p,q} + {}^3\mathcal{T}_{p,q}{}^3A_p^{p,q} = -{}^4A_0^{p-1,q}. \tag{43}$$

References

[1] T. Kailath, "A View of Three Decades of Linear Filtering Theory," *IEEE Trans. Inf. Theory*, IT-20, no. 2, pp. 146–181, 1974.

[2] G. Cybenko, "The Numerical Stability of the Levinson-Durbin Algorithm for Toeplitz Systems of Equations," *SIAM J. Sci. Stat. Comput.*, 1, no. 3, pp. 303 – 319, Sep. 1980.

[3] M. Porsani and T. J. Ulrych, "Levinson-Type Extensions for Non-Toeplitz Systems," *IEEE Trans. on Signal Processing*, 39, no. 2, pp. 366–375, February 1991.

[4] P. E. Caines, *Linear Stochastic Systems.* John Wiley & Sons, Inc., 1988.

[5] B. Friedlander, Morf, T. Kailath and L. Ljung, "Extended Levinson- and Chandrasekhar-type equations for a General discrete-Time Linear Estimation Problem," *IEEE Trans. Automatioc Control*, AC-23, pp. 653–659, 1978.

[6] T. Kailath, S. Y. Kung and M. Morf, "Displacement Ranks of Matrices and Linear Equations," *J. Math Anay. App.*, 68, pp. 395–407, 1979.

[7] S. Haykin, *Adaptive Filter Theory.* Prentice-Hall, 1986.

[8] S. M. Kay, *Modern Spectral Estimation.* Prentice-Hall, Inc., 1988.

[9] J. Pan, "Fast Recursive Estimation of System Order and Parameters for Adaptive Control and IIR Filtering," Electrical Engineering Department, University of Maryland at College Park, Ph.D. Thesis, August, 1991.

[10] J. J. Shynk, "Adaptive IIR Filtering," presented at IEEE ASSP Magazine, April 1989.

[11] J. D. Markel and A. H. Gray. Jr., *Linear Prediction of Speech.* Springer-Verlag, 1976.

[12] L. Ljung, "Issues in System Identification," *IEEE Control Systems Magazine*, 11, no. 1, pp. 25–229, 1991.

[13] B. Friedlander, "Lattice Filters for Adaptive Processing," *Proc. IEEE* , 70, pp. 829–867, Aug. 1982.

[14] S. Karaboyas and N. Kalouptsidis, "Efficient Adaptive lgorithms for ARX Identification," *IEEE Trans on ASSP*, 39, no. 3, pp. 571–582, March 1991.

[15] J. Pan and W. S. Levine, "A Fast Time and Order Recursive Parameter Estimation Algorithm for a Set of ARX Models," in *Proc. of American Control Conference.* Boston, MA: pp. 2832–2838, 1991.

[16] T. Kailath, *Linear Systems.* Prentice-Hall, Inc., 1980.